QA:QA

Civilian Radioactive Waste Management System
Management & Operating Contractor

SOFTWARE USER'S MANUAL
EQ3/6, Version 8.0

Thomas W. Wolery,

Russell L Jarek

Software Tracking Number: 10813-8.0-00

Software Activity Number: SNL-2002-099

Software Media Number: 10813-PC-8.0-00

Software Document Number: 10813-UM-8.0-00

January 2003

Prepared for:
U.S. Department of Energy
Office of Civilian Radioactive Waste Management
Office of Repository Development
1261 Town Center Drive
Las Vegas, Nevada 89144

Prepared by:
Sandia National Laboratories
P.O. Box 5800
Albuquerque, New Mexico 87185

ACKNOWLEDGMENT

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under contract DE-AC04-94AL85000. This work is supported by Office of Repository Development.

Civilian Radioactive Waste Management System
Management & Operating Contractor

SOFTWARE USER'S MANUAL

EQ3/6, Version 8.0

Software Tracking Number:	10813-8.0-00
Software Activity Number:	SNL-2002-099
Software Media Number:	10813-PC-8.0-00
Software Document Number:	10813-UM-8.0-00

January 2003

Authored by:

Thomas W. Wolery
Lawrence Livermore National Labs

Date

Authored by:

Russell L Jarek
Sandia National Labs

Date

Verified by:

Darren Jolley
Independent Technical Reviewer

Date

Approved by:

Daniel A. Thomas
Responsible Manager

Date

Reviewed by:

Dianne P. Spence
ITSMA

Date

Change History

Revision Number	Effective Date	Description and Reason for Revision
0		Original Issue

INTENTIONALLY LEFT BLANK

CONTENTS

CONTENTS (Continued)

CONTENTS (Continued)

FIGURES

TABLES

ACRONYMS AND ABBREVIATIONS

AE	Algebraic Equation
ATDT	Automated Technical Data Tracking
DTN	Data Tracking Number
EBS	Engineered Barrier Systems
ITP	Installation Test Plan
LLNL	Lawrence Livermore National Laboratory
TDS	Total Dissolved Solids
ODE	Ordinary Differential Equation
OS	Operating System
QA	Quality Assurance
SI	Saturation Index
SNL	Sandia National Laboratories
TDS	Total Dissolved Solids
TIC	Technical Information Center
UZ	Unsaturated Zone
VTP	Validation Test Plan
WF	Waste Form
WP	Waste Package

INTENTIONALLY LEFT BLANK

1. PURPOSE AND SCOPE

The goal of this Software User's Manual document is to explain relevant information that allows the use of the computer code EQ3/6 Version 8.0, developed by Dr. Thomas J. Wolery at the Lawrence Livermore National Laboratory (LLNL). This document is required in compliance with the QA procedure on *Software Management*, AP-SI.1Q, Rev. 3/ICN 4. EQ3/6 Version 8.0 is a software package utilized to perform geochemical modeling computations encompassing fluid-mineral interactions and/or solution-mineral-equilibria in aqueous systems. The software package is composed of two major components: EQ3NR, a speciation-solubility code; EQ6, a reaction path modeling code to simulate water/rock interaction or fluid mixing in either a pure reaction progress mode or a time-dependent or kinetic mode. Supporting software includes EQPT, a data file preprocessor, along with several supporting thermodynamic data files, as well as file converter programs to update old input files. The software deals with the concepts of thermodynamic equilibrium, irreversible mass transfer, and reaction kinetics. The supporting thermodynamic data files contain both standard state and activity coefficient-related data that allows the use of activity models such as Davies, B-dot, or Pitzer's equations for determination of activity coefficients.

The scope of this software is to support various interdisciplinary efforts of the Yucca Mountain Project (YMP) in the modeling of the physical and chemical environment related to Engineered Barrier Systems (EBS). This includes modeling alteration of Waste Package (WP) materials, Waste Form (WF) degradation, criticality issues related to radionuclide accumulations, and other interrelated areas such as the Unsaturated Zone (UZ).

2. USER INTERACTIONS

Interactions with the software begin with the activation (double clicking) of the shortcut link that will initiate a "command prompt" window and execute a setup script. Instructions detailing code installation and startup procedures are given in the *Installation Test Plan EQ3/6 Version 8.0* (ITP) (Jarek 2002; SDN: 10813-ITP-8.0-00). The user should see the success of this script indicated thusly:

```
Configuring to run EQ3/6 version 8.0
Done
```

If errors are reported, something is wrong with the shortcut file, the configuration batch file (eq36cfg.bat), or some other aspect of the software installation. Refer to the ITP document (Jarek 2002; SDN: 10813-ITP-8.0-00) to correct any problems. It is important to use the appropriate shortcut for the type of Windows OS used and to have the correct data in the shortcut's "Properties" (see Section 4.1 of the ITP). Other configuration data are present in the configuration batch file; these must also be correct (see Section 4.3 of the ITP).

EQ3/6 problems are run in the resulting command prompt window. In order to run EQ3NR or EQ6, the user must specify one or more problems on an input file and choose an appropriate supporting data file. Problems may be "stacked" on an input file by simply following one problem specification with another. In order to create or modify an input file, some editing program must be used. None is provided with EQ3/6 itself. Such an editing program may work either inside or outside the command prompt window. WordPad (which operates outside this window) and is included in the Windows OS is one example of such a program. To facilitate the construction of input files, many examples are included with the software. Finding one that serves as a good template is the recommended approach to preparing the user's own input files. The details of input file form and content for both EQ3NR and EQ6 are described later in this manual. EQ3NR input files use the filename extension ".3i"; EQ6 input files use ".6i".

Any data file used by EQ3/6 exists in two parallel forms: an ASCII formatted form known as a "data0" file (which can be created or modified by a text editor) and a parallel unformatted form known as a "data1" file. The latter is sometimes referred to as a "binary" file; however, it is technically merely unformatted. The two forms of a data file share a three-letter "datafilekey," which serves as the filename extension for each (e.g., ymp for data0.ymp and data1.ymp). A data1 file is created from the corresponding data0 file by running the EQPT program (described further in this section). In addition to creating an unformatted equivalent data file, EQPT performs a large number of checks. For example, reactions are checked for mass and charge balance, and Pitzer interaction coefficients are checked for duplicate and invalid species combinations.

The EQ3NR and EQ6 codes actually use the data1 files, not the data0 files. As the EQ3/6 software is distributed, the data1 files are included and the user need take no action with regard to these unless the data0 files are modified or new data0 files are created. If so, the EQPT code must be run on those data0 files to replace or create the corresponding data1 files. The form and content of data0 files is little changed from that for the Version 7 code. Version 7 level data0 files can be used unchanged with the Version 8 code. However, they must be run through the Version 8 EQPT code to be used with the Version 8 EQ3NR and EQ6, as the data1 file structure is slightly different for the two version levels. Limited differences in the possible form and content of data0 files between the two versions (related to new Version 8 capabilities) will be discussed later in this manual.

NOTE: The data0 and data1 files distributed with EQ3/6 Version 8.0 software are unqualified. Any YMP user needs to obtain the most recent qualified and controlled data0 file(s) from the Automated Technical Data Tracking (ATDT) System for producing quality-affecting work products. The most recent of these are data0.ymp.R2 (DTN: MO0210SPATHDYN.000) and the Pitzer database file data0.ypf.R0 (DTN: SN0210T0510102.001).

All of the codes in the EQ3/6 package (excluding the command prompt shortcuts) are run in the command prompt window. With proper shortcut configuration, a configuration batch file runs automatically when the command prompt window opens. Note that the EQ3/6 software can only be run in a window opened by clicking on the appropriate EQ3/6 command prompt shortcut. An ordinary command prompt window will not provide the properly configured environment.

In common usage, the principal codes in the EQ3/6 package are not run directly but through smaller interface codes. For example, RUNEQPT runs EQPT, RUNEQ3 runs EQ3NR, and RUNEQ6 runs EQ6. The interface codes contain OS-dependent coding (which is the same for all versions of Windows). One of their key functions is to simplify the handling of file names. They permit output files to be automatically named after the corresponding input files.

The EQPT code is run by entering a command of the form:

```
>runeqpt datafilekey(s)
```

where argument(s) following “runeqpt” consist of one or more three-letter data file keys (e.g., cmp, ymp, ypf). This command is run in the database (“db”) directory described in the ITP (Jarek 2002). Normally, this directory is c:\eq3_6v8.0\db. Thus, entering “runeqpt ymp” specifies that data1.ymp is to be created using data0.ymp as input. If an error is encountered while processing a data0 file, RUNEQPT will delete the corresponding data1 file so that it cannot be used. In addition to the data1 file, EQPT also writes an output file and an “slist” file. Both files (which are ASCII) contain listings of the species on the data0 file.

The EQ3NR and EQ6 codes are respectively run by commands of the form:

```
>runeq3 datafilekey inputfile(s)

>runeq6 datafilekey inputfile(s),
```

respectively. The *datafilekey* is the usual three-letter data file key discussed above. Note that only one data file may be used per run. Failure to specify a valid key (one for which the corresponding data1 file exists in the database directory) is an error. The *inputfile(s)* are one or more EQ3NR (for runeq3) or EQ6 (for runeq6) input files. Specific examples for running EQ3NR include “runeq3 cmp j13ww.3i” and “runeq3 ymp *.3i”. Examples for EQ6 include “runeq6 cmp micro.6i” and “runeq6 ymp *.6i”.

All EQ3NR and EQ6 output files are ASCII files. EQ3NR writes two output files, the normal output (“.3o”) file and a pickup (“.3p”) file. The EQ3NR pickup file is used to communicate data to EQ6. It is either a complete EQ6 input file or the bottom half of one. EQ6 writes several output files, a normal output (“.6o”) file, a pickup (“.6p) file, a summary output or “tab” (“.6t”) file, and a scrambled tab (“.6tx”) file. The EQ6 pickup file is always a complete EQ6 input file, and can be used to extend a previous run. The tab file provides a more succinct summary than the output file and may be useful in plotting data. The scrambled tab file is used to produce a tab file that extends over more than one run. All these output files are described in more detail later in this manual under Sections 3.5 and 3.6.

The format and content of EQ3NR and EQ6 input files has changed for Version 8. The new format and content are described later in this manual. The twin format styles (W, compact, and D, menu-style) have been retained for version 8; however, only the latter is recommended. Version level 6, 7.0, and 7.2 input files may be converted to Version 8 level using software in the Version 8 package. EQ3NR input files may be converted using XCIF3, EQ6 input files using XCIF6. XCIF3 and XCIF6 are interface codes similar to RUNEQPT, RUNEQ3, and RUNEQ6. XCIF3 runs a code called XCON3, XCIF6 runs one called XCON6. Usage of these codes is described in Section 2.3.

An overview of the input/output flow into the different program modules is presented in Figure 1.

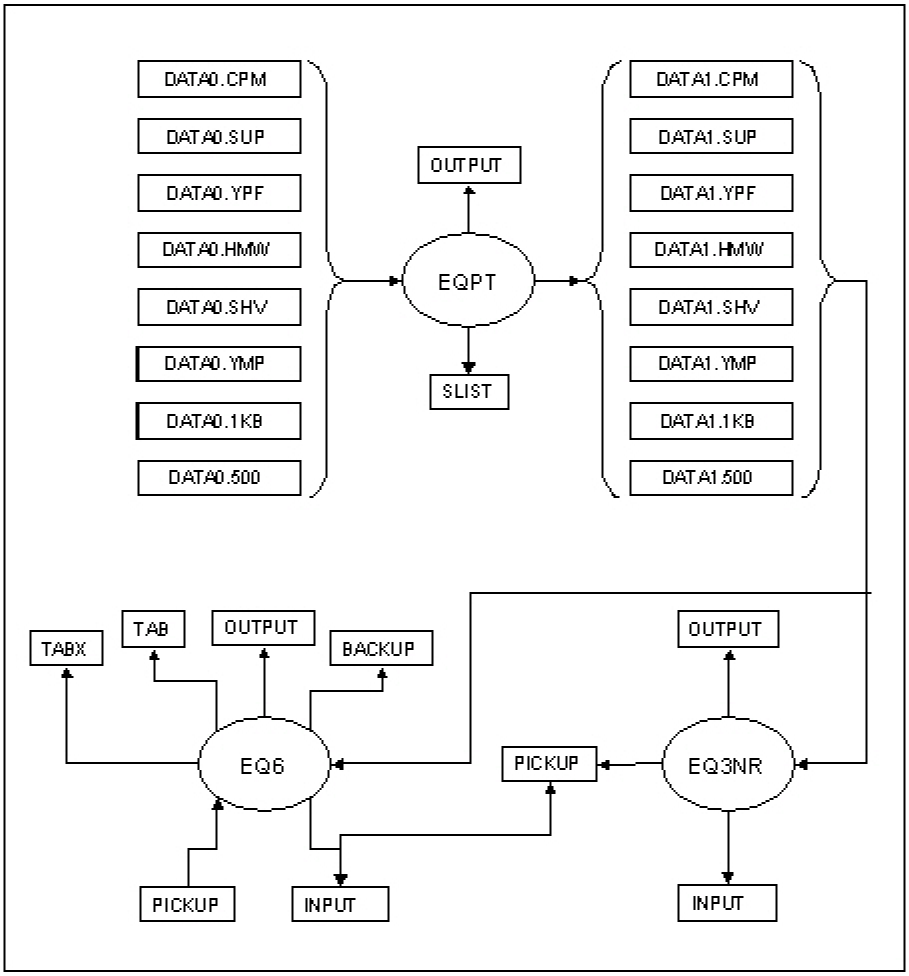

Figure 1. Simplified flowchart of *EQ3/6* Version 8.0 showing the association between inputs/output and database elements and the main codes. (modified after Wolery, 1992a)

2.1 CONCEPTUAL BACKGROUND

Users are expected to be familiar with the necessary concepts employed in geochemical modeling of aqueous systems. Users of the version 7 series EQ3/6 (Versions 7.0 through 7.2c) will likely have the necessary background. Others should read the next subsection (2.1.1) of this manual for an introduction to speciation-solubility modeling of aqueous systems. Standard aqueous geochemistry texts and monographs may also be helpful. Particularly recommended are: Nordstrom and Munoz (1985); Stumm and Morgan (1996); and Langmuir (1997).

2.1.1 Basis Species: Key Concepts

Users need a thorough understanding of the concept of basis species, and can refer to this section's subsections for details. In general, each strict basis species except the fictive $O_2(g)$ represents a chemical element, the fictive $O_2(g)$ itself serving as the redox species (which can be conceptually tied to charge balance as each of the other strict basis species can be so tied to a mass balance). A strict basis species may be comprised of only the associated chemical element plus oxygen and hydrogen. Basis species typically are simple [e.g., Na^+, K^+, H^+, Ca^{2+}, Mg^{2+}, Cl^-, F^-, SO_4^{2-}, HCO_3^-, $SiO_2(aq)$, $B(OH)_3(aq)$]. In general, H_2O is the basis species associated with oxygen, H^+ that associated with hydrogen.

An auxiliary basis species usually represents a chemical element in a different oxidation state (e.g., Fe^{3+} versus Fe^{2+}), though that is not essential. Usually a species is defined as an auxiliary basis species because there is some expectation of possible disequilibria with the corresponding strict basis species, or because analytical methods exist that permit discrimination between the two. The user may treat an auxiliary basis species in one of two ways. It can be used as an active basis species by treating it as such on the input file (e.g., by specifying a concentration for it). Alternatively, it can be treated as a dependent (non-basis) species. This is the default condition (when no corresponding concentration or alternative constraint is applied on the input file).

To know what are the basis species on a given supporting data file, consult the "slist" file produced by the EQPT data file preprocessor (discussed below), or directly examine the relevant "data0" form of the supporting data file. The strict basis species comprise the first group of aqueous species, ending with the fictive aqueous species $O_2(g)$. The auxiliary basis species follow. After that come the dependent (non-basis) species.

2.1.1.1 Basis Species

In the EQ3/6 system, there is a set of master or basis species. This concept revolves around the notion that one such species is associated with each chemical element and its associated mass balance (e.g., Na^+ for Na). If oxidation-reduction is considered, one additional species such as $O_2(g)$ for e^- must be added, which is associated with charge balance. Every remaining species (aqueous, mineral, or gas) is formally associated with a reaction that destroys it. For example, an aqueous complex is paired with its dissociation reaction, and a mineral with its dissolution reaction. The basis species are used as a set of generalized "building blocks" in writing chemical reactions. The reactions are then written in terms of only the single associated species and the set of basis aqueous species.

To know what are the basis species on a given supporting data file, consult the "slist" file produced by the EQPT data file preprocessor (discussed below), or directly examine the relevant "data0" form of the supporting data file. The strict basis species comprise the first group of aqueous species, ending with the fictive aqueous species $O_2(g)$. The auxiliary basis species follow. After that come the dependent (non-basis) species.

We will call a basis set as defined above a *strict basis* set. It is the minimal basis set required for chemical modeling. The number of species in this set, in the general case including a redox species, is given by:

$$s_B = \varepsilon_T + 1 \tag{2–1}$$

where ε_T is the number of chemical elements in the system of interest. The redox species itself will be denoted as the s_Bth species. In the case of systems in which there is no oxidation-reduction, we will simply treat the redox species as being inactive.

A speciation-solubility problem concerning an aqueous solution deals only with mass balances involving species in one (aqueous) phase. Therefore, the basis set in EQ3NR consists entirely of aqueous species. These are defined (at least initially) on the supporting data file. One is $H_2O(l)$, the solvent. The redox species used in EQ3/6 is $O_2(g)$, which is treated in this context as a fictive aqueous species; the conventional e^- used by some other modeling codes is another example of such. The other basis species are simple species likely to dominate their respective mass balance relationships, at least in many instances.

Basis species are usually chosen as mono-elemental species such as Na^+ and Ca^{2+}. Some are also comprised of oxygen and/or hydrogen (e.g., SO_4^{2-} and $B(OH)_3(aq)$). No basis species on a supporting data file is permitted to be comprised of more than one chemical element other than oxygen or hydrogen. The purpose of this restriction is to avoid certain problems that would otherwise arise in defining the total concentrations of the basis species. Such problems do not arise in the case of dealing with elemental oxygen and hydrogen because no meaningful analytical values exist for the total concentrations of the associated basis species, $H_2O(l)$ and H^+ or of these elements themselves. The concentration of water as measured by its mole fraction is implicitly fixed by the concentrations of the solute components. The concentration of the hydrogen ion is analytically determined via the pH or some other approach not involving a total concentration.

Using a strict basis set, all mass balance relationships can be defined in terms of chemical elements and the coefficients describing the elemental compositions of all species. The charge balance relationship can be defined in terms of the electrical charges of the species. All non-basis species appearing in these balance equations are related to the basis species via the associated chemical reactions. The concentrations of these non-basis species are then determined by the concentrations of the basis species through the associated mass action equations, assuming that the activity coefficients appearing in these equations are known. Thus, if the concentrations of the basis species are known, they may be used to span (compute) the complete speciation of the system. In mathematics, a set with such properties is usually called a basis, which is where the term *basis set* in the present context is derived.

In thermodynamic modeling, one deals in an algebraic sense with *n* equations in *n* unknowns. The use of a basis set which is strict requires assuming that the concentration of every non-basis species appearing in a balance equation satisfies a corresponding mass action equation. This has the effect of requiring the modeled system to be in a complete state of internal chemical equilibrium. There is simply no mechanism in this construction to deal with even one simple reaction in a state of disequilibria. The concept of internal equilibrium as used here refers to a system that excludes any non-basis species that do not appear in the balance equations. Thus, an aqueous solution may be in a state of internal equilibrium, but still supersaturated with respect to calcite. The mineral in this context is a non-basis species, but it does not appear in the balance equations that describe the aqueous solution. The system consisting of the same aqueous solution plus the mineral, however, is not in a state of internal equilibrium.

In EQ3NR, the modeled system consists exclusively of the aqueous solution. Systems including other phases are treated in EQ6. Nevertheless, it is apparent from the above example that in order to model systems with some internal disequilibria, one must expand the basis set beyond the confines of the strict basis. In the example given above, we would make the mineral a basis species. However, since the associated reaction is presumed to be in disequilibria, the associated mass action equation is not used as a governing constraint. In order to maintain a balance of *n* equations in *n* unknowns, it is necessary to introduce a new mass balance equation for the new basis species. In the present example, this is just a statement of how much of the mineral is present in the system. Note that this is a new kind of mass balance relation not related to a chemical element.

The same principle holds in modeling an aqueous solution. One might wish to compute a model in which Fe^{3+} is not in equilibrium with Fe^{2+}. If Fe^{2+} is already in the strict basis set, one must add Fe^{3+} to the basis set (or vice versa). In this case, the situation is more complicated, as the new basis species may have its own ion pairs and complexes appearing in the associated mass balance. This is a simple concept. However, it requires rethinking the description of mass balances, as the number of mass balance equations now exceeds the number of chemical elements. As we will show, a better concept is to associate the mass balance relations with corresponding basis species, not with the chemical elements.

We now show how to develop this more generalized concept for defining mass balance relations. Consider the following reaction:

$$HgCl_3^- = Hg^{2+} + 3Cl^- \tag{2–2}$$

This is represented in the software by paired arrays of reaction coefficients (floating point numbers) and names of the corresponding species (character variables). If you ask the question, how many chlorides is the non-basis species on the left hand side equivalent to, a human being will invariably answer the question by looking at the subscript "3" in the chemical formula of the species. In the software, this is equivalent to looking up the elemental composition of the species in the appropriate data array. Thus, this mechanism produces the required coefficient for evaluating the contribution of this species to a mass balance relation based on a chemical element. This is really the answer to the question, to how many chlorines (not chlorides) is this species equivalent. This is not what is presently desired, and a different approach is required.

The original question can be more accurately answered by looking at the reaction coefficients. Since reactions must satisfy mass and charge balance to be valid, the required information must be available there. A human being would probably answer the question by looking at the coefficient on the right hand side of the reaction. However, the coefficient of the non-basis species on the left hand side must also be considered to obtain the correct answer in the general case. To emphasize this, we note that the reaction can also be written as:

$$2HgCl_3^- = 2Hg^{2+} + 6Cl^- \tag{2–3}$$

Although a reaction whose coefficients have not been reduced to the lowest common denominator is unlikely to be written in any of the EQ3/6 data files, it is not prohibited. In addition, some reactions written with a unit coefficient for the associated non-basis species require fractional coefficients. An example is:

$$Fe^{3+} + \tfrac{1}{2}\, H_2O(l) = Fe^{2+} + H^+ + \tfrac{1}{4}\, O_2(g) \tag{2–4}$$

One might reasonably wish to avoid the fractions and write instead:

$$4Fe^{3+} + 2H_2O(l) = 4Fe^{2+} + 4H^+ + O_2(g) \tag{2–5}$$

Furthermore, certain actions taken by the code as it executes, such as basis switching, may cause a reaction to be rewritten, and there is no general restriction requiring the new reaction to have a unit coefficient for the associated non-basis species.

In the software, the coefficients of products are defined as positive numbers and those of reactants as negative ones. For the basis species, these coefficients are symbolized by $b_{s'r}$, where s' denotes a basis species and r the reaction. The non-basis species associated with the r^{th} aqueous reaction is denoted by s'', and its reaction coefficient is symbolized by $b_{s''r}$. Thus, the factor giving the stoichiometric equivalence of such a species to the s'^{th} basis species is given by:

$$u_{s''s'} = -\frac{b_{s'r}}{b_{s''r}} \tag{2–6}$$

In a speciation-solubility problem, the mass balance equation for the s'^{th} basis species is then:

$$m_{T,s'} = m_{s'} + \sum_{r=1}^{r_T} u_{s''s'} m_{s''} \tag{2–7}$$

where r is the reaction associated with the $s''^{\text{ th}}$ species ($r = s'' - s_B$; see Section 9.2) and r_T is the number of reactions for the dissociation of non-basis aqueous species. Considerable care must be used in the application of such a formulation. Mathematically, it is quite rigorous. Physically, however, there are some potential problems. The quantity on the left hand side may or may not correspond to something that can be obtained by chemical analysis and therefore have physical as well as mathematical meaning. The formulation can be applied to any basis species. In the case of $O_2(g)$ or e^-, these have no physical meaning, as these are only hypothetical aqueous species in the first place. In the case of H^+, the total concentration has no physical significance. Its value is uniquely established only because one normally chooses to put this species in the basis set instead of OH^-. In the case of $H_2O(l)$, the computed total concentration is also technically non-physical and depends on which of H^+ or OH^- is chosen as a basis species.

It was pointed out earlier that basis species on the EQ3/6 data files are restricted in composition to no more than one chemical element other that oxygen and hydrogen. This is done to protect the physical meaning of the total concentrations of basis species other than $H_2O(l)$, H^+, and $O_2(g)$, for which there is no possibility of physical meaning, anyway. To illustrate the problem, consider the following three reactions in which $HgCl_3^-$ is used as a basis species in place of Hg^{2+}:

$$Hg^{2+} + 3Cl^- = HgCl_3^- \tag{2–8}$$

$$HgBr_3^- + 3Cl^- = HgCl_3^- + 3Br^- \tag{2–9}$$

Consider the mass balance of chloride and the contribution to it from Hg^{2+}. Applying eq (2–6) to the first reaction above gives a stoichiometric coefficient of –3. The same result is obtained for the bromide complex in the second reaction. The chloride complex itself has a stoichiometric coefficient of zero.

In EQ3NR and EQ6, the chloride complex in the above example is likely to strongly dominate the mass balance of dissolved mercury, giving an incentive to consider switching it into the basis in place of the mercuric ion. The codes deal with this situation by continuing to define the stoichiometric factors appearing in the mass balance relations in terms the reactions as they were written prior to basis switching, modified only for stoichiometric factors relating the new basis species to the old ones.

2.1.1.2 Organization and Treatment of Basis Species

The set of basis species on an EQ3/6 data file is divided into two parts: the strict basis and the *auxiliary* basis. The species in the strict basis set correspond one-to-one with the chemical elements, except for O_2(g), which is used as a hypothetical aqueous species, and which corresponds to charge balance. These species appear first in the overall list of aqueous species. The solvent, H_2O(l), is the first aqueous species. The hypothetical aqueous species O_2(g) is the last. The species in the strict basis set are not associated with any reactions, as are all other species.

The auxiliary basis species follow the strict basis species. For the most part, they represent chemical elements in different oxidation states. However, they may also include any species that do not readily equilibrate with other basis species according to the associated reactions. Auxiliary basis species are used like strict basis species as “building blocks” in writing reactions for various species on the data file. In EQ3NR, an auxiliary basis species may be treated as either a basis species or a non-basis species. The choice is up to the user in each case. By default, an auxiliary basis species is eliminated from the active basis set (except for O_2(aq) and H_2(aq), which are special cases). Any reactions for other species written in terms of this species will be rewritten to reflect this. However, an auxiliary basis species is treated as an active basis species if the user provides an appropriate matching input on the input file, such as a total concentration. The non-basis aqueous species follow the auxiliary basis species. In the present version of EQ3/6, a species defined as a non-basis species on the data file cannot be treated as an active basis species unless it is switched with an existing member of this set. This prevents defining an additional mass balance relation for this species. If it is desired to use such a species in the active basis set for the purpose of defining an additional such relation, it is necessary to modify the data file, moving the species into the auxiliary basis set.

An input model constraint, such as a total concentration, is required for each master aqueous species in order to perform a speciation-solubility calculation. However, as discussed in Section 3.3.9, there are limitations on the constraints that can be placed on a given basis species, depending on whether it is in the strict basis or the auxiliary basis. The user needs to be keenly aware of which species are in which set. Users should consult either the relevant **data0** data file or the **slist** (species list) file written by EQPT when it processes this file.

The user may specify selected examples of basis switching on the input file. This provides a means of changing the set of basis species at run time. For example, a strict basis species may be exchanged with an auxiliary basis species. This puts the latter in the strict basis, the former in the auxiliary set. A basis species may also be switched with a non-basis species. A basis switch causes reactions to be re-written in terms of the new basis set. Judicious basis switching can improve the code numerics, sometimes making the difference in whether the iteration converges or not. As a rule, a basis species should not make up an insignificant part of the associated mass balance when a total concentration is used as the input constraint. The compositional restrictions on basis species on the EQ3/6 data file do not apply to basis switching made when running EQ3NR or EQ6.

A basis switch involving a non-basis species causes the corresponding input constraints (see Section 3.3.9), to be reassigned to the species brought into the basis set. In a switch involving a basis species constrained to satisfy a specified total concentration, the total concentration is recomputed by a stoichiometric adjustment to match the species moved into the basis. In this case, the problem itself is not really changed; it is merely expressed in different terms.

Setting the auxiliary basis species to “suppress” causes the species to be eliminated from the active basis set. As noted above, this is the default condition for most such species. Elimination from the active basis set causes reactions originally written in terms of the eliminated species to be rewritten. For example, consider the following reaction:

$$\mathrm{FeSO_4^{+} = Fe^{3+} + SO_4^{2-}} \qquad (2\text{–}10)$$

Elimination of Fe^{3+} from the basis set causes this to be rewritten as:

$$\mathrm{FeSO_4^{+} + ½\,H_2O(l) = Fe^{2+} + H^{+} + ¼\,O_2(g) + SO_4^{2-}} \qquad (2\text{–}11)$$

Thus, $FeSO_4^+$ now appears to be, and is treated as, a complex of Fe^{2+}. Elimination thus has the effect of combining mass balances. In this case, Fe^{3+} and its complexes are folded into the mass balance for Fe^{2+}. Note that the reaction for $FeSO_4^+$ on the data file must be the first, not the second, of the two above reactions. Otherwise, $FeSO_4^+$ would have been considered a complex of Fe^{2+} from the start. If one did not eliminate Fe^{3+} from the active basis set, $FeSO_4^+$ would be incorrectly folded in to the mass balance for Fe^{2+}, and missing from that of Fe^{3+}.

In the majority of cases, there are only one or two auxiliary basis species for a given chemical element. When there is more than one such auxiliary basis species, the species can be “chained.” The first such species must be related through its associated reaction to the corresponding strict basis species. This can also be done in the case of the second, third, etc., such basis species. However, the second such auxiliary basis species could be directly related to the first such auxiliary basis species. The third could be directly related to the first or second, etc. Of course, a non-basis species can be directly related to any basis species.

Chaining is not significant in the case of most chemical elements. Carbon, however, is an exception. A large number of organic species are now present on the **com** and **sup** data files because of the inclusion of such in SUPCRT92 from the work of Shock and Helgeson (1990). Several of these are treated as auxiliary basis species, the majority as non-basis species of which all are directly related to one of the organic species in the auxiliary basis. The basic problem with organics in a geochemical modeling code is that they may often be treated as complexes of HCO_3^- when this is not what is desired. In the “R7” versions of the **com** and **sup** data files, each of the several organic species in the auxiliary basis set is directly related to HCO_3^-. In order to model organic-free systems without the unexpected appearance of organics in the model, it is necessary to enter on the **input** file a zero concentration for each organic species in the auxiliary basis set. In future versions of the data files, one such species (probably 'acetic acid(aq)') may be set up as a sort of master organic species. Only this organic species will be directly related to HCO_3^-. All other organic species in the auxiliary basis set will be directly related to this master organic species. Then all organics can be kept out of a computed model by entering on the **input** file a zero concentration for just the master organic species.

2.1.1.3 Stoichiometric Conversions of Analytical Data

The analytical data used to define speciation-solubility problems in EQ3NR pertain to the basis species on the supporting data file employed in a given run. To use the code correctly, one must know what the species are, and it is often necessary to correct the analytical data one is provided in order to provide a stoichiometric match.

For example, the river water test case of Nordstrom et al. (1979) includes the following data:

- Si 8.52 mg/L.
- B 0.050 mg/L.
- PO_4^{3-} 0.210 mg/L.

The species on the **data0.com** data file which respectively correspond to these components are SiO_2(aq), $B(OH)_3$(aq), and HPO_4^{2-}. The problem is that, for example, 8.52 mg/L of Si is not equivalent to 8.52 mg/L of SiO_2(aq). A stoichiometric conversion, of the sort common in analytical chemistry, must be done. This is illustrated in the present case by:

$$C_{mg/L,SiO_2(aq)} = \frac{M_{w,SiO_2(aq)}}{M_{w,Si}} C_{mg/L,Si} \tag{2–12}$$

where $M_{w,i}$ is the molecular weight of the component labeled i. The atomic weight of Si is 28.086 g/mole and that of O is 15.999g /mole. The molecular weight of SiO_2(aq) is therefore 60.0840 g/mole. The ratio of the molecular weights is therefore 2.1393, and the 8.52 mg/L of Si is therefore equivalent to 18.23 mg/L of SiO_2(aq). By following this method, it can be shown that the 0.050 mg/L of B is equivalent to 0.286 mg/L of $B(OH)_3$(aq), and that the 0.210 mg/L of PO_4^{3-} is equivalent to 0.212 mg/l of HPO_4^{2-}. Corrections are analogous if the data are concentrations in mg/kg of solution.

The situation is much simpler if the analytical data are reported as molalities or molarities, as no conversion is generally necessary. For example, 0.0001 molal Si is equivalent to 0.0001 molal SiO_2(aq).

The code user must make any necessary stoichiometric conversions before entering the data on the **input** file. EQ3NR contains no provisions for direct input of data corresponding to dissolved components other than the basis species appearing on the data file used, so it is not possible for it to make these conversions for the user.

2.2 OPERATING THE SOFTWARE

The EQ3NR code is normally run using a command of the form:

```
>runeq3 datafilekey inputfilename(s)
```

where the *datafilekey* is a three-character key associated with a particular supporting data file. The associated ASCII data file has a name of the form data0.*datafilekey*. Some of the most relevant data file keys are:

cmp - “Composite: (non-Pitzer) data file

hmw - Harvie-Moller-Weare (1984) data file

ymp - Yucca Mountain Project non-Pitzer data file (obtain from ATDT)

ypf - Yucca Mountain Project Pitzer data file (obtain from ATDT)

Thus, the test case library input file swmaj.3i, which is designed to run with the composite data file, would be run by the command:

```
>runeq3 cmp swmaj.3i
```

The species named on swmaj.3i must therefore match those on the data0.cmp data file. Furthermore, the species so named also have to be of the proper type. Any confusion typically centers on which species are basis species, and if so, whether strict or auxiliary. The EQ6 code is typically run by entering an analogous command; e.g.,

```
>runeq6 cmp microhcl.6i
```

The *datafilekey* directs runeq3 or runeq6 to the corresponding “data1” file, such as data1.cmp. The “data1” file is a binary form of the corresponding “data0” file that is created by running the EQPT data file preprocessor on the “data0” file. EQPT is normally run by a command of the form:

```
>runeqpt datafilekey(s)
```

or

```
>runeqpt all
```

(which runs EQPT on all of the data0 files in the current directory).

The EQPT preprocessor produces in addition to a "data1" file an output file of the form output.*datafilekey* and the species list or "slist" file of the form slist.*datafilekey*. The results of running EQPT on the various common data files are normally included in the EQ3/6 software as distributed. Therefore, it is often unnecessary for many users to run EQPT in the first place. See the ITP document (Jarek 2002) for more details on the use of EQPT. Descriptions of each database are given in their respective "data0" ASCII file headers.

2.3 INPUT FILE CONVERSION

The use of the XCIF3 and XCIF6 input file converters will be briefly described here. Instructions are also given in the ITP document (Jarek 2002). The command form for the former is:

```
>xcif3 oldversion newversion formatletter filename(s)
```

For example, the Version 7.2 level (e.g., for EQ3/6 7.2, 7.2a, 7.2b, or 7.2c) input file swmaj.3i (in either "W" or "D" format) could be converted to Version 8 level "D" format by entering the command:

```
>xcif3 7.2 8 D swmaj.3i
```

The XCIF6 converter operates similarly on EQ3/6 input files. The analogous conversion is achieved by the command:

```
>xcif6 oldversion newversion formatletter filename(s)
```

A specific example of conversion to Version 8 level "D" format is given by:

```
>xcif6 7.2 8 D evapsw.6i
```

where evapsw.6i is presumed to initially be at Version 7.2 level (in either "W" or "D" format). Such conversions (forward and backward) may be made among Version levels 6, 7, 7.2, and 8 (note: there is no "D" format for Version 6 level input files).

The XCIF3 and XCIF6 converters provide a path for existing Version 7.2 level users to migrate their existing input files to Version 8 level. Such users could also create new Version 8 level input files by first making them up at the version 7.2 level and then using the converters to move them to Version 8 level. This approach is only recommended as a temporary solution. Some existing code users may find this exercise helpful in learning about the Version 8 level files.

2.4 CONSTRUCTING AND EDITING INPUT FILES

Any text editor may be used to modify and construct EQ3/6 input files. These files are ordinary text files (equivalent to files with the usual "txt" filename extension). Among the possible text editors are WordPad and Notepad, which are typically included in a Windows operating system. Of these two, WordPad is probably more suitable.

Users may wish to create filename associations so that when icons for “.3i” and “.6i” files are clicked on, the relevant files are opened in WordPad (or another Windows-based editor). Such an association for an unassociated extension can be created by double-clicking on a file of the relevant type. A Windows dialog box will open and allow the user to choose an application to open the file (find WordPad or another standard text editor and select it). The checkbox to create a permanent association (“Always use this program to open this type of file”) will be normally be checked by default. Click “Okay”; the association will be made and the file will open. Close the file. The next time a file of this type is double-clicked, it will automatically open in WordPad (or another editor if that was chosen instead).

In general, the user should not attempt to create input files from scratch. Rather, he or she should examine the test case library or other set of pre-existing input files to find an example that is close to the desired problem. That example should be copied, the copy renamed to that for the desired input file, and its contents edited to specify the desired problem. In some cases, it may be necessary to copy some element (block of lines) from a second or even a third pre-existing file in order to construct the desired file. However, that is rarely necessary.

EQ3NR input files may be stacked, such that one problem directly follows another on the same file. This is recommended only in the case of two such problems when the aim is to create an EQ6 input file for fluid-mixing using the appropriate advanced PICKUP file option [iopt(19), Option 3].

2.4.1 About the Menu-Style (“D”) Format

The menu-style input file format uses pipe signs (“|”) to define fields on each line. Typically, the first such sign is in column one and the last one in column 80. The first field on a line starts immediately after the first pipe sign. Each line contains at least one field. Lines are in turn organized into blocks. Each block represents some coherent piece of input, similar to a dialogue window. A simple example from the EQ3NR input file is:

```
|------------------------------------------------------------------------------|
|Temperature (C)         | 2.50000E+01| (tempc)                                |
|------------------------------------------------------------------------------|
```

The purpose here is to specify the temperature (Celsius). The temperature line, here shown surrounded by two separator lines, contains three fields. The first field is a label describing the associated input, the second is for the input value, and the third is another label noting the associated internal code variable. If this block were to be replaced by a dialogue window, the two labels would appear as labels, and an input box would replace the input field. A number in an input field need not follow any special format, as long as the form of the number itself is valid under the rules of Fortran. In particular, the number need not occupy a specific position in the input field, and the number of decimal places, if any, is not fixed. Thus, the following examples are equivalent to the one above:

```
|------------------------------------------------------------------------------|
|Temperature (C)         |  25.0      | (tempc)                                |
|------------------------------------------------------------------------------|

|------------------------------------------------------------------------------|
|Temperature (C)         |     25.    | (tempc)                                |
|------------------------------------------------------------------------------|
```

```
|------------------------------------------------------------------------------|
|Temperature (C)         | 2.50E+01 | (tempc)                                 |
|------------------------------------------------------------------------------|
```

The last example is used to show that the pipe signs themselves are not required to be in fixed positions. An input file can be "prettified" (including putting the pipe signs in optimal places for best appearance) by running it through XCIF3 (EQ3NR input files) or XCIF6 (EQ6 input files).

A slightly more complex example of an input line block mimics a "radio button" dialog box:

```
|------------------------------------------------------------------------------|
|iopr(4) - Print a Table of Aqueous Species Concentrations, Activities, etc.:  |
|  [ ] (-3) Omit species with molalities < 1.e-8                               |
|  [ ] (-2) Omit species with molalities < 1.e-12                              |
|  [ ] (-1) Omit species with molalities < 1.e-20                              |
|  [x] ( 0) Omit species with molalities < 1.e-100                             |
|  [ ] ( 1) Include all species                                                |
|------------------------------------------------------------------------------|
```

Here the user selects one of a number (here five choices for a specified option. In this example, this is an output file print option (the "iopr(4)" option, common to both EQ3NR and EQ6) for a table of aqueous species concentrations, activities, and so forth. The option allows the user to control the size of the table by ignoring minor species at any of four pre-selected concentration levels. In a block of this type, the "[ ]" functions as a radio button, being turned on when an "x" appears inside, off when a space is present. An "X" or "*" can be substituted in place of an "x". The default option in such a case is always marked by the "(0") label, which corresponds to a zero value for the associated code variable [here iopr(4)]. If no option is selected, the default applies. If two options are selected, the one with the higher value for the associated code variable applies. If an illegal character (anything other than an "x", "X", or "*") is used to select an option, the code will flag an error condition.

A comment line may appear anywhere in a Version 8 input file. Such a line is characterized by an asterisk ("*") in column 1. A comment line may be used for any purpose. The examples from the test case library all contain "standard" comment lines listing any necessary string inputs, apart from species names. Examples will be discussed below as they are encountered. Here is an example:

```
* Valid alter/suppress strings (ukxm(kxmod(n))) are:                           *
*    Suppress            Replace             AugmentLogK                       *
*    AugmentG                                                                  *
*------------------------------------------------------------------------------*
```

The XCIF3 and XCIF6 input file converters might be used to "prettify" input files. For example, if "abc.3i" is an input file already at version 8 level and in "D" format, the command:

```
>xcif3 8 8 D abc.3i
```

will put the pipe signs in "standard" positions and insure the presence of "standard" comment lines that give the string inputs (apart from species names) that may be needed for use in corresponding input fields.

2.5 CONSTRAINTS AND/OR SPECIAL INSTRUCTIONS

There is a constraint that this software can only run from a "command prompt" or DOS window as described above, where the user must enter commands from the keyboard. On-screen monitoring of the code simulations can be tracked while software is running a problem until completion. To stop software execution at any time, press Ctrl-C to break the process.

3. INPUT/OUTPUT OPTIONS

3.1 EQ3/6 INPUT DATABASE FILES

As was noted, one of several possible supporting data files must be specified in order to run EQ3NR or EQ6. Each data file has two forms, the ASCII data0 form and the corresponding unformatted ("binary") data1 form. A data1 file (e.g., data1.ymp) is created by running the EQPT code on the corresponding data0 file (e.g., data0.ymp). The latter is created by running EQPT on the former. When running EQ3NR or EQ6 with RUNEQ3 or RUNEQ6, respectively, the user specifies the data1 file to use by means of the corresponding three-letter key that also serves as the filename extension of that data1 file.

The user must choose a supporting data file that is consistent with the input file(s) to be run. In general, there are two elements of consistency. The first is that the data file and the input file(s) must correspond to the same category of activity coefficient model (e.g., Pitzer vs. non-Pitzer). RUNEQ3 and RUNEQ6 will screen out any mismatches here. The second element of consistency is that the supporting data file must contain the species referenced on the input file(s). In particular, the names must match. This can be a problem in that there is no universally accepted set of names. Thus, the undissociated aqueous silica species may be "SiO2(aq)" on one data file, and "H4SiO4(aq)" on another. Thus, the code user must have some familiarity with the species names used on the data file to be used. The most convenient source to consult is often the output or "slist" file that was created when EQPT was run on the data0 file.

The Version 7 level structure of the data0 files has been largely preserved in the Version 8 software. The original 80-column maximum line length is retained. Version 7 level data0 files are fully compatible with the Version 8 software. However, the data1 formats have changed somewhat so that the data1 files produced by the Version 7 EQPT (Daveler and Wolery, 1992) cannot be used with the Version 8 EQ3NR and EQ6. Rather, one must create new data1 files using the Version 8 EQPT.

In general, the data0 file consists of the following elements:

- Title including Options
- Miscellaneous parameters: Nominal temperature limits, temperature grid for gridded parameters: pressure, Debye-Hückel constants, the B-dot parameter (if applicable), coefficients for the activity coefficient of CO_2 (if applicable), and the special log K grid for relating Eh to log fO_2

- Activity coefficient parameters specific to individual aqueous species, pairs, or triplets (ion sizes for non-Pitzer data files, interaction coefficient values for Pitzer data files

- Chemical elements block

- Aqueous species blocks (starting with blocks for strict basis species, then auxiliary basis species, then non-basis species)

- Pure minerals blocks

- Pure non-aqueous liquids blocks

- Gas species blocks

- Solid solutions blocks

- References block

The auxiliary basis and non-basis species blocks, and the pure minerals, pure non-aqueous liquids, and gas species blocks each contain an associated chemical reaction, in which the associated species is destroyed (e.g., dissociation or dissolution). With each such reaction is included the associated log K grid. New species may be added by including new blocks in the appropriate sections. If a new chemical element is added, a corresponding new strict basis species composed of only that element with or without oxygen and hydrogen must also be added, and in the corresponding position among the strict basis species. A block for an auxiliary basis species that references another auxiliary basis species in its associated reaction must follow the block for that species. Otherwise, there are no restrictions on the position of a species block within the relevant "superblock" of species of the same type.

The Version 8 EQPT uses dynamic dimensioning for virtually all arrays. Therefore, such actions as adding new species will not cause a problem in exceeding static limits, which was an occasional problem with earlier versions of EQPT.

Comment lines are used frequently in the data0 file. These begin with an asterisk ("*") in the first column. EQPT does not process any data on these lines. The general purpose of comment lines is to provide additional information. In some cases, this information is mainly vestigial. Some of these correspond to lines that were once used in EQ3/6 data0 files, but not in those corresponding to the current or more recent version levels. Typically, such lines have been retained as comment lines in an attempt to facilitate use of newer EQ3/6 data files with other software.,

3.1.1 Title including Options Block

The title is the first element of the data file. It has no fixed length, but the reader should note that the first five lines will be stamped on any EQ3NR and EQ6 output files when the corresponding data1 file is used. The data file name should appear on the first line, followed by the revision number and date of last revision. Any other highly relevant information such as data tracking numbers (DTNs) might also best be given in this section.

Users who modify the data0 files provided with the package should be careful to use the title to note the differences between their versions and versions provided with the software or obtained from some other standard source. A flag denoting that such differences exist should appear in the first five lines of the title to facilitate traceability.

Unless a data0 file is fully compatible with Version 7 level format, the title must contain a small block of keystrings and associated values associated with features unique to Version 8. Normally this block appears at the end of the title, though it (or any of its possible components) may appear anywhere in the title. A sample block (used for data files with the new high-temperature Pitzer capability) from the prototype data0.ypf data file is:

```
BEGIN CONFIGURATION DATA BLOCK
Do not change the data in this block unless you know what you
are doing.
INTERPRET 500 AS NO DATA= YES
* YES or NO
SPARSE GRID RANGE CONDITION=IGNORE
* IGNORE, WARN, or ERROR
Pitzer data parameters:
   PITZER DATA BLOCK ORG.= NEW
   PITZER TEMP FUNCTION= LIVERMORE
   NO. OF PITZER TEMP FUNC TERMS= 4
END CONFIGURATION DATA
```

The default for "INTERPRET 500 AS NO DATA" is YES. The value 500 was originally used in EQ3/6 to indicate no data. It was originally thought that no log K value would exceed such a value. However, some redox reactions have log K values that do exceed this limit. In very old versions of EQ3/6, any value greater than or equal to 500 was treated as "no data". By at least the Version 7 series of the software, only values within 0.0001 of 500 were treated as "no data". Of course, 500 remains a perfectly possible value for a log K. Setting the "INTERPRET 500 AS NO DATA" option to NO enforces the use of the string "No_Data" to indicate "no data". A value of 500 is then treated as a value of 500.

The default for "SPARSE GRID RANGE CONDITION" is IGNORE. A grid range is defined to be sparse if it contains less than three data points. Setting this option to WARN causes a warning message to be issued by EQPT if an instance of such sparseness is encountered. Setting this option to ERROR causes an error message to be issued.

The option sub-block:

```
Pitzer data parameters:
   PITZER DATA BLOCK ORG.= NEW
   PITZER TEMP FUNCTION= LIVERMORE
   NO. OF PITZER TEMP FUNC TERMS= 4
```

must be included to use the new high-temperature Pitzer capability. That is the primary meaning of "PITZER DATA BLOCK ORG.= NEW". The "PITZER TEMP FUNCTION= LIVERMORE" refers to the generalized temperature function:

$$X(T) = a_1 + a_2\left(\frac{1}{T} - \frac{1}{298.15}\right) + a_3 \ln\left(\frac{T}{298.15}\right) + a_4(T - 298.15) + a_5\left(T^2 - (298.15)^2\right) \tag{3–1}$$

where X is the quantity represented, T is the absolute temperature, and a_1, a_2, a_3, etc., are the relevant coefficients. This formula is "25°C-centric", as $X = a_1$ at 25°C (298.15K). This function is used to represent the temperature dependence of the Pitzer interaction coefficients. Certain other temperature functions, some of them not 25°C-centric, may someday be available as alternatives. The "NO. OF PITZER TEMP FUNC TERMS= 4" input directs that only the first four terms (the first four coefficients) are to be used.

Certain other options can also be embedded in the data file title. The following examples are taken from the 7Feb1997 version of data0.1kb, a high-pressure (1 kilobar) data file:

```
Standard data grid parameters:
   NO. OF TEMPERATURE RANGES =  3
   NO. OF POINTS IN RANGE 1  =  9
   NO. OF POINTS IN RANGE 2  =  9
   NO. OF POINTS IN RANGE 3  =  9
Data grid flags:
   ENTHALPY  = ON
   VOLUME    = ON
```

The first sub-block of options allows the standard temperature grid to be changed from the classic 0, 25, 60, 100°C in range 1, and 100, 150, 200, 250, 300°C in range 2. This would correspond to:

```
Standard data grid parameters:
   NO. OF TEMPERATURE RANGES =  2
   NO. OF POINTS IN RANGE 1  =  4
   NO. OF POINTS IN RANGE 2  =  5
```

In general, high-pressure data files demand a wider temperature range, extending to higher temperatures. In the case of data0.1kb, the standard temperature grid is given by:

```
Temperature grid (degC)
         0.0000   25.0000   50.0000   75.0000  100.0000  125.0000
       150.0000  175.0000  200.0000  225.0000  250.0000  275.0000
       300.0000  325.0000  350.0000  375.0000  400.0000  425.0000
       450.0000  475.0000  500.0000  525.0000  550.0000  575.0000
       600.0000
```

It thus extends to 600°C. For this data file, the corresponding pressure is a fixed 1 kilobar. Although changing the standard temperature grid is often desirable for a high-pressure data file, the temperature grid may be changed from the classic one for any data file.

The second sub-block is:

```
Data grid flags:
   ENTHALPY  = ON
   VOLUME    = ON
```

The option "ENTHALPY = ON" indicates that enthalpic properties data grids are present on the data0 file. These additional grids include enthalpic Debye-Hückel parameters, apparent standard molar enthalpies for basis species, and standard molar enthalpies of reaction for auxiliary basis species, non-basis aqueous species, pure minerals, pure non-aqueous liquids, and gas species. At present, EQ3/6 does not use any of this enthalpy data. The option "VOLUME   = ON" indicates that volumetric properties data grids are present on the data0 file. These correspond to those listed above for enthalpic properties. At present, EQ3/6 only uses the volumetric data to make thermodynamic pressure corrections for pressures off the data file standard (e.g., using the 1 kb data file for an EQ3NR or EQ6 run with a specified pressure of 1.2 kb).

3.1.2 Miscellaneous Parameters

The temperature limits along with the temperature grid and various parameters are in the first block titled “Miscellaneous Parameters” as exemplified here from data0.ymp.R2 (DTN: MO0210SPATHDYN.000):

```
+--------------------------------------------------------------------
Miscellaneous parameters
+--------------------------------------------------------------------
Temperature limits (degC)
         0.0000  100.0000
temperatures
         0.0100   25.0000   60.0000  100.0000
       150.0000  200.0000  250.0000  300.0000
pressures
* P(T) along H2O vaporization boundary [source: 84haa/gal]
         1.0132    1.0132    1.0132    1.0132
         4.7572   15.5365   39.7365   85.8378
debye huckel a (adh)                   [source: 74hel/kir2,91joh/nor]
         0.4939    0.5114    0.5465    0.5995
         0.6855    0.7994    0.9593    1.2180
debye huckel b (bdh)                   [source: 74hel/kir2,91joh/nor]
         0.3253    0.3288    0.3346    0.3421
         0.3525    0.3639    0.3766    0.3925
bdot                                   [source: 69hel]
         0.0394    0.0410    0.0438    0.0460
         0.0470    0.0470    0.0340    0.0000
cco2   (coefficients for the 81dru polynomial)
        -1.0312              0.0012806
          255.9                 0.4445
      -0.001606
log k for eh reaction
* eh reaction: 2 H2O(liq) <==> O2(g) + 4 H+ + 4 e- (92joh/oel)
* [sources: as provided within data blocks for the individual species]
       -91.0448  -83.1049  -74.0534  -65.8641
       -57.8929  -51.6848  -46.7256  -42.6828
```

The first parameter, “Temperature limits (degC)”, sets the range over which the temperature can be varied in the calculations utilizing a particular database. This value is typically set from 0–300ºC (as is the range of the temperature grid to be discussed next), but is shown to be set to a range 0–100ºC for which the data will be qualified.

The next parameter, “temperature”, defines the temperature grid points to be utilized by the rest of the database through the log K values. These provide the temperature points at which log K data will be specified and then interpolated by a polynomial fitting, this being performed by EQPT.

The other parameters labeled “pressures”, “debye huckel a (adh)”, “debye huckel b (bdh)”, “bdot”, and “log k for eh reaction” are all given in reference to the respective “temperature” grid points. Only the activity coefficient of CO_2 does not utilize this gridding.

3.1.3 Pitzer Coefficients

Similar blocks exist for Pitzer interaction coefficients for pairs and triplets of aqueous species. These blocks come in two formats. The first is the older one used in Version 7 and older EQ3/6 Pitzer data files. This format uses the 25°C values of their parameters along with optional 25°C values for the first and second temperature derivatives. An example from the 02Dec1996 "R6" version of the data0.hmw data file is for the Ca^{2+}, Cl^{-} pair:

```
+-------------------------------------------------------------
Ca++                       Cl-
  2            -1
*
     beta0 =   0.31590   beta1 =   1.61400   beta2 =   0.00000
                         alpha1 =   2.0       alpha2 =  12.0
*
      cphi = -0.00034
    source = 84har/mol
*
*
    db0/dt =  0.000E+00  d2b0/dt2 =  0.000E+00
    db1/dt =  0.000E+00  d2b1/dt2 =  0.000E+00
    db2/dt =  0.000E+00  d2b2/dt2 =  0.000E+00
     dc/dt =  0.000E+00   d2c/dt2 =  0.000E+00
    source =
+-------------------------------------------------------------
```

Another example from the same data file is for the Na^{+}, Ca^{2+}, Cl^{-} triplet:

```
+-------------------------------------------------------------
Na+                        Ca++                      Cl-
*
     theta =  0.07000       psi = -0.00700
    source = 84har/mol
*
*
    dth/dt =   0.0000    d2th/dt2 =  0.000E+00
   dpsi/dt =   0.0000    d2ps/dt2 =  0.000E+00
    source =
+-------------------------------------------------------------
```

The various types of Pitzer coefficients are discussed in Appendix B.2.

Version 8 of EQ3/6 includes a new high-temperature Pitzer treatment that represents the temperature dependence in a different format (coefficients for a specified temperature function, to be discussed below). In general, the temperature coefficients for an interaction parameter are labeled a_1, a_2, a_3, a_4, and so forth as required. The following example, with its section header, from a data file of this type is for the Ca^{2+}, Cl^{-} pair (from a prototype version of data0.ypf):

```
+--------------------------------------------------------------------
ca combinations: beta(n)(ca) and Cphi(ca) [optional: alpha(n)(ca)]
+--------------------------------------------------------------------
Ca++                       Cl-
  alpha(1) = 2.0
  alpha(2) = 12.0
  beta(0):
    a1 = 0.298461082547347
    a2 = -707.82352924451
    a3 = -3.00707919201924
    a4 = 2.73921619920597E-03
```

```
  beta(1):
    a1 = 1.7106586765016
    a2 = -3510.47408002665
    a3 = -28.1054180935404
    a4 = 5.83893049870609E-02
  beta(2):
    a1 = 0
    a2 = 0
    a3 = 0
    a4 = 0
  Cphi:
    a1 = 3.03712219573499E-03
    a2 = 38.3888745241957
    a3 = 0.062048303720889
    a4 = 1.48987725388134E-05
* Source: refit of 89gre/mol
+----------------------------------------------------------------
```

Note that the alpha(1) and alpha(2) parameters are not temperature dependent. Their appearance in this data block is optional. If they do not appear here, the usual values for the pair charge type will be applied. These usual values are 2.0 and anything for most pairs (for which beta(2) is not normally used), and 1.4 and 12.0 for 2:2, 2:3, 3:3, 2:4 and other higher charge combinations (for which beta(2) is normally used).

In the new format, the blocks for interaction coefficients are distinct according to pair or triplet type, and thus distinguished by the set of associated interaction coefficients. Common types of blocks are grouped together. The old format was problematic in the way that it treated theta parameters. It was also problematic in its treatment of pairs and triplets including electrically neutral species, for which the format was not originally designed to handle.

The theta coefficients are now tied to cation-cation and anion-anion pairs. In the old scheme, they were included in the triplet blocks (as shown in an example given above). The problem is that theta parameters are unique to pairs of ions of the same charge sign, not to triplets containing such pairs. So, the same theta parameter would be specified independently for each such triplet (e.g., Na^+, Ca^{2+}, Cl^- and Na^+, Ca^{2+}, SO_4^{2-}). Basically, it was left to the EQPT code to enforce the use of a single value in the face of multiple entries. Thetas are now specified in a block and its header (also from the prototype data0.ypf) exemplified by:

```
+----------------------------------------------------------------
cc' and aa' combinations: theta(cc') and theta(aa')
+----------------------------------------------------------------
Na+                      Ca++
  theta:
    a1 = 4.99999999999996E-02
    a2 = 1.48831977595669E-10
    a3 = 7.98530455911449E-13
    a4 = -1.04732281527349E-15
* Source: refit of 89gre/mol
+----------------------------------------------------------------
```

Thus for triplets containing two ions of the same charge sign and one of opposite charge sign, the block is reduced to contain only the psi parameter as shown in the following example:

```
+---------------------------------------------------------------
Na+                       Ca++                      Cl-
  psi:
    a1 = -3.00000000000001E-03
    a2 = 4.21408136523772E-12
    a3 = 2.18984170216161E-14
    a4 = -2.77789990205555E-17
* Source: refit of 89gre/mol
+---------------------------------------------------------------
```

New block types have been implemented for pairs and triplets involving electrically neutral species (for nn, nn’, na, nc, nnn’ and nca combinations, where n = neutral, n’ = a different neutral, a = anion, and c = cation). Examples, with their headers, include:

```
+---------------------------------------------------------------
nc and na combinations: lambda(nc) and lambda(na)
+---------------------------------------------------------------
CO2(aq)                   Ca++
  lambda:
    a1 = 0.
    a2 = 0.
    a3 = 0.
    a4 = 0.
* Source:
+-------------------------------------------------------------
```

and

```
+---------------------------------------------------------------
nca combinations: zeta(nca)
+---------------------------------------------------------------
CO2(aq)                   Ca++                      Cl-
  zeta:
    a1 = 0.
    a2 = 0.
    a3 = 0.
    a4 = 0.
* Source:
+-------------------------------------------------------------
```

There are no actual data in these two examples. They merely illustrate the format.

The coefficient data in the new Pitzer interaction coefficient blocks does not require a fixed format. The line for say a_1 only requires that the string “a1 =” appear somewhere on the line followed by a valid number.

3.1.4 Aqueous Species Block

The strict basis species block is illustrated by the following one for HCO_3^-, taken from data0.ymp.R2 (DTN: MO0210SPATHDYN.000):

```
+----------------------------------------------------------------------
basis species
+----------------------------------------------------------------------
HCO3-
     sp.type =  basis
*    EQ3/6   =  ymp.R2, ymp.R0, com, ree, alt, sup
     YMP Qualification status = Q
*    mol.wt. =  61.017 g/mol
*    DHazero =   4.0
     charge  =  -1.0
****
     3 element(s):
      1.0000 C              1.0000 H              3.0000 O
****
*    gflag = 1 [25C,1bar: reported delG0f used]
*    P-T extrapolation algorithm: Cp(T), V(P) integration
*    P-T extrapolation alg. ref.: 88tan/hel, 92joh/oel
*    reference-state data source: 88sho/hel
*         delG0f =   -140.282 kcal/mol
*         delH0f =   -164.898 kcal/mol
*         S0PrTr =     23.530 cal/(mol*K)
*    EOS parameters source: 88sho/hel
*         a1     =      7.5621(10**-1)
*         a2     =      1.1505(10**2)
*         a3     =      1.2346(10**0)
*         a4     =     -2.8266(10**4)
*         c1     =     12.9395(10**0)
*         c2     =     -4.7579(10**4)
*         omega  =      1.2733(10**5)
+----------------------------------------------------------------------
```

Note the presence of many comment lines. In addition, the "sp.type" line and the "YMP Qualification status" line are unused by either the Version 7 or Version 8 level EQPT. They are actually unnecessary for the Version 8 EQPT. Thus, from that code's viewpoint, the above block could be simplified to just:

```
+----------------------------------------------------------------------
HCO3-
charge  =  -1.0
****
3 element(s):
      1.0000 C              1.0000 H              3.0000 O
+----------------------------------------------------------------------
```

Here one comment line used as a spacer has been retained for esthetic reasons.

A strict basis species does not have an associated chemical reaction and attendant thermodynamic data. Therefore, the essential information in the data block for such a species is limited to the species name (limited to 24 characters), electrical charge number, and elemental composition.

A more general aqueous species block (for an auxiliary basis species or a non-basis species) contains additional data. This type of aqueous species block is illustrated by the following one for CaF^+, taken from data0.ymp.R2 (DTN: MO0210SPATHDYN.000):

```
+--------------------------------------------------------------------
aqueous species
+--------------------------------------------------------------------
CaF+
     sp.type =  aqueous
*    EQ3/6   =  ymp.R2, ymp.R0, com, ree, alt, sup
     YMP Qualification status = Q
*    mol.wt. =  59.076 g/mol
*    DHazero =   4.0
     charge  =   1.0
****
     2 element(s):
      1.0000 Ca            1.0000 F
****
     3 species in aqueous dissociation reaction:
    -1.0000  CaF+                        1.0000  Ca++
     1.0000  F-
*
**** logK grid [0-25-60-100C @1bar; 150-200-250-300C @Psat-H2O]:
        -0.6546   -0.6817   -0.8624   -1.1708
        -1.6504   -2.2173   -2.8881   -3.7282
*
*    gflag = 1 [25C,1bar: reported delG0f used]
*    P-T extrapolation algorithm: Cp(T), V(P) integration
*    P-T extrapolation alg. ref.: 88tan/hel, 92joh/oel
*    reference-state data source: 97sve/sho
*         delG0f =   -200.390 kcal/mol
*         delH0f =   -208.600 kcal/mol
*         S0PrTr =     -9.000 cal/(mol*K)
*    EOS parameters source: 97sve/sho
*         a1     =      0.1568(10**-1)
*         a2     =     -7.3958(10**2)
*         a3     =      8.6499(10**0)
*         a4     =     -2.4732(10**4)
*         c1     =     30.1743(10**0)
*         c2     =      3.0968(10**4)
*         omega  =      0.6911(10**5)
+--------------------------------------------------------------------
```

The essential elements are given by:

```
+--------------------------------------------------------------------
CaF+
charge  =   1.0
****
2 element(s):
      1.0000 Ca            1.0000 F
****
3 species in aqueous dissociation reaction:
    -1.0000  CaF+                        1.0000  Ca++
     1.0000  F-
*
**** logK grid [0-25-60-100C @1bar; 150-200-250-300C @Psat-H2O]:
        -0.6546   -0.6817   -0.8624   -1.1708
        -1.6504   -2.2173   -2.8881   -3.7282
+--------------------------------------------------------------------
```

Here four of the comment lines (three of them spacers) have been retained for esthetic reasons.

3.1.5 Pure Mineral Block

A pure mineral block is quite similar in form and content. Such a block is illustrated by the following, again with its header, for albite taken from data0.ymp.R2 (DTN: MO0210SPATHDYN.000):

```
+--------------------------------------------------------------------
solids
+--------------------------------------------------------------------
Albite                  NaAlSi3O8
     sp.type =  solid
*    EQ3/6   =  ymp.R2, ymp.R0, com, ree, alt, sup
     YMP Qualification status = Q
*    mol.wt. =   262.223 g/mol
     V0PrTr  =   100.250 cm**3/mol [source: 78hel/del]
****
     4 element(s):
      1.0000 Al             1.0000 Na             8.0000 O
      3.0000 Si
****
     6 species in aqueous dissociation reaction:
    -1.0000  Albite                      -4.0000  H+
     1.0000  Al+++                        1.0000  Na+
     2.0000  H2O                          3.0000  SiO2(aq)
*
**** logK grid [0-25-60-100C @1bar; 150-200-250-300C @Psat-H2O]:
         3.9014    2.8495    1.4033    0.0345
        -1.2965   -2.3537   -3.2926   -4.2839
*
*    gflag = 1 [25C,1bar: reported delG0f used]
*    P-T extrapolation algorithm: Cp(T), const-V(P) integration
*    P-T extrapolation alg. ref.: 78hel/del, 78hel/del
*    reference-state data source: 78hel/del
*         delG0f =   -886.308 kcal/mol
*         delH0f =   -939.680 kcal/mol
*         S0PrTr =     49.510 cal/(mol*K)
*    Cp coefficients [source: 78hel/del ]
*         T**0   =  0.61700000E+02
*         T**1   =  0.13900000E-01
*         T**-2  = -0.15010000E+07
*         Tlimit =  199.85C
*         T**0   =  0.81880000E+02
*         T**1   =  0.35540000E-02
*         T**-2  = -0.50154000E+07
*         Tlimit =  926.85C
+--------------------------------------------------------------------
```

The essential elements in this example are given by:

```
+--------------------------------------------------------------------
Albite                  NaAlSi3O8
V0PrTr  =   100.250 cm**3/mol [source: 78hel/del]
****
     4 element(s):
      1.0000 Al             1.0000 Na             8.0000 O
      3.0000 Si
****
     6 species in aqueous dissociation reaction:
    -1.0000  Albite                      -4.0000  H+
     1.0000  Al+++                        1.0000  Na+
     2.0000  H2O                          3.0000  SiO2(aq)
*
**** logK grid [0-25-60-100C @1bar; 150-200-250-300C @Psat-H2O]:
         3.9014    2.8495    1.4033    0.0345
        -1.2965   -2.3537   -3.2926   -4.2839
+--------------------------------------------------------------------
```

Here once again four of the comment lines (three of them spacers) have been retained for esthetic reasons.

Data blocks for pure non-aqueous liquid and gas species have the same form as those for pure minerals. Therefore, they will not be discussed further here.

3.1.6 Solid Solutions Block

Data blocks for solid solutions are quite different. The following example, with its section header, for a hexagonal six-component carbonate is taken from data0.ymp.R2 (DTN: MO0210SPATHDYN.000):

```
+--------------------------------------------------------------------
solid solutions
+--------------------------------------------------------------------
Carbonate-Calcite       (Ca,Mn,Zn,Mg,Fe,Sr)CO3
     sp.type =  ss   ideal

  6 components
     1.0000  Calcite                      1.0000  Magnesite
     1.0000  Rhodochrosite                1.0000  Siderite
     1.0000  Smithsonite                  1.0000  Strontianite
  type  = 1
  0 model parameter(s)
  1 site parameter(s)
  1.000     0.000     0.000     0.000     0.000     0.000
+--------------------------------------------------------------------
```

The solid solutions currently used in EQ3/6 data files are ideal site-mixing models treating mixing over only one site. The above data block is an example of this (“type = 1”). For discussion of other possible models, see Appendix B.3.3–B.3.7. The data block gives the end-member components (which must all be pure minerals with data blocks present in the minerals superblock) along with one or more “model” parameters and one or more “site” parameters. There are no “model” parameters for an ideal solution. Only one site parameter is currently used. This is the site stoichiometric factor (the number of moles of site per mole of solid solution).

In adding any new species (or solid solution) blocks to a data0 file, the user should use an existing block as a template, using the copy-paste-edit technique. This will allow the user to proceed without having to determine exact line formats.

3.2 EQ3NR AND EQ6 INPUT FILES

A text input file is the primary means of specifying a problem to run using EQ3/6 Version 8.0. The choice of supporting data file (discussed below) completes the specification. EQ3NR input files use the “3i” filename extension; EQ6 input files use the “6i” extension. In the Version 7 series software, two formats were available: compact or “W” and menu-style or “D”. The Version 8.0 of this software retains both formats. However, only the menu-style format should be used. This format has been substantially improved from the Version 7 equivalent. The deprecated compact format, which requires adhering to strict line formatting, can still be accessed by converting a “D” format input file to “W” format using the XCIF3 and XCIF6 utilities for EQ3NR and EQ6 input files, respectively. Because “W” format is deprecated, it will not be described here except to note that it is very similar to the Version 7 equivalent.

3.2.1 Input and Output Default Values

Following the usual rules of Fortran, blank inputs are treated as blanks for strings and zeros for numbers. In certain instances, zero values for numbers are overridden by internal non-zero defaults. The number of instances where non-zero defaults may be provided in EQ3NR and EQ6 input files has deliberately been kept small. In general, such instances are restricted to a handful of numerical control parameters including convergence tolerances and limits on iteration numbers in convergence processes. In the descriptions of the EQ3NR and EQ6 input files given below, the values of any non-zero defaults will be noted. Otherwise, there are no defaults. When EQ3NR and EQ6 run, they write values for all relevant parameters, including assumed defaults, on their OUTPUT files (.3o files for EQ3NR, .6o files for EQ6).

3.2.2 Input File Processing and Error Control

Both EQ3NR and EQ6 write on their OUTPUT files a line-by-line echo of the input file as they read it. This facilitates locating any errors on the input file, as the last line successfully read and processed is included in this echo. The lines on the menu-style input files are read as 80-character strings, which precludes read format errors at this stage. Read format errors can potentially occur when data are read from the relevant input fields. However, any such errors are trapped by the program, which then handles its own error processing. Users thus see EQ3/6 error messages, not standard Fortran error messages.

All EQ3/6 error messages (not just those related to input file processing) are complete messages in transparent English, normally consisting of 1 to 5 sentences. An example is:

```
* Error - (EQ3NR/intbs3) The species Salicylic_acid(aq) is
       on the input file, but it isn't on the supporting data file.
```

Cryptic error codes and the like are eschewed. It is presumed that the user understands the concepts necessary to understand these messages, such as what is a computer file, what are auxiliary basis species and how are they used, and what species names may appear on an input file. There is a small possibility that the user may encounter a system error message. Any message that seems to be in “computer-ese” is probably a system message.

EQ3/6 messages related to the input file, like such messages more generally, fall into three categories: errors, warnings, and notes. When an input file is being processed, messages of any of these three types may be generated. Errors are just that, and will at some point (not necessarily immediately) lead to run termination. Warnings indicate conditions that the user should examine to see if these might be actual errors. Notes indicate conditions that the user should at least know about. An error message will contain the string “Error” (as in the example shown above), a warning the string “Warning”, and a note the string “Note”. Warnings and notes follow the same format as error messages.

3.3 THE EQ3NR INPUT FILE

The following is an example of an EQ3NR input file (in menu-style or “D” format). This is the file swmaj.3i from the EQ3NR test case library. This is presented here in its entirety (occupying about four pages) to better acquaint the reader with the overall form. The various parts of this input file, which is a reasonably typical example, will be discussed below. Note that the file begins and ends with a separator line. Separator lines are never doubled up. Labels are included in all instances to provide cross-references to code variable names. Some of the important code variable names are presented in bold type in this Section and in Appendices B and D.

```
|------------------------------------------------------------------------------|
| Title                  | (utitl(n))                                          |
|------------------------------------------------------------------------------|
|EQ3NR input file name= swmaj.3i                                               |
|Description= "Sea water, major ions only"                                     |
|Version level= 8.0                                                            |
|Revised 02/14/97    Revisor= T.J. Wolery                                      |
|This is part of the EQ3/6 Test Case Library                                   |
|                                                                              |
|  Sea water, including only the major ions. This is a considerably            |
|pared-down version of swtst.3i, which contains the full benchmark sea water   |
|test case of Nordstrom et al. (1979, Table III).                              |
|                                                                              |
|  Purpose: to test the code on a small problem involving a moderately         |
|concentrated solution, using the B-dot equation to calculate the activity     |
|coefficients of the aqueous species. Input file swmajd.3i runs the same       |
|problem using the Davies equation instead of the B-dot equation. Input file   |
|swmajp.3i runs the same problem using Pitzer's equations.                     |
|                                                                              |
|                                                                              |
|                            References                                        |
|                                                                              |
|Nordstrom, D.K., et al., 1979, A comparison of computerized chemical models   |
|  for equilibrium calculations in aqueous systems, in Jenne, E.A., editor,    |
|  Chemical Modeling in Aqueous Systems, ACS Symposium Series, v. 93,          |
|  American Chemical Society, Washington, DC, p. 857-892.                      |
|                                                                              |
|------------------------------------------------------------------------------|
|Special Basis Switches (for model definition only)       | (nsbswt)           |
|------------------------------------------------------------------------------|
|Replace |None                                            | (usbsw(1,n))       |
|   with |None                                            | (usbsw(2,n))       |
|------------------------------------------------------------------------------|
|Temperature (C)         | 2.50000E+01| (tempc)                                |
|------------------------------------------------------------------------------|
|Pressure option (jpres3):                                                     |
|  [x] ( 0) Data file reference curve value                                    |
|  [ ] ( 1) 1.013-bar/steam-saturation curve value                             |
|  [ ] ( 2) Value (bars) | 0.00000E+00| (press)                                |
|------------------------------------------------------------------------------|
|Density (g/cm3)         | 1.02336E+00| (rho)                                  |
|------------------------------------------------------------------------------|
|Total dissolved solutes option (itdsf3):                                      |
|  [x] ( 0) Value (mg/kg.sol) | 0.00000E+00| (tdspkg)                          |
|  [ ] ( 1) Value (mg/L)      | 0.00000E+00| (tdspl)                           |
|------------------------------------------------------------------------------|
|Electrical balancing option (iebal3):                                         |
|  [x] ( 0) No balancing is done                                               |
|  [ ] ( 1) Balance on species |None                    | (uebal)              |
|------------------------------------------------------------------------------|
|Default redox constraint (irdxc3):                                            |
|  [ ] (-3) Use O2(g) line in the aqueous basis species block                  |
|  [ ] (-2) pe (pe units)      | 0.00000E+00| (pei)                            |
|  [x] (-1) Eh (volts)         | 5.00000E-01| (ehi)                            |
|  [ ] ( 0) Log fO2 (log bars) | 0.00000E+00| (fo2lgi)                         |
|  [ ] ( 1) Couple (aux. sp.)  |None                    | (uredox)             |
|------------------------------------------------------------------------------|
|Aqueous Basis Species/Constraint Species        |Conc., etc. |Units/Constraint|
| (uspeci(n)/ucospi(n))                          | (covali(n))|(ujf3(jflgi(n)))|
|------------------------------------------------------------------------------|
|Na+                                             | 1.07680E+04|mg/kg.sol       |
|K+                                              | 3.99100E+02|mg/kg.sol       |
|Ca++                                            | 4.12300E+02|mg/kg.sol       |
|Mg++                                            | 1.29180E+03|mg/kg.sol       |
|H+                                              | 8.22000E+00|pH              |
|HCO3-                                           | 2.02200E-03|Molality        |
|Cl-                                             | 1.93530E+04|mg/kg.sol       |
```

```
|SO4--                                           | 2.71200E+03|mg/kg.sol       |
|------------------------------------------------------------------------------|
* Valid jflag strings (ujf3(jflgi(n))) are:                                    *
*    Suppressed          Molality            Molarity                          *
*    mg/L                mg/kg.sol           Alk., eq/kg.H2O                   *
*    Alk., eq/L          Alk., eq/kg.sol     Alk., mg/L CaCO3                  *
*    Alk., mg/L HCO3-    Log activity        Log act combo                     *
*    Log mean act        pX                  pH                                *
*    pHCl                Hetero. equil.      Homo. equil.                      *
*    Make non-basis                                                            *
*------------------------------------------------------------------------------*
|Create Ion Exchangers  | (net)                                                |
|------------------------------------------------------------------------------|
|Advisory: no exchanger creation blocks follow on this file.                   |
|Option: on further processing (writing a PICKUP file or running XCON3 on the  |
|present file), force the inclusion of at least one such block (qgexsh):       |
|  [ ] (.true.)                                                                |
|------------------------------------------------------------------------------|
|Ion Exchanger Compositions      | (neti)                                      |
|------------------------------------------------------------------------------|
|Exchanger phase |None                    | (ugexpi(n))                        |
|------------------------------------------------------------------------------|
|->|Moles/kg.H2O    |  0.0000    | (cgexpi(n))                                 |
|------------------------------------------------------------------------------|
|->|Exchange site   |None    | (ugexji(j,n))                                   |
|------------------------------------------------------------------------------|
|--->|Exchange species        |Eq. frac.   | (this is a table header)          |
|------------------------------------------------------------------------------|
|--->|None                    | 0.00000E+00| (ugexsi(i,j,n), egexsi(i,j,n))    |
|------------------------------------------------------------------------------|
|Solid Solution Compositions     | (nxti)                                      |
|------------------------------------------------------------------------------|
|Solid Solution          |None                    | (usoli(n))                 |
|------------------------------------------------------------------------------|
|->|Component               |Mole frac.  | (this is a table header)            |
|------------------------------------------------------------------------------|
|->|None                    | 0.00000E+00| (umemi(i,n), xbari(i,n))            |
|------------------------------------------------------------------------------|
|Alter/Suppress Options  | (nxmod)                                             |
|------------------------------------------------------------------------------|
|Species                                         |Option          |Alter value |
| (uxmod(n))                                     |(ukxm(kxmod(n)))| (xlkmod(n))|
|------------------------------------------------------------------------------|
|None                                            |None            | 0.00000E+00|
|------------------------------------------------------------------------------|
* Valid alter/suppress strings (ukxm(kxmod(n))) are:                           *
*    Suppress            Replace             AugmentLogK                       *
*    AugmentG                                                                  *
*------------------------------------------------------------------------------*
|Iopt Model Option Switches ("( 0)" marks default choices)                     |
|------------------------------------------------------------------------------|
|iopt(4) - Solid Solutions:                                                    |
|  [x] ( 0) Ignore                                                             |
|  [ ] ( 1) Permit                                                             |
|------------------------------------------------------------------------------|
|iopt(11) - Auto Basis Switching in pre-N-R Optimization:                      |
|  [x] ( 0) Turn off                                                           |
|  [ ] ( 1) Turn on                                                            |
|------------------------------------------------------------------------------|
|iopt(17) - PICKUP File Options:                                               |
|  [ ] (-1) Don't write a PICKUP file                                          |
|  [x] ( 0) Write a PICKUP file                                                |
|------------------------------------------------------------------------------|
|iopt(19) - Advanced EQ3NR PICKUP File Options:                                |
|  [x] ( 0) Write a normal EQ3NR PICKUP file                                   |
|  [ ] ( 1) Write an EQ6 INPUT file with Quartz dissolving, relative rate law  |
|  [ ] ( 2) Write an EQ6 INPUT file with Albite dissolving, TST rate law       |
|  [ ] ( 3) Write an EQ6 INPUT file with Fluid 1 set up for fluid mixing       |
|------------------------------------------------------------------------------|
|Iopg Activity Coefficient Option Switches ("( 0)" marks default choices)      |
|------------------------------------------------------------------------------|
```

```
|iopg(1) - Aqueous Species Activity Coefficient Model:                         |
|  [ ] (-1) The Davies equation                                                |
|  [x] ( 0) The B-dot equation                                                 |
|  [ ] ( 1) Pitzer's equations                                                 |
|  [ ] ( 2) HC + DH equations                                                  |
|------------------------------------------------------------------------------|
|iopg(2) - Choice of pH Scale (Rescales Activity Coefficients):                |
|  [ ] (-1) "Internal" pH scale (no rescaling)                                 |
|  [x] ( 0) NBS pH scale (uses the Bates-Guggenheim equation)                  |
|  [ ] ( 1) Mesmer pH scale (numerically, pH = -log m(H+))                     |
|------------------------------------------------------------------------------|
|Iopr Print Option Switches ("( 0)" marks default choices)                     |
|------------------------------------------------------------------------------|
|iopr(1) - Print All Species Read from the Data File:                          |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print                                                              |
|------------------------------------------------------------------------------|
|iopr(2) - Print All Reactions:                                                |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print the reactions                                                |
|  [ ] ( 2) Print the reactions and log K values                               |
|  [ ] ( 3) Print the reactions, log K values, and associated data             |
|------------------------------------------------------------------------------|
|iopr(3) - Print the Aqueous Species Hard Core Diameters:                      |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print                                                              |
|------------------------------------------------------------------------------|
|iopr(4) - Print a Table of Aqueous Species Concentrations, Activities, etc.:  |
|  [ ] (-3) Omit species with molalities < 1.e-8                               |
|  [ ] (-2) Omit species with molalities < 1.e-12                              |
|  [ ] (-1) Omit species with molalities < 1.e-20                              |
|  [x] ( 0) Omit species with molalities < 1.e-100                             |
|  [ ] ( 1) Include all species                                                |
|------------------------------------------------------------------------------|
|iopr(5) - Print a Table of Aqueous Species/H+ Activity Ratios:                |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print cation/H+ activity ratios only                               |
|  [ ] ( 2) Print cation/H+ and anion/H+ activity ratios                       |
|  [ ] ( 3) Print ion/H+ activity ratios and neutral species activities        |
|------------------------------------------------------------------------------|
|iopr(6) - Print a Table of Aqueous Mass Balance Percentages:                  |
|  [ ] (-1) Don't print                                                        |
|  [x] ( 0) Print those species comprising at least 99% of each mass balance   |
|  [ ] ( 1) Print all contributing species                                     |
|------------------------------------------------------------------------------|
|iopr(7) - Print Tables of Saturation Indices and Affinities:                  |
|  [ ] (-1) Don't print                                                        |
|  [x] ( 0) Print, omitting those phases undersaturated by more than 10 kcal   |
|  [ ] ( 1) Print for all phases                                               |
|------------------------------------------------------------------------------|
|iopr(8) - Print a Table of Fugacities:                                        |
|  [ ] (-1) Don't print                                                        |
|  [x] ( 0) Print                                                              |
|------------------------------------------------------------------------------|
|iopr(9) - Print a Table of Mean Molal Activity Coefficients:                  |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print                                                              |
|------------------------------------------------------------------------------|
|iopr(10) - Print a Tabulation of the Pitzer Interaction Coefficients:         |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print a summary tabulation                                         |
|  [ ] ( 2) Print a more detailed tabulation                                   |
|------------------------------------------------------------------------------|
|iopr(17) - PICKUP file format ("W" or "D"):                                   |
|  [x] ( 0) Use the format of the INPUT file                                   |
|  [ ] ( 1) Use "W" format                                                     |
|  [ ] ( 2) Use "D" format                                                     |
|------------------------------------------------------------------------------|
|Iodb Debugging Print Option Switches ("( 0)" marks default choices)           |
|------------------------------------------------------------------------------|
|iodb(1) - Print General Diagnostic Messages:                                  |
```

```
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print Level 1 diagnostic messages                                  |
|  [ ] ( 2) Print Level 1 and Level 2 diagnostic messages                      |
|------------------------------------------------------------------------------|
|iodb(3) - Print Pre-Newton-Raphson Optimization Information:                  |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print summary information                                          |
|  [ ] ( 2) Print detailed information (including the beta and del vectors)    |
|  [ ] ( 3) Print more detailed information (including matrix equations)       |
|  [ ] ( 4) Print most detailed information (including activity coefficients)  |
|------------------------------------------------------------------------------|
|iodb(4) - Print Newton-Raphson Iteration Information:                         |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print summary information                                          |
|  [ ] ( 2) Print detailed information (including the beta and del vectors)    |
|  [ ] ( 3) Print more detailed information (including the Jacobian)           |
|  [ ] ( 4) Print most detailed information (including activity coefficients)  |
|------------------------------------------------------------------------------|
|iodb(6) - Print Details of Hypothetical Affinity Calculations:                |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print summary information                                          |
|  [ ] ( 2) Print detailed information                                         |
|------------------------------------------------------------------------------|
|Numerical Parameters                                                          |
|------------------------------------------------------------------------------|
| Beta convergence tolerance      | 0.00000E+00| (tolbt)                       |
| Del convergence tolerance       | 0.00000E+00| (toldl)                       |
| Max. Number of N-R Iterations   |   0        | (itermx)                      |
|------------------------------------------------------------------------------|
|Ordinary Basis Switches (for numerical purposes only)    | (nobswt)           |
|------------------------------------------------------------------------------|
|Replace |None                                            | (uobsw(1,n))       |
|   with |None                                            | (uobsw(2,n))       |
|------------------------------------------------------------------------------|
|Sat. flag tolerance     | 0.00000E+00| (tolspf)                               |
|------------------------------------------------------------------------------|
|Aq. Phase Scale Factor  | 1.00000E+00| (scamas)                               |
|------------------------------------------------------------------------------|
|End of problem                                                                |
|------------------------------------------------------------------------------|
```

The elements of an EQ3NR input file are:

- Title
- Special basis switches
- Temperature
- Pressure option
- Density
- Total dissolved salts (TDS) option
- Electrical balancing option
- Redox option
- Basis species constraints (concentrations, etc.)
- Ion exchanger creation flag

- *If the above flag is set to true*: Ion exchanger creation blocks (one or more)
- Ion exchanger compositions
- Solid solution compositions
- Alter/suppression options
- Iopt (model) options
- Iopg (activity coefficient) options
- Iopr (print) options
- Iodb (debug print) options
- Numerical parameters
- Ordinary basis switches
- Saturation flag tolerance
- Aqueous phase scale factor

We will consider these in order.

3.3.1 Title

The title block is intended to allow the user to document the problem defined on the input file. In the present example, the title is:

```
|------------------------------------------------------------------------------|
| Title                  | (utitl(n))                                          |
|------------------------------------------------------------------------------|
|EQ3NR input file name= swmaj.3i                                               |
|Description= "Sea water, major ions only"                                    |
|Version level= 8.0                                                            |
|Revised 02/14/97    Revisor= T.J. Wolery                                      |
|This is part of the EQ3/6 Test Case Library                                   |
|                                                                              |
|  Sea water, including only the major ions. This is a considerably            |
|pared-down version of swtst.3i, which contains the full benchmark sea water   |
|test case of Nordstrom et al. (1979, Table III).                              |
|                                                                              |
|  Purpose: to test the code on a small problem involving a moderately         |
|concentrated solution, using the B-dot equation to calculate the activity     |
|coefficients of the aqueous species. Input file swmajd.3i runs the same       |
|problem using the Davies equation instead of the B-dot equation. Input file   |
|swmajp.3i runs the same problem using Pitzer's equations.                     |
|                                                                              |
```

```
|                                                                            |
|                            References                                      |
|                                                                            |
|Nordstrom, D.K., et al., 1979, A comparison of computerized chemical models |
|  for equilibrium calculations in aqueous systems, in Jenne, E.A., editor,  |
|  Chemical Modeling in Aqueous Systems, ACS Symposium Series, v. 93,        |
|  American Chemical Society, Washington, DC, p. 857-892.                    |
|                                                                            |
|----------------------------------------------------------------------------|
```

Note that the header is separated from the main body by a separator line, a common but not universal practice. The format of the first three lines should be followed exactly. The first line includes the name of the input file, the second contains an abbreviated description of the problem, and the third notes the version level. XCIF3 and XCIF6 use the version level line as a quick means of identifying the level. The format of the fourth line, intended to identify the date and the input file creator/reviser, is recommended, but not necessary. The fifth line here identifies this file as a member of the test case library. If this file is used as a template for an ordinary file, the user should remove this line, or at least the words, “This is part of the EQ3/6 Test Case Library,” as the user’s input file will not be a member of that library. The remaining lines should be used to describe the problem being defined, along with any pertinent references. If the data entered on this input file are associated with any data tracking numbers (DTNs), this is a good place to note those numbers. The number of lines in the main body of a title is not fixed, but may not exceed **200**. This is a static dimension.

3.3.2 Special Basis Switches

The next block allows specification of special basis switches. A special basis switch is one in which a strict basis species and a corresponding auxiliary basis species exchange roles. The net effect is as though they had the new roles on the supporting data file. In the present example, the block is:

```
|----------------------------------------------------------------------------|
|Special Basis Switches (for model definition only)      | (nsbswt)          |
|----------------------------------------------------------------------------|
|Replace |None                                            | (usbsw(1,n))      |
|   with |None                                            | (usbsw(2,n))      |
|----------------------------------------------------------------------------|
```

This is not an interesting case, as no switches are specified. The following modification would replace $NH_3(aq)$ in the strict basis set with NO_3^-:

```
|----------------------------------------------------------------------------|
|Special Basis Switches (for model definition only)      | (nsbswt)          |
|----------------------------------------------------------------------------|
|Replace |NH3(aq)                                         | (usbsw(1,n))      |
|   with |NO3-                                            | (usbsw(2,n))      |
|----------------------------------------------------------------------------|
```

Multiple switches can be specified by repeating as necessary the “Replace” and “with” lines in the main body of this block; e.g.,

```
|------------------------------------------------------------------------------|
|Special Basis Switches (for model definition only)       | (nsbswt)           |
|------------------------------------------------------------------------------|
|Replace |NH3(aq)                                          | (usbsw(1,n))       |
|   with |NO3-                                             | (usbsw(2,n))       |
|Replace |Fe++                                             | (usbsw(1,n))       |
|   with |Fe+++                                            | (usbsw(2,n))       |
|Replace |CrO4--                                           | (usbsw(1,n))       |
|   with |Cr+++                                            | (usbsw(2,n))       |
|------------------------------------------------------------------------------|
```

The roles defined by special basis switching apply to the interpretation of all subsequent inputs on the input file. Note that special basis switching may affect model definition, due to the way that mass balances are defined for basis species. This will be addressed below in the discussion of the basis species constraint block. There is no set limit to the number of special basis switches, but they may not exceed the number of basis species on the supporting data file that is employed in the calculation.

3.3.3 Temperature

Next comes the temperature block, which is quite simple and was discussed previously (Section 2.4.1) as an example of input file blocks:

```
|------------------------------------------------------------------------------|
|Temperature (C)         | 2.50000E+01| (tempc)                                |
|------------------------------------------------------------------------------|
```

3.3.4 Pressure Option

This is followed by the pressure option block:

```
|------------------------------------------------------------------------------|
|Pressure option (jpres3):                                                     |
|  [x] ( 0) Data file reference curve value                                    |
|  [ ] ( 1) 1.013-bar/steam-saturation curve value                             |
|  [ ] ( 2) Value (bars) | 0.00000E+00| (press)                                |
|------------------------------------------------------------------------------|
```

The default option (checked here) sets the pressure at the reference curve value for the data file employed. For the data files most commonly used, this corresponds to the 1.013-bar/steam-saturation curve, though some data files have other reference pressure curves (commonly a fixed pressure at all temperatures, such as 500 bars, 1 kb, 2kb, or 5kb). Option 1 is intended to allow the 1.013-bar/steam-saturation curve to be used with a data file that has a different reference pressure curve, although it might be used for a file that uses the same curve for its pressure reference. Options 1 and 2 should be used only with data files that support thermodynamic pressure corrections, such as the “shv”, “500”, “1kb”, “2kb”, and “5kb” data files. Note that the commonly used cmp, hmw, sup, ymp, and ypf data files do not support pressure corrections.

3.3.5 Density

Next is the density block:

```
|------------------------------------------------------------------------------|
|Density (g/cm3)         | 1.02336E+00| (rho)                                  |
|------------------------------------------------------------------------------|
```

The density of the aqueous solution is used to convert volumetrically based units of concentration (e.g., mg/L, molarity) to molality. The default value is 1.0 g/cm^3.

3.3.6 Total Dissolved Salts (TDS) Option

The total dissolved solids (TDS) block follows:

```
|------------------------------------------------------------------------------|
|Total dissolved solutes option (itdsf3):                                      |
|  [x] ( 0) Value (mg/kg.sol) | 0.00000E+00| (tdspkg)                          |
|  [ ] ( 1) Value (mg/L)      | 0.00000E+00| (tdspl)                           |
|------------------------------------------------------------------------------|
```

Only one of two possible entries may be used, as determined by the checkmark. If TDS in mg/L is used, the code converts it to TDS in mg/kg.sol (mg/kg of solution). That in turn is used in converting concentrations in molarity, mg/L, and mg/kg.sol to molalities. The default value of TDS in mg/kg.sol (used in such conversions) is zero. For a discussion of the conversion of other units of concentration to molality, see Appendix B.1.2.

3.3.7 Electrical Balancing Option

Next comes the electrical balancing option:

```
|------------------------------------------------------------------------------|
|Electrical balancing option (iebal3):                                         |
|  [x] ( 0) No balancing is done                                               |
|  [ ] ( 1) Balance on species |None                    | (uebal)              |
|------------------------------------------------------------------------------|
```

In the case of simple speciation-solubility calculations using real water analyses, balancing is not recommended. The code will then calculate the apparent electrical imbalance. A large calculated electrical imbalance may indicate incomplete or erroneous chemical analysis, or a mistake on the user's part in interpreting reported analytical results. A common trap is thinking that "HCO3 mg/L" is the mg/L of HCO_3^-, when in standard drinking water analytical procedure this is actually the HCO3 alkalinity in equivalent mg/L of $CaCO_3$ (See Appendix B.1.3.5).

Electrical balancing is specified by marking the appropriate checkbox and providing the name of the basis species on which balancing is to be made. The input concentration (or pH in the case of balancing on H^+) is then adjusted to achieve an electrically balanced solution model. Electrical balancing is appropriate when using the code to solve certain kinds of problems, for example, what is the pH at a certain temperature of pH buffer solution. Electrical balancing can also be done as an arbitrary means of correcting for the cumulative effects of small analytical errors. It is generally wise to electrically balance a model aqueous solution before using EQ6 to simulate a process such as evaporation, where the amount of solvent water may become quite small. In such a case, the charge imbalance would be fixed, but would become large relevant to the amount of remaining water. Electrical balancing on a given species may fail because the concentration of that species would have to be less than zero. The code will normally diagnose this for the user. If this happens, the user should then try balancing on an ion of opposite charge.

Here is an example of the input block where electrical balancing on the chloride anion is specified:

```
|------------------------------------------------------------------------------|
|Electrical balancing option (iebal3):                                         |
|  [ ] ( 0) No balancing is done                                               |
|  [x] ( 1) Balance on species |Cl-                     | (uebal)               |
|------------------------------------------------------------------------------|
```

3.3.8 Redox Option

This is followed by the redox option:

```
|------------------------------------------------------------------------------|
|Default redox constraint (irdxc3):                                            |
|  [ ] (-3) Use O2(g) line in the aqueous basis species block                  |
|  [ ] (-2) pe (pe units)      | 0.00000E+00| (pei)                            |
|  [x] (-1) Eh (volts)         | 5.00000E-01| (ehi)                            |
|  [ ] ( 0) Log fO2 (log bars) | 0.00000E+00| (fo2lgi)                         |
|  [ ] ( 1) Couple (aux. sp.)  |None                    | (uredox)             |
|------------------------------------------------------------------------------|
```

Internally, the code uses the log oxygen fugacity (log fO_2) as the primary redox variable. A value may be directly specified. The redox potential (Eh) is also common in geochemical modeling. Note that the value must be entered in volts, not millivolts. Eh values are more commonly reported in millivolts. The "pe" or electron activity function (intended to be analogous to pH) is much less commonly used, but can be used as an input to EQ3NR. The "–3" option shown above can be used to specify the log fO_2 as the concentration for the fictive basis species $O_2(g)$ on a line in the basis species constraint block, which follows the present block. The "1" option allows the log fO_2 to be calculated from data for a redox couple such as Fe^{2+}/Fe^{3+}. One of the members of the couple must be an auxiliary basis species whose reaction links it to the other species (usually a strict basis species, but it could be another auxiliary basis species). The former species must be named on the "Couple (aux. sp.)" line. On most supporting data files, Fe^{3+} is an auxiliary basis species linked to Fe^{2+}. The redox option block for using this couple would then be:

```
|------------------------------------------------------------------------------|
|Default redox constraint (irdxc3):                                            |
|  [ ] (-3) Use O2(g) line in the aqueous basis species block                  |
|  [ ] (-2) pe (pe units)      | 0.00000E+00| (pei)                            |
|  [ ] (-1) Eh (volts)         | 5.00000E-01| (ehi)                            |
|  [ ] ( 0) Log fO2 (log bars) | 0.00000E+00| (fo2lgi)                         |
|  [x] ( 1) Couple (aux. sp.)  |Fe+++                   | (uredox)             |
|------------------------------------------------------------------------------|
```

To complete the option, concentrations or alternative constraints must be placed on both members of the couple in the basis species constraint block. For further discussion of redox options in EQ3NR, see Appendix B.1.3.6 and Appendix B.4.8.12–B.4.8.14.

3.3.9 Basis Species Constraints

The basis species constraint block then follows. In the present example, this is:

```
|------------------------------------------------------------------------------|
|Aqueous Basis Species/Constraint Species        |Conc., etc. |Units/Constraint|
| (uspeci(n)/ucospi(n))                          | (covali(n))|(ujf3(jflgi(n)))|
|------------------------------------------------------------------------------|
|Na+                                             | 1.07680E+04|mg/kg.sol       |
|K+                                              | 3.99100E+02|mg/kg.sol       |
|Ca++                                            | 4.12300E+02|mg/kg.sol       |
|Mg++                                            | 1.29180E+03|mg/kg.sol       |
|H+                                              | 8.22000E+00|pH              |
|HCO3-                                           | 2.02200E-03|Molality        |
|Cl-                                             | 1.93530E+04|mg/kg.sol       |
|SO4--                                           | 2.71200E+03|mg/kg.sol       |
|------------------------------------------------------------------------------|
* Valid jflag strings (ujf3(jflgi(n))) are:                                    *
*    Suppressed          Molality            Molarity                          *
*    mg/L                mg/kg.sol           Alk., eq/kg.H2O                   *
*    Alk., eq/L          Alk., eq/kg.sol     Alk., mg/L CaCO3                  *
*    Alk., mg/L HCO3-    Log activity        Log act combo                     *
*    Log mean act        pX                  pH                                *
*    pHCl                Hetero. equil.      Homo. equil.                      *
*    Make non-basis                                                            *
*------------------------------------------------------------------------------*
```

This block is used to specify concentrations or alternative constraint inputs for the basis species appearing in the problem (apart from H_2O and, usually, the fictive aqueous species $O_2(g)$). To know what are the basis species on a given supporting data file, consult the “slist” file produced by the EQPT data file preprocessor, or directly examine the relevant “data0” form of the data file. The fictive aqueous species $O_2(g)$ can appear in this block if the “–3” redox option switch discussed above is checked. However, this is not usually done.

If a chemical element is present in the model solution, it is generally represented by the corresponding strict basis species. Thus, 2.0 mg/L Fe in an analysis would normally be interpreted (and input to the code) as 2.0 mg/L Fe^{2+}. The user may treat an auxiliary basis species in one of two ways. It can be used as an active basis species by treating it as such in the basis species constraint block (e.g., by specifying a concentration or other constraint for it). Alternatively, it can be treated as a dependent (non-basis) species. This is the default condition (when no corresponding concentration or alternative constraint is applied).

This block consists of three parts. A header label comes first, and that is separated from the main body by a separator line. In the main body, the input line for each basis species in the main body contains three fields. The first contains the basis species name, the second an associated number (usually a concentration of some kind), and a string defining the type of constraint to apply. Thus for Na^+, a concentration of 10768.0 mg/kg.sol is specified. The possible strings appear below the main body in a group of comment lines. When such string inputs are required, such comment lines are generally found immediately below. If these types of comment lines (which are also used for other kinds of input line blocks) are missing from a file, they can be added by “prettifying” the file with XCIF3 or XCIF6 (Section 2.3).

The main body of the above block may include as few or as many basis species as desired, though the names in the list must match basis species found on the supporting thermodynamic data file that is used. In fact, the Version 8 code allows a non-basis species on that file to be promoted to basis species status simply by including it as such on the input file. It can have no dependent species. This capability may be useful for example for the odd organic species. In any event, the number of lines in the main body in the above example is not fixed; such lines may be added or deleted as needed. Note that a line can be retained for future reference by turning it into a comment line.

Most of the string options shown above simply define the unit of concentration matching the number that is given in the second field. For a discussion of such units, see for example Appendix B.1.2; brief summaries of the concentration units are given in Table 1. In certain instances, analytical data may be reported in a form that requires stoichiometric corrections (see Section 2.1.1.3) to match the basis species present on the supporting data file used (e.g., mg/L Si to mg/L SiO_2).

Table 1. Brief Definition of Concentration Units Usable in EQ3/6

EQ3/6 Concentration Unit	Equivalent Unit(s)	Definition
Molality	mg/kg.sol	Milligrams solute per kilogram of water solvent
Molarity	mg/L	Milligrams solute per liter aqueous solution
pX	–(Log activity)	$-\log_{10}$(activity of species X)

For a detailed discussion of other types of constraints that may be placed on basis species, see Appendix B.1.3.3–B.1.3.5. Note that any type of alkalinity constraint can only be applied to HCO_3^-. For a discussion of alkalinity and its various forms, see Appendix B.1.3.5. New to the Version 8 code is the ability to input alkalinity in units of equivalent ppm $CaCO_3$. Similarly, pH can only be applied to H^+. Note that pX is analogous to pH, and applied to H^+, it is the pH. Applied to "X", it is pX [e.g., for Cl^-, it is $pCl = -\log a(Cl^-)$].

The "Suppressed" option can only be applied to auxiliary basis species. This is equivalent to entering a zero concentration for such a species. When an auxiliary basis species is suppressed in this manner, it and its dependent species are eliminated from the calculated model. When the "Suppressed" option is employed, the field where a concentration is input is ignored. If this option is employed for a strict basis species, an error message is generated.

If no input is made for an auxiliary basis species, it is treated by default as a dependent species (usually of a strict basis species, but possibly of another auxiliary basis species, depending on the linking defined by the associated reaction on the supporting data file). Using the "Make non-basis" option does the same thing, albeit more explicitly. This option can only be applied to an auxiliary basis species. If this option is employed for a strict basis species, an error message is generated.

There is an asymmetry between strict and auxiliary basis species in the way that mass balances are treated. Consider the case of N species, assuming that $NH_3(aq)$ is the strict basis species and NO_3^- and NO_2^- are the only auxiliary basis species. If a total concentration or equivalent constraint is provided for $NH_3(aq)$ but not for the other two, the mass balances for the other two (which are treated as non-basis species) are folded into that for the NH_3(aq). From an analytical chemistry point of view, this treatment makes little sense for a non-metallic element like nitrogen, though it would ordinarily be quite sensible for a metallic element like Fe. Nitrate and nitrite are not some form of ammonia, so this treatment in a speciation/solubility calculation may be problematic unless the contributions from these species are very small. Ordinarily, the user should enter separate concentrations or equivalent constraints for auxiliary basis species of non-metallic elements, suppress them, or at the very least check the calculated speciation for sensibility (e.g., has most or all the ammonia been "converted" to nitrate?)

The asymmetry affect comes in to play if a concentration or equivalent input is made for an auxiliary basis species but not for the corresponding strict basis species. The mass balance for the strict basis species is not folded into that for the auxiliary basis species. Nor are the mass balances for any other auxiliary basis species folded in unless those species are linked on the supporting data file directly to the auxiliary basis species in question via their associated reactions. If the user defines a model for which a strict basis species is "cut out" of the model, the EQ3NR will issue a warning message. Because special basis switching can exchange the roles of strict and auxiliary basis species, it is thus capable of affecting the model definition.

The "Log act combo" and "Log mean act" options apply to alternative constraints for an ionic basis species for which the number entered in lieu of a concentration corresponds to a combined thermodynamic activity function involving the ion and a specified ion of opposite charge. The Appendix B.1.3.4 discusses such functions, which apply to electrically neutral combinations of ions. An example of a "Log act combo" (log activity combination) is log $a(Ca^{2+})$ + 2 log $a(Cl^-)$, which could be used to constrain either Ca^{2+} or Cl^-. An example of "Log mean act" (log mean activity) is log $a_{\pm}(CaCl_2)$. Because the ion of opposite charge must be defined as part of the constraint, an extra line appears in the main body of the basis species constraints block. The following lines, taken from the EQ3NR test case library input file swlmahcl.3i, illustrate the use of the "Log mean act" option applied to H+:

```
|H+                                               |-4.33610E+00|Log mean act    |
|->|Cl-                                              | (ucospi(n))              |
```

Here the "`|->|`" on the second line serves to suggest indentation and a subordinate role to the line immediately above. Note that for a constraint of this type, a separate input must be made for the ion of opposite charge. In the present example, that line was:

```
|Cl-                                              | 1.93530E+04|mg/kg.sol       |
```

The "pHCl" option involves a similar type of quantity (pHCl = pH + pCl). However, this option may only be applied to H^+. A second line to specify the ion of opposite charge is unnecessary, as its identity is implicit in the definition of the quantity.

The "Homo. equil." (Homogeneous equilibrium) option can only be applied to auxiliary basis species (else the code generates an error message). The case is illustrated by the following situation. Suppose an aqueous solution contains both SO_4^{2-} and HS^-. An analytical technique measures the total concentration of the former, which is a strict basis species. This measurement excludes any contributions from the latter and any of its dependent species (such as H_2S(aq)). No measurement is available for the total concentration of HS^-, which happens to be an auxiliary basis species.

The total and individual concentrations of HS^- can be calculated from the input total concentration for SO_4^{2-} plus an input value for a redox variable (normally the log oxygen fugacity). When the input total concentration of SO_4^{2-} is distributed between SO_4^{2-} and its dependent species, HS^- and its dependent species are not included in the associated mass balance. Rather, they are included in a separate mass balance. This option does not require an extra line of input. Rather, it uses the reaction associated with HS^- on the supporting data file. Normally this reaction is one of oxidation of HS^- to SO_4^{2-}. If the reaction did not involve SO_4^{2-}, the HS^- would be calculated from what does appear in this reaction.

The “Hetero. equil.” (Heterogeneous equilibrium) option may be applied to any type of basis species. Equilibrium may be specified with either a mineral or a gas species. The basis species must appear in the reaction associated with the mineral or gas. A second input line is required to specify the identity of the mineral or gas. If a gas is used, the log fugacity must be entered in the “concentration” field. Otherwise, any number in the “concentration” field is ignored. In the following example (taken from the EQ3NR test case library input file swco2.3i), the concentration of HCO_3^- is constrained by a log fugacity of –3.5 for CO_2(g).

```
|HCO3-                                         |-3.50000E+00|Hetero. equil.  |
|->|CO2(g)                                         | (ucospi(n))             |
```

If equilibrium with a mineral, say calcite, were desired instead, one would change the above to:

```
|HCO3-                                          |     0.     |Hetero. equil.  |
|->|Calcite                                        | (ucospi(n))             |
```

This example would also require an input for Ca^{2+} in the basis species constraints block.

If the basis species does not appear in the reaction for the specified mineral or gas, the code writes an error message. A solid solution may be used as a specified mineral. In such a case, the basis species need appear only in the reaction associated with at least one end-member. The composition of the solid solution must be defined in terms of the mole fractions of the end-members. However, this specification is made in another type of input block. If a solid solution is used in this manner and the necessary composition is not in the input file, the code generates an error message.

3.3.10 Ion Exchanger Creation Flag

The next block contains a flag noting whether one or more ion exchanger creation blocks follow on this file. Such a creation block is rather long, and it was decided that it would be better to not require an unused block as a minimal element of an input file. The flag block has the following form:

```
*-----------------------------------------------------------------------------*
|Create Ion Exchangers  | (net)                                               |
|-----------------------------------------------------------------------------|
|Advisory: no exchanger creation blocks follow on this file.                  |
|Option: on further processing (writing a PICKUP file or running XCON3 on the |
|present file), force the inclusion of at least one such block (qgexsh):      |
|  [ ] (.true.)                                                               |
|-----------------------------------------------------------------------------|
```

Note that there is a checkbox for whether or not one or more creation blocks follow. If an input file named say abc.3i has this box unchecked and it is desired to add a blank creation block to the file, simply checking the box and running it through XCIF3 as follows:

```
>xcif3 8 8 D abc.3i
```

will put the blank creation block in the file.

3.3.11 Ion Exchanger Creation Blocks

If the box noted above is checked, one or more ion exchanger creation blocks follow. In the current example, that box is not checked and no such creation blocks are present. A block will be presented and discussed here, however. The following simple example is taken from the EQ3NR test case library input file swv1sx.3i:

```
|-----------------------------------------------------------------------------|
|Exchanger phase |Exchanger 1             | (ugexp(n))                        |
|-----------------------------------------------------------------------------|
|->|Mol. wt. (Z)    |  100.00    | (mwtges(n))                                |
|-----------------------------------------------------------------------------|
|->|Exchange model  |Vanselow                | (ugexmo(n))                    |
|-----------------------------------------------------------------------------|
|->|Ref. temp. (C)  |  25.000    | (tgexp(n))                                 |
|-----------------------------------------------------------------------------|
|->|Exchange site   |Site A  | (ugexj(j,n))                                   |
|-----------------------------------------------------------------------------|
|--->|Stoich. number  |  1.0000    | (cgexj(j,n))                             |
|-----------------------------------------------------------------------------|
|--->|Electr. charge  | -1.0000    | (zgexj(j,n))                             |
|-----------------------------------------------------------------------------|
|--->|Reaction        |Ca++ = Na+              | (ugexr(i,j,n)                |
|-----------------------------------------------------------------------------|
|----->|Parameter |Value       |Units   | (this is a table header)            |
|-----------------------------------------------------------------------------|
|----->|K func.   | 3.00000E+00|LogK/eq | (xlkgex(i,j,n), uxkgex(i,j,n))      |
|----->|DelH0r    | 0.00000E+00|kcal/eq | (xhfgex(i,j,n), uhfgex(i,j,n))      |
|----->|DelV0r    | 0.00000E+00|cm3/eq  | (xvfgex(i,j,n), uvfgex(i,j,n))      |
|-----------------------------------------------------------------------------|
* Valid exchange model strings (ugexmo(n)) are:                               *
*    Vanselow                    Gapon                                        *
*-----------------------------------------------------------------------------*
* Valid units strings (uxkgex(i,j,n)/uhfgex(i,j,n)/uvfgex(i,j,n)) are:        *
*    LogK/eq             kcal/eq             kJ/eq                            *
*    cm3/eq                                                                   *
*-----------------------------------------------------------------------------*
```

All the data in a creation block could be on a supporting data file instead. However, such a path was not chosen due in part to the somewhat experimental nature of the models but also in part to the wide range of possible exchangers.

The first necessary input is a name for the exchanger phase. In the example given here, that is simply “Exchanger 1”. Next, a molecular weight is specified (the value here of 100 g/mole is entirely arbitrary). After that, an exchange model type must be specified as one of several allowed strings. The allowed strings are noted in “standard” comment lines below the main block (note: when more than one creation block is present, these comment lines only appear after the last such block). Currently the only choices are “Vanselow” and “Gapon”. For a discussion of these models, see for example Viani and Bruton (1992) or Sposito (1989, pp. 182–184). The models are equivalent for homovalent exchange (e.g., Na^+ or K^+). For heterovalent exchange (e.g., Na^+ for Ca^{2+}), the Vanselow model is likely to be preferred (cf. Viani and Bruton, 1992).

A reference temperature for the thermodynamic properties is then required. This is normally 25°C, but could be any desired temperature.

The remainder of the main body of an ion exchanger creation block consists of one or more sub-blocks, one for each exchange site. In the present example, there is only one exchange site, called “Site A”. The stoichiometric number is the number of moles of this site per mole of exchanger phase. The electrical charge is the charge number for the unfilled site. In the present example, this is –1. Filling each instance of the site with a monovalent cation such as Na^+ then leads to electrical neutrality.

Next follows one or more sub-sub-blocks for the possible exchange reactions. In the present example, only one such reaction is present. The reaction is described in a shorthand, which in the present example is “$Ca^{++} = Na^+$”. This implies the reaction in which Ca^{2+} on the exchange site is replaced by Na^+. For each such reaction, a log K, enthalpy of reaction, and volume of reaction function is entered. For each value, a unit’s string must be supplied. These strings are given in standard comment lines below those for the exchange model types. Note that the three quantities are defined on a per equivalent basis. The “K func.” input must always use the string “LogK/eq”. The DelH0r (enthalpy) input can use either “kcal/eq” or “kJ/eq”, depending on the desired units. The DelV0r (volume) input must use the “cm3/eq” string, as the units must be cm^3/eq. The number of generic ion exchanger phases is presently limited to **10**. The number of sites (site types) for a given such phase is **4**. The number of species per such site is **12**. These are static dimensions.

3.3.12 Ion Exchanger Compositions

The next block describes ion exchanger compositions. A blank block is required at a minimum (e.g., whether or not any ion exchangers have been created). The blank form of the block is given by:

```
|------------------------------------------------------------------------------|
|Ion Exchanger Compositions      | (neti)                                      |
|------------------------------------------------------------------------------|
|Exchanger phase |None                    | (ugexpi(n))                        |
|------------------------------------------------------------------------------|
|->|Moles/kg.H2O    |  0.0000    | (cgexpi(n))                                 |
|------------------------------------------------------------------------------|
|->|Exchange site   |None    | (ugexji(j,n))                                   |
|------------------------------------------------------------------------------|
|--->|Exchange species        |Eq. frac.   | (this is a table header)        |
|------------------------------------------------------------------------------|
|--->|None                    | 0.00000E+00| (ugexsi(i,j,n), egexsi(i,j,n))  |
|------------------------------------------------------------------------------|
```

The name must match that of a created ion exchanger phase. The amount of exchanger is specified in moles/kg.H_2O ("molality"). A composition of each site in terms of exchange species may then given (or not, as discussed below), using as many lines as are needed.

A real example is more helpful to further discussion. The following one is again taken from the EQ3NR test case library input file swv1sx.3i:

```
|------------------------------------------------------------------------------|
|Ion Exchanger Compositions      | (neti)                                      |
|------------------------------------------------------------------------------|
|Exchanger phase |Exchanger 1             | (ugexpi(n))                        |
|------------------------------------------------------------------------------|
|->|Moles/kg.H2O    | 0.10000    | (cgexpi(n))                                 |
|------------------------------------------------------------------------------|
|->|Exchange site   |Site A  | (ugexji(j,n))                                   |
|------------------------------------------------------------------------------|
|--->|Exchange species        |Eq. frac.   | (this is a table header)        |
|------------------------------------------------------------------------------|
|--->|Na+                     | 0.00000E+00| (ugexsi(i,j,n), egexsi(i,j,n))  |
|--->|Ca++                    | 0.00000E+00| (ugexsi(i,j,n), egexsi(i,j,n))  |
|------------------------------------------------------------------------------|
```

Here for "Exchanger 1" the only real input is a concentration of 0.1 moles/kg.H_2O. The composition of the exchange site is not defined (zeroes are entered). This requires some explanation of what EQ3NR is expected to do in this instance. Essentially, the input for this problem is sufficient to completely define the aqueous solution composition, including the concentrations of aqueous Na^+ and Ca^{2+}. The amount of exchange substrate is also defined. What the code then does is calculate the composition of the exchange site (leaving the aqueous concentrations of exchangeable ions undisturbed).

If a non-zero composition is entered for an exchange site, the code operates in a different manner (thus solving a different problem). In that case, the specified composition rules, and the resulting exchanger phase may be out of equilibrium with the aqueous solution. The code will then calculate the degree of disequilibria.

A very important fact to keep in mind is that the model system created by EQ3NR consists of an aqueous solution at a minimum, supplemented by ion exchanger phases if any are created and the specified to be present in the system.

3.3.13 Solid Solution Compositions

Next comes a very similar type of block for solid solutions. The blank form of this is:

```
|------------------------------------------------------------------------------|
|Solid Solution Compositions     | (nxti)                                      |
|------------------------------------------------------------------------------|
|Solid Solution          |None                    | (usoli(n))                 |
|------------------------------------------------------------------------------|
|->|Component               |Mole frac.  | (this is a table header)            |
|------------------------------------------------------------------------------|
|->|None                    | 0.00000E+00| (umemi(i,n), xbari(i,n))            |
|------------------------------------------------------------------------------|
```

Solid solutions must be defined (created) on the supporting data file. Note that one cannot specify a mass or concentration for a solid solution, as one can for an ion exchanger phase. EQ3NR does not create a model system in which any concentration or mass of solid solutions is present. Entering a composition for a solid solution is useful for only two purposes. The first is to allow the solid solution to be used in defining a “Hetero. equil.” constraint in the basis species constraint block. The second is to facilitate the calculation of the degree of disequilibrium of such a composition with the aqueous solution (which may be zero if enough “Hetero. equil.” options are employed). The number of solid solutions for which such compositions may be entered is limited to **50**. This corresponds to a static dimension.

3.3.14 Alter/Suppression Options

Next comes the alter/suppress options block:

```
|------------------------------------------------------------------------------|
|Alter/Suppress Options  | (nxmod)                                             |
|------------------------------------------------------------------------------|
|Species                                       |Option          |Alter value |
| (uxmod(n))                                   |(ukxm(kxmod(n)))| (xlkmod(n))|
|------------------------------------------------------------------------------|
|None                                          |None            | 0.00000E+00|
|------------------------------------------------------------------------------|
* Valid alter/suppress strings (ukxm(kxmod(n))) are:                           *
*    Suppress            Replace             AugmentLogK                       *
*    AugmentG                                                                  *
*------------------------------------------------------------------------------*
```

This consists of a two-part header, a main body (here one line with “None” in the first field), followed by a standard block of comment lines giving the strings that may be used in the second field of a line in the main body.

The following made-up example is more illustrative:

```
|------------------------------------------------------------------------------|
|Alter/Suppress Options  | (nxmod)                                             |
|------------------------------------------------------------------------------|
|Species                                         |Option          |Alter value |
| (uxmod(n))                                     |(ukxm(kxmod(n)))| (xlkmod(n))|
|------------------------------------------------------------------------------|
|Phenol(aq)                                      |Suppress        | 0.00000E+00|
|Tridymite                                       |Suppress        | 0.00000E+00|
|Quartz                                          |Replace         |-3.90000E+00|
|Acetate                                         |AugmentLogK     | 0.10000E+00|
|Formate                                         |AugmentG        | 1.00000E+00|
|------------------------------------------------------------------------------|
* Valid alter/suppress strings (ukxm(kxmod(n))) are:                           *
*    Suppress            Replace             AugmentLogK                       *
*    AugmentG                                                                  *
*------------------------------------------------------------------------------*
```

Here the aqueous species Phenol(aq) and the mineral Tridymite are suppressed. Phenol(aq) therefore will not appear as a species in the computed EQ3NR model. However, EQ3NR does not create model systems in which minerals are actually present, so the option for Tridymite has no specific action here though, EQ3NR does pass on these options to EQ6. There, neither Phenol(aq) nor Tridymite would be allowed to form. Here also the log K value for the reaction associated with Quartz is replaced by the specified value (–3.9, in the third field on the Quartz line). This value pertains to the current run temperature. Here the log K value for Acetate is augmented by 0.1 units, and the log K for Formate is changed as if the Gibbs energy of Formate were incremented by 1 kcal/mol.

The number of alter/suppress options is limited to **100**. This corresponds to a static dimension.

3.3.15 Iopt (Model) Options

Beginning here, there follows four consecutive “superblocks” of option switches, the “iopt” model options (this subsection), the “iopg” activity coefficient options (§ 3.3.16), the “iopr” print options(§ 3.3.17), and the “iodb” debugging print options(§ 3.3.18). The full sets of these option switches are shared by EQ3NR and EQ6. Some pertain to both codes, others only to one or the other. Only the ones that pertain to a code appear on a corresponding input file. Prior to the version 8 code, EQ3NR and EQ6 had their own independent sets of these option switches.

The “iopt” model options superblock in the present example is given by:

```
|------------------------------------------------------------------------------|
|Iopt Model Option Switches ("( 0)" marks default choices)                     |
|------------------------------------------------------------------------------|
|iopt(4) - Solid Solutions:                                                    |
|  [x] ( 0) Ignore                                                             |
|  [ ] ( 1) Permit                                                             |
|------------------------------------------------------------------------------|
|iopt(11) - Auto Basis Switching in pre-N-R Optimization:                      |
|  [x] ( 0) Turn off                                                           |
|  [ ] ( 1) Turn on                                                            |
|------------------------------------------------------------------------------|
|iopt(17) - PICKUP File Options:                                               |
|  [ ] (-1) Don't write a PICKUP file                                          |
|  [x] ( 0) Write a PICKUP file                                                |
|------------------------------------------------------------------------------|
```

```
|iopt(19) - Advanced EQ3NR PICKUP File Options:                              |
|  [x] ( 0) Write a normal EQ3NR PICKUP file                                 |
|  [ ] ( 1) Write an EQ6 INPUT file with Quartz dissolving, relative rate law  |
|  [ ] ( 2) Write an EQ6 INPUT file with Albite dissolving, TST rate law      |
|  [ ] ( 3) Write an EQ6 INPUT file with Fluid 1 set up for fluid mixing      |
|-----------------------------------------------------------------------------|
```

Only the four switches shown above pertain to EQ3NR. Solid solutions can be turned off or on as desired. However, not all supporting data files contain any solid solutions. “Auto Basis Switching in pre-N-R Optimization” should generally be turned off as shown. This option allows automatic basis switching in the pre-Newton-Raphson stage of the calculations. The PICKUP file (sample in Appendix C) option should be turned on if the user intends to feed the resulting calculation model to EQ6. If so, one of the “Advanced PICKUP File Options” may be helpful. The “normal” PICKUP file corresponds to just the bottom half of an EQ6 input file (this will be discussed later in regard to the EQ6 input file). Any of the advanced options results in the writing of a full EQ6 input file. The resulting top half of it can then be used as a template. Option 3 works best when two EQ3NR problems are stacked on a single input file. The aqueous solution in the first problem is then defined in the top half of the PICKUP file as a “reactant” to be reacted (mixed) with the aqueous solution in the second problem.

3.3.16 Iopg (Activity Coefficient) Options

This is followed by the “iopg” activity coefficient options superblock. In the present example, this is given by:

```
|-----------------------------------------------------------------------------|
|Iopg Activity Coefficient Option Switches ("( 0)" marks default choices)     |
|-----------------------------------------------------------------------------|
|iopg(1) - Aqueous Species Activity Coefficient Model:                        |
|  [ ] (-1) The Davies equation                                               |
|  [x] ( 0) The B-dot equation                                                |
|  [ ] ( 1) Pitzer's equations                                                |
|  [ ] ( 2) HC + DH equations                                                 |
|-----------------------------------------------------------------------------|
|iopg(2) - Choice of pH Scale (Rescales Activity Coefficients):               |
|  [ ] (-1) "Internal" pH scale (no rescaling)                                |
|  [x] ( 0) NBS pH scale (uses the Bates-Guggenheim equation)                 |
|  [ ] ( 1) Mesmer pH scale (numerically, pH = -log m(H+))                    |
|-----------------------------------------------------------------------------|
```

The first switch chooses the activity coefficient model. These activity models are explained and defined in detail in Appendix B.2. It must be consistent with the chosen supporting data file, else the code will issue an error message (actually, RUNEQ3 or RUNEQ6 will filter out any inconsistent input files, such that they are not run, and then list them). Note that Option 2 (“HC + DH equations”) is currently inoperative and is not a valid choice.

The second switch handles computed ionic activity coefficient scales for consistency with the specified pH scale. Normally this should be the default (NBS pH scale). The “iopg” options are also shared by EQ3NR and EQ6, as are the “iopr” and “iodb” options described below.

3.3.17 Iopr (Print) Options

After that is the “iopr” print options superblock. In the present example, this is given by:

```
|------------------------------------------------------------------------------|
|Iopr Print Option Switches ("( 0)" marks default choices)                     |
|------------------------------------------------------------------------------|
|iopr(1) - Print All Species Read from the Data File:                          |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print                                                              |
|------------------------------------------------------------------------------|
|iopr(2) - Print All Reactions:                                                |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print the reactions                                                |
|  [ ] ( 2) Print the reactions and log K values                               |
|  [ ] ( 3) Print the reactions, log K values, and associated data             |
|------------------------------------------------------------------------------|
|iopr(3) - Print the Aqueous Species Hard Core Diameters:                      |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print                                                              |
|------------------------------------------------------------------------------|
|iopr(4) - Print a Table of Aqueous Species Concentrations, Activities, etc.:  |
|  [ ] (-3) Omit species with molalities < 1.e-8                               |
|  [ ] (-2) Omit species with molalities < 1.e-12                              |
|  [ ] (-1) Omit species with molalities < 1.e-20                              |
|  [x] ( 0) Omit species with molalities < 1.e-100                             |
|  [ ] ( 1) Include all species                                                |
|------------------------------------------------------------------------------|
|iopr(5) - Print a Table of Aqueous Species/H+ Activity Ratios:                |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print cation/H+ activity ratios only                               |
|  [ ] ( 2) Print cation/H+ and anion/H+ activity ratios                       |
|  [ ] ( 3) Print ion/H+ activity ratios and neutral species activities        |
|------------------------------------------------------------------------------|
|iopr(6) - Print a Table of Aqueous Mass Balance Percentages:                  |
|  [ ] (-1) Don't print                                                        |
|  [x] ( 0) Print those species comprising at least 99% of each mass balance   |
|  [ ] ( 1) Print all contributing species                                     |
|------------------------------------------------------------------------------|
|iopr(7) - Print Tables of Saturation Indices and Affinities:                  |
|  [ ] (-1) Don't print                                                        |
|  [x] ( 0) Print, omitting those phases undersaturated by more than 10 kcal   |
|  [ ] ( 1) Print for all phases                                               |
|------------------------------------------------------------------------------|
|iopr(8) - Print a Table of Fugacities:                                        |
|  [ ] (-1) Don't print                                                        |
|  [x] ( 0) Print                                                              |
|------------------------------------------------------------------------------|
|iopr(9) - Print a Table of Mean Molal Activity Coefficients:                  |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print                                                              |
|------------------------------------------------------------------------------|
|iopr(10) - Print a Tabulation of the Pitzer Interaction Coefficients:         |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print a summary tabulation                                         |
|  [ ] ( 2) Print a more detailed tabulation                                   |
|------------------------------------------------------------------------------|
|iopr(17) - PICKUP file format ("W" or "D"):                                   |
|  [x] ( 0) Use the format of the INPUT file                                   |
|  [ ] ( 1) Use "W" format                                                     |
|  [ ] ( 2) Use "D" format                                                     |
|------------------------------------------------------------------------------|
```

These print switches are mainly used to control the degree of output generated by the code. In most instances, the most pleasing results will be obtained by using the default values.

3.3.18 Iodb (Debug Print) Options

Next comes the “iodb” debugging print options superblock. In the present example, this is given by:

```
|------------------------------------------------------------------------------|
|Iodb Debugging Print Option Switches ("( 0)" marks default choices)           |
|------------------------------------------------------------------------------|
|iodb(1) - Print General Diagnostic Messages:                                  |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print Level 1 diagnostic messages                                  |
|  [ ] ( 2) Print Level 1 and Level 2 diagnostic messages                      |
|------------------------------------------------------------------------------|
|iodb(3) - Print Pre-Newton-Raphson Optimization Information:                  |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print summary information                                          |
|  [ ] ( 2) Print detailed information (including the beta and del vectors)    |
|  [ ] ( 3) Print more detailed information (including matrix equations)       |
|  [ ] ( 4) Print most detailed information (including activity coefficients)  |
|------------------------------------------------------------------------------|
|iodb(4) - Print Newton-Raphson Iteration Information:                         |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print summary information                                          |
|  [ ] ( 2) Print detailed information (including the beta and del vectors)    |
|  [ ] ( 3) Print more detailed information (including the Jacobian)           |
|  [ ] ( 4) Print most detailed information (including activity coefficients)  |
|------------------------------------------------------------------------------|
|iodb(6) - Print Details of Hypothetical Affinity Calculations:                |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print summary information                                          |
|  [ ] ( 2) Print detailed information                                         |
|------------------------------------------------------------------------------|
```

These option switches should be turned on only to assist debugging or to observe numerical convergence processes. If a run fails and no useful diagnostics are obtained (normally a rare occurrence), the user should try setting the first switch to Option 1 or 2 and re-running the problem.

3.3.19 Numerical Parameters

The next block contains values for some numerical parameters:

```
|------------------------------------------------------------------------------|
|Numerical Parameters                                                          |
|------------------------------------------------------------------------------|
| Beta convergence tolerance      | 0.00000E+00| (tolbt)                       |
| Del convergence tolerance       | 0.00000E+00| (toldl)                       |
| Max. Number of N-R Iterations   |   0        | (itermx)                      |
|------------------------------------------------------------------------------|
```

Ordinarily, the user should accept the default values. The default for the two convergence tolerances is 1 x 10^{-6}. The default value for the maximum number of Newton-Raphson iterations is 200. Should a run fail to converge, there is a rather limited possibility of coaxing it to work by using larger values for these parameters. For example, one might try loosening the two conversion tolerances to say 1 x 10^{-5}, or increasing the maximum number of Newton-Raphson iterations to say 500.

Further explaination of the several techniques used to aid convergence are discussed in Appendix B.4.5.

3.3.20 Ordinary Basis Switching

Then comes a block for ordinary basis switching:

```
|------------------------------------------------------------------------------|
|Ordinary Basis Switches (for numerical purposes only)    | (nobswt)           |
|------------------------------------------------------------------------------|
|Replace |None                                             | (uobsw(1,n))       |
|   with |None                                             | (uobsw(2,n))       |
|------------------------------------------------------------------------------|
```

This block is similar to that for special basis switching. Ordinary basis switching cannot affect model definition. It can affect only numerical behavior. This type of switching involves the replace of a basis species by another species; typically a dependent species that dominates the mass balance associated with that basis species. For example, replacing UO_2^{2+} by $UO_2(CO_3)_2^{2-}$ might aid convergence properties (note: in the Version 8 code, the mass balance will still be defined in terms of the original basis species, here UO_2^{2+}). Users will ordinarily have no need for this feature. Recall that the iopt(11) switch allows automatic basis switching. Only the number of basis species pertinent to a given problem limits the number of ordinary basis switches. See Section 2.1.1 “Basis Species: Key Concepts” for further explanation of basis species.

3.3.21 Saturation Flag Tolerance

A block for the saturation flag tolerance follows:

```
|------------------------------------------------------------------------------|
|Sat. flag tolerance     | 0.00000E+00| (tolspf)                               |
|------------------------------------------------------------------------------|
```

This tolerance controls the writing of the string flags “SATD” (saturated) and “SSATD” (supersaturated) in the saturation state tables written on the output file. If the thermodynamic affinity (kcal) of a species is within ±tolspf, the species is flagged as saturated. If that affinity is greater than +tolspf, the species is flagged as supersaturated. The default value is 0.0005. In fact, meaningful judgment of whether a species is truly saturated or supersaturated is more accurately accomplished on a case-by-case basis, accounting for uncertainties from a variety of sources. Thus, the exact value of tolspf tends not to have great meaning.

3.3.22 Aqueous Phase Scale Factor

This is followed in turn by the aqueous phase scale factor and an “end of problem” marker:

```
|------------------------------------------------------------------------------|
|Aq. Phase Scale Factor  | 1.00000E+00| (scamas)                               |
|------------------------------------------------------------------------------|
|End of problem                                                                |
|------------------------------------------------------------------------------|
```

This scale factor determines the absolute model system mass written on the EQ3NR PICKUP file. The default value of the scale factor is 1.0. That results in the system being scaled to 1 kg of solvent water. Choosing a scale factor of 0.1 would result in it being scaled to 0.1 kg of solvent water.

3.4 THE EQ6 INPUT FILE

An EQ6 input file is longer and more complex than the one for EQ3NR. However, there are many common elements. Furthermore, an EQ6 an input file is divided into top and bottom halves. The bottom half is always created as either an EQ3NR "pickup" file or as part of an EQ6 "pickup" file. Ordinarily, the user should never edit the bottom half of an EQ6 inputs file. There is a possible exception: suppressing minerals in the alter/suppress options block (the same block that appears on the EQ3NR input file, Section 3.3.14). That said, users should be able to copy and paste top and bottom halves together to create a new EQ6 input file. Such operations can be done using WordPad or another text editor, or by using command line copy commands. These are basic operating system (OS) operations; consult the documentation for your OS for detailed instructions if needed.

The EQ6 user is expected to be familiar with the basic concepts and terminology of reaction path modeling. Much detailed conceptual and computational information can be found in Appendix D in sections D.1 and D.2 In particular, the user should fully understand the concepts of reaction progress (cf. Appendix D.1.3.1) and reaction rates and mass transfer (cf. Appendix D.1.3.2). The user should also have a good familiarity with the types of rate laws commonly employed in geochemical modeling (cf. Appendix D.1.3.3).

The following is an example of an EQ6 input file (in menu-style or "D" format). This is the file micro.6i from the EQ6 test case library, in which the mineral microline reacted with an HCl solution having an initial pH of 4. This is presented here in its entirety (occupying about seven and a half pages) to better acquaint the reader with the overall form. The various parts of this input file, which is a relatively simple but sufficiently typical example, will be discussed below.

Note the part:

```
|------------------------------------------------------------------------------|
* Start of the bottom half of the INPUT file                                   *
*------------------------------------------------------------------------------*
| Secondary Title        | (utitl2(n))                                         |
|------------------------------------------------------------------------------|
```

which occurs about halfway through the file. This marks the start of the bottom half of the file.

```
|------------------------------------------------------------------------------|
| Main Title             | (utitl1(n))                                         |
|------------------------------------------------------------------------------|
|EQ6 input file name= micro.6i                                                 |
|Description= "Microcline dissolution in pH 4 HCl"                             |
|Version level= 8.0                                                            |
|Revised 02/18/97    Revisor= T.J. Wolery                                      |
|This is part of the EQ3/6 Test Case Library                                   |
|                                                                              |
|  React "Maximum_Microcline" (KAlSi3O8) with a pH 4.0 HCl solution at 25C.    |
|There is no time frame in this calculation. Precipitation of quartz, chal-    |
|cedony, and tridymite is prevented by means of nxmod suppress options that    |
|are inherited from the initializing EQ3NR input file, ph4hcl.3i. This is      |
|test problem 3 of INTERA (1983, p. 65-73); it is similar to test problem 5    |
|of Parkhurst, Thorstenson, and Plummer (1980). INTERA (1983) reported that    |
|the product minerals formed were gibbsite, kaolinite, and muscovite. The run  |
|terminates when the solution becomes saturated with microcline.               |
|                                                                              |
|  The original problem called for suppression of only quartz. Chalcedony and  |
|tridymite were apparently not on the data file used by INTERA (1983).         |
|                                                                              |
|  Purpose: to compare against results obtained in a previous comparison of    |
|EQ3/6 with PHREEQE (Parkhurst, Thorstenson, and Plummer, 1980) made by        |
|INTERA (1983).                                                                |
|                                                                              |
|  This problem has no redox aspect. The option switch iopt(15) is set to 1    |
|to indicate this to the code.                                                 |
|                                                                              |
|                                                                              |
|                            References                                        |
|                                                                              |
|INTERA Environmental Consultants, Inc., 1983, Geochemical Models Suitable     |
|  for Performance Assessment of Nuclear Waste Storage: Comparison of          |
|  PHREEQE and EQ3/EQ6: Office of Nuclear Waste Isolation, Battelle Project    |
|  Management Division, Columbus, Ohio, ONWI-473, 114 p.                       |
|                                                                              |
|Parkhurst, D.L., Thorstenson, D.C., and Plummer, L.N., 1980, PHREEQE-         |
|  A Computer Program for Geochemical Calculations: Water Resources            |
|  Investigations Report 80-96, U.S. Geological Survey, Reston, Virginia,      |
|  210 p.                                                                      |
|                                                                              |
|------------------------------------------------------------------------------|
|Temperature option (jtemp):                                                   |
|  [x] ( 0) Constant temperature:                                              |
|             Value (C)         | 2.50000E+01| (tempcb)                         |
|  [ ] ( 1) Linear tracking in Xi:                                             |
|             Base Value (C)    | 0.00000E+00| (tempcb)                         |
|             Derivative        | 0.00000E+00| (ttk(1))                         |
|  [ ] ( 2) Linear tracking in time:                                           |
|             Base Value (C)    | 0.00000E+00| (tempcb)                         |
|             Derivative        | 0.00000E+00| (ttk(1))                         |
|  [ ] ( 3) Fluid mixing tracking (fluid 2 = special reactant):                |
|             T of fluid 1 (C)  | 0.00000E+00| (tempcb)                         |
|             T of fluid 2 (C)  | 0.00000E+00| (ttk(2))                         |
|             Mass ratio factor | 0.00000E+00| (ttk(1))                         |
|------------------------------------------------------------------------------|
|Pressure option (jpress):                                                     |
|  [x] ( 0) Follow the data file reference pressure curve                      |
|  [ ] ( 1) Follow the 1.013-bar/steam-saturation curve                        |
|  [ ] ( 2) Constant pressure:                                                 |
|             Value (bars)      | 0.00000E+00| (pressb)                         |
|  [ ] ( 3) Linear tracking in Xi:                                             |
|             Base Value (bars) | 0.00000E+00| (pressb)                         |
|             Derivative        | 0.00000E+00| (ptk(1))                         |
|  [ ] ( 4) Linear tracking in time:                                           |
|             Base Value (bars) | 0.00000E+00| (pressb)                         |
|             Derivative        | 0.00000E+00| (ptk(1))                         |
|------------------------------------------------------------------------------|
```

```
|Reactants (Irreversible Reactions) | (nrct)                                   |
|------------------------------------------------------------------------------|
|Reactant        |Maximum_Microcline      | (ureac(n))                         |
|------------------------------------------------------------------------------|
|->|Type         |Pure mineral            | (urcjco(jcode(n)))                 |
|------------------------------------------------------------------------------|
|->|Status       |Reacting                | (urcjre(jreac(n)))                 |
|------------------------------------------------------------------------------|
|->|Amount remaining (moles) | 1.00000E+00| (morr(n))                          |
|------------------------------------------------------------------------------|
|->|Amount destroyed (moles) | 0.00000E+00| (modr(n))                          |
|------------------------------------------------------------------------------|
|->|Surface area option (nsk(n)):                                             |
|->|  [x] ( 0) Constant surface area:                                         |
|->|             Value (cm2)       | 0.00000E+00| (sfcar(n))                   |
|->|  [ ] ( 1) Constant specific surface area:                                |
|->|             Value (cm2/g)     | 0.00000E+00| (ssfcar(n))                  |
|->|  [ ] ( 2) n**2/3 growth law- current surface area:                      |
|->|             Value (cm2)       | 0.00000E+00| (sfcar(n))                   |
|------------------------------------------------------------------------------|
|->|Surface area factor      | 0.00000E+00| (fkrc(n))                          |
|------------------------------------------------------------------------------|
|->|Forward rate law          |Relative rate equation  | (urcnrk(nrk(1,n)))    |
|------------------------------------------------------------------------------|
|--->|dXi(n)/dXi (mol/mol)      | 1.00000E+00| (rkb(1,1,n))                    |
|------------------------------------------------------------------------------|
|--->|d2Xi(n)/dXi2 (mol/mol2)   | 0.00000E+00| (rkb(2,1,n))                    |
|------------------------------------------------------------------------------|
|--->|d3Xi(n)/dXi3 (mol/mol3)   | 0.00000E+00| (rkb(3,1,n))                    |
|------------------------------------------------------------------------------|
|->|Backward rate law         |Partial equilibrium     | (urcnrk(nrk(2,n)))    |
|------------------------------------------------------------------------------|
* Valid reactant type strings (urcjco(jcode(n))) are:                          *
*    Pure mineral                Solid solution                                *
*    Special reactant            Aqueous species                               *
*    Gas species                 Generic ion exchanger                         *
*------------------------------------------------------------------------------*
* Valid reactant status strings (urcjre(jreac(n))) are:                        *
*    Saturated, reacting         Reacting                                      *
*    Exhausted                   Saturated, not reacting                       *
*------------------------------------------------------------------------------*
* Valid forward rate law strings (urcnrk(nrk(1,n))) are:                       *
*    Use backward rate law       Relative rate equation                        *
*    TST rate equation           Linear rate equation                          *
*------------------------------------------------------------------------------*
* Valid backward rate law strings (urcnrk(nrk(2,n))) are:                      *
*    Use forward rate law        Partial equilibrium                           *
*    Relative rate equation      TST rate equation                             *
*    Linear rate equation                                                      *
*------------------------------------------------------------------------------*
|Starting, minimum, and maximum values of key run parameters.                  |
|------------------------------------------------------------------------------|
|Starting Xi value        | 0.00000E+00| (xistti)                              |
|------------------------------------------------------------------------------|
|Maximum Xi value         | 1.00000E+00| (ximaxi)                              |
|------------------------------------------------------------------------------|
|Starting time (seconds)  | 0.00000E+00| (tistti)                              |
|------------------------------------------------------------------------------|
|Maximum time (seconds)   | 1.00000E+38| (timmxi)                              |
|------------------------------------------------------------------------------|
|Minimum value of pH      |-1.00000E+38| (phmini)                              |
|------------------------------------------------------------------------------|
|Maximum value of pH      | 1.00000E+38| (phmaxi)                              |
|------------------------------------------------------------------------------|
|Minimum value of Eh (v)  |-1.00000E+38| (ehmini)                              |
|------------------------------------------------------------------------------|
|Maximum value of Eh (v)  | 1.00000E+38| (ehmaxi)                              |
|------------------------------------------------------------------------------|
|Minimum value of log fO2 |-1.00000E+38| (o2mini)                              |
|------------------------------------------------------------------------------|
|Maximum value of log fO2 | 1.00000E+38| (o2maxi)                              |
```

```
|------------------------------------------------------------------------------|
|Minimum value of aw      |-1.00000E+38| (awmini)                              |
|------------------------------------------------------------------------------|
|Maximum value of aw      | 1.00000E+38| (awmaxi)                              |
|------------------------------------------------------------------------------|
|Maximum number of steps  |         200| (kstpmx)                              |
|------------------------------------------------------------------------------|
|Print interval parameters.                                                    |
|------------------------------------------------------------------------------|
|Xi print interval        | 1.00000E+00| (dlxprn)                              |
|------------------------------------------------------------------------------|
|Log Xi print interval    | 1.00000E+00| (dlxprl)                              |
|------------------------------------------------------------------------------|
|Time print interval      | 1.00000E+38| (dltprn)                              |
|------------------------------------------------------------------------------|
|Log time print interval  | 1.00000E+38| (dltprl)                              |
|------------------------------------------------------------------------------|
|pH print interval        | 1.00000E+38| (dlhprn)                              |
|------------------------------------------------------------------------------|
|Eh (v) print interval    | 1.00000E+38| (dleprn)                              |
|------------------------------------------------------------------------------|
|Log fO2 print interval   | 1.00000E+38| (dloprn)                              |
|------------------------------------------------------------------------------|
|aw print interval        | 1.00000E+38| (dlaprn)                              |
|------------------------------------------------------------------------------|
|Steps print interval     |         100| (ksppmx)                              |
|------------------------------------------------------------------------------|
|Plot interval parameters.                                                     |
|------------------------------------------------------------------------------|
|Xi plot interval         | 1.00000E+38| (dlxplo)                              |
|------------------------------------------------------------------------------|
|Log Xi plot interval     | 1.00000E+38| (dlxpll)                              |
|------------------------------------------------------------------------------|
|Time plot interval       | 1.00000E+38| (dltplo)                              |
|------------------------------------------------------------------------------|
|Log time plot interval   | 1.00000E+38| (dltpll)                              |
|------------------------------------------------------------------------------|
|pH plot interval         | 1.00000E+38| (dlhplo)                              |
|------------------------------------------------------------------------------|
|Eh (v) plot interval     | 1.00000E+38| (dleplo)                              |
|------------------------------------------------------------------------------|
|Log fO2 plot interval    | 1.00000E+38| (dloplo)                              |
|------------------------------------------------------------------------------|
|aw plot interval         | 1.00000E+38| (dlaplo)                              |
|------------------------------------------------------------------------------|
|Steps plot interval      |       10000| (ksplmx)                              |
|------------------------------------------------------------------------------|
|Iopt Model Option Switches ("( 0)" marks default choices)                     |
|------------------------------------------------------------------------------|
|iopt(1) - Physical System Model Selection:                                    |
|  [x] ( 0) Closed system                                                      |
|  [ ] ( 1) Titration system                                                   |
|  [ ] ( 2) Fluid-centered flow-through open system                            |
|------------------------------------------------------------------------------|
|iopt(2) - Kinetic Mode Selection:                                             |
|  [x] ( 0) Reaction progress mode (arbitrary kinetics)                        |
|  [ ] ( 1) Reaction progress/time mode (true kinetics)                        |
|------------------------------------------------------------------------------|
|iopt(3) - Phase Boundary Searches:                                            |
|  [x] ( 0) Search for phase boundaries and constrain the step size to match   |
|  [ ] ( 1) Search for phase boundaries and print their locations              |
|  [ ] ( 2) Don't search for phase boundaries                                  |
|------------------------------------------------------------------------------|
|iopt(4) - Solid Solutions:                                                    |
|  [x] ( 0) Ignore                                                             |
|  [ ] ( 1) Permit                                                             |
|------------------------------------------------------------------------------|
|iopt(5) - Clear the ES Solids Read from the INPUT File:                       |
|  [x] ( 0) Don't do it                                                        |
|  [ ] ( 1) Do it                                                              |
|------------------------------------------------------------------------------|
```

```
|iopt(6) - Clear the ES Solids at the Initial Value of Reaction Progress:      |
|  [x] ( 0) Don't do it                                                        |
|  [ ] ( 1) Do it                                                              |
|------------------------------------------------------------------------------|
|iopt(7) - Clear the ES Solids at the End of the Run:                          |
|  [x] ( 0) Don't do it                                                        |
|  [ ] ( 1) Do it                                                              |
|------------------------------------------------------------------------------|
|iopt(9) - Clear the PRS Solids Read from the INPUT file:                      |
|  [x] ( 0) Don't do it                                                        |
|  [ ] ( 1) Do it                                                              |
|------------------------------------------------------------------------------|
|iopt(10) - Clear the PRS Solids at the End of the Run:                        |
|  [x] ( 0) Don't do it                                                        |
|  [ ] ( 1) Do it, unless numerical problems cause early termination           |
|------------------------------------------------------------------------------|
|iopt(11) - Auto Basis Switching in pre-N-R Optimization:                      |
|  [x] ( 0) Turn off                                                           |
|  [ ] ( 1) Turn on                                                            |
|------------------------------------------------------------------------------|
|iopt(12) - Auto Basis Switching after Newton-Raphson Iteration:               |
|  [x] ( 0) Turn off                                                           |
|  [ ] ( 1) Turn on                                                            |
|------------------------------------------------------------------------------|
|iopt(13) - Calculational Mode Selection:                                      |
|  [x] ( 0) Normal path tracing                                                |
|  [ ] ( 1) Economy mode (if permissible)                                      |
|  [ ] ( 2) Super economy mode (if permissible)                                |
|------------------------------------------------------------------------------|
|iopt(14) - ODE Integrator Corrector Mode Selection:                           |
|  [x] ( 0) Allow Stiff and Simple Correctors                                  |
|  [ ] ( 1) Allow Only the Simple Corrector                                    |
|  [ ] ( 2) Allow Only the Stiff Corrector                                     |
|  [ ] ( 3) Allow No Correctors                                                |
|------------------------------------------------------------------------------|
|iopt(15) - Force the Suppression of All Redox Reactions:                      |
|  [ ] ( 0) Don't do it                                                        |
|  [x] ( 1) Do it                                                              |
|------------------------------------------------------------------------------|
|iopt(16) - BACKUP File Options:                                               |
|  [ ] (-1) Don't write a BACKUP file                                          |
|  [x] ( 0) Write BACKUP files                                                 |
|  [ ] ( 1) Write a sequential BACKUP file                                     |
|------------------------------------------------------------------------------|
|iopt(17) - PICKUP File Options:                                               |
|  [ ] (-1) Don't write a PICKUP file                                          |
|  [x] ( 0) Write a PICKUP file                                                |
|------------------------------------------------------------------------------|
|iopt(18) - TAB File Options:                                                  |
|  [ ] (-1) Don't write a TAB file                                             |
|  [x] ( 0) Write a TAB file                                                   |
|  [ ] ( 1) Write a TAB file, prepending TABX file data from a previous run    |
|------------------------------------------------------------------------------|
|iopt(20) - Advanced EQ6 PICKUP File Options:                                  |
|  [x] ( 0) Write a normal EQ6 PICKUP file                                     |
|  [ ] ( 1) Write an EQ6 INPUT file with Fluid 1 set up for fluid mixing       |
|------------------------------------------------------------------------------|
|Iopr Print Option Switches ("( 0)" marks default choices)                     |
|------------------------------------------------------------------------------|
|iopr(1) - Print All Species Read from the Data File:                          |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print                                                              |
|------------------------------------------------------------------------------|
|iopr(2) - Print All Reactions:                                                |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print the reactions                                                |
|  [ ] ( 2) Print the reactions and log K values                               |
|  [ ] ( 3) Print the reactions, log K values, and associated data             |
|------------------------------------------------------------------------------|
```

```
|iopr(3) - Print the Aqueous Species Hard Core Diameters:                      |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print                                                              |
|------------------------------------------------------------------------------|
|iopr(4) - Print a Table of Aqueous Species Concentrations, Activities, etc.:  |
|  [ ] (-3) Omit species with molalities < 1.e-8                               |
|  [ ] (-2) Omit species with molalities < 1.e-12                              |
|  [ ] (-1) Omit species with molalities < 1.e-20                              |
|  [x] ( 0) Omit species with molalities < 1.e-100                             |
|  [ ] ( 1) Include all species                                                |
|------------------------------------------------------------------------------|
|iopr(5) - Print a Table of Aqueous Species/H+ Activity Ratios:                |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print cation/H+ activity ratios only                               |
|  [ ] ( 2) Print cation/H+ and anion/H+ activity ratios                       |
|  [ ] ( 3) Print ion/H+ activity ratios and neutral species activities        |
|------------------------------------------------------------------------------|
|iopr(6) - Print a Table of Aqueous Mass Balance Percentages:                  |
|  [ ] (-1) Don't print                                                        |
|  [x] ( 0) Print those species comprising at least 99% of each mass balance   |
|  [ ] ( 1) Print all contributing species                                     |
|------------------------------------------------------------------------------|
|iopr(7) - Print Tables of Saturation Indices and Affinities:                  |
|  [ ] (-1) Don't print                                                        |
|  [x] ( 0) Print, omitting those phases undersaturated by more than 10 kcal   |
|  [ ] ( 1) Print for all phases                                               |
|------------------------------------------------------------------------------|
|iopr(8) - Print a Table of Fugacities:                                        |
|  [x] (-1) Don't print                                                        |
|  [ ] ( 0) Print                                                              |
|------------------------------------------------------------------------------|
|iopr(9) - Print a Table of Mean Molal Activity Coefficients:                  |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print                                                              |
|------------------------------------------------------------------------------|
|iopr(10) - Print a Tabulation of the Pitzer Interaction Coefficients:         |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print a summary tabulation                                         |
|  [ ] ( 2) Print a more detailed tabulation                                   |
|------------------------------------------------------------------------------|
|iopr(17) - PICKUP file format ("W" or "D"):                                   |
|  [x] ( 0) Use the format of the INPUT file                                   |
|  [ ] ( 1) Use "W" format                                                     |
|  [ ] ( 2) Use "D" format                                                     |
|------------------------------------------------------------------------------|
|Iodb Debugging Print Option Switches ("( 0)" marks default choices)           |
|------------------------------------------------------------------------------|
|iodb(1) - Print General Diagnostic Messages:                                  |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print Level 1 diagnostic messages                                  |
|  [ ] ( 2) Print Level 1 and Level 2 diagnostic messages                      |
|------------------------------------------------------------------------------|
|iodb(2) - Kinetics Related Diagnostic Messages:                               |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print Level 1 kinetics diagnostic messages                         |
|  [ ] ( 2) Print Level 1 and Level 2 kinetics diagnostic messages             |
|------------------------------------------------------------------------------|
|iodb(3) - Print Pre-Newton-Raphson Optimization Information:                  |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print summary information                                          |
|  [ ] ( 2) Print detailed information (including the beta and del vectors)    |
|  [ ] ( 3) Print more detailed information (including matrix equations)       |
|  [ ] ( 4) Print most detailed information (including activity coefficients)  |
|------------------------------------------------------------------------------|
|iodb(4) - Print Newton-Raphson Iteration Information:                         |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print summary information                                          |
|  [ ] ( 2) Print detailed information (including the beta and del vectors)    |
|  [ ] ( 3) Print more detailed information (including the Jacobian)           |
|  [ ] ( 4) Print most detailed information (including activity coefficients)  |
|------------------------------------------------------------------------------|
```

```
|iodb(5) - Print Step-Size and Order Selection:                                |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print summary information                                          |
|  [ ] ( 2) Print detailed information                                         |
|------------------------------------------------------------------------------|
|iodb(6) - Print Details of Hypothetical Affinity Calculations:                |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print summary information                                          |
|  [ ] ( 2) Print detailed information                                         |
|------------------------------------------------------------------------------|
|iodb(7) - Print General Search (e.g., for a phase boundary) Information:      |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print summary information                                          |
|------------------------------------------------------------------------------|
|iodb(8) - Print ODE Corrector Iteration Information:                          |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print summary information                                          |
|  [ ] ( 2) Print detailed information (including the betar and delvcr vectors)|
|------------------------------------------------------------------------------|
|Mineral Sub-Set Selection Suppression Options | (nxopt)                       |
|------------------------------------------------------------------------------|
|Option  |Sub-Set Defining Species| (this is a table header)                   |
|------------------------------------------------------------------------------|
|None    |None                    | (uxopt(n), uxcat(n))                       |
|------------------------------------------------------------------------------|
* Valid mineral sub-set selection suppression option strings (uxopt(n)) are:   *
*    None        All         Alwith      Allwith                               *
*------------------------------------------------------------------------------*
|Exceptions to the Mineral Sub-Set Selection Suppression Options | (nxopex)    |
|------------------------------------------------------------------------------|
|Mineral                 | (this is a table header)                            |
|------------------------------------------------------------------------------|
|None                    | (uxopex(n))                                         |
|------------------------------------------------------------------------------|
|Fixed Fugacity Options | (nffg)                                               |
|------------------------------------------------------------------------------|
|Gas                     |Moles to Add |Log Fugacity | --                      |
| (uffg(n))              | (moffg(n))  | (xlkffg(n)) | --                      |
|------------------------------------------------------------------------------|
|None                    |  0.00000E+00|  0.00000E+00| --                      |
|------------------------------------------------------------------------------|
|Numerical Parameters                                                          |
|------------------------------------------------------------------------------|
|Max. finite-difference order               |   6        | (nordmx)            |
|Beta convergence tolerance                 | 0.00000E+00| (tolbt)             |
|Del convergence tolerance                  | 0.00000E+00| (toldl)             |
|Max. No. of N-R iterations                 |   0        | (itermx)            |
|Search/find convergence tolerance          | 0.00000E+00| (tolxsf)            |
|Saturation tolerance                       | 0.00000E+00| (tolsat)            |
|Max. No. of Phase Assemblage Tries         |   0        | (ntrymx)            |
|Zero order step size (in Xi)               | 0.00000E+00| (dlxmx0)            |
|Max. interval in Xi between PRS transfers  | 0.00000E+00| (dlxdmp)            |
|------------------------------------------------------------------------------|
* Start of the bottom half of the INPUT file                                   *
*------------------------------------------------------------------------------*
| Secondary Title        | (utitl2(n))                                         |
|------------------------------------------------------------------------------|
|EQ3NR input file name= ph4hcl.3i                                              |
|Description= "A pH 4 HCl solution, with traces of K, Al, and Si"              |
|Version level= 8.0                                                            |
|Revised 02/14/97    Revisor= T.J. Wolery                                      |
|This is part of the EQ3/6 Test Case Library                                   |
|                                                                              |
|  Dilute HCl solution, pH 4.00, with traces of potassium, aluminum, and       |
|silica. This problem is part of test problem 3 of INTERA (1983), who report   |
|a comparison study of EQ3/6 with PHREEQE (Parkhurst, Thorstenson, and         |
|Plummer, 1980). Note that precipitation of quartz, chalcedony, and tridymite  |
|is prevented by means of nxmod suppress options. This has no effect on the    |
|EQ3NR calculation. It is simply needed for the subsequent EQ6 problem. EQ3NR  |
|passes this on to EQ6 on the PICKUP file.                                     |
|                                                                              |
```

```
|  Purpose: to provide a PICKUP file for construction of the EQ6 test cases    |
|micro.6i and microft.6i.                                                      |
|                                                                              |
|  This problem is actually redox-indifferent. The auxiliary basis species     |
|O2(aq) and H2(aq) have therefore been suppressed. The EQ6 option switch       |
|iopt(15) should be set to 1.                                                  |
|                                                                              |
|                                                                              |
|                            References                                        |
|                                                                              |
|INTERA Environmental Consultants, Inc., 1983, Geochemical Models Suitable     |
|  for Performance Assessment of Nuclear Waste Storage: Comparison of          |
|  PHREEQE and EQ3/EQ6: Office of Nuclear Waste Isolation, Battelle Project    |
|  Management Division, Columbus, Ohio, ONWI-473, 114 p.                       |
|                                                                              |
|Parkhurst, D.L., Thorstenson, D.C., and Plummer, L.N., 1980, PHREEQE-         |
|  A Computer Program for Geochemical Calculations: Water Resources            |
|  Investigations Report 80-96, U.S. Geological Survey, Reston, Virginia,      |
|  210 p.                                                                      |
|                                                                              |
|------------------------------------------------------------------------------|
|Special Basis Switches (for model definition only)       | (nsbswt)           |
|------------------------------------------------------------------------------|
|Replace |None                                            | (usbsw(1,n))       |
|   with |None                                            | (usbsw(2,n))       |
|------------------------------------------------------------------------------|
|Original temperature (C) | 2.50000E+01| (tempci)                              |
|------------------------------------------------------------------------------|
|Original pressure (bars) | 1.01320E+00| (pressi)                              |
|------------------------------------------------------------------------------|
|Create Ion Exchangers  | (net)                                                |
|------------------------------------------------------------------------------|
|Advisory: no exchanger creation blocks follow on this file.                   |
|Option: on further processing (writing a PICKUP file or running XCON6 on the  |
|present file), force the inclusion of at least one such block (qgexsh):       |
|  [ ] (.true.)                                                                |
|------------------------------------------------------------------------------|
|Alter/Suppress Options  | (nxmod)                                             |
|------------------------------------------------------------------------------|
|Species                                         |Option          |Alter value |
| (uxmod(n))                                     |(ukxm(kxmod(n)))| (xlkmod(n))|
|------------------------------------------------------------------------------|
|Quartz                                          |Suppress        | 0.00000E+00|
|Chalcedony                                      |Suppress        | 0.00000E+00|
|Tridymite                                       |Suppress        | 0.00000E+00|
|------------------------------------------------------------------------------|
* Valid alter/suppress strings (ukxm(kxmod(n))) are:                           *
*    Suppress            Replace             AugmentLogK                       *
*    AugmentG                                                                  *
*------------------------------------------------------------------------------*
|Iopg Activity Coefficient Option Switches ("( 0)" marks default choices)      |
|------------------------------------------------------------------------------|
|iopg(1) - Aqueous Species Activity Coefficient Model:                         |
|  [ ] (-1) The Davies equation                                                |
|  [x] ( 0) The B-dot equation                                                 |
|  [ ] ( 1) Pitzer's equations                                                 |
|  [ ] ( 2) HC + DH equations                                                  |
|------------------------------------------------------------------------------|
|iopg(2) - Choice of pH Scale (Rescales Activity Coefficients):                |
|  [ ] (-1) "Internal" pH scale (no rescaling)                                 |
|  [x] ( 0) NBS pH scale (uses the Bates-Guggenheim equation)                  |
|  [ ] ( 1) Mesmer pH scale (numerically, pH = -log m(H+))                     |
|------------------------------------------------------------------------------|
|Matrix Index Limits                                                           |
|------------------------------------------------------------------------------|
|No. of chem. elements   |    6| (kct)                                         |
|No. of basis species    |    7| (kbt)                                         |
|Index of last pure min. |    7| (kmt)                                         |
|Index of last sol-sol.  |    7| (kxt)                                         |
|Matrix size             |    7| (kdim)                                        |
|PRS data flag           |    0| (kprs)                                        |
```

```
|------------------------------------------------------------------------------|
|Mass Balance Species (Matrix Row Variables)     |Units/Constraint| --         |
| (ubmtbi(n))                                    |(ujf6(jflgi(n)))| --         |
|------------------------------------------------------------------------------|
|H2O                     Aqueous solution        |Moles           | --         |
|Al+++                   Aqueous solution        |Moles           | --         |
|Cl-                     Aqueous solution        |Moles           | --         |
|H+                      Aqueous solution        |Moles           | --         |
|K+                      Aqueous solution        |Moles           | --         |
|SiO2(aq)                Aqueous solution        |Moles           | --         |
|O2(g)                   Aqueous solution        |Moles           | --         |
|------------------------------------------------------------------------------|
* Valid jflag strings (ujf6(jflgi(n))) are:                                    *
*    Moles               Make non-basis                                        *
*------------------------------------------------------------------------------*
|Mass Balance Totals (moles)                                                   |
|------------------------------------------------------------------------------|
|Basis species (info. only)      |Equilibrium System    |Aqueous Solution      |
| (ubmtbi(n))                    | (mtbi(n))            | (mtbaqi(n))          |
|------------------------------------------------------------------------------|
|H2O                     Aqueous | 5.550843506189440E+01| 5.550843506189440E+01|
|Al+++                   Aqueous | 1.000000000000050E-12| 1.000000000000050E-12|
|Cl-                     Aqueous | 1.011629092597910E-04| 1.011629092597910E-04|
|H+                      Aqueous | 1.011629052597910E-04| 1.011629052597910E-04|
|K+                      Aqueous | 1.000000000000001E-12| 1.000000000000001E-12|
|SiO2(aq)                Aqueous | 9.999999999999994E-13| 9.999999999999994E-13|
|O2(g)                   Aqueous | 2.331489422794826E-18| 2.331489422794826E-18|
|Electrical imbalance            |-4.065758146820642E-20|-4.065758146820642E-20|
|------------------------------------------------------------------------------|
|Ordinary Basis Switches (for numerical purposes only)    | (nobswt)           |
|------------------------------------------------------------------------------|
|Replace |None                                            | (uobsw(1,n))       |
|   with |None                                            | (uobsw(2,n))       |
|------------------------------------------------------------------------------|
|Matrix Column Variables and Values                                            |
|------------------------------------------------------------------------------|
|Basis species (uzveci(n))                       |Log moles (zvclgi(n)) | --   |
|------------------------------------------------------------------------------|
|H2O                     Aqueous solution        | 1.744358983526984E+00| --   |
|Al+++                   Aqueous solution        |-1.204400338249425E+01| --   |
|Cl-                     Aqueous solution        |-3.994987866885238E+00| --   |
|H+                      Aqueous solution        |-3.994987444367077E+00| --   |
|K+                      Aqueous solution        |-1.200000137411354E+01| --   |
|SiO2(aq)                Aqueous solution        |-1.200000049015453E+01| --   |
|O2(g)                   Aqueous solution        |-7.000000000000000E-01| --   |
|------------------------------------------------------------------------------|
|Phases and Species in the PRS                                                 |
|------------------------------------------------------------------------------|
|Phase           |None                    | (uprphi(n))                        |
|------------------------------------------------------------------------------|
|->|No. of Moles    | 0.000000000000000E+00| (mprphi(n))                       |
|------------------------------------------------------------------------------|
|--->|Species                 |No. of Moles          | --                      |
|--->| (uprspi(i,n))          | (mprspi(i,n))        | --                      |
|------------------------------------------------------------------------------|
|--->|None                    | 0.000000000000000E+00| --                      |
|------------------------------------------------------------------------------|
|End of problem                                                                |
|------------------------------------------------------------------------------|
```

The elements of an EQ6 input file are:

Top Half

Main Title

Temperature option

Pressure Option

Reactants Superblock:

For each of zero or more reactants

Reactant name

Type

Status

Amount remaining

Amount destroyed

Surface area option

Surface area factor

Forward (e.g., dissolution) rate law type and parameters

Backward (e.g., precipitation) rate law type and parameters

Run Parameter Superblock:

Starting, minimum, and maximum values of key run parameters:

Starting Xi (reaction progress variable) value

Maximim Xi value

Starting time (seconds)

Maximum time (seconds)

Minimum pH

Maximum pH

Minimum Eh (volts)

Maximum Eh (volts)

Minimum log fO_2

Maximum log fO_2

Minimum activity of water (a_w)

Maximum activity of water (a_w)

Maximum number of steps

Print interval parameters:

Xi print interval

Log Xi print interval

Time (seconds) print interval

Log time (seconds) print interval

  - pH print interval
  - Eh (volts) print interval
  - Log fO_2 print interval
  - Activity of water (aw) print interval
  - Steps print interval
  - Plot interval parameters:
  - Xi plot interval
  - Log Xi plot interval
  - Time (seconds) plot interval
  - Log time (seconds) plot interval
  - pH plot interval
  - Eh (volts) plot interval
  - Log fO_2 plot interval
  - Activity of water (aw) plot interval
  - Steps plot interval
- Iopt (model) options
- Iopr (print) options
- Iodb (debug print) options
- Mineral sub-set selection suppression options
- Fixed fugacity options
- Numerical parameters

Bottom Half

- Secondary Title
- Special basis switches
- Original temperature (C)
- Original pressure (bars)
- Ion exchanger creation flag
- *If the above flag is set to true*: one or more ion exchanger creation blocks
- Alter/suppression options
- Iopg (activity coefficient) options
- Matrix index limits

Mass balance species (matrix row values)

Mass balance totals

Ordinary basis switches

Matrix column variables and values

Phases and species in the PRS (physically removed system)

These will be considered in order.

3.4.1 Main Title

The main title block is intended to allow the user to document the problem defined on the input file. In the present example, the main title is:

```
|------------------------------------------------------------------------------|
| Main Title             | (utitl1(n))                                         |
|------------------------------------------------------------------------------|
|EQ6 input file name= micro.6i                                                 |
|Description= "Microcline dissolution in pH 4 HCl"                             |
|Version level= 8.0                                                            |
|Revised 02/18/97    Revisor= T.J. Wolery                                      |
|This is part of the EQ3/6 Test Case Library                                   |
|                                                                              |
|  React "Maximum_Microcline" (KAlSi3O8) with a pH 4.0 HCl solution at 25C.    |
|There is no time frame in this calculation. Precipitation of quartz, chal-    |
|cedony, and tridymite is prevented by means of nxmod suppress options that    |
|are inherited from the initializing EQ3NR input file, ph4hcl.3i. This is      |
|test problem 3 of INTERA (1983, p. 65-73); it is similar to test problem 5    |
|of Parkhurst, Thorstenson, and Plummer (1980). INTERA (1983) reported that    |
|the product minerals formed were gibbsite, kaolinite, and muscovite. The run  |
|terminates when the solution becomes saturated with microcline.               |
|                                                                              |
|  The original problem called for suppression of only quartz. Chalcedony and  |
|tridymite were apparently not on the data file used by INTERA (1983).         |
|                                                                              |
|  Purpose: to compare against results obtained in a previous comparison of    |
|EQ3/6 with PHREEQE (Parkhurst, Thorstenson, and Plummer, 1980) made by        |
|INTERA (1983).                                                                |
|                                                                              |
|  This problem has no redox aspect. The option switch iopt(15) is set to 1    |
|to indicate this to the code.                                                 |
|                                                                              |
|                                                                              |
|                            References                                        |
|                                                                              |
|INTERA Environmental Consultants, Inc., 1983, Geochemical Models Suitable     |
|  for Performance Assessment of Nuclear Waste Storage: Comparison of          |
|  PHREEQE and EQ3/EQ6: Office of Nuclear Waste Isolation, Battelle Project    |
|  Management Division, Columbus, Ohio, ONWI-473, 114 p.                       |
|                                                                              |
|Parkhurst, D.L., Thorstenson, D.C., and Plummer, L.N., 1980, PHREEQE-         |
|  A Computer Program for Geochemical Calculations: Water Resources            |
|  Investigations Report 80-96, U.S. Geological Survey, Reston, Virginia,      |
|  210 p.                                                                      |
|                                                                              |
|------------------------------------------------------------------------------|
```

This is analogous to the title on the EQ3NR input file (Section 3.3.1), with which the user should already be familiar. The special form of the first three lines (starting with ‘EQ6 input file name=” should be preserved, with users changing only the input file name on the first line and the short description on the second line. The use of the fourth line (with appropriate changes by the user) is encouraged. The string “This is part of the EQ3/6 Test Case Library.” should not be retained when the user uses such a file as a template for an input file that is not part of that library.

If the case-insensitive string “TABFILEASCSV” is included *anywhere* in the main title of an EQ6 input file, the EQ6 “tab” file will be output as a “.csv” (comma separated value) file instead of as the usual text file. Once renamed with a “csv” filename extension, Microsoft Excel or other spreadsheet application can open such a file.

The maximum number of lines in a title on an EQ6 input file is the same as for a title on an EQ3NR input file, **200**. This is a static dimension.

3.4.2 Temperature Option

A block for the temperature option follows this, which is this example is given by:

```
|------------------------------------------------------------------------------|
|Temperature option (jtemp):                                                   |
|  [x] ( 0) Constant temperature:                                              |
|             Value (C)         | 2.50000E+01| (tempcb)                         |
|  [ ] ( 1) Linear tracking in Xi:                                             |
|             Base Value (C)    | 0.00000E+00| (tempcb)                         |
|             Derivative        | 0.00000E+00| (ttk(1))                         |
|  [ ] ( 2) Linear tracking in time:                                           |
|             Base Value (C)    | 0.00000E+00| (tempcb)                         |
|             Derivative        | 0.00000E+00| (ttk(1))                         |
|  [ ] ( 3) Fluid mixing tracking (fluid 2 = special reactant):                |
|             T of fluid 1 (C)  | 0.00000E+00| (tempcb)                         |
|             T of fluid 2 (C)  | 0.00000E+00| (ttk(2))                         |
|             Mass ratio factor | 0.00000E+00| (ttk(1))                         |
|------------------------------------------------------------------------------|
```

The possibilities here exceed those in EQ3NR (Section 3.3.3) where the temperature is simply some value. In EQ6, the temperature may change along the reaction path. Most EQ6 runs tend to be isothermal (Option 0). For Option 1 (“Linear tracking in Xi”) the temperature (C) is given by:

$$T = T_b + k_1 \xi \tag{3–2}$$

where T_b (tempcb) is the base temperature (the temperature at zero reaction progress), k_1 [ttk(1)] is the tracking constant (the slope $dT/d\xi$), and ξ (“Xi”) is the overall reaction progress variable. Analogously, for Option 2 (“Linear tracking in time”), the temperature (C) is given by:

$$T = T_b + k_1 t \tag{3–3}$$

where T_b (tempcb) is the base temperature (the temperature at zero time in this instance), k_1 [ttk(1)] is the tracking constant (the slope dT/dt), and t is the time (seconds).

Option 3 (“Fluid mixing tracking”) is intended for use in fluid-mixing calculations in which the two fluids (1 and 2) have different temperatures. Here the temperature (C) is given by:

$$T = \frac{(T_b k_1 + \xi k_2)}{(\xi + k_1)} \tag{3–4}$$

where T_b (tempcb) is both the base temperature (the temperature at zero reaction progress) and also the temperature of fluid 1, k_2 [ttk(2)] is the temperature of fluid 2, and k_1 [ttk(1)] is mass ratio factor (the ratio of the mass of fluid 1 to that of fluid 2 at $\xi = 1$, normally 1.0) This is an approximation that assumes equal heat capacities for the two fluids.

3.4.3 Pressure Option

Next comes a fairly analogous block for the pressure:

```
|------------------------------------------------------------------------------|
|Pressure option (jpress):                                                     |
|  [x] ( 0) Follow the data file reference pressure curve                      |
|  [ ] ( 1) Follow the 1.013-bar/steam-saturation curve                        |
|  [ ] ( 2) Constant pressure:                                                 |
|             Value (bars)      | 0.00000E+00| (pressb)                        |
|  [ ] ( 3) Linear tracking in Xi:                                             |
|             Base Value (bars) | 0.00000E+00| (pressb)                        |
|             Derivative        | 0.00000E+00| (ptk(1))                        |
|  [ ] ( 4) Linear tracking in time:                                           |
|             Base Value (bars) | 0.00000E+00| (pressb)                        |
|             Derivative        | 0.00000E+00| (ptk(1))                        |
|------------------------------------------------------------------------------|
```

Options 0 and 1 should be understood from the discussion of pressure in the section on the EQ3NR input file (Section 3.3.4). The other options (2–4) are analogous to Options 0–2 for the temperature discussed in the previous section (3.4.2), respectively.

3.4.4 Reactants Superblock

The reactants superblock follows. The full superblock corresponding to the present example is a relatively simple one:

```
|------------------------------------------------------------------------------|
|Reactants (Irreversible Reactions) | (nrct)                                   |
|------------------------------------------------------------------------------|
|Reactant        |Maximum_Microcline      | (ureac(n))                         |
|------------------------------------------------------------------------------|
|->|Type         |Pure mineral           | (urcjco(jcode(n)))                  |
|------------------------------------------------------------------------------|
|->|Status       |Reacting               | (urcjre(jreac(n)))                  |
|------------------------------------------------------------------------------|
|->|Amount remaining (moles) | 1.00000E+00| (morr(n))                          |
|------------------------------------------------------------------------------|
|->|Amount destroyed (moles) | 0.00000E+00| (modr(n))                          |
|------------------------------------------------------------------------------|
|->|Surface area option (nsk(n)):                                              |
|->|  [x] ( 0) Constant surface area:                                          |
|->|             Value (cm2)       | 0.00000E+00| (sfcar(n))                   |
|->|  [ ] ( 1) Constant specific surface area:                                 |
|->|             Value (cm2/g)     | 0.00000E+00| (ssfcar(n))                  |
|->|  [ ] ( 2) n**2/3 growth law- current surface area:                        |
|->|             Value (cm2)       | 0.00000E+00| (sfcar(n))                   |
|------------------------------------------------------------------------------|
|->|Surface area factor      | 0.00000E+00| (fkrc(n))                          |
|------------------------------------------------------------------------------|
```

```
|->|Forward rate law          |Relative rate equation  | (urcnrk(nrk(1,n)))    |
|------------------------------------------------------------------------------|
|--->|dXi(n)/dXi (mol/mol)      | 1.00000E+00| (rkb(1,1,n))                     |
|------------------------------------------------------------------------------|
|--->|d2Xi(n)/dXi2 (mol/mol2)   | 0.00000E+00| (rkb(2,1,n))                     |
|------------------------------------------------------------------------------|
|--->|d3Xi(n)/dXi3 (mol/mol3)   | 0.00000E+00| (rkb(3,1,n))                     |
|------------------------------------------------------------------------------|
|->|Backward rate law         |Partial equilibrium     | (urcnrk(nrk(2,n)))    |
|------------------------------------------------------------------------------|
* Valid reactant type strings (urcjco(jcode(n))) are:                          *
*    Pure mineral                Solid solution                                *
*    Special reactant            Aqueous species                               *
*    Gas species                 Generic ion exchanger                         *
*------------------------------------------------------------------------------*
* Valid reactant status strings (urcjre(jreac(n))) are:                        *
*    Saturated, reacting         Reacting                                      *
*    Exhausted                   Saturated, not reacting                       *
*------------------------------------------------------------------------------*
* Valid forward rate law strings (urcnrk(nrk(1,n))) are:                       *
*    Use backward rate law       Relative rate equation                        *
*    TST rate equation           Linear rate equation                          *
*------------------------------------------------------------------------------*
* Valid backward rate law strings (urcnrk(nrk(2,n))) are:                      *
*    Use forward rate law        Partial equilibrium                           *
*    Relative rate equation      TST rate equation                             *
*    Linear rate equation                                                      *
*------------------------------------------------------------------------------*
```

There is a short header to the superblock. Then come one or more reactant blocks. If there are zero reactants in the problem specified, the one such block must contain null inputs, such as "None" or all blanks for string inputs and zeros for numeric inputs. Certain problems do not require a reactant, for example equilibrating a system in disequilibria or just changing the temperature and/or pressure. The reactants superblock terminates with four standard comment line blocks containing the allowed strings for those string inputs not corresponding to species names.

A reactant block may vary in form and content in two ways. A reactant has a type defined by one of the strings given in the first of the four standard comment blocks. The input data required for a reactant varies first according this type. In the case of a "Special reactant", "Solid solution", or "Generic ion exchanger", data in addition to those shown above for a "Pure mineral" are required. These data are contained in additional sub-blocks that will be discussed below. Also, a forward (dissolution) and a backward (precipitation) rate law treatment must be defined by one of the strings in the third and fourth standard comment blocks shown above, respectively. Each possible rate law treatment requires somewhat different input data. Again, the possible cases differing from those shown in the above example will be discussed below.

Each reactant block begins with a line specifying the name of the reactant (here "Maximum Microcline"). The reactant type is then specified (here this is "Pure Mineral"). There are six possible types of reactants (as shown in the first standard comment block). The reactant name for every type except "Special reactant" or "Ion exchanger" must match a name of the cited type on the supporting data file. A reactant name for a "Generic ion exchanger" must match that of an exchanger defined in an ion exchanger creation block present on the EQ6 input file in the bottom half of the file. A special reactant is one that the user makes up on the input file.

After the reactant type comes the reactant status. There are four possible states, given by the possibilities in the second standard comment block. “Reacting” is the status when a problem is first defined (i.e., this is the only value a user should use when constructing an input file to start a problem). The other three states are only achievable after an initial problem has been run some finite number of steps. These may be seen on EQ6 “pickup” files, which allow a problem to be restarted.

This is followed by the amount remaining in moles. When a problem is first made up, this is the amount made available for reaction by the user. This is followed in turn by the amount destroyed in moles. That should always be zero when a problem is first made up. The amount remaining will decrease during a run, and the amount destroyed will increase. This will be noticeable upon the examination of any EQ6 “pickup” files.

Next the user must specify additional information for special reactants, solid solutions, and generic ion exchangers. This includes compositional data in each case. The present example does not include these reactant types. After the present example of a reactant block has been discussed, such blocks for these other reactant types will be discussed with relevant examples.

A surface area options sub-block follows. This is intended to be used with reactants that are solids of some type, but it is a required element of any reactant block. The surface area is only used in connection with true kinetic rate laws like the TST form (i.e., not with specified relative rates). This sub-block is followed by a surface area factor, which is an arbitrary multiplier to the surface area in a true kinetic rate law. This factor can be interpreted as the fraction of kinetically active surface area.

Next come sub-blocks for the forward (dissolution) and backward) (precipitation) rate laws. The form of these sub-blocks varies according to the forward and backward rate law choices. The example in the present instance is a common one. Here the forward (dissolution) rate law is based on an arbitrary relative rate law of the form:

$$v_j^{rel} = k_{+,1j} \tag{3–5}$$

where v_j^{rel} is the relative reaction rate of the j^{th} reactant ($d\xi_j/d\xi$, where ξ_j is the reaction progress variable for the reaction associated with this reactant), ξ is the overall reaction progress variable, and $k_{+,1j}$ [rkb(1,1,n)] is a rate constant (for the forward direction, as is implied by the subscript “+”) that here is equivalent to the relative rate value. A relative rate is not an actual rate. For a full discussion of the concept of relative rates, actual rates, and reaction progress variables, see Appendix D.1.3.1 and D.1.3.2, or Helgeson (1968).

In many EQ6 simulations, the overall reaction progress ξ is specified to increase from zero to a specified maximum value $\xi_{\max}$, which often has a value of 1.0. When this maximum value is achieved, the run terminates. Specifying a relative rate of x (also often with a value of 1.0) for a reactant means that the progress variable for the reaction associated with the reactant goes from zero to $x\xi_{\max}$. Reactions for “reactants” are defined such that the reaction coefficient for the associated “reactant” is –1.0 (the sign is negative because the reactant is consumed in the reaction). Thus, the number of moles of a reactant that are consumed in going from $\xi = 0$ to

$\xi = \xi_{max}$ is $x\xi_{max}$. Because x and ξ_{max} are both often set to 1.0, the number of moles of reactant consumed at the end of the run is also 1.0.

The backward (precipitation) rate law specified in this example is technically a partial equilibrium constraint, not an actual rate law (the precipitation rate is what it needs to be in order to prevent supersaturation). This corresponds t the classic treatment of reactant precipitation in geochemical reaction path modeling (e.g., Helgeson, 1968, 1970; Helgeson et al, 1969; Helgeson et al., 1970).

In this example, the relative rate is directly specified by the rkb(1,1,n) input. The rkb(2,1,n) and rkb(3,1,n) inputs imply that the relative rate in this instance can change with increasing overall reaction progress. In the version 7 level code, these inputs corresponded to the first and second derivatives of the relative rate (at zero reaction progress) and were used in an expanded Taylor's series description of the rate (see Appendix D.1.3.3). However, such expanded treatment is no longer permitted, as there is little justification for such a thing. Hence the rkb(2,1,n) and rkb(3,1,n) inputs for the case of a relative rate law are not used by the version 8 code and their appearance on the input file in this context is purely vestigial .

Another example of a reactant data block, this time for a special reactant, is presented below. Only the part down to the beginning of the surface area option sub-block is shown, as this is all that is needed to discuss the treatment of this reactant type. This is taken from the EQ6 test case library input file swtitr.6i:

```
|------------------------------------------------------------------------------|
|Reactant        |Aqueous H2SO4, 0.1 N    | (ureac(n))                         |
|------------------------------------------------------------------------------|
|->|Type         |Special reactant        | (urcjco(jcode(n)))                 |
|------------------------------------------------------------------------------|
|->|Status       |Reacting                | (urcjre(jreac(n)))                 |
|------------------------------------------------------------------------------|
|->|Amount remaining (moles) | 1.00000E+00| (morr(n))                          |
|------------------------------------------------------------------------------|
|->|Amount destroyed (moles) | 0.00000E+00| (modr(n))                          |
|------------------------------------------------------------------------------|
|->|Molar volume (cm3/mol)   | 0.00000E+00| (vreac(n))                         |
|------------------------------------------------------------------------------|
|->|Composition                                                                |
|------------------------------------------------------------------------------|
|--->|Element |Stoich. Number        | (this is a table header)                |
|------------------------------------------------------------------------------|
|--->|O       | 5.570895364010290E+01| (uesri(i,n), cesri(i,n))                |
|--->|H       | 1.111168701265550E+02| (uesri(i,n), cesri(i,n))                |
|--->|C       | 1.059405461284710E-05| (uesri(i,n), cesri(i,n))                |
|--->|S       | 5.000000274353950E-02| (uesri(i,n), cesri(i,n))                |
|------------------------------------------------------------------------------|
|->|Reaction                                                                   |
|------------------------------------------------------------------------------|
|--->|Species                 |Reaction Coefficient  | (this is a table header)|
|------------------------------------------------------------------------------|
|--->|Aqueous H2SO4, 0.1 N    |-1.000000000000000E+00| (ubsri(i,n), cbsri(i,n))|
|--->|HCO3-                   | 1.059405461284710E-05| (ubsri(i,n), cbsri(i,n))|
|--->|SO4--                   | 5.000000274353950E-02| (ubsri(i,n), cbsri(i,n))|
|--->|H+                      | 1.000105995416918E-01| (ubsri(i,n), cbsri(i,n))|
|--->|H2O                     | 5.550842446647936E+01| (ubsri(i,n), cbsri(i,n))|
|--->|O2(g)                   | 2.486902427776272E-04| (ubsri(i,n), cbsri(i,n))|
|------------------------------------------------------------------------------|
|->|Surface area option (nsk(n)):                                              |
```

Here the special reactant is an aqueous solution 0.1 N H_2SO_4. A special reactant could also be a solid of some kind, even a whole rock. In essence, it is little more than a stoichiometry. The additional information appears after the amount destroyed. The molar volume (cm^3/mol) must be entered. This is followed by a "Composition" sub-block, which defines the elemental composition of the special reactant (moles element per mole of reactant). That is followed in turn by a "Reaction" sub-block. This can be omitted on the input file, in which case EQ6 will generate a default reaction from the composition data. At the present time, no thermodynamic properties for the reaction are defined. Hence special reactants can have no thermodynamic stability, and cannot saturate the aqueous system being modeled in an EQ6 run. The remaining data in this example are the same as in the main example under discussion here.

Still another example of a reactant data block, this time for a solid solution, is presented below. Again, only the part down to the beginning of the surface area option sub-block is shown. This is taken from the EQ6 test case library input file j13wtuff.6i:

```
|------------------------------------------------------------------------------|
|Reactant        |Plagioclase             | (ureac(n))                         |
|------------------------------------------------------------------------------|
|->|Type         |Solid solution          | (urcjco(jcode(n)))                 |
|------------------------------------------------------------------------------|
|->|Status       |Reacting                | (urcjre(jreac(n)))                 |
|------------------------------------------------------------------------------|
|->|Amount remaining (moles) | 4.53300E-04| (morr(n))                          |
|------------------------------------------------------------------------------|
|->|Amount destroyed (moles) | 0.00000E+00| (modr(n))                          |
|------------------------------------------------------------------------------|
|->|Composition                                                                |
|------------------------------------------------------------------------------|
|--->|Component              |Mole frac.  | (this is a table header)           |
|------------------------------------------------------------------------------|
|--->|Albite_high            | 8.30000E-01| (ucxri(i,n), rxbari(i,n))          |
|--->|Anorthite              | 1.70000E-01| (ucxri(i,n), rxbari(i,n))          |
|------------------------------------------------------------------------------|
|->|Surface area option (nsk(n)):                                              |
```

The additional information is contained in a "Composition" sub-block. This sub-block consists of a header, followed by the input composition. This composition is specified in terms of mole fractions of the end-members. The solid solution itself is defined on the supporting data file. That definition includes the identity of the allowed end-members.

Yet another example of a reactant data block, this time for a generic ion exchanger, is presented below. Again, only the part down to the beginning of the surface area option sub-block is again shown. This is taken from the EQ6 test case library input file swxrca.6i:

```
|------------------------------------------------------------------------------|
|Reactant        |Exchanger 1             | (ureac(n))                         |
|------------------------------------------------------------------------------|
|->|Type         |Generic ion exchanger   | (urcjco(jcode(n)))                 |
|------------------------------------------------------------------------------|
|->|Status       |Reacting                | (urcjre(jreac(n)))                 |
|------------------------------------------------------------------------------|
|->|Amount remaining (moles) | 1.00000E+00| (morr(n))                          |
|------------------------------------------------------------------------------|
|->|Amount destroyed (moles) | 0.00000E+00| (modr(n))                          |
|------------------------------------------------------------------------------|
|->|Exchange model  |Vanselow                | (ugermo(n))                       |
|------------------------------------------------------------------------------|
|->|Composition                                                                |
|------------------------------------------------------------------------------|
```

```
|--->|Exchange site   |Site A  | (ugerji(j,n))                              |
|--------------------------------------------------------------------------|
|----->|Component              |Eq. frac.   | (this is a table header)      |
|--------------------------------------------------------------------------|
|----->|Na+                    | 1.00000E-02| (ugersi(i,j,n), egersi(i,j,n))  |
|----->|Ca++                   | 9.90000E-01| (ugersi(i,j,n), egersi(i,j,n))  |
|--------------------------------------------------------------------------|
|->|Surface area option (nsk(n)):                                          |
```

The additional information includes the exchange model (which is checked against that in the matching ion exchanger creation block that must appear on the bottom half of the input file) and a “Composition” sub-block. The “Composition” sub-block, apart from its head, includes one or more sub-sub-blocks for the various exchange sites. In the present example there is only one of these. Note that the composition is specified in terms of equivalent fractions. Also, following exchanger notation on EQ3/6 input files, “Na+” and “Ca++” refer to the ions on the exchanger, not the associated aqueous ions.

Note that the reactant types “Gas species” and “Aqueous species” do not require any additional input data. Care should be taken when using “Aqueous species” reactants to ensure that electroneutrality is preserved. For example, if Na^+ is defined as a reactant, Cl^- could also be defined as a reactant and added at the same rate.

The number of reactants is limited to **40**. This is a static dimension. Within this limit, there are no restrictions on the number of reactants of any of the allowed types.

The allowed treatments for forward (dissolution) and backward (precipitation) rates are specified in the last two of four standard comment line blocks at the end of the reactants superblock. These two blocks are repeated here for easy reference:

```
*--------------------------------------------------------------------------*
* Valid forward rate law strings (urcnrk(nrk(1,n))) are:                   *
*    Use backward rate law       Relative rate equation                    *
*    TST rate equation           Linear rate equation                      *
*--------------------------------------------------------------------------*
* Valid backward rate law strings (urcnrk(nrk(2,n))) are:                  *
*    Use forward rate law        Partial equilibrium                       *
*    Relative rate equation      TST rate equation                         *
*    Linear rate equation                                                  *
*--------------------------------------------------------------------------*
```

Each reactant block contains one sub-block for the forward direction rate law and another for the backward direction rate law.

For reactants of the type “Pure mineral”, “Solid solution, or Generic ion exchanger”, the thermodynamic driving force (affinity) determines which rate law treatment is used during a given point in the reaction progress calculation. The forward (dissolution) rate law applies to a mineral reactant when the aqueous solution is undersaturated with respect to that phase; the backward (precipitation) rate law applies when the solution is supersaturated.

Currently the software does not associate actual thermodynamic driving forces (affinities) with reactants of the type “Special reactant”, “Aqueous species”, or “Gas species”. These reactants are treated as mere stoichiometries with no thermodynamic stability. Thus only the rate law treatment for the forward direction is actually used, though a sub-block for the treatment in the backward direction is also required. As the EQ3/6 convention for reaction rates defines positive rates for the forward direction and negative rates for the backward direction, the specification of a negative rate in the forward rate law data block suffices to achieve removal of the associated stoichiometry from the modeled system.

It may be argued with some theoretical validity that a single rate law description should be applied to reaction rates in both directions. However, the rate laws commonly employed in geochemical modeling at the present time, despite theoretical underpinnings, tend to be little more than empirical explanations of rate behavior in only one direction. Put more plainly, what does a good job of describing dissolution rates often does poorly when extrapolated to precipitation, and vice versa. Therefore, EQ6 allows the use of different rate laws for the two directions.

A consistent treatment for both directions can be obtained by specifying a forward rate law, then choosing "Use forward rate law" for the backward rate law. Alternatively, one can choose "Use backward rate law" for the forward rate law and specify a backward rate law. Use of this approach requires choosing an actual rate law treatment for the one direction that is inherently consistent with thermodynamic driving forces. At present, only the "TST rate equation" (described below) satisfies this requirement. Obviously, choosing "Use forward rate law" and "Use backward rate law" together is not a valid combination. EQ6 will flag invalid combinations and issue appropriate error messages.

To facilitate the discussion of the rate law sub-blocks, those from the main example (EQ6 test case input file micro.6i) are repeated here:

```
|------------------------------------------------------------------------------|
|->|Forward rate law         |Relative rate equation  | (urcnrk(nrk(1,n)))     |
|------------------------------------------------------------------------------|
|--->|dXi(n)/dXi (mol/mol)      | 1.00000E+00| (rkb(1,1,n))                    |
|------------------------------------------------------------------------------|
|--->|d2Xi(n)/dXi2 (mol/mol2)   | 0.00000E+00| (rkb(2,1,n))                    |
|------------------------------------------------------------------------------|
|--->|d3Xi(n)/dXi3 (mol/mol3)   | 0.00000E+00| (rkb(3,1,n))                    |
|------------------------------------------------------------------------------|
|->|Backward rate law        |Partial equilibrium     | (urcnrk(nrk(2,n)))     |
|------------------------------------------------------------------------------|
```

As noted previously, the "Relative rate equation" now only allows a constant relative rate to be specified (as "**rkb**(1,1,n)". The inputs for "**rkb**(2,1,n)" and "**rkb**(3,1,n") will be ignored. The "Relative rate equation" option is also applicable to the backward rate law. Such usage (not employed in the above example) would require an analogous block exemplified by:

```
|------------------------------------------------------------------------------|
|->|Backward rate law        |Relative rate equation  | (urcnrk(nrk(2,n)))     |
|------------------------------------------------------------------------------|
|--->|dXi(n)/dXi (mol/mol)      |-1.00000E+00| (rkb(1,2,n))                    |
|------------------------------------------------------------------------------|
|--->|d2Xi(n)/dXi2 (mol/mol2)   | 0.00000E+00| (rkb(2,2,n))                    |
|------------------------------------------------------------------------------|
|--->|d3Xi(n)/dXi3 (mol/mol3)   | 0.00000E+00| (rkb(3,2,n))                    |
|------------------------------------------------------------------------------|
```

The specified relative rate is negative, in keeping with the EQ3/6 convention that forward rates are positive and backward rates are negative.

A sub-block of the above type cannot be used to specify the removal from the modeled system of a reactant of type “Special reactant”, “Aqueous species”, or “Gas species”. Rather, one must specify a negative relative rate in a “Relative rate equation” block for the forward rate. This is particularly important to remember when setting up an evaporation simulation using the “Aqueous species” reactant “H2O”.

The backward rate law specification in the main example (micro.6i) is “Partial equilibrium”, and the relevant block is just:

```
|-----------------------------------------------------------------------------|
|->|Backward rate law        |Partial equilibrium     | (urcnrk(nrk(2,n)))    |
|-----------------------------------------------------------------------------|
```

As noted in the earlier discussion, this treatment does not involve a true rate law, but rather a constraint that the rate be whatever is necessary to avoid supersaturation. A corresponding reaction rate value for use in rate law integration is not calculated as part of the process. Rather, the relevant algebraic constraint based on mass action (partial equilibrium) is applied to achieve the desired result. A corresponding rate value could be estimated using finite differences; however, the software presently does not do this.

The rate law sub-blocks for applying the rate law for one direction to another are comparably simple:

```
|-----------------------------------------------------------------------------|
|->|Forward rate law         |Use backward rate law   | (urcnrk(nrk(1,n)))    |
|-----------------------------------------------------------------------------|
```

and

```
|-----------------------------------------------------------------------------|
|->|Backward rate law        | Use forward rate law   | (urcnrk(nrk(2,n)))    |
|-----------------------------------------------------------------------------|
```

Only three actual rate law types are permitted in each case (forward and backward): “Relative rate equation”, TST rate equation, and “Linear rate equation”. The “Relative rate equation” has already been discussed. In contrast to that option, the “TST rate equation” and “Linear rate equation” options involve specification of actual kinetic rate laws.

The TST (transition state theory) rate law is probably the most important of the kinetic rate laws used in EQ6. For an introduction to the theory suitable for geochemists, see Aagaard and Helgeson (1982) or Lasaga (1981). The TST rate law is a function of the chemistry of the aqueous solution, but is not an explicit function of time.

The general form of the TST rate law for the “forward” (“+”) case is:

$$v_j = f_j s_j \sum_{i=1}^{i_{T,+,j}} k_{+,ij} \left(\prod_{n=1}^{n_{T,+,ij}} a_n^{-N_{+,nij}} \right) \left(1 - e^{-\frac{A_{+,j}}{\sigma_{+,ij} RT}} \right) \quad (3\text{–}6)$$

Here s_j is the total surface area of the phase dissolving in the j^{th} irreversible reaction. The factor f_j is a fudge factor representing the proportion of effective to total surface area; normally, it is taken as unity. The net forward form provides for treating $i_{T,+j}$ parallel mechanisms. For each, there is a rate constant ($k_{+,ij}$) a *kinetic activity product*, and a term that depends on the affinity ($A_{+,j}$). The kinetic activity product (which can be symbolized as $q_{+,ij}$; Delany, Puigdomenech, and Wolery 1986) depends on the thermodynamic activities of $n_{T,+,ij}$ species, each raised to a non-zero power characteristic of the mechanism ($-N_{+,nij}$). The kinetic activity product has a value of unity if $n_{T,+,ij} = 0$. The kinetic activity product most often reflects the dependence of the reaction mechanism on *pH*, and usually consists of just the activity of the hydrogen ion raised to some power (cf. Delany, Puigdomenech and Wolery 1986). The affinity factor goes to zero when the affinity goes to zero, forcing the reaction rate to do likewise. This factor also depends on the gas constant (R), the absolute temperature (T), and a stoichiometric factor ($\sigma_{+,ij}$) that relates the affinity of a macroscopic reaction ($A_{+,j}$) to that of the corresponding microscopic or elementary reaction ($A_{+,j}/\sigma_{+,ij}$). The stoichiometric factor is usually taken as having a value of unity.

The net reverse form parallels the net forward form:

$$-v_j = f_j s_j \sum_{i=1}^{i_{T,-,j}} k_{-,ij} \left(\prod_{n=1}^{n_{T,-,ij}} a_n^{-N_{-,nij}} \right) \left(1 - e^{-\frac{A_{-,j}}{\sigma_{-,ij} RT}} \right) \tag{3–7}$$

where $-v_j$ is the reaction rate in the backward ("–") direction (e.g., the precipitation rate of a mineral). The quantities shown in the above equation are analogous to those appearing in the forward direction form of the rate law.

Transition state theory has the strongest theoretical foundation among the various rate laws that have been applied to the kinetics of mineral dissolution and growth (see for example Lasaga, 1981; Aagaard and Helgeson 1982; Delany, Puigdomenech, and Wolery 1986). Nevertheless, the reader should be aware that in aqueous geochemistry it has been used primarily merely to provide a mathematical form, which can be used to explain some measurements. Such explanations may not be unique, even within the scope of the theory itself. The actual mechanisms, and even the number of mechanisms required to explain available data, may be open to debate. A TST expression that accurately describes dissolution rates may or may not extrapolate to correctly predict precipitation rates, and vice versa.

The following example of a forward rate law sub-block (for the reactant quartz) is taken from the EQ6 test case library input file j13wtuff.6i:

```
|------------------------------------------------------------------------------|
|->|Forward rate law         |TST rate equation       | (urcnrk(nrk(1,n)))    |
|------------------------------------------------------------------------------|
|--->|Mechanism  1                                                             |
|------------------------------------------------------------------------------|
|----->|sigma(i,+,n)           | 1.00000E+00| csigma(i,1,n)                     |
|------------------------------------------------------------------------------|
|----->|k(i,+,n) (mol/cm2/sec)  | 1.20000E-13| rkb(i,1,n)                       |
|------------------------------------------------------------------------------|
|----->|Ref. Temperature (C)    | 1.50000E+02| trkb(i,1,n)                      |
|------------------------------------------------------------------------------|
```

```
|----->|Temperature dependence option (iact(i,1,n)):                          |
|----->|  [x] ( 0) No temperature dependence                                  |
|----->|  [ ] ( 1) Constant activation energy:                                |
|----->|             Value (kcal/mol)  | 0.00000E+00| (eact(i,1,n))         |
|----->|  [ ] ( 2) Constant activation enthalpy:                              |
|----->|             Value (kcal/mol)  | 0.00000E+00| (hact(i,1,n))         |
|------------------------------------------------------------------------------|
|----->|Kinetic activity product species (ndact(i,1,n))                       |
|------------------------------------------------------------------------------|
|------->|Species                                     |-N(j,i,+,n)         |
|------->| (udac(j,i,1,n))                            | (cdac(j,i,1,n))    |
|------------------------------------------------------------------------------|
|------->|None                                        | 0.00000E+00        |
|------------------------------------------------------------------------------|
```

One or more sub-sub-blocks appear in such a sub-block, each corresponding to a distinct reaction mechanism (hence the “Mechanism” header). In the above example, there is only one such sub-sub-block. Within such a sub-sub-block there is one “Kinetic activity product”) sub-sub-sub-block. The main body of this can be expanded to allow the specification of multiple species whose activities appear in the kinetic activity product. The code will accommodate up to **four** mechanisms per TST rate law and up to **four** species in a kinetic activity product. These limits are static dimensions.

The corresponding backward rate law sub-block (for the same reactant, taken from the same EQ6 test case library input file) is completely analogous:

```
|------------------------------------------------------------------------------|
|->|Backward rate law         |TST rate equation       | (urcnrk(nrk(2,n)))    |
|------------------------------------------------------------------------------|
|--->|Mechanism  1                                                            |
|------------------------------------------------------------------------------|
|----->|sigma(i,-,n)           | 1.00000E+00| csigma(i,2,n)                   |
|------------------------------------------------------------------------------|
|----->|k(i,-,n) (mol/cm2/sec)  | 0.00000E+00| rkb(i,2,n)                     |
|------------------------------------------------------------------------------|
|----->|Ref. Temperature (C)    | 1.50000E+02| trkb(i,2,n)                    |
|------------------------------------------------------------------------------|
|----->|Temperature dependence option (iact(i,2,n)):                          |
|----->|  [x] ( 0) No temperature dependence                                  |
|----->|  [ ] ( 1) Constant activation energy:                                |
|----->|             Value (kcal/mol)  | 0.00000E+00| (eact(i,2,n))         |
|----->|  [ ] ( 2) Constant activation enthalpy:                              |
|----->|             Value (kcal/mol)  | 0.00000E+00| (hact(i,2,n))         |
|------------------------------------------------------------------------------|
|----->|Kinetic activity product species (ndact(i,2,n))                       |
|------------------------------------------------------------------------------|
|------->|Species                                     |-N(j,i,-,n)         |
|------->| (udac(j,i,2,n))                            | (cdac(j,i,2,n))    |
|------------------------------------------------------------------------------|
|------->|None                                        | 0.00000E+00        |
|------------------------------------------------------------------------------|
```

The “Linear rate equation” rate laws are relatively simple in form:

$$v_j = f_j s_j k_{+,1j} \tag{3–8}$$

and,

$$-v_j = f_j s_j k_{-,1j} \tag{3–9}$$

In essence, the rates are constant except for the effect of a surface area factor (which may or may not be constant). The rate law sub-blocks are very similar in form to the TST equivalents, but lack the kinetic activity product sub-sub-block. Here the TST sub-blocks given above have been converted to “Linear rate equation” sub-blocks. The example for the forward direction is:

```
|------------------------------------------------------------------------------|
|->|Forward rate law          |Linear rate equation    | (urcnrk(nrk(1,n)))    |
|------------------------------------------------------------------------------|
|--->|Mechanism  1                                                             |
|------------------------------------------------------------------------------|
|----->|sigma(i,+,n)            | 1.00000E+00| csigma(i,1,n)                   |
|------------------------------------------------------------------------------|
|----->|k(i,+,n) (mol/cm2/sec)  | 1.20000E-13| rkb(i,1,n)                      |
|------------------------------------------------------------------------------|
|----->|Ref. Temperature (C)    | 1.50000E+02| trkb(i,1,n)                     |
|------------------------------------------------------------------------------|
|----->|Temperature dependence option (iact(i,1,n)):                           |
|----->|  [x] ( 0) No temperature dependence                                   |
|----->|  [ ] ( 1) Constant activation energy:                                 |
|----->|             Value (kcal/mol)  | 0.00000E+00| (eact(i,1,n))            |
|----->|  [ ] ( 2) Constant activation enthalpy:                               |
|----->|             Value (kcal/mol)  | 0.00000E+00| (hact(i,1,n))            |
|------------------------------------------------------------------------------|
```

The example for the backward direction is:

```
|------------------------------------------------------------------------------|
|->|Backward rate law         |Linear rate equation    | (urcnrk(nrk(2,n)))    |
|------------------------------------------------------------------------------|
|--->|Mechanism  1                                                             |
|------------------------------------------------------------------------------|
|----->|sigma(i,-,n)            | 1.00000E+00| csigma(i,2,n)                   |
|------------------------------------------------------------------------------|
|----->|k(i,-,n) (mol/cm2/sec)  | 0.00000E+00| rkb(i,2,n)                      |
|------------------------------------------------------------------------------|
|----->|Ref. Temperature (C)    | 1.50000E+02| trkb(i,2,n)                     |
|------------------------------------------------------------------------------|
|----->|Temperature dependence option (iact(i,2,n)):                           |
|----->|  [x] ( 0) No temperature dependence                                   |
|----->|  [ ] ( 1) Constant activation energy:                                 |
|----->|             Value (kcal/mol)  | 0.00000E+00| (eact(i,2,n))            |
|----->|  [ ] ( 2) Constant activation enthalpy:                               |
|----->|             Value (kcal/mol)  | 0.00000E+00| (hact(i,2,n))            |
|------------------------------------------------------------------------------|
```

Only one mechanism sub-sub-block should appear in a “Linear rate equation” sub-block.

3.4.5 Run Parameters Superblock

The discussion will now return to the main example (the EQ6 test case library input file micro.6i). Following the reactants superblock is the key run parameters superblock. This consists of a header followed by a one-line block for each key run parameter:

```
*------------------------------------------------------------------------------*
|Starting, minimum, and maximum values of key run parameters.                  |
|------------------------------------------------------------------------------|
|Starting Xi value        | 0.00000E+00| (xistti)                              |
|------------------------------------------------------------------------------|
|Maximum Xi value         | 1.00000E+00| (ximaxi)                              |
|------------------------------------------------------------------------------|
```

```
|Starting time (seconds)  | 0.00000E+00| (tistti)                            |
|---------------------------------------------------------------------------|
|Maximum time (seconds)   | 1.00000E+38| (timmxi)                            |
|---------------------------------------------------------------------------|
|Minimum value of pH      |-1.00000E+38| (phmini)                            |
|---------------------------------------------------------------------------|
|Maximum value of pH      | 1.00000E+38| (phmaxi)                            |
|---------------------------------------------------------------------------|
|Minimum value of Eh (v)  |-1.00000E+38| (ehmini)                            |
|---------------------------------------------------------------------------|
|Maximum value of Eh (v)  | 1.00000E+38| (ehmaxi)                            |
|---------------------------------------------------------------------------|
|Minimum value of log fO2 |-1.00000E+38| (o2mini)                            |
|---------------------------------------------------------------------------|
|Maximum value of log fO2 | 1.00000E+38| (o2maxi)                            |
|---------------------------------------------------------------------------|
|Minimum value of aw      |-1.00000E+38| (awmini)                            |
|---------------------------------------------------------------------------|
|Maximum value of aw      | 1.00000E+38| (awmaxi)                            |
|---------------------------------------------------------------------------|
|Maximum number of steps  |         200| (kstpmx)                            |
|---------------------------------------------------------------------------|
```

Starting and maximum values are provided for both the overall reaction progress variable (“Xi”) and the model time (“time”). Minimum and maximum values are specified for the pH, the Eh (volts), the log fO_2, and the activity of water (“aw”). A maximum value is provided for the number of steps taken along the reaction path. Achieving a minimum or maximum value specified in this section causes run termination. Furthermore, the code will adjust the step size as needed to avoid exceeding any specified minimum or maximum value. Note that -1×10^{38} and $+1 \times 10^{38}$ are used in the example as the effective equivalents of $-\infty$ and $+\infty$, respectively. Any value of sufficiently large magnitude will do, however. The maximum number of steps can be specified as zero if one simply wants to equilibrate the starting system without continuing on to a reaction path calculation.

3.4.6 Print Interval Parameters Superblock

A superblock for print interval parameters follows this. These parameters control the frequency at which a complete description of the state of the system is written to the output files (the “output” or .6o files plus the “tab” or .6t) files. Print intervals may be specified in terms of reaction progress, log reaction progress, time (s), log time (s), pH, Eh (volts), log fO_2, or the activity of water (“aw”).

```
|---------------------------------------------------------------------------|
|Print interval parameters.                                                 |
|---------------------------------------------------------------------------|
|Xi print interval        | 1.00000E+00| (dlxprn)                            |
|---------------------------------------------------------------------------|
|Log Xi print interval    | 1.00000E+00| (dlxprl)                            |
|---------------------------------------------------------------------------|
|Time print interval      | 1.00000E+38| (dltprn)                            |
|---------------------------------------------------------------------------|
|Log time print interval  | 1.00000E+38| (dltprl)                            |
|---------------------------------------------------------------------------|
|pH print interval        | 1.00000E+38| (dlhprn)                            |
|---------------------------------------------------------------------------|
```

```
|Eh (v) print interval    | 1.00000E+38| (dleprn)                                  |
|------------------------------------------------------------------------------|
|Log fO2 print interval   | 1.00000E+38| (dloprn)                                  |
|------------------------------------------------------------------------------|
|aw print interval        | 1.00000E+38| (dlaprn)                                  |
|------------------------------------------------------------------------------|
|Steps print interval     |         100| (ksppmx)                                  |
|------------------------------------------------------------------------------|
```

3.4.7 Plot Interval Parameters Superblock

A superblock for corresponding plot interval parameters follows in turn. These parameters presently do nothing, as EQ6 currently lacks a formal plot file. The "tab" or .6t file presently functions as a rudimentary plot file. The frequency at which data are written to this file is determined by the above "print interval" parameters (Section 3.4.6), not these "plot interval" parameters.

```
|------------------------------------------------------------------------------|
|Plot interval parameters.                                                     |
|------------------------------------------------------------------------------|
|Xi plot interval         | 1.00000E+38| (dlxplo)                                  |
|------------------------------------------------------------------------------|
|Log Xi plot interval     | 1.00000E+38| (dlxpll)                                  |
|------------------------------------------------------------------------------|
|Time plot interval       | 1.00000E+38| (dltplo)                                  |
|------------------------------------------------------------------------------|
|Log time plot interval   | 1.00000E+38| (dltpll)                                  |
|------------------------------------------------------------------------------|
|pH plot interval         | 1.00000E+38| (dlhplo)                                  |
|------------------------------------------------------------------------------|
|Eh (v) plot interval     | 1.00000E+38| (dleplo)                                  |
|------------------------------------------------------------------------------|
|Log fO2 plot interval    | 1.00000E+38| (dloplo)                                  |
|------------------------------------------------------------------------------|
|aw plot interval         | 1.00000E+38| (dlaplo)                                  |
|------------------------------------------------------------------------------|
|Steps plot interval      |       10000| (ksplmx)                                  |
|------------------------------------------------------------------------------|
```

3.4.8 Iopt (Model) Options

This is followed by a superblocks for the iopt model option switches, the iopr print option switches, and the iodb debugging print option switches. As noted previously in the discussion of the EQ3NR input file (Section 3.3.15), these switches (along with the iopg activity coefficient option switches, Section 3.3.16) are shared with EQ3NR. Some pertain to both EQ3NR and EQ6, others to only one or the other. Recall that the "(0)" choices are the default values.

The superblock for the iopt model option switches on the EQ6 input file takes the form:

```
|------------------------------------------------------------------------------|
|Iopt Model Option Switches ("( 0)" marks default choices)                     |
|------------------------------------------------------------------------------|
|iopt(1) - Physical System Model Selection:                                    |
|  [x] ( 0) Closed system                                                      |
|  [ ] ( 1) Titration system                                                   |
|  [ ] ( 2) Fluid-centered flow-through open system                            |
|------------------------------------------------------------------------------|
```

```
|iopt(2) - Kinetic Mode Selection:                                            |
|  [x] ( 0) Reaction progress mode (arbitrary kinetics)                       |
|  [ ] ( 1) Reaction progress/time mode (true kinetics)                       |
|-----------------------------------------------------------------------------|
|iopt(3) - Phase Boundary Searches:                                           |
|  [x] ( 0) Search for phase boundaries and constrain the step size to match  |
|  [ ] ( 1) Search for phase boundaries and print their locations             |
|  [ ] ( 2) Don't search for phase boundaries                                 |
|-----------------------------------------------------------------------------|
|iopt(4) - Solid Solutions:                                                   |
|  [x] ( 0) Ignore                                                            |
|  [ ] ( 1) Permit                                                            |
|-----------------------------------------------------------------------------|
|iopt(5) - Clear the ES Solids Read from the INPUT File:                      |
|  [x] ( 0) Don't do it                                                       |
|  [ ] ( 1) Do it                                                             |
|-----------------------------------------------------------------------------|
|iopt(6) - Clear the ES Solids at the Initial Value of Reaction Progress:     |
|  [x] ( 0) Don't do it                                                       |
|  [ ] ( 1) Do it                                                             |
|-----------------------------------------------------------------------------|
|iopt(7) - Clear the ES Solids at the End of the Run:                         |
|  [x] ( 0) Don't do it                                                       |
|  [ ] ( 1) Do it                                                             |
|-----------------------------------------------------------------------------|
|iopt(9) - Clear the PRS Solids Read from the INPUT file:                     |
|  [x] ( 0) Don't do it                                                       |
|  [ ] ( 1) Do it                                                             |
|-----------------------------------------------------------------------------|
|iopt(10) - Clear the PRS Solids at the End of the Run:                       |
|  [x] ( 0) Don't do it                                                       |
|  [ ] ( 1) Do it, unless numerical problems cause early termination          |
|-----------------------------------------------------------------------------|
|iopt(11) - Auto Basis Switching in pre-N-R Optimization:                     |
|  [x] ( 0) Turn off                                                          |
|  [ ] ( 1) Turn on                                                           |
|-----------------------------------------------------------------------------|
|iopt(12) - Auto Basis Switching after Newton-Raphson Iteration:              |
|  [x] ( 0) Turn off                                                          |
|  [ ] ( 1) Turn on                                                           |
|-----------------------------------------------------------------------------|
|iopt(13) - Calculational Mode Selection:                                     |
|  [x] ( 0) Normal path tracing                                               |
|  [ ] ( 1) Economy mode (if permissible)                                     |
|  [ ] ( 2) Super economy mode (if permissible)                               |
|-----------------------------------------------------------------------------|
|iopt(14) - ODE Integrator Corrector Mode Selection:                          |
|  [x] ( 0) Allow Stiff and Simple Correctors                                 |
|  [ ] ( 1) Allow Only the Simple Corrector                                   |
|  [ ] ( 2) Allow Only the Stiff Corrector                                    |
|  [ ] ( 3) Allow No Correctors                                               |
|-----------------------------------------------------------------------------|
|iopt(15) - Force the Suppression of All Redox Reactions:                     |
|  [ ] ( 0) Don't do it                                                       |
|  [x] ( 1) Do it                                                             |
|-----------------------------------------------------------------------------|
|iopt(16) - BACKUP File Options:                                              |
|  [ ] (-1) Don't write a BACKUP file                                         |
|  [x] ( 0) Write BACKUP files                                                |
|  [ ] ( 1) Write a sequential BACKUP file                                    |
|-----------------------------------------------------------------------------|
|iopt(17) - PICKUP File Options:                                              |
|  [ ] (-1) Don't write a PICKUP file                                         |
|  [x] ( 0) Write a PICKUP file                                               |
|-----------------------------------------------------------------------------|
|iopt(18) - TAB File Options:                                                 |
|  [ ] (-1) Don't write a TAB file                                            |
|  [x] ( 0) Write a TAB file                                                  |
|  [ ] ( 1) Write a TAB file, prepending TABX file data from a previous run   |
|-----------------------------------------------------------------------------|
```

```
|iopt(20) - Advanced EQ6 PICKUP File Options:                              |
|  [x] ( 0) Write a normal EQ6 PICKUP file                                 |
|  [ ] ( 1) Write an EQ6 INPUT file with Fluid 1 set up for fluid mixing   |
|--------------------------------------------------------------------------|
```

The **iopt(1)** and **iopt(2)** switches are of special importance. The former controls the nature of the modeled system: closed system, titration system, or fluid-centered, flow-through open system. The closed system is self-evident. The titration system is similar. The difference is that in a closed system, equilibration of a reactant causes the transfer of any remaining unreacted mass to the equilibrium system (ES). The ES consists of the aqueous solution plus any phases in equilibrium with it, such that the any mass is included in the mass balances defined for that system. The titration system differs from the closed system in that such equilibration does not result in such as transfer of any remaining unreacted mass. Such unreacted mass will continue to be “titrated” into the ES. The fluid-centered, flow-through open system follows the evolution of a packet of water flowing through a reactive medium, leaving any product phases behind. These products can then no longer interact with the fluid packet. Some further details are provided in Appendix D.2.3

The **iopt(2)** switch is important because it determines whether or not the modeled system has a time element or not. In reaction-progress mode, there is no model time. Only relative rates can be specified for the reactants. System evolution is measured only as a function of the overall reaction progress variable (“Xi”). In reaction-progress/time mode, model time is employed and actual rate laws can be specified for reactants. Note that this mode is also required if temperature or pressure are specified to change as a function of time. The overall reaction progress variable (“Xi”) remains an element of this mode.

The meaning of the other iopt model options switches should be self-evident from the example presented. The switches **iopt(5)**, **iopt(6)**, and **iopt(7)** permit the user to delete any minerals present in the equilibrium system (ES), hence separating them from the aqueous solution. The switches **iopt(9)** and **iopt(10)** allow deletion of minerals from the physically removed system (PRS), which is where minerals go that are left behind by the fluid packet in the fluid-centered, flow-through open system model. The choices shown for the other switches other than **iopt(1)** and **iopt(2)** are reasonable ones for users to employ in most instances. However, the **iopt(15)** switch (“Force the Suppression of All Redox Reactions”) should normally be set to “Don’t do it”. The code is normally able to determine whether or not there is a redox aspect to a given problem. Note that **iopt(11)**, **iopt(12)**, **iopt(13)** and **iopt(14)** only affect code numerics, not model definition.

Iopt(14) controls the new corrector in the ordinary differential equation (ODE) solver. Detail of these ODE integration methods used are supplied in Appendix E.

3.4.9 Iopr (Print) Options

A corresponding superblock for the iopr print option switches follows:

```
|------------------------------------------------------------------------------|
|Iopr Print Option Switches ("( 0)" marks default choices)                     |
|------------------------------------------------------------------------------|
|iopr(1) - Print All Species Read from the Data File:                          |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print                                                              |
|------------------------------------------------------------------------------|
|iopr(2) - Print All Reactions:                                                |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print the reactions                                                |
|  [ ] ( 2) Print the reactions and log K values                               |
|  [ ] ( 3) Print the reactions, log K values, and associated data             |
|------------------------------------------------------------------------------|
|iopr(3) - Print the Aqueous Species Hard Core Diameters:                      |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print                                                              |
|------------------------------------------------------------------------------|
|iopr(4) - Print a Table of Aqueous Species Concentrations, Activities, etc.:  |
|  [ ] (-3) Omit species with molalities < 1.e-8                               |
|  [ ] (-2) Omit species with molalities < 1.e-12                              |
|  [ ] (-1) Omit species with molalities < 1.e-20                              |
|  [x] ( 0) Omit species with molalities < 1.e-100                             |
|  [ ] ( 1) Include all species                                                |
|------------------------------------------------------------------------------|
|iopr(5) - Print a Table of Aqueous Species/H+ Activity Ratios:                |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print cation/H+ activity ratios only                               |
|  [ ] ( 2) Print cation/H+ and anion/H+ activity ratios                       |
|  [ ] ( 3) Print ion/H+ activity ratios and neutral species activities        |
|------------------------------------------------------------------------------|
|iopr(6) - Print a Table of Aqueous Mass Balance Percentages:                  |
|  [ ] (-1) Don't print                                                        |
|  [x] ( 0) Print those species comprising at least 99% of each mass balance   |
|  [ ] ( 1) Print all contributing species                                     |
|------------------------------------------------------------------------------|
|iopr(7) - Print Tables of Saturation Indices and Affinities:                  |
|  [ ] (-1) Don't print                                                        |
|  [x] ( 0) Print, omitting those phases undersaturated by more than 10 kcal   |
|  [ ] ( 1) Print for all phases                                               |
|------------------------------------------------------------------------------|
|iopr(8) - Print a Table of Fugacities:                                        |
|  [x] (-1) Don't print                                                        |
|  [ ] ( 0) Print                                                              |
|------------------------------------------------------------------------------|
|iopr(9) - Print a Table of Mean Molal Activity Coefficients:                  |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print                                                              |
|------------------------------------------------------------------------------|
|iopr(10) - Print a Tabulation of the Pitzer Interaction Coefficients:         |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print a summary tabulation                                         |
|  [ ] ( 2) Print a more detailed tabulation                                   |
|------------------------------------------------------------------------------|
|iopr(17) - PICKUP file format ("W" or "D"):                                   |
|  [x] ( 0) Use the format of the INPUT file                                   |
|  [ ] ( 1) Use "W" format                                                     |
|  [ ] ( 2) Use "D" format                                                     |
|------------------------------------------------------------------------------|
```

These options should be self-evident, as with the EQ3NR print options (Section 3.3.17). In most instances, the default (“(0)’) values are recommended.

3.4.10 Iodb (Debug Print) Options

This is followed by a superblock for the iodb debugging print option switches:

```
|------------------------------------------------------------------------------|
|Iodb Debugging Print Option Switches ("( 0)" marks default choices)           |
|------------------------------------------------------------------------------|
|iodb(1) - Print General Diagnostic Messages:                                  |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print Level 1 diagnostic messages                                  |
|  [ ] ( 2) Print Level 1 and Level 2 diagnostic messages                      |
|------------------------------------------------------------------------------|
|iodb(2) - Kinetics Related Diagnostic Messages:                               |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print Level 1 kinetics diagnostic messages                         |
|  [ ] ( 2) Print Level 1 and Level 2 kinetics diagnostic messages             |
|------------------------------------------------------------------------------|
|iodb(3) - Print Pre-Newton-Raphson Optimization Information:                  |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print summary information                                          |
|  [ ] ( 2) Print detailed information (including the beta and del vectors)    |
|  [ ] ( 3) Print more detailed information (including matrix equations)       |
|  [ ] ( 4) Print most detailed information (including activity coefficients)  |
|------------------------------------------------------------------------------|
|iodb(4) - Print Newton-Raphson Iteration Information:                         |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print summary information                                          |
|  [ ] ( 2) Print detailed information (including the beta and del vectors)    |
|  [ ] ( 3) Print more detailed information (including the Jacobian)           |
|  [ ] ( 4) Print most detailed information (including activity coefficients)  |
|------------------------------------------------------------------------------|
|iodb(5) - Print Step-Size and Order Selection:                                |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print summary information                                          |
|  [ ] ( 2) Print detailed information                                         |
|------------------------------------------------------------------------------|
|iodb(6) - Print Details of Hypothetical Affinity Calculations:                |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print summary information                                          |
|  [ ] ( 2) Print detailed information                                         |
|------------------------------------------------------------------------------|
|iodb(7) - Print General Search (e.g., for a phase boundary) Information:      |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print summary information                                          |
|------------------------------------------------------------------------------|
|iodb(8) - Print ODE Corrector Iteration Information:                          |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print summary information                                          |
|  [ ] ( 2) Print detailed information (including the betar and delvcr vectors)|
|------------------------------------------------------------------------------|
```

In general, as with the EQ3NR debug options (Section 3.3.18), these option switches should all be set to “Don’t print”. They are ordinarily employed only in debugging failed runs or for observing the numerical processes. If the user encounters a failed run and the cause is not evident, setting the **iodb(1)** switch in particular to one of the “Print” settings and re-running the problem may generate a useful diagnostic.

3.4.11 Mineral Suppression Options

Next is a superblock for mineral sub-set selection suppression options:

```
|------------------------------------------------------------------------------|
|Mineral Sub-Set Selection Suppression Options | (nxopt)                       |
|------------------------------------------------------------------------------|
|Option  |Sub-Set Defining Species| (this is a table header)                   |
|------------------------------------------------------------------------------|
|None    |None                    | (uxopt(n), uxcat(n))                       |
|------------------------------------------------------------------------------|
* Valid mineral sub-set selection suppression option strings (uxopt(n)) are:   *
*    None        All         Alwith      Allwith                               *
*------------------------------------------------------------------------------*
|Exceptions to the Mineral Sub-Set Selection Suppression Options | (nxopex)    |
|------------------------------------------------------------------------------|
|Mineral                 | (this is a table header)                            |
|------------------------------------------------------------------------------|
|None                    | (uxopex(n))                                         |
|------------------------------------------------------------------------------|
```

These options are similar in function to the suppress part of the alter/suppress functions discussed in the section on the EQ3NR input file (Section 3.3.14) and which are included on the bottom half of the EQ6 input file yet to be discussed. However, the present options permit whole categories of minerals to be suppressed. Exceptions may also be specified. The following example suppresses all minerals (with no exceptions):

```
|------------------------------------------------------------------------------|
|Mineral Sub-Set Selection Suppression Options | (nxopt)                       |
|------------------------------------------------------------------------------|
|Option  |Sub-Set Defining Species| (this is a table header)                   |
|------------------------------------------------------------------------------|
|All     |                        | (uxopt(n), uxcat(n))                       |
|------------------------------------------------------------------------------|
* Valid mineral sub-set selection suppression option strings (uxopt(n)) are:   *
*    None        All         Alwith      Allwith                               *
*------------------------------------------------------------------------------*
|Exceptions to the Mineral Sub-Set Selection Suppression Options | (nxopex)    |
|------------------------------------------------------------------------------|
|Mineral                 | (this is a table header)                            |
|------------------------------------------------------------------------------|
|None                    | (uxopex(n))                                         |
|------------------------------------------------------------------------------|
```

The following example suppresses all Ca++ minerals except calcite and margarite:

```
|------------------------------------------------------------------------------|
|Mineral Sub-Set Selection Suppression Options | (nxopt)                       |
|------------------------------------------------------------------------------|
|Option  |Sub-Set Defining Species| (this is a table header)                   |
|------------------------------------------------------------------------------|
|Allwith |Ca++                    | (uxopt(n), uxcat(n))                       |
|------------------------------------------------------------------------------|
* Valid mineral sub-set selection suppression option strings (uxopt(n)) are:   *
*    None        All         Alwith      Allwith                               *
*------------------------------------------------------------------------------*
|Exceptions to the Mineral Sub-Set Selection Suppression Options | (nxopex)    |
|------------------------------------------------------------------------------|
|Mineral                 | (this is a table header)                            |
|------------------------------------------------------------------------------|
|Calcite                 | (uxopex(n))                                         |
|Margarite               | (uxopex(n))                                         |
|------------------------------------------------------------------------------|
```

Note that the “Allwith” (or “Alwith”) option requires specification of a data file basis species, not a chemical element as was the case in the version 7 code.

The number of mineral sub-set selection suppression options is limited to **40**. The number of exceptions is limited to **100**. These are static dimensions.

3.4.12 Fixed Fugacity Superblock

A superblock for fixed fugacity options then follows:

```
|------------------------------------------------------------------------------|
|Fixed Fugacity Options | (nffg)                                               |
|------------------------------------------------------------------------------|
|Gas                     |Moles to Add |Log Fugacity | --                      |
| (uffg(n))              | (moffg(n))  | (xlkffg(n)) | --                      |
|------------------------------------------------------------------------------|
|None                    |  0.00000E+00|  0.00000E+00| --                      |
|------------------------------------------------------------------------------|
```

These options are employed to simulate equilibrium with gases in a large external gas reservoir (such as the atmosphere). In this example, no such option is employed. One could fix the fugacities of O_2(g) and CO_2(g) at the atmospheric values by entering:

```
|------------------------------------------------------------------------------|
|Fixed Fugacity Options | (nffg)                                               |
|------------------------------------------------------------------------------|
|Gas                     |Moles to Add |Log Fugacity | --                      |
| (uffg(n))              | (moffg(n))  | (xlkffg(n)) | --                      |
|------------------------------------------------------------------------------|
|O2(g)                   |  0.50000E+00| -0.70000E+00| --                      |
|CO2(g)                  |  0.50000E+00| -3.50000E+00| --                      |
|------------------------------------------------------------------------------|
```

Note that in addition to the name of a gas species and a log fugacity value, one must also enter the number of moles in a buffer mass. A gas fugacity is fixed by imposing equilibrium with a fictive mineral. A minimum initial buffer mass is specified as the “Moles to Add”. Here 0.5 moles have been chosen for both gases in the example. This is usually a reasonable value. Values higher than 5 moles should be avoided, as they may degrade the code numerics. A value of zero may suffice if maintenance of the corresponding fugacity value acts only to increase the buffer mass. If the buffer mass is lost, the fugacity will not be fixed at the desired value. A warning message will be written if a fixed fugacity value cannot be maintained.

The number of fixed-fugacity options is limited to **20**. This corresponds to a static dimension.

3.4.13 Numerical Parameters

This is followed in turn by the numerical parameters superblock:

```
|------------------------------------------------------------------------------|
|Numerical Parameters                                                          |
|------------------------------------------------------------------------------|
|Max. finite-difference order              |   6        | (nordmx)             |
|Beta convergence tolerance                | 0.00000E+00| (tolbt)              |
|Del convergence tolerance                 | 0.00000E+00| (toldl)              |
|Max. No. of N-R iterations                |   0        | (itermx)             |
|Search/find convergence tolerance         | 0.00000E+00| (tolxsf)             |
|Saturation tolerance                      | 0.00000E+00| (tolsat)             |
|Max. No. of Phase Assemblage Tries        |   0        | (ntrymx)             |
|Zero order step size (in Xi)              | 0.00000E+00| (dlxmx0)             |
|Max. interval in Xi between PRS transfers | 0.00000E+00| (dlxdmp)             |
|------------------------------------------------------------------------------|
```

This is similar to that for the EQ3NR input file (Section 3.3.19). Zero values are replaced by non-zero defaults in all cases. In general, users should take the defaults. The maximum finite-difference order can have any value from 1 to 8, with 6 being the default and recommended value. The beta and del convergence tolerances have shared default values of 1 x 10^{-6} for **iopt(1)** = 0 (reaction-progress mode) and 1 x 10^{-8} for **iopt(1)** = 1 (reaction-progress/time mode). In the former mode, values are restricted to between 1 x 10^{-10} and 1 x 10^{-4}, in the latter, between 1 x 10^{-10} and 1 x 10^{-6}. The default value for the maximum number of Newton Raphson iterations is 200. The default value for the search/find convergence tolerance is 1 x 10^{-6}. The default value for the saturation tolerance (on the affinity function, in kcal/mol) is 0.0005, and values are constrained to between 0.00005 and 0.005. The maximum number of phase assemblage tries has a default value of 100.

The zero-order step size (in units of reaction progress) has a complex set of defaults. In reaction-progress/time mode (**iopt(2)** = 1) or if using the fluid-centered, flow-through open system mode (**iopt(1)** = 2), the default value is 1 x 10^{-9}. In reaction-progress mode (**iopt(1)** = 0) the default value is usually 1 x 10^{-8}, but greater values may pertain in certain instances. For example, if there are no reactants, the default value is 1 x 10^{-2}. If economy mode (**iopt(13)** = 1) is selected and there is at least one reactant, the default is 1 x 10^{-6}. If super economy mode (**iopt(13)** = 2) is selected and there is at least one reactant, the default is the linear print interval, if a positive value for that is specified. If not, and the specified maximum reaction progress exceeds the starting value of reaction progress, the default is one-tenth the difference between these two reaction progress variables. If not that, the default is 1 x 10^{-8}. In any case, the minimum value for the zero-order step size is 1 x 10^{-12}.

The maximum interval in reaction progress between PRS transfers ordinarily has a default value of 1 x 10^{38} (effectively, infinity). However, if solid solutions or generic ion exchangers are present in the model system, the default value is 1 x 10^{-6}, except that if the maximum reaction progress parameter is greater than zero, the default value is 1/25 of that. If the fluid-centered flow-through system mode (**iopt(1)** = 2) and either solid solutions or generic ion exchangers are present, the maximum interval in reaction progress may not be less than the linear print interval in overall reaction progress.

Note that all values actually used in a run are written on the output file before any equilibrium or reaction-path calculations are made. This allows the actual values to be verified.

Should a run fail due to numerical difficulties, there is a rather limited possibility of coaxing it to work by using larger values for one or more of these parameters. For example, one might try increasing the maximum number of Newton-Raphson iterations from say 200 to 500. One might also try increasing certain other parameters, including the two convergence tolerances (which ordinarily should share a common value), the saturation tolerance, and the zero-order step size by say an order of magnitude at a time. Ordinarily, only one such parameter (considering the two convergence tolerances as one) should be changed per attempt. In general, tinkering with the numerical parameters in EQ6 is not likely to be a fruitful procedure for the average user.

The numerical parameters superblock is the last part of the top half of the EQ6 input files. As such, it is ordinarily the last part of the EQ6 input file that a user would directly define or modify.

3.4.14 Bottom Half of EQ6 Input File

The bottom half of the EQ6 input file is normally either an EQ3NR “pickup” file, the bottom half of such a file produced using one of the “advanced pickup file options”, or the bottom half of an EQ6 “pickup” file. A complete “pickup” file example is provided in Appendix C.

This section of the EQ6 input file begins with comment lines noting the start of the bottom half followed by the secondary title. In the present example, that part is given by:

```
|------------------------------------------------------------------------------|
* Start of the bottom half of the INPUT file                                   *
*------------------------------------------------------------------------------*
| Secondary Title        | (utitl2(n))                                         |
|------------------------------------------------------------------------------|
|EQ3NR input file name= ph4hcl.3i                                              |
|Description= "A pH 4 HCl solution, with traces of K, Al, and Si"              |
|Version level= 8.0                                                            |
|Revised 02/14/97    Revisor= T.J. Wolery                                      |
|This is part of the EQ3/6 Test Case Library                                   |
|                                                                              |
|  Dilute HCl solution, pH 4.00, with traces of potassium, aluminum, and       |
|silica. This problem is part of test problem 3 of INTERA (1983), who report  |
|a comparison study of EQ3/6 with PHREEQE (Parkhurst, Thorstenson, and        |
|Plummer, 1980). Note that precipitation of quartz, chalcedony, and tridymite |
|is prevented by means of nxmod suppress options. This has no effect on the   |
|EQ3NR calculation. It is simply needed for the subsequent EQ6 problem. EQ3NR |
|passes this on to EQ6 on the PICKUP file.                                     |
|                                                                              |
|  Purpose: to provide a PICKUP file for construction of the EQ6 test cases   |
|micro.6i and microft.6i.                                                      |
|                                                                              |
|  This problem is actually redox-indifferent. The auxiliary basis species    |
|O2(aq) and H2(aq) have therefore been suppressed. The EQ6 option switch      |
|iopt(15) should be set to 1.                                                  |
|                                                                              |
|                                                                              |
|                            References                                        |
|                                                                              |
|INTERA Environmental Consultants, Inc., 1983, Geochemical Models Suitable    |
|  for Performance Assessment of Nuclear Waste Storage: Comparison of         |
|  PHREEQE and EQ3/EQ6: Office of Nuclear Waste Isolation, Battelle Project   |
|  Management Division, Columbus, Ohio, ONWI-473, 114 p.                       |
|                                                                              |
```

```
|Parkhurst, D.L., Thorstenson, D.C., and Plummer, L.N., 1980, PHREEQE-         |
|  A Computer Program for Geochemical Calculations: Water Resources            |
|  Investigations Report 80-96, U.S. Geological Survey, Reston, Virginia,      |
|  210 p.                                                                      |
|                                                                              |
|------------------------------------------------------------------------------|
```

The secondary title is the title inherited from the previous EQ3NR or EQ6 run. The purpose of the secondary title is to aid traceability.

A detailed description of the remaining parts of the bottom half of the EQ6 input file is eschewed here, as this half is intended to be created by a previous code run and user modification is discouraged, except for one block noted below. The overall purpose of the bottom half is to carry over sufficient information from the previous code run to allow a subsequent, derivative run. Changing the data here is dangerous because the results may lose validity. In general, only the most advanced users should consider making changes here. In the Version 7 code (Wolery and Daveler, 1992), user-made changes here were required to perform fluid mixing calculations. In the present version, such changes are rendered unnecessary by the new EQ3NR advanced pickup file option **iopt(19)** = 3.

The promised exception to user modification lies in the following block:

```
|------------------------------------------------------------------------------|
|Alter/Suppress Options  | (nxmod)                                             |
|------------------------------------------------------------------------------|
|Species                                         |Option          |Alter value |
| (uxmod(n))                                     |(ukxm(kxmod(n)))| (xlkmod(n))|
|------------------------------------------------------------------------------|
|Quartz                                          |Suppress        | 0.00000E+00|
|Chalcedony                                      |Suppress        | 0.00000E+00|
|Tridymite                                       |Suppress        | 0.00000E+00|
|------------------------------------------------------------------------------|
* Valid alter/suppress strings (ukxm(kxmod(n))) are:                           *
*    Suppress            Replace             AugmentLogK                       *
*    AugmentG                                                                  *
*------------------------------------------------------------------------------*
```

This is the same alter/suppress options block that should be familiar from the discussion of the EQ3NR input file. Here it is reasonable for the code user to suppress additional minerals, provided that they have not already formed in an EQ6 run leading to the present one. As minerals do not actually form in an EQ3NR calculation, one can always suppress additional minerals when the bottom half of the EQ6 input file is obtained from an EQ3NR "pickup" file. Users should not here suppress aqueous species, or use any of the alter options. Any such options should be made in the EQ3NR runs leading to the present EQ6 run.

The principle here is a general one. If the user desires to make an intrinsic change to the definition of a chemical model made in a previous run, such as changing the activity coefficient model (iopg(1)) or altering the thermodynamic data, one should always go back and make the change as early as possible in the chain of runs. This usually means going back and redoing the EQ3NR run that initiates this chain.

The reader will note that in addition to the alter/suppress options block, other elements appearing on the EQ3NR input file also appear on the bottom half of the EQ6 input file, including any generic ion exchanger creation blocks (Sections 3.3.10–3.3.12), both the special and ordinary basis switching blocks (Sections 3.3.2 and 3.3.20, respectively), and the iopg activity coefficient option switch superblock (Section 3.3.16). Some elements of the bottom half of the EQ6 input files do not appear on the EQ3NR input file. These include blocks dealing with mass balances and other system descriptors that were calculated in a prior EQ3NR or EQ6 run.

The EQ6 input file ends with the following block:

```
|------------------------------------------------------------------------------|
|End of problem                                                                |
|------------------------------------------------------------------------------|
```

In reality, this just marks the end of a problem. Problems may be stacked on an EQ6 input file, just as they may be on an EQ3NR input file. However, stacking more than one problem on an EQ6 input file is not recommended.

3.5 EQ3NR OUTPUT FILES

EQ3NR (cf. Appendix B.1) is a speciation-solubility code. As such, it calculates from various input constraints (usually including total concentrations of dissolved components, pH, Eh, etc.) a chemical model of an aqueous solution. Such a model includes a detailed speciation model, saturation indices (log *Q/K*) for various minerals, and equilibrium gas fugacities. The detailed speciation model includes the concentrations of individual chemical species (as distinct from analytical totals) and the corresponding thermodynamic activities and thermodynamic activity coefficients. Because of EQ3NR's high flexibility, many quantities of interest can be either inputs or outputs. For example, the pH is normally an input, but this can also be an output if certain other constraints are placed on the hydrogen ion (e.g., electrical balancing, equilibrium with a gas or mineral). A complete EQ3NR output file is presented in Appendix B.4.

The EQ3NR code produces two kinds of output files. The first is the normal "output" file, which normally carries a ".3o" filename extension. The second is the "pickup" file, which provides information to be carried forward into an EQ6 calculation. A pickup file, which normally carries a ".3p" filename extension, is either a complete EQ6 input file (if one of the "advanced pickup file" options is chosen) or the bottom half of such a file. The EQ3NR pickup file will not receive further discussion here, as it has effectively been covered in the section on input files, Section 3.4.14. Note that the bottom half of the primary example, the EQ6 test case library input file micro.6i, is the pickup file obtained by running the EQ3NR test case library input file ph4hcl.3i. Also, a complete pickup file is presented in Appendix C.

The EQ3NR output (".3o") file consists of six parts. The example to be discussed here is that obtained by running the EQ3NR test case library input file swmaj.3i, which was the primary example presented in the section on the EQ3NR input file.

3.5.1 Header Section and Input File Echo

The first part is a header section, followed by a line-by-line echo of the input file (which constitutes the second part). The following piece of the swmaj.3o file shows this header followed by about the first dozen lines of the input file echo:

```
 EQ3/6, Version 8.0 (EQ3/6-V8-REL-V8.0-PC)
 EQ3NR Speciation-Solubility Code (EQ3/6-V8-EQ3NR-EXE-R43-PC)
 Supported by the following EQ3/6 libraries:
   EQLIB (EQ3/6-V8-EQLIB-LIB-R43-PC)
   EQLIBG (EQ3/6-V8-EQLIBG-LIB-R43-PC)
   EQLIBU (EQ3/6-V8-EQLIBU-LIB-R43-PC)

 Copyright (c) 1987, 1990-1993, 1995, 1997, 2002 The Regents of the
 University of California, Lawrence Livermore National Laboratory.
 All rights reserved.

 This work is subject to additional statements and
 disclaimers which may be found in the README.txt file
 included in the EQ3/6 software transmittal package.


 Run  15:47:54  06Sep2002


 Reading the data1 file header section ...

 Reading the rest of the DATA1 file ...

   The data file title is:

     data0.com.V8.R6
     CII: GEMBOCHS.V2-EQ8-data0.com.V8.R6
     THERMODYNAMIC DATABASE
     generated by GEMBOCHS.V2-Jewel.src.R5 03-dec-1996 14:19:25
     Output package:  eq3
     Data set:        com

   Continuing to read the DATA1 file ...

 * Note - (EQLIB/inbdot) The following aqueous species have been assigned
       a default hard core diameter of  4.000 x 10**-8 cm:

        Cd(N3)2(aq)                CuSO4(aq)

   Done reading the DATA1 file.


 Reading problem   1 from the input file ...

|------------------------------------------------------------------------------|
| Title                  | (utitl(n))                                          |
|------------------------------------------------------------------------------|
|EQ3NR input file name= swmaj.3i                                               |
|Description= "Sea water, major ions only"                                     |
|Version level= 8.0                                                            |
|Revised 02/14/97    Revisor= T.J. Wolery                                      |
|This is part of the EQ3/6 Test Case Library                                   |
|                                                                              |
|  Sea water, including only the major ions. This is a considerably            |
|pared-down version of swtst.3i, which contains the full benchmark sea water   |
|test case of Nordstrom et al. (1979, Table III).                              |
|                                                                              |
```

The header information includes version identification for both the code and the supporting thermodynamic data file, a copyright statement, a time and date stamp, and any code messages generated prior to reading the input file.

3.5.2 Recap of Input Data

The third section consists of a recap of all the input data, including any defaults or overrides. For the example chosen, this is (the last few lines of the input file echo are included here):

```
|Aq. Phase Scale Factor  | 1.00000E+00| (scamas)                               |
|------------------------------------------------------------------------------|
|End of problem                                                                |
|------------------------------------------------------------------------------|

   Done reading problem   1.


 The activity coefficients of aqueous species will be
 calculated using the B-dot equation.


 Temperature=  25.00 C


 jpres3=    0 (Pressure option switch)

   Pressure=   1.0132     bars (data file reference curve value)


       --- Numbers of Phases, Species, and Groups Thereof---

       Entity               Date Base    Dimension   Current Problem

   Chemical Elements             81          81           9
   Basis Species                201         211          51
   Phases                      1135        1159          87
   Species                     3031        3523         364
   Aqueous Species             1769        1769         262
   Pure Minerals               1120        1120          84
   Pure Liquids                   1           3           1
   Gas Species                   93          93          17
   Solid Soutions                12          12           0


 iopt(1)=   0 (Used only by EQ6)
 iopt(2)=   0 (Used only by EQ6)
 iopt(3)=   0 (Used only by EQ6)
 iopt(4)=   0 (Solid solutions)
 iopt(5)=   0 (Used only by EQ6)
 iopt(6)=   0 (Used only by EQ6)
 iopt(7)=   0 (Not used)
 iopt(8)=   0 (Not used)
 iopt(9)=   0 (Not used)
 iopt(10)=  0 (Not used)
 iopt(11)=  0 (Auto basis switching, in pre-Newton-Raphson optimization)
 iopt(12)=  0 (Used only by EQ6)
 iopt(13)=  0 (Not used)
 iopt(14)=  0 (Not used)
 iopt(15)=  0 (Used only by EQ6)
 iopt(16)=  0 (Not used)
 iopt(17)=  0 (pickup file options)
 iopt(18)=  0 (Used only by EQ6)
 iopt(19)=

   iopg(1)=   0 (Aqueous species activity coefficient model)
   iopg(2)=   0 (pH scale)


 iopr(1)=   0 (List all species)
 iopr(2)=   0 (List all reactions)
 iopr(3)=   0 (List HC diameters)
 iopr(4)=   0 (Aqueous species concentration print cut-off)
 iopr(5)=   0 (Ion/H+ activity ratios)
 iopr(6)=   0 (Mass balance percentages)
 iopr(7)=   0 (Affinity print cut-off)
 iopr(8)=   0 (Fugacities)
 iopr(9)=   0 (Mean molal activity coefficients)
 iopr(10)=  0 (Pitzer coefficients tabulation)
 iopr(11)=  0 (Not used)
 iopr(12)=  0 (Not used)
```

```
iopr(13)=  0 (Not used)
iopr(14)=  0 (Not used)
iopr(15)=  0 (Not used)
iopr(16)=  0 (Not used)
iopr(17)=  0 (pickup file format)


  iodb(1)=   0 (General diagnostics)
  iodb(2)=   0 (Used only by EQ6)
  iodb(3)=   0 (pre-Newton-Raphson optimization iterations)
  iodb(4)=   0 (Newton-Raphson iterations)
  iodb(5)=   0 (Used only by EQ6)
  iodb(6)=   0 (Hypothetical affinity iterations)
  iodb(7)=   0 (Used only by EQ6)


irdxc3=   -1 (Default redox constraint switch)

  The default redox state is constrained by Eh =  0.50000 volts.


iebal3=    0 (Electrical balancing option switch)

  No electrical balancing adjustment will be made.
  The imbalance will be calculated.


Solution density =  1.02336 g/ml


itdsf3=    0 (Total dissolved solutes option switch)

  Total dissolved salts =       0.00 mg/kg.sol


tolbt  =  1.00000E-06 (convergence tolerance on residual functions)
toldl  =  1.00000E-06 (convergence tolerance on correction terms)
tolspf =  5.00000E-05 (saturation print flag tolerance, does not affect
                        convergence)


itermx = 200 (maximum number of iterations)


scamas =  1.00000E+00 (scale factor for aqueous solution
                        mass written on the pickup file)


                    --- Original Input Constraints ---

    Species                    coval   jflag   Type of Input

 Na+                        1.07680E+04   3  Total mg/kg.sol
 K+                         3.99100E+02   3  Total mg/kg.sol
 Ca++                       4.12300E+02   3  Total mg/kg.sol
 Mg++                       1.29180E+03   3  Total mg/kg.sol
 H+                         8.22000E+00  20  pH
 HCO3-                      2.02200E-03   0  Total molality
 Cl-                        1.93530E+04   3  Total mg/kg.sol
 SO4--                      2.71200E+03   3  Total mg/kg.sol
 H2(aq)                                  27  Homogenous equilibrium
 O2(aq)                                  27  Homogenous equilibrium


                    --- Modified Input Constraints ---

    Species                    coval   jflag   Type of Input

 H2O                        0.00000E+00   0  Total molality
 Ca++                       1.02874E-02   0  Total molality
 Cl-                        5.45882E-01   0  Total molality
```

```
  H+                          8.22000E+00  20  pH
  HCO3-                       2.02200E-03   0  Total molality
  K+                          1.02076E-02   0  Total molality
  Mg++                        5.31496E-02   0  Total molality
  Na+                         4.68382E-01   0  Total molality
  SO4--                       2.82313E-02   0  Total molality
  HS-                                      30  Make non-basis
  Acetic_acid(aq)                          30  Make non-basis
  S2--                                     30  Make non-basis
  S2O3--                                   30  Make non-basis
  Acetone(aq)                              30  Make non-basis
  Benzene(aq)                              30  Make non-basis
  Butanoic_acid(aq)                        30  Make non-basis
  CO(aq)                                   30  Make non-basis
  ClO-                                     30  Make non-basis
  ClO2-                                    30  Make non-basis
  ClO3-                                    30  Make non-basis
  ClO4-                                    30  Make non-basis
  Ethane(aq)                               30  Make non-basis
  Ethanol(aq)                              30  Make non-basis
  Ethylene(aq)                             30  Make non-basis
  Ethyne(aq)                               30  Make non-basis
  Formic_acid(aq)                          30  Make non-basis
  Glycolic_acid(aq)                        30  Make non-basis
  H2(aq)                                   27  Homogenous equilibrium
  HSO5-                                    30  Make non-basis
  Lactic_acid(aq)                          30  Make non-basis
  Malonic_acid(aq)                         30  Make non-basis
  Methane(aq)                              30  Make non-basis
  O2(aq)                                   27  Homogenous equilibrium
  Oxalic_acid(aq)                          30  Make non-basis
  Pentanoic_acid(aq)                       30  Make non-basis
  Phenol(aq)                               30  Make non-basis
  Propanoic_acid(aq)                       30  Make non-basis
  S2O4--                                   30  Make non-basis
  S2O6--                                   30  Make non-basis
  S2O8--                                   30  Make non-basis
  S3--                                     30  Make non-basis
  S3O6--                                   30  Make non-basis
  S4--                                     30  Make non-basis
  S4O6--                                   30  Make non-basis
  S5--                                     30  Make non-basis
  S5O6--                                   30  Make non-basis
  SO3--                                    30  Make non-basis
  Succinic_acid(aq)                        30  Make non-basis
  Toluene(aq)                              30  Make non-basis
  o-Phthalate                              30  Make non-basis


                   --- Inactive Species ---

    None
```

Various relevant statistics are included here. Note that most types of input concentration values are converted to molalities.

3.5.3 Iterative Calculations

The fourth section consists of a rundown on the iterative calculations made by the code. For the present example, this is:

```
- - BEGIN ITERATIVE CALCULATIONS  - - - - - - - - - - - - - - - - - - - - - - -


 Starting Pre-Newton-Raphson Optimization.

             Completed pass   1 in   3 cycles.
             Completed pass   2 in   1 cycles.

   Done. Optimization ended within requested limits.


 Starting hybrid Newton-Raphson iteration.

   Done. Hybrid Newton-Raphson iteration converged in   4 iterations.
```

3.5.4 Principal Results

The fifth section contains the principal results of the calculations. For the present example, this is:

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -


           --- Elemental Composition of the Aqueous Solution ---

   Element         mg/L         mg/kg.sol        Moles/kg.H2O

     O          0.91080E+06    0.89001E+06    5.5627418346E+01
     Ca          421.93         412.30        1.0287440065E-02
     Cl          19805.         19353.        5.4588228648E-01
     H          0.11451E+06    0.11190E+06    1.1101873052E+02
     C           24.854         24.286        2.0220000148E-03
     K           408.42         399.10        1.0207605432E-02
     Mg          1322.0         1291.8        5.3149561015E-02
     Na          11020.         10768.        4.6838226438E-01
     S           926.41         905.26        2.8231299257E-02



           --- Numerical Composition of the Aqueous Solution ---

   Species                    mg/L         mg/kg.sol         Molality

 H2O                        0.10234E+07    0.10000E+07    5.5508427149E+01
 Ca++                        421.93         412.30        1.0287440065E-02
 Cl-                         19805.         19353.        5.4588228648E-01
 H+                         -0.15037       -0.14694       -1.4578369723E-04
 HCO3-                       126.26         123.38        2.0220000148E-03
 K+                          408.42         399.10        1.0207605432E-02
 Mg++                        1322.0         1291.8        5.3149561015E-02
 Na+                         11020.         10768.        4.6838226438E-01
 SO4--                       2775.4         2712.0        2.8231299257E-02
 O2(g)                      0.15899E-20    0.15536E-20    4.8552512423E-26
 H2(aq)                     0.63521E-33    0.62071E-33    3.0791131785E-37
 O2(aq)                     0.13270E-14    0.12967E-14    4.0524430287E-20

   Some of the above data may not be physically significant.
```

```
          --- Sensible Composition of the Aqueous Solution ---

  Species                     mg/L         mg/kg.sol         Molality

Ca++                         421.93         412.30        1.0287440065E-02
Cl-                          19805.         19353.        5.4588228648E-01
OH-                          2.5373         2.4794        1.4578369723E-04
HCO3-                        126.26         123.38        2.0220000148E-03
K+                           408.42         399.10        1.0207605432E-02
Mg++                         1322.0         1291.8        5.3149561015E-02
Na+                          11020.         10768.        4.6838226438E-01
SO4--                        2775.4         2712.0        2.8231299257E-02
H2(aq)                      0.63521E-33    0.62071E-33    3.0791131785E-37
O2(aq)                      0.13270E-14    0.12967E-14    4.0524430287E-20

  The above data have physical significance, but some may be
  inconsistent with certain analytical methods or reporting schemes.



                   Oxygen fugacity=  3.70025E-17 bars
               Log oxygen fugacity=  -16.432

                 Activity of water=  9.82326E-01
             Log activity of water= -7.74451E-03

            Mole fraction of water=  9.80899E-01
        Log mole fraction of water= -8.37582E-03

     Activity coefficient of water=  1.00145E+00
 Log activity coefficient of water=  6.31308E-04

               Osmotic coefficient=  0.91574
Stoichiometric osmotic coefficient=  0.88513

                 Sum of molalities=  1.08093E+00
  Sum of stoichiometric molalities=  1.11831E+00

                Ionic strength (I)=  6.22681E-01 molal
     Stoichiometric ionic strength=  6.96657E-01 molal

               Ionic asymmetry (J)=  3.68962E-02 molal
    Stoichiometric ionic asymmetry=  3.54128E-02 molal

                  Solvent fraction=   1.0000     kg.H2O/kg.sol
                   Solute fraction= -2.08167E-17 kg.solutes/kg.sol


          --- The pH, Eh, pe-, and Ah on various pH scales ---

                              pH     Eh, volts        pe-       Ah, kcal

NBS pH scale                8.2200     0.5000     8.4522E+00     11.5312
Mesmer pH scale             8.1079     0.5066     8.5642E+00     11.6840

  The single ion activities and activity coefficients listed below
  are consistent with the NBS pH scale.

    The pHCl=      8.6852


          --- HCO3-CO3-OH Total Alkalinity ---

                0.216779E-02 eq/kg.H2O
                 108.390     mg/kg.sol CaCO3
                 132.149     mg/kg.sol HCO3-
                 110.922     mg/L CaCO3
                 135.236     mg/L HCO3-


          --- Extended Total Alkalinity ---
```

```
                0.216779E-02 eq/kg.H2O
                 108.390     mg/kg.sol CaCO3
                 132.149     mg/kg.sol HCO3-
                 110.922     mg/L CaCO3
                 135.236     mg/L HCO3-


          --- Electrical Balance Totals ---

                                 eq/kg.H2O

       Sigma(mz) cations=     5.5991392766E-01
        Sigma(mz) anions=    -5.5896272440E-01
            Total charge=     1.1188766521E+00
             Mean charge=     5.5943832603E-01
        Charge imbalance=     9.5120325837E-04


        The electrical imbalance is:

             0.0850 per cent of the total charge
             0.1700 per cent of the mean charge



               --- Distribution of Aqueous Solute Species ---

   Species                  Molality    Log Molality   Log Gamma  Log Activity

Cl-                        5.2436E-01      -0.2804      -0.1848      -0.4652
Na+                        4.4488E-01      -0.3518      -0.1891      -0.5408
Mg++                       4.0722E-02      -1.3902      -0.5325      -1.9226
NaCl(aq)                   1.6481E-02      -1.7830       0.0000      -1.7830
SO4--                      1.3219E-02      -1.8788      -0.7334      -2.6122
K+                         9.9812E-03      -2.0008      -0.2180      -2.2188
Ca++                       9.2681E-03      -2.0330      -0.6390      -2.6720
MgSO4(aq)                  7.5308E-03      -2.1232       0.0000      -2.1232
NaSO4-                     6.6504E-03      -2.1772      -0.1559      -2.3331
MgCl+                      4.6384E-03      -2.3336      -0.1891      -2.5227
HCO3-                      1.2792E-03      -2.8930      -0.1559      -3.0490
CaSO4(aq)                  6.7125E-04      -3.1731       0.0000      -3.1731
NaHCO3(aq)                 3.6669E-04      -3.4357       0.0000      -3.4357
CaCl+                      2.2715E-04      -3.6437      -0.1891      -3.8328
MgHCO3+                    1.7914E-04      -3.7468      -0.1891      -3.9359
KSO4-                      1.6014E-04      -3.7955      -0.1559      -3.9514
MgCO3(aq)                  7.9159E-05      -4.1015       0.0000      -4.1015
KCl(aq)                    6.6287E-05      -4.1786       0.0000      -4.1786
CaCl2(aq)                  5.6761E-05      -4.2460       0.0000      -4.2460
CO3--                      3.3748E-05      -4.4718      -0.6860      -5.1578
CaHCO3+                    3.2721E-05      -4.4852      -0.1891      -4.6743
CaCO3(aq)                  3.1430E-05      -4.5027       0.0000      -4.5027
CO2(aq)                    1.0502E-05      -4.9787       0.0622      -4.9165
NaCO3-                     9.3693E-06      -5.0283      -0.1559      -5.1842
OH-                        2.4352E-06      -5.6135      -0.1694      -5.7828
CaOH+                      7.5743E-08      -7.1207      -0.1891      -7.3097
NaOH(aq)                   7.5268E-08      -7.1234       0.0000      -7.1234
H+                         7.7993E-09      -8.1079      -0.1121      -8.2200
KOH(aq)                    3.4156E-09      -8.4665       0.0000      -8.4665
HSO4-                      2.0081E-09      -8.6972      -0.1559      -8.8531
HCl(aq)                    4.4139E-10      -9.3552       0.0000      -9.3552
Mg4(OH)4++++               1.2872E-12     -11.8904      -2.7012     -14.5915
KHSO4(aq)                  5.7888E-13     -12.2374       0.0000     -12.2374
O2(aq)                     4.0524E-20     -19.3923       0.0622     -19.3301
H2SO4(aq)                  8.4505E-21     -20.0731       0.0000     -20.0731
ClO-                       8.3994E-26     -25.0757      -0.1559     -25.2317
HClO(aq)                   1.3108E-26     -25.8825       0.0000     -25.8825
HO2-                       2.5187E-30     -29.5988      -0.1559     -29.7547
Formate                    9.5500E-37     -36.0200      -0.1694     -36.1894
Mg(For)+                   3.2149E-37     -36.4928      -0.1891     -36.6819
H2(aq)                     3.0791E-37     -36.5116       0.0622     -36.4494
```

```
Na(For)(aq)              2.0878E-37     -36.6803       0.0000     -36.6803
Ca(For)+                 5.7255E-38     -37.2422      -0.1891     -37.4313
HSO5-                    2.3556E-38     -37.6279      -0.1559     -37.7838
K(For)(aq)               4.1873E-39     -38.3781       0.0000     -38.3781
SO3--                    1.3016E-39     -38.8855      -0.6860     -39.5715
HSO3-                    3.7134E-41     -40.4302      -0.1559     -40.5861
Formic_acid(aq)          2.2061E-41     -40.6564       0.0000     -40.6564
ClO2-                    1.7892E-43     -42.7473      -0.1559     -42.9032
Oxalate                  6.9763E-44     -43.1564      -0.7334     -43.8898
CO(aq)                   5.0533E-44     -43.2964       0.0000     -43.2964
ClO3-                    2.8070E-47     -46.5518      -0.1694     -46.7211
H2SO3(aq)                1.5910E-47     -46.7983       0.0000     -46.7983
SO2(aq)                  1.1529E-47     -46.9382       0.0000     -46.9382
H-Oxalate                2.0078E-48     -47.6973      -0.1559     -47.8532
HClO2(aq)                1.1132E-48     -47.9534       0.0000     -47.9534
S2O8--                   8.1608E-54     -53.0883      -0.7334     -53.8217
ClO4-                    2.1625E-55     -54.6650      -0.1694     -54.8344
Oxalic_acid(aq)          1.5744E-55     -54.8029       0.0000     -54.8029
S2O6--                   2.0980E-63     -62.6782      -0.7334     -63.4116
Mg(For)2(aq)             9.9727E-73     -72.0012       0.0000     -72.0012
Ca(For)2(aq)             1.7760E-73     -72.7505       0.0000     -72.7505
Na(For)2-                1.0142E-73     -72.9939      -0.1559     -73.1498
K(For)2-                 1.9439E-75     -74.7113      -0.1559     -74.8672
Formaldehyde(aq)         3.0808E-79     -78.5113       0.0000     -78.5113
S2O5--                   5.6414E-86     -85.2486      -0.7334     -85.9820

   Species with molalities less than  1.000-100 are not listed.



     --- Major Species by Contribution to Aqueous Mass Balances ---


Species Accounting for 99% or More of Aqueous Ca++

    Species                   Factor    Molality     Per Cent

  Ca++                         1.00    9.2681E-03     90.09
  CaSO4(aq)                    1.00    6.7125E-04      6.52
  CaCl+                        1.00    2.2715E-04      2.21
  CaCl2(aq)                    1.00    5.6761E-05    5.517E-01
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  Subtotal                             1.0223E-02     99.38


Species Accounting for 99% or More of Aqueous Cl-

    Species                   Factor    Molality     Per Cent

  Cl-                          1.00    5.2436E-01     96.06
  NaCl(aq)                     1.00    1.6481E-02      3.02
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  Subtotal                             5.4084E-01     99.08


Species Accounting for 99% or More of Aqueous HCO3-

    Species                   Factor    Molality     Per Cent

  HCO3-                        1.00    1.2792E-03     63.27
  NaHCO3(aq)                   1.00    3.6669E-04     18.13
  MgHCO3+                      1.00    1.7914E-04      8.86
  MgCO3(aq)                    1.00    7.9159E-05      3.91
  CO3--                        1.00    3.3748E-05      1.67
  CaHCO3+                      1.00    3.2721E-05      1.62
  CaCO3(aq)                    1.00    3.1430E-05      1.55
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  Subtotal                             2.0021E-03     99.02


Species Accounting for 99% or More of Aqueous K+
```

```
    Species                   Factor    Molality     Per Cent

  K+                           1.00    9.9812E-03     97.78
  KSO4-                        1.00    1.6014E-04      1.57
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  Subtotal                             1.0141E-02     99.35


Species Accounting for 99% or More of Aqueous Mg++

    Species                   Factor    Molality     Per Cent

  Mg++                         1.00    4.0722E-02     76.62
  MgSO4(aq)                    1.00    7.5308E-03     14.17
  MgCl+                        1.00    4.6384E-03      8.73
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  Subtotal                             5.2891E-02     99.51


Species Accounting for 99% or More of Aqueous Na+

    Species                   Factor    Molality     Per Cent

  Na+                          1.00    4.4488E-01     94.98
  NaCl(aq)                     1.00    1.6481E-02      3.52
  NaSO4-                       1.00    6.6504E-03      1.42
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  Subtotal                             4.6801E-01     99.92


Species Accounting for 99% or More of Aqueous SO4--

    Species                   Factor    Molality     Per Cent

  SO4--                        1.00    1.3219E-02     46.82
  MgSO4(aq)                    1.00    7.5308E-03     26.68
  NaSO4-                       1.00    6.6504E-03     23.56
  CaSO4(aq)                    1.00    6.7125E-04      2.38
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  Subtotal                             2.8071E-02     99.43


Species Accounting for 99% or More of Aqueous H2(aq)

    Species                   Factor    Molality     Per Cent

  H2(aq)                       1.00    3.0791E-37    100.00
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  Subtotal                             3.0791E-37    100.00


Species Accounting for 99% or More of Aqueous O2(aq)

    Species                   Factor    Molality     Per Cent

  O2(aq)                       1.00    4.0524E-20    100.00
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  Subtotal                             4.0524E-20    100.00



               --- Aqueous Redox Reactions ---

  Couple                         Eh, volts      pe-      log fO2   Ah, kcal

DEFAULT                            0.500    8.4522E+00   -16.432    11.531

   Couples required to satisfy the default redox constraint are not listed.
```

```
     --- Saturation States of Aqueous Reactions Not Fixed at Equilibrium ---

  Reaction                           Log Q/K    Affinity, kcal

None


          --- Saturation States of Pure Solids ---

      Phase                      Log Q/K    Affinity, kcal

    Anhydrite                    -0.97782     -1.33402
    Aragonite                     0.50594      0.69025     SSATD
    Arcanite                     -5.24901     -7.16114
    Artinite                     -1.92896     -2.63165
    Bassanite                    -1.62659     -2.21913
    Bischofite                   -7.29176     -9.94804
    Bloedite                     -5.78205     -7.88836
    Brucite                      -1.79613     -2.45043
    CaSO4:0.5H2O(beta)           -1.79469     -2.44847
    Calcite                       0.65034      0.88725     SSATD
    Dolomite                      3.23395      4.41202     SSATD
    Dolomite-dis                  1.68955      2.30502     SSATD
    Dolomite-ord                  3.23395      4.41202     SSATD
    Epsomite                     -2.62677     -3.58366
    Gaylussite                   -4.61443     -6.29540
    Glauberite                   -3.50913     -4.78745
    Gypsum                       -0.81741     -1.11518
    Halite                       -2.59152     -3.53558
    Hexahydrite                  -2.85453     -3.89439
    Huntite                       1.94325      2.65115     SSATD
    Hydromagnesite               -3.38940     -4.62410
    Ice                          -0.14644     -0.19979
    Kainite                      -6.93066     -9.45540
    Kalicinite                   -5.55145     -7.57376
    Kieserite                    -4.27560     -5.83314
    Lansfordite                  -1.63122     -2.22545
    Magnesite                     0.95480      1.30262     SSATD
    Mg1.25SO4(OH)0.5:0.5H2O      -6.17326     -8.42209
    Mg1.5SO4(OH)                 -6.53902     -8.92109
    Mirabilite                   -2.63156     -3.59019
    Monohydrocalcite             -0.19110     -0.26072
    Na2CO3                       -7.09285     -9.67667
    Na2CO3:7H2O                  -5.91076     -8.06397
    Na4Ca(SO4)3:2H2O             -6.79373     -9.26859
    Nahcolite                    -3.47800     -4.74499
    Natron                       -5.59830     -7.63767
    Nesquehonite                 -1.77033     -2.41524
    Oxychloride-Mg               -5.53657     -7.55346
    Pentahydrite                 -3.18638     -4.34713
    Periclase                    -6.82578     -9.31232
    Picromerite                  -7.19153     -9.81130
    Pirssonite                   -4.75010     -6.48049
    Starkeyite                   -3.56594     -4.86496
    Sylvite                      -3.52987     -4.81575
    Syngenite                    -4.74167     -6.46899
    Thenardite                   -3.38481     -4.61785
    Thermonatrite                -6.88069     -9.38723

       Phases with affinities less than -10 kcal are not listed.


          --- Saturation States of Pure Liquids ---

      Phase                      Log Q/K    Affinity, kcal

    H2O                          -0.00774     -0.01057

       Phases with affinities less than -10 kcal are not listed.
```

```
          --- Summary of Saturated and Supersaturated Phases ---

               There are no saturated phases.
               There are    7 supersaturated phases.


                    --- Fugacities ---

     Gas                    Log Fugacity    Fugacity

   H2O(g)                     -1.59314    2.55185E-02
   CO2(g)                     -3.44761    3.56768E-04
   HCl(g)                    -14.99068    1.02170E-15
   O2(g)                     -16.43177    3.70025E-17
   Chlorine                  -30.02809    9.37359E-31
   H2(g)                     -33.34436    4.52522E-34
   CO(g)                     -40.28963    5.13300E-41
   SO2(g)                    -47.10819    7.79487E-48
   Na(g)                     -68.35618    4.40376E-69
   K(g)                      -70.75162    1.77164E-71
   H2S(g)                   -110.73308    1.84892-111
   CH4(g)                   -113.90751    1.23733-114
   Mg(g)                    -118.07480    8.41784-119
   Ca(g)                    -141.65256    2.22557-142
   C(g)                     -173.70344    1.97950-174
   S2(g)                    -180.42919    3.72225-181
   C2H4(g)                  -190.83131    1.47465-191

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

The tables in this section of the output should be understood by a code user with the qualifications to run this code (described in Sections 2.1 and 8). However, some aspects of the second (“numerical composition of the aqueous solution”) and third (“sensible composition of the aqueous solution”) tables deserve some note here. The “numerical composition” is based on standard mass balance relationships (cf. B.1.3.1 and B.3.1) applied to the data file basis species set, as possibly modified by special basis switching. Some of the totals here may lack physical meaning. The totals here for H_2O, H^+, and O_2(g) typically fall into this camp. The “sensible composition” is designed to overcome that problem by listing results that are physically meaningful (though they may not necessarily correspond to protocols used to report analytical results).

3.5.5 End of EQ3NR Run

The sixth and final section is exemplified by:

```
 The pickup file has been written.

 No further input found.


          Start time = 15:47:54  06Sep2002
            End time = 15:47:55  06Sep2002

           Run time =  0.750     seconds

 Normal exit
```

3.6 EQ6 OUTPUT FILES

EQ6 is a reaction-path code. As such, it calculates how a model aqueous system changes as it reacts with a set of “reactants” and/or undergoes changes in temperature and pressure. The code is also capable of simply equilibrating an initial system that contains one or more elements of dis-equilibrium. In this case, the number of reaction path steps is zero. Refer to Appendix D – EQ6 Supplimentary Information, for information about calculational details.

The EQ6 code produces three kinds of output files. The first is the normal “output” file, which normally carries a “.6o” filename extension. The second is the “tab” file, which normally carries a “.6t” filename extension. This contains summary tables of the run that may be useful in plotting results. If the string “TABFILEASCSV” is included in the input file table, this file will be written as a “.csv” or comma-separated-value file. By changing the “.6t” to “.csv”, the user will be able to open this file directly into Microsoft Excel or other spreadsheet program. The user may also find a “.6tx” file, which is a “scrambled” tab file. This is a device to allow a subsequent run to produce a “complete” tab file. The third kind of output file is the “pickup” file, which provides information to be carried forward into a subsequent EQ6 calculation. Such a pickup file, which normally carries a “.6p” filename extension, is a complete EQ6 input file. The EQ6 pickup file will not receive further discussion here, as it has effectively been covered in the section on EQ6 input files. If for some reason a run terminates abnormally, the user may find a “backup” file that is a “pickup” file written at some point prior to the abnormal termination. This may provide a more economical means of restarting the run.

The EQ6 output (“.6o”) file consists of five parts. The example to be discussed here (micro.6o) is that obtained by running the EQ6 test case library input file micro.6i, which was the primary example presented in the section on the EQ6 input file.

3.6.1 Header, Input Echo and Input Recap

The first part is a header section. This is virtually identical to that employed on the EQ3NR output file described previously (Section 3.5.1), and an example will not be shown here. This is followed by the second part, a line-by-line echo of the input file (also in Section 3.5.1). Again, an example will not be shown here. The third part (which is analogous to the third part of the EQ3NR output file, Section 3.5.2) lists the code inputs, along with any defaults or overrides (the last few lines of the echo of the input file are included here):

```
|--->|None                   | 0.000000000000000E+00| --                   |
|-----------------------------------------------------------------------------|
|End of problem                                                               |
|-----------------------------------------------------------------------------|

   Done reading problem   1.


 The phase Quartz has been user-suppressed.

 The phase Chalcedony has been user-suppressed.

 The phase Tridymite has been user-suppressed.
```

```
                 --- Inactive Species ---

    Chalcedony
    Quartz
    Tridymite


 The activity coefficients of aqueous species will be calculated using
   the B-dot equation.


      --- Numbers of Phases, Species, and Groups Thereof---

      Entity               Date Base    Dimension   Current Problem

   Chemical Elements            81          81           6
   Basis Species               201         259           7
   Phases                     1135        1159          29
   Species                    3031        3523           0
   Aqueous Species            1769        1769          22
   Pure Minerals              1120        1120          26
   Pure Liquids                  1           3           1
   Gas Species                  93          93           2
   Solid Soutions               12          12           0


   Temperature=    25.0000 C

   Pressure= the data file reference curve value at any temperature


 xistti=  0.00000E+00 (Initial value of Xi)
 ximaxi=  1.00000E+00 (Maximum value of Xi)
 tistti=  0.00000E+00 (Initial value of time, sec)
 timmxi=  1.00000E+38 (Maximum value of time, sec)
 phmini= -1.00000E+38 (Minimum value of pH)
 phmaxi=  1.00000E+38 (Maximum value of pH)
 ehmini= -1.00000E+38 (Minimum value of Eh, v)
 ehmaxi=  1.00000E+38 (Maximum value of Eh, v)
 o2mini= -1.00000E+38 (Minimum value of log fO2)
 o2maxi=  1.00000E+38 (Maximum value of log fO2)
 awmini= -1.00000E+38 (Minimum value of aw)
 awmaxi=  1.00000E+38 (Maximum value of aw)
 kstpmx=          200 (Maximum number of steps this run)


   dlxprn=  1.00000E+00 (Print interval in Xi)
   dlxprl=  1.00000E+00 (Print interval in log Xi)
   dltprn=  1.00000E+38 (Print interval in time, sec)
   dltprl=  1.00000E+38 (Print interval in log time)
   dlhprn=  1.00000E+38 (Print interval in pH units)
   dleprn=  1.00000E+38 (Print interval in Eh, v)
   dloprn=  1.00000E+38 (Print interval in log fO2)
   dlaprn=  1.00000E+38 (Print interval in aw)
   ksppmx=          100 (Print interval in steps)


     dlxplo=  1.00000E+38 (Plot interval in Xi)
     dlxpll=  1.00000E+38 (Plot interval in log Xi)
     dltplo=  1.00000E+38 (Plot interval in time)
     dltpll=  1.00000E+38 (Plot interval in log time)
     dlhplo=  1.00000E+38 (Plot interval in pH units)
     dleplo=  1.00000E+38 (Plot interval in Eh, v)
     dloplo=  1.00000E+38 (Plot interval in log fO2)
     dlaplo=  1.00000E+38 (Plot interval in aw)
     ksplmx=        10000 (Plot interval in steps)


   dlxdmp=  1.00000E+38 (PRS transfer interval in Xi)
```

```
    dlxmx0=  1.00000E-08 (Zero-order step size in Xi)
    dlxmax=  1.00000E+38 (Maximum step size)


nordmx=  6 (Maximum dimensioned order)


iopt(1)=   0 (Physical system model)
iopt(2)=   0 (Kinetic mode)
iopt(3)=   0 (Suppress phase boundary searches)
iopt(4)=   0 (Solid solutions)
iopt(5)=   0 (Clear ES solids read from the input file)
iopt(6)=   0 (Clear ES solids at the starting point)
iopt(7)=   0 (Clear ES solids at the end of the run)
iopt(8)=   0 (Not used)
iopt(9)=   0 (Clear PRS solids read from the input file)
iopt(10)=  0 (Clear PRS solids at the end of the run)
iopt(11)=  0 (Auto basis switching, in pre-Newton-Raphson optimization)
iopt(12)=  0 (Auto basis switching, after Newton-Raphson iteration)
iopt(13)=  0 (Calculational mode)
iopt(14)=  0 (ODE integrator corrector mode)
iopt(15)=  1 (Force global redox suppression)
iopt(16)=  0 (Backup file options)
iopt(17)=  0 (Pickup file options)
iopt(18)=  0 (Tab file options)


  iopg(1)=   0 (Aqueous species activity coefficient model)
  iopg(2)=   0 (pH scale)


iopr(1)=   0 (List all species)
iopr(2)=   0 (List all reactions)
iopr(3)=   0 (List HC diamaters)
iopr 4)=   0 (Aqueous species concentration print cut-off)
iopr(5)=   0 (Ion/H+ activity ratios)
iopr(6)=   0 (Mass balance percentages)
iopr(7)=   0 (Affinity print cut-off)
iopr(8)=  -1 (Fugacities)
iopr(9)=   0 (Mean molal activity coefficient)
iopr(10)=  0 (Pitzer coefficients tabulation)
iopr(17)=  0 (Pickup file format)


  iodb(1)=   0 (General diagnostics)
  iodb(2)=   0 (Kinetics diagnostics)
  iodb(3)=   0 (Pre-Newton-Raphson optimization)
  iodb(4)=   0 (Newton-Raphson iterations)
  iodb(5)=   0 (Order/scaling calculations)
  iodb(6)=   0 (Hypothetical affinity iterations)
  iodb(7)=   0 (Search iterations)
  iodb(8)=   0 (ODE corrector iterations)


tolbt =  1.00000E-06 (Residual function convergence tolerance)
toldl =  1.00000E-06 (Correction term convergence tolerance)
tolxsf=  1.00000E-06 (Search/find tolerance (general, relative))
tolxst=  1.00000E-08 (Search/find tolerance on time (relative))
tolxsu=  1.00000E-05 (Search/find tolerance on pH, Eh, etc. (absolute))
tolsat=  5.00000E-04 (Saturation tolerance)
tolsst=  1.00000E-03 (Supersaturation tolerance)


  sscrew(1)=  1.000E-04 (Matrix variable step size parameter)
  sscrew(2)=    0.00000 (Not used)
  sscrew(3)=  1.000E-04 (Rate function step size parameter)
  sscrew(4)=  1.000E-06 (Rate function corrector parameter)
  sscrew(5)=    4.00000 (Under-relaxation parameter (Newton-Raphson))
  sscrew(6)=    4.00000 (economy mode step size)
```

```
 zklogu=     -7.000 (threshhold log mass for solids)
 zklogl=      2.000 (Log mass decrement for PRS shift)
 zkfac =      0.980 (Shift adjustment factor)
 zklgmn=     -7.009 (Minimum log mass after a shift)


   itermx= 200 (Newton-Raphson iteration limit)
   ntrymx= 100 (Phase assemblage try limit)
   npslmx=   8 (Critical phase instability slide limit)
   nsslmx=   8 (Critical redox instability slide limit)


 There is no redox balance constraint; hence, the fO2,
 Eh, pe, and Ah are all undefined.


                     --- Reactants/Rate Laws ---


   Forward direction

 Maximum_Microcline         Specified relative rate

                       rkb1=  1.00000E+00
                       rkb2=  0.00000E+00
                       rkb3=  0.00000E+00



   Backward direction

 Maximum_Microcline         Instantaneous equilibrium
```

3.6.2 Iterative Calculations and Principal Results

The fourth part includes both calculation summaries as well as detailed results at the so-called print points. It is thus equivalent to the fourth and fifth parts of the EQ3NR output file (Sections 3.5.3 and 3.5.4, respectively). The print points include those generated by print interval controls (see the section on the EQ6 input file) as well as certain other points, such as the initial and final points of the run and any points at which certain events occur (e.g., a mineral starts to precipitate). The initial part for the example used here summarizes the initial iterative calculation, the first print point, and a calculation summary for the first step of reaction progress:

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - -


 Stepping to Xi=  0.0000E+00, delxi=  0.0000E+00, nord= 0

 Attempted phase assemblage number   1

    1  H2O
    2  Al+++
    3  Cl-
    4  H+
    5  K+
    6  SiO2(aq)
    7  O2(g)

 Steps completed=     0, iter=   2

 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -


                   Xi=  0.00000E+00
               Log Xi= -99999.00000
```

```
Temperature=  25.00 C

Pressure=   1.0132     bars


Start or restart of the run.


                    --- Reactant Summary ---


    Definitions and conventions

         Delta x = x now - x at start
         Affinity is + for forward direction (destruction),
                     - for reverse direction (formation)
         Rates are + for forward direction (destruction),
                   - for reverse direction (formation)


    Reactant                  Moles     Delta moles    Mass, g    Delta mass, g

 Maximum_Microcline         1.0000E+00   0.0000E+00   2.7833E+02   0.0000E+00


                   Mass remaining=  2.7833E+02 grams
                   Mass destroyed=  0.0000E+00 grams


    Reactant                   Affinity     Rel. Rate
                               kcal/mol      mol/mol

 Maximum_Microcline              59.7814   1.0000E+00


  Affinity of the overall irreversible reaction= 59.7814 kcal.
  Contributions from irreversible reactions with no thermodynamic data
  are not included.


          --- Elemental Composition of the Aqueous Solution ---

  Element         mg/L         mg/kg.sol        Moles/kg.H2O

    O          0.88810E+06    0.88810E+06    5.5508435062E+01
    Al         0.26981E-07    0.26981E-07    1.0000000000E-12
    Cl          3.5865         3.5865        1.0116290926E-04
    H          0.11190E+06    0.11190E+06    1.1101697129E+02
    K          0.39098E-07    0.39098E-07    1.0000000000E-12
    Si         0.28085E-07    0.28085E-07    1.0000000000E-12


          --- Sensible Composition of the Aqueous Solution ---

  Species                    mg/L         mg/kg.sol         Molality

Al+++                      0.26981E-07    0.26981E-07    1.0000000000E-12
Cl-                         3.5865         3.5865        1.0116290926E-04
H+                         0.10197        0.10197        1.0116290526E-04
K+                         0.39098E-07    0.39098E-07    1.0000000000E-12
SiO2(aq)                   0.60084E-07    0.60084E-07    1.0000000000E-12

  The above data have physical significance, but some may be
  inconsistent with certain analytical methods or reporting schemes.
```

```
          --- The pH on various pH scales ---

                                         pH

     NBS pH scale                     4.0000
     Mesmer pH scale                  3.9950

  The single ion activities and activity coefficients listed below
  are consistent with the NBS pH scale.

    The pHCl=      8.0001



                  Oxygen fugacity=  1.99526E-01 bars
              Log oxygen fugacity= -0.70000

                Activity of water=  9.99996E-01
            Log activity of water= -1.57685E-06

           Mole fraction of water=  9.99996E-01
       Log mole fraction of water= -1.58297E-06

     Activity coefficient of water=  1.00000E+00
 Log activity coefficient of water=  6.11677E-09

              Osmotic coefficient=  0.99613
Stoichiometric osmotic coefficient=  0.99612

                Sum of molalities=  2.02324E-04
  Sum of stoichiometric molalities=  2.02326E-04

               Ionic strength (I)=  1.01161E-04 molal
     Stoichiometric ionic strength=  1.01163E-04 molal

              Ionic asymmetry (J)=  3.70880E-12 molal
    Stoichiometric ionic asymmetry=  4.00000E-12 molal

                 Solvent fraction=   1.0000     kg.H2O/kg.sol
                  Solute fraction=  3.68845E-06 kg.solutes/kg.sol


                    Mass of solution=   1000.0     g
                     Mass of solvent=   1000.0     g
                     Mass of solutes=  3.68847E-03 g


          --- HCO3-CO3-OH Total Alkalinity ---

                0.102319E-09 eq/kg.H2O
                0.511594E-05 mg/kg.sol CaCO3
                0.623736E-05 mg/kg.sol HCO3-
                0.511594E-05 mg/L CaCO3
                0.623736E-05 mg/L HCO3-


          --- Extended Total Alkalinity ---

                0.102319E-09 eq/kg.H2O
                0.511594E-05 mg/kg.sol CaCO3
                0.623736E-05 mg/kg.sol HCO3-
                0.511594E-05 mg/L CaCO3
                0.623736E-05 mg/L HCO3-



       --- Aqueous Solution Charge Balance ---

         Actual Charge imbalance= -4.5138E-20 eq
       Expected Charge imbalance= -4.0658E-20 eq
              Charge discrepancy= -4.4800E-21 eq
```

```
             Sigma |equivalents|=  2.0232E-04 eq

         Actual Charge imbalance= -4.5137E-20 eq/kg.solu
       Expected Charge imbalance= -4.0657E-20 eq/kg.solu
              Charge discrepancy= -4.4800E-21 eq/kg.solu
             Sigma |equivalents|=  2.0232E-04 eq/kg.solu

     Relative charge discrepancy= -2.2143E-17


               --- Distribution of Aqueous Solute Species ---

   Species                  Molality    Log Molality   Log Gamma  Log Activity

H+                         1.0116E-04      -3.9950      -0.0050      -4.0000
Cl-                        1.0116E-04      -3.9950      -0.0051      -4.0001
HCl(aq)                    2.1377E-09      -8.6701       0.0000      -8.6701
OH-                        1.0232E-10      -9.9900      -0.0051      -9.9951
SiO2(aq)                   1.0000E-12     -12.0000       0.0000     -12.0000
K+                         1.0000E-12     -12.0000      -0.0051     -12.0051
Al+++                      9.0364E-13     -12.0440      -0.0450     -12.0890
AlOH++                     9.4231E-14     -13.0258      -0.0203     -13.0461
Al(OH)2+                   2.0969E-15     -14.6784      -0.0051     -14.6835
HAlO2(aq)                  3.0066E-17     -16.5219       0.0000     -16.5219
KCl(aq)                    3.1640E-18     -17.4998       0.0000     -17.4998
HSiO3-                     1.1286E-18     -17.9475      -0.0051     -17.9525
AlO2-                      1.0783E-19     -18.9673      -0.0051     -18.9723
KOH(aq)                    3.4268E-23     -22.4651       0.0000     -22.4651
Al2(OH)2++++               1.6318E-24     -23.7873      -0.0809     -23.8682
H2SiO4--                   1.1488E-27     -26.9397      -0.0203     -26.9600
Al3(OH)4(5+)               9.5242E-35     -34.0212      -0.1262     -34.1474
H6(H2SiO4)4--              2.4002E-54     -53.6198      -0.0203     -53.6400
H4(H2SiO4)4----            1.3839E-68     -67.8589      -0.0811     -67.9400

   Species with molalities less than  1.000-100 are not listed.



     --- Major Species by Contribution to Aqueous Mass Balances ---


Species Accounting for 99% or More of Aqueous Al+++

    Species                   Factor    Molality     Per Cent

  Al+++                        1.00    9.0364E-13     90.36
  AlOH++                       1.00    9.4231E-14      9.42
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  Subtotal                             9.9787E-13     99.79


Species Accounting for 99% or More of Aqueous Cl-

    Species                   Factor    Molality     Per Cent

  Cl-                          1.00    1.0116E-04    100.00
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  Subtotal                             1.0116E-04    100.00


Species Accounting for 99% or More of Aqueous K+

    Species                   Factor    Molality     Per Cent

  K+                           1.00    1.0000E-12    100.00
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  Subtotal                             1.0000E-12    100.00
```

```
Species Accounting for 99% or More of Aqueous SiO2(aq)

    Species                  Factor    Molality     Per Cent

  SiO2(aq)                    1.00    1.0000E-12    100.00
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  Subtotal                            1.0000E-12    100.00



                    --- Summary of Solid Phases (ES) ---

  Phase/End-member            Log moles     Moles        Grams     Volume, cm3

None


           --- Grand Summary of Solid Phases (ES + PRS + Reactants) ---

  Phase/End-member            Log moles     Moles        Grams     Volume, cm3

Maximum_Microcline               0.0000   1.0000E+00   2.7833E+02   0.0000E+00


                         Mass, grams       Volume, cm3

          Created         0.00000E+00       0.00000E+00
          Destroyed       0.00000E+00       0.00000E+00
          Net             0.00000E+00       0.00000E+00

         These volume totals may be incomplete because of missing
         partial molar volume data in the data base.


     --- Saturation States of Aqueous Reactions Not Fixed at Equilibrium ---

  Reaction                           Log Q/K    Affinity, kcal

None



          --- Saturation States of Pure Solids ---

      Phase                      Log Q/K    Affinity, kcal

    Diaspore                     -7.24932     -9.89014
    Ice                          -0.13870     -0.18923

       Phases with affinities less than -10 kcal are not listed.



          --- Saturation States of Pure Liquids ---

      Phase                      Log Q/K    Affinity, kcal

    H2O                           0.00000      0.00000     SATD

       Phases with affinities less than -10 kcal are not listed.


          --- Summary of Saturated and Supersaturated Phases ---

               There is 1 saturated phase.
               There are no supersaturated phases.


- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

```
Stepping to Xi=  1.0000E-08, delxi=  1.0000E-08, nord= 0
Steps completed=     1, iter=   3, ncorr= 0
Most rapidly changing is zvclg1(SiO2(aq))=     -7.5229
```

Note that the output for a print point consists of tables that resemble (and include examples overlapping with) those in the fifth section of the EQ3NR output file (Section 3.5.4). A user sufficiently knowledgeable to run this code should be able to understand the output in these tables (refer to Sections 2.1 and 8.). Users not so knowledgeable should refer to Appendix D.

Typically the output on stepping covers a number of steps between print points, as exemplified by the following snippet from micro.6o:

```
Stepping to Xi=  2.2315E-05, delxi=  1.2315E-05, nord= 3
Steps completed=    22, iter=   4, ncorr= 0
Most rapidly changing is zvclg1(Diaspore)=     -4.7601

Stepping to Xi=  4.2540E-05, delxi=  2.0225E-05, nord= 4
Steps completed=    23, iter=   4, ncorr= 0
Most rapidly changing is zvclg1(Diaspore)=     -4.3982

Stepping to Xi=  6.4143E-05, delxi=  2.1603E-05, nord= 4
iter= 3

  Have 1 supersaturated phases.

--- The extent of supersaturation exceeds the normal tolerance ---
--- Cutting the step size and trying again ---

Stepping to Xi=  4.7940E-05, delxi=  5.4007E-06, nord= 4
Steps completed=    24, iter=   3, ncorr= 0
Most rapidly changing is zvclg1(Al+++)=     -5.7794

Stepping to Xi=  5.6071E-05, delxi=  8.1302E-06, nord= 3
Steps completed=    25, iter=   3, ncorr= 0
Most rapidly changing is zvclg1(Al+++)=     -5.9629

Stepping to Xi=  5.8113E-05, delxi=  2.0427E-06, nord= 5
Steps completed=    26, iter=   3, ncorr= 0
Most rapidly changing is zvclg1(Al+++)=     -6.0153

Stepping to Xi=  5.8601E-05, delxi=  4.8762E-07, nord= 3
iter= 3

  Have 1 supersaturated phases.

  The phase to be added is Kaolinite (17)


Attempted phase assemblage number   2

   1  H2O
   2  Al+++
   3  Cl-
   4  H+
   5  K+
   6  SiO2(aq)
   7  O2(g)
   8  Diaspore
   9  Kaolinite

Steps completed=    27, iter=   8, ncorr= 0
```

3.6.3 End of EQ6 Run

The fifth and final part of the EQ6 output file is exemplified by:

```
   All irreversible reaction rates are now zero.

   Each reactant is now saturated or exhausted.

 ---  The reaction path has terminated normally ---


          67 steps were taken
       Xi increased from
               0.00000E+00 to  1.49579E-04
       The average value of delxi was  2.23252E-06
       The average matrix dimension was    8

       The pH increased from 4.0000 to 7.3666
       The aw decreased from 1.0000 to 1.0000
       The mass of solvent water decreased from 1.0000 to 1.0000 kg

 No further input found.

          Start time = 15:48:14  06Sep2002
            End time = 15:48:16  06Sep2002

           Run time =   2.01     seconds

 Normal exit
```

This is analogous to the sixth and final part of the EQ3NR input file (Section 3.5.5).

3.7 EQ6 TABULATED ("TAB") OUTPUT FILE

The normal "tab" or ".6t" file is exemplified by the micro.6t file. This is shown on the following pages in landscape mode, as it is typically written assuming more than 80 characters per line. The contents of this file should be understandable by the knowledgeable user. The normal "tab" file for Version 8.0 is unchanged from that in the Version 7 code series.

```
Note: This main title is a carry-over from a previous run.
Description= "Microcline dissolution in pH 4 HCl"
Version level= 8.0
Revised 02/18/97    Revisor= T.J. Wolery
This is part of the EQ3/6 Test Case Library

  React "Maximum_Microcline" (KAlSi3O8) with a pH 4.0 HCl solution at 25C.
There is no time frame in this calculation. Precipitation of quartz, chal-
cedony, and tridymite is prevented by means of nxmod suppress options that
are inherited from the initializing EQ3NR input file, ph4hcl.3i. This is
test problem 3 of INTERA (1983, p. 65-73); it is similar to test problem 5
of Parkhurst, Thorstenson, and Plummer (1980). INTERA (1983) reported that
the product minerals formed were gibbsite, kaolinite, and muscovite. The run
terminates when the solution becomes saturated with microcline.

  The original problem called for suppression of only quartz. Chalcedony and
tridymite were apparently not on the data file used by INTERA (1983).

  Purpose: to compare against results obtained in a previous comparison of
EQ3/6 with PHREEQE (Parkhurst, Thorstenson, and Plummer, 1980) made by
INTERA (1983).

  This problem has no redox aspect. The option switch iopt(15) is set to 1
to indicate this to the code.


                            References

INTERA Environmental Consultants, Inc., 1983, Geochemical Models Suitable
  for Performance Assessment of Nuclear Waste Storage: Comparison of
  PHREEQE and EQ3/EQ6: Office of Nuclear Waste Isolation, Battelle Project
  Management Division, Columbus, Ohio, ONWI-473, 114 p.

Parkhurst, D.L., Thorstenson, D.C., and Plummer, L.N., 1980, PHREEQE-
  A Computer Program for Geochemical Calculations: Water Resources
  Investigations Report 80-96, U.S. Geological Survey, Reston, Virginia,
  210 p.

Running EQ3/6-V8.0-EQ6-EXE-R43-PC


data0.com.V8.R6
CII: GEMBOCHS.V2-EQ8-data0.com.V8.R6
THERMODYNAMIC DATABASE
generated by GEMBOCHS.V2-Jewel.src.R5 03-dec-1996 14:19:25
Output package:  eq3
Data set:        com
```

```
+--------------------------------------------------------------------



      xi     log xi    time, d  log days    tempc     press       ph     log fo2      eh        pe      kg h2o   tot aff

 0.000E+00 -999.0000 0.000E+00 -999.0000   25.0000    1.0132    4.0000   -0.7000 -999.0000 -999.0000    1.0000   59.7814
 1.000E-08   -8.0000 0.000E+00 -999.0000   25.0000    1.0132    4.0002   -0.7000 -999.0000 -999.0000    1.0000   30.5418
 1.000E-07   -7.0000 0.000E+00 -999.0000   25.0000    1.0132    4.0017   -0.7000 -999.0000 -999.0000    1.0000   23.7126
 1.000E-06   -6.0000 0.000E+00 -999.0000   25.0000    1.0132    4.0171   -0.7000 -999.0000 -999.0000    1.0000   16.8098
 7.134E-06   -5.1467 0.000E+00 -999.0000   25.0000    1.0132    4.1387   -0.7000 -999.0000 -999.0000    1.0000   10.3504
 1.000E-05   -5.0000 0.000E+00 -999.0000   25.0000    1.0132    4.1482   -0.7000 -999.0000 -999.0000    1.0000    9.5369
 5.860E-05   -4.2321 0.000E+00 -999.0000   25.0000    1.0132    4.4114   -0.7000 -999.0000 -999.0000    1.0000    4.9878
 8.782E-05   -4.0564 0.000E+00 -999.0000   25.0000    1.0132    4.8853   -0.7000 -999.0000 -999.0000    1.0000    4.1014
 1.000E-04   -4.0000 0.000E+00 -999.0000   25.0000    1.0132    5.9294   -0.7000 -999.0000 -999.0000    1.0000    2.4466
 1.019E-04   -3.9918 0.000E+00 -999.0000   25.0000    1.0132    7.3676   -0.7000 -999.0000 -999.0000    1.0000    0.4541
 1.496E-04   -3.8251 0.000E+00 -999.0000   25.0000    1.0132    7.3666   -0.7000 -999.0000 -999.0000    1.0000    0.0000



   log xi   time, d   log days  log alk  log tot   log tot   log tot   log a h2o
                                          co3--     so4--     s--

 -999.0000 0.000E+00 -999.0000   -9.9900    0.0000    0.0000    0.0000    0.0000
   -8.0000 0.000E+00 -999.0000   -9.9899    0.0000    0.0000    0.0000    0.0000
   -7.0000 0.000E+00 -999.0000   -9.9884    0.0000    0.0000    0.0000    0.0000
   -6.0000 0.000E+00 -999.0000   -9.9728    0.0000    0.0000    0.0000    0.0000
   -5.1467 0.000E+00 -999.0000   -9.8509    0.0000    0.0000    0.0000    0.0000
   -5.0000 0.000E+00 -999.0000   -9.8414    0.0000    0.0000    0.0000    0.0000
   -4.2321 0.000E+00 -999.0000   -9.5785    0.0000    0.0000    0.0000    0.0000
   -4.0564 0.000E+00 -999.0000   -9.1048    0.0000    0.0000    0.0000    0.0000
   -4.0000 0.000E+00 -999.0000   -8.0606    0.0000    0.0000    0.0000    0.0000
   -3.9918 0.000E+00 -999.0000   -6.6224    0.0000    0.0000    0.0000    0.0000
   -3.8251 0.000E+00 -999.0000   -6.6234    0.0000    0.0000    0.0000    0.0000



                   log molality of dissolved elements

   log xi   time, d   log days     Al        Cl        K         Si

 -999.0000 0.000E+00 -999.0000  -12.0000   -3.9950  -12.0000  -12.0000
   -8.0000 0.000E+00 -999.0000   -8.0000   -3.9950   -8.0000   -7.5229
   -7.0000 0.000E+00 -999.0000   -7.0000   -3.9950   -7.0000   -6.5229
   -6.0000 0.000E+00 -999.0000   -6.0000   -3.9950   -6.0000   -5.5229
   -5.1467 0.000E+00 -999.0000   -5.1470   -3.9950   -5.1467   -4.6696
   -5.0000 0.000E+00 -999.0000   -5.1744   -3.9950   -5.0000   -4.5229
```

```
  -4.2321 0.000E+00 -999.0000   -5.9195   -3.9950   -4.2321   -3.7552
  -4.0564 0.000E+00 -999.0000   -7.1598   -3.9950   -4.0564   -3.7552
  -4.0000 0.000E+00 -999.0000   -8.9125   -3.9950   -4.0000   -3.6990
  -3.9918 0.000E+00 -999.0000   -8.3632   -3.9950   -3.9918   -3.6908
  -3.8251 0.000E+00 -999.0000   -8.5305   -3.9950   -3.9908   -3.5244


               ppm (mg/kg) of dissolved elements

  log xi   time, d   log days     Al        Cl        K         Si

-999.0000 0.000E+00 -999.0000 .270E-07  3.59      .391E-07  .281E-07
  -8.0000 0.000E+00 -999.0000 .270E-03  3.59      .391E-03  .843E-03
  -7.0000 0.000E+00 -999.0000 .270E-02  3.59      .391E-02  .843E-02
  -6.0000 0.000E+00 -999.0000 .270E-01  3.59      .391E-01  .843E-01
  -5.1467 0.000E+00 -999.0000 .192      3.59      .279      .601
  -5.0000 0.000E+00 -999.0000 .181      3.59      .391      .843
  -4.2321 0.000E+00 -999.0000 .325E-01  3.59      2.29      4.93
  -4.0564 0.000E+00 -999.0000 .187E-02  3.59      3.43      4.93
  -4.0000 0.000E+00 -999.0000 .330E-04  3.59      3.91      5.62
  -3.9918 0.000E+00 -999.0000 .117E-03  3.59      3.98      5.72
  -3.8251 0.000E+00 -999.0000 .795E-04  3.59      3.99      8.40


                  solid solution product compositions



                  log of moles of product minerals



  log xi   time, d   log days  Diaspore



  -5.1467 0.000E+00 -999.0000   -8.2826
  -5.0000 0.000E+00 -999.0000   -5.4804



  log xi   time, d   log days  Diaspore  Kaolinit
                                         e
```

```
   -4.2321 0.000E+00 -999.0000   -4.2419   -7.2590



   log xi   time, d   log days  Kaolinit
                                e


   -4.0564 0.000E+00 -999.0000   -4.3578
   -4.0000 0.000E+00 -999.0000   -4.3010



   log xi   time, d   log days  Kaolinit  Muscovit
                                e         e


   -3.9918 0.000E+00 -999.0000   -4.2929   -8.9992



   log xi   time, d   log days  Kaolinit  Maximum_  Muscovit
                                e         Microcli  e
                                          ne

   -3.8251 0.000E+00 -999.0000   -5.4289   -0.0001   -4.3247



                  log of destroyed moles of reactants

   log xi   time, d   log days  Maximum_
                                Microcli
                                ne

 -999.0000 0.000E+00 -999.0000 -999.0000
   -8.0000 0.000E+00 -999.0000   -8.0000
   -7.0000 0.000E+00 -999.0000   -7.0000
   -6.0000 0.000E+00 -999.0000   -6.0000
   -5.1467 0.000E+00 -999.0000   -5.1467
   -5.0000 0.000E+00 -999.0000   -5.0000
   -4.2321 0.000E+00 -999.0000   -4.2321
   -4.0564 0.000E+00 -999.0000   -4.0564
   -4.0000 0.000E+00 -999.0000   -4.0000
   -3.9918 0.000E+00 -999.0000   -3.9918
```

```
  -3.8251 0.000E+00 -999.0000   -3.8251



     xi     log xi   time, d   log days   g des     g cre     g net     cc des    cc cre    cc net

 0.000E+00 -999.0000 0.000E+00 -999.0000 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00 0.000E+00
 1.000E-08   -8.0000 0.000E+00 -999.0000 2.783E-06 0.000E+00-2.783E-06 0.000E+00 0.000E+00 0.000E+00
 1.000E-07   -7.0000 0.000E+00 -999.0000 2.783E-05 0.000E+00-2.783E-05 0.000E+00 0.000E+00 0.000E+00
 1.000E-06   -6.0000 0.000E+00 -999.0000 2.783E-04 0.000E+00-2.783E-04 0.000E+00 0.000E+00 0.000E+00
 7.134E-06   -5.1467 0.000E+00 -999.0000 1.986E-03 3.130E-07-1.985E-03 0.000E+00 0.000E+00 0.000E+00
 1.000E-05   -5.0000 0.000E+00 -999.0000 2.783E-03 1.984E-04-2.585E-03 0.000E+00 0.000E+00 0.000E+00
 5.860E-05   -4.2321 0.000E+00 -999.0000 1.631E-02 3.451E-03-1.286E-02 0.000E+00 0.000E+00 0.000E+00
 8.782E-05   -4.0564 0.000E+00 -999.0000 2.444E-02 1.133E-02-1.312E-02 0.000E+00 0.000E+00 0.000E+00
 1.000E-04   -4.0000 0.000E+00 -999.0000 2.783E-02 1.291E-02-1.493E-02 0.000E+00 0.000E+00 0.000E+00
 1.019E-04   -3.9918 0.000E+00 -999.0000 2.836E-02 1.315E-02-1.521E-02 0.000E+00 0.000E+00 0.000E+00
 1.496E-04   -3.8251 0.000E+00 -999.0000 4.163E-02 2.783E+02 2.783E+02 0.000E+00 0.000E+00 0.000E+00



                  affinities of irreversible reactions

   log xi   time, d   log days  Maximum_
                                Microcli
                                ne

 -999.0000 0.000E+00 -999.0000   59.7814
   -8.0000 0.000E+00 -999.0000   30.5418
   -7.0000 0.000E+00 -999.0000   23.7126
   -6.0000 0.000E+00 -999.0000   16.8098
   -5.1467 0.000E+00 -999.0000   10.3504
   -5.0000 0.000E+00 -999.0000    9.5369
   -4.2321 0.000E+00 -999.0000    4.9878
   -4.0564 0.000E+00 -999.0000    4.1014
   -4.0000 0.000E+00 -999.0000    2.4466
   -3.9918 0.000E+00 -999.0000    0.4541
   -3.8251 0.000E+00 -999.0000    0.0000
```

3.8 POST-PROCESSING OF OUTPUT FILES

The user should be aware of efficient and reliable post-processing software able to extract and organize most of the data from EQ3NR and EQ6 output files. This qualified software, GETEQDATA V1.0.1 (STN: 10809-1.0.1-00), is available from Yucca Mountain's Software Configuration Management. If special needs arise for data extraction that are unavailable in the latest version of this software, a request can be made to the code's author for its possible inclusion. Contact: Russell Jarek at Sandia National Laboratories, rljarek@sandia.gov.

4. FILE FORMATS

All input and output file formats are ASCII text files (with exception of the "data1" file generated by EQPT in binary format), readable and editable by many simple text-editing programs e.g., Notepad, WordPad or MS Word. The database input is converted from text ("data0" filenames) to binary format by the program EQPT (described at end of Section 2.1) and it is the binary file ("data1" filenames) that is directly utilized by EQ3NR and EQ6.

The problem description input files for EQ3NR ("*.3i" files) and EQ6 ("*.6i" files) are free-formatted but input information must be specific and correctly ordered. These input files are typically generated by the modification of pre-existing files. Many examples of different test input files are available for EQ3NR and EQ6 from the installation CD-ROM subdirectories (Software Media Number 10813-PC-8.0-00) "3Tlib" and "6Tlib," respectively. Note that the character "*" indicates a comment line and is not read as an input.

5. ALLOWABLE/TOLERABLE RANGES FOR INPUTS AND OUTPUTS

The maximum aqueous solution concentration in a "Pitzer" run cannot exceed an ionic strength of 100 molal. The maximum temperature for calculations with the attached thermodynamic databases (and those currently to be qualified for the Yucca Mountain Project) cannot exceed 300°C; note this does not preclude use of this code at higher temperatures if a future database supports them. The user must assure on using the appropriate thermodynamic databases for calculations out of the Liquid-Vapor Pressure (LVP) saturation curve for pure H_2O (check input parameters described in Section 3.1.2). The user must also ensure that all "basis" species needed to define the composition of solid or aqueous phases (Section 2.1.1) in the problem of interest are detailed in the input file. Otherwise, solid (or aqueous) phases with missing "defined" components will not be considered in the calculation.

The input file limitations are summarized in Table 2.

Table 2. Input File Range Limitations

Description of the Input Range	EQ3NR or EQ6	Limiting Value	Section Reference
Lines in the main body of a title	both	200	3.3.1, 3.4.1
Generic ion exchanger phases	EQ3NR	10	3.3.11
Sites on an ion exchanger phase	EQ3NR	4	3.3.11
Species per site on ion exchanger phase	EQ3NR	12	3.3.11
Solid solution components	EQ3NR	50	3.3.13
Alter/Suppress options	EQ3NR	100	3.3.14
Activity coefficient option not valid: "HC & DH equations"	EQ3NR	≠ option 2	3.3.16
Reactants	EQ6	40	3.4.4
Mechanisms per TST rate law	EQ6	4	3.4.4
Species in a kinetic activity product	EQ6	4	3.4.4
Mineral sub-set selection suppression options	EQ6	40	3.4.11
Exceptions to the mineral suppression "allwith" options	EQ6	100	3.4.11
Fixed-fugacity options	EQ6	20	3.4.12

6. ANTICIPATED ERRORS AND USER RESPONSE

For error messages related to the input file see Section 3.2.2.

There are three different summary outputs indicative of an error at the end of an EQ3/6 run, these are:

```
1) The following input files were not run because
   they don't exist:

2) * Error - The data file "data1.xyz" doesn't
         exist in the EQ3/6 data file directory c:\eq3_6v8.0\db
         defined in the environment variable EQ36DA.

3) The following input files were run, but EQ3NR error
   messages were generated:
```

The first indicates that the input file was not found. This may be due to a misspelling, an incorrect directory being referenced, or that the file was not successfully copied from the distribution CD-ROM. The second error message indicates that the database file with the extension "xyz" was not located in the indicated directory. This may be a misspelling of the database file's extension or indicate that the desired database is located in a directory other than the default "Eq3_6v8.0\DB". Be sure the database has been converted from the *data0.xyz* into the binary *data1.xyz* with the EQPT program. The third error message indicates a problem with the input file and may be caused either by specifying an invalid option, or an inability for the code to converge with the input speciation and concentration. The user is directed to read the messages in the command prompt screen where further specific information will be provided.

Should a reproducible software error (i.e. bug) or technical error be discovered, initially notify the software code developer: Thomas J. Wolery, Lawrence Livermore National Labs, Livermore, California. The potential error will be checked for its validity, e.g. that it's not outside the scope of the programming code itself. If a valid error, then this should be reported by the user to the Software Configuration Management office via the appropriate implementing procedure (e.g. AP-SI.1Q) for corrective action.

7. HARDWARE AND SOFTWARE ENVIRONMENT

The hardware platform for EQ3/6 is any personal computer running a 32-bit Windows Operating System (e.g. Windows 95 and onward). Support for UNIX computers has been discontinued.

Hard disk drive space requirements will depend on the users typically output file size. EQ3NR output files (extension .3o) are quite small, typically <100 KB, whereas EQ6 output files (extension .6o) range from a typical 1 MB to tens of MB depending upon the time (reaction path) and printout intervals.

8. REQUIRED TRAINING

The use of the code does not require any formal training, however, it is strongly recommended that the user:

1. Have the ability to utilize the Microsoft Windows OS environment and be knowledgeable with basic DOS commands.

2. Have fundamental expertise on the essentials of geochemical modeling, particularly on the subject of aqueous speciation, solution-mineral-equilibria, and the kinetic description of heterogeneous mineral-fluid interactions. Unrealistic results may arise from poorly selected input parameters in defining a geochemical problem that could conduct to erroneous interpretations.

3. Have some understanding of mathematical methods as applied to code computations in solving solution-mineral-equilibria of multi-component systems and FORTRAN 90 (or other appropriate) computer language.

9. SAMPLE PROBLEMS

Many sample problems are supplied with the EQ3/6 Version 8.0 installation package as detailed in the *Installation Test Plan EQ3/6 Version 8.0* (ITP) (Jarek 2002) and in the *Validation Test Plan EQ3/6 Version 8.0* (VTP) (Jove-Colon 2002a). These are located on the installation CD-ROM (Software Media Number: 10813-8.0-00) in the folders "3TLIB" and "6TLIB" for EQ3NR and EQ6 sample inputs, respectively. They should also be on the user's local hard drive with the same directory names located under the EQ3/6 software installation directory. These sample problems are tutorial in nature and demonstrate various functionalities of the code. The users should modify these examples for their specific purposes. The installation validation should be performed following installation as instructed in the ITP (Jarek 2002); validation is beyond the scope of this user's manual. If the user desires further detailed examples to perform, see the VTP (Jove-Colon 2002a) and the results in the corresponding *Validation Test Report EQ3/6 Version 8.0* (Jove-Colon 2002b).

10. INSTALLATION PROCEDURES

The software will be officially installed and tested in compliance with the current procedure AP-SI.1Q in conjunction with the document *Installation Test Plan EQ3/6 Version 8.0* (Jarek 2002; SDN: 10813-ITP-8.0-00), which address code installation procedures in detail.

11. REFERENCES

11.1 CITED DOCUMENTS

Aagaard, P. and Helgeson, H.C. 1982. "Thermodynamic and kinetic constraints on reaction rates among minerals and aqueous solutions. I. Theoretical considerations." *American Journal of Science*, *282*, 237–285. New Haven, CT: Scholarly Publications. TIC: 225516.

Daveler, S.A. and Wolery, T.J. 1992. *EQPT, A Data File Preprocessor for the EQ3/6 Software Package: User's Guide and Related Documentation (Version 7.0).* UCRL-MA-110662 PT II. Livermore, California: Lawrence Livermore National Laboratory. TIC: 205240.

Delany, J.M., Puigdomenech, I. and Wolery, T.J. 1986. *Precipitation Kinetics Option for the EQ6 Geochemical Reaction Path Code.* UCRL-53642. Livermore, California: Lawrence Livermore National Laboratory. TIC: 203008.

Helgeson, H. C. 1968. "Evaluation of irreversible reactions in geochemical processes involving minerals and aqueous solutions– I. Thermodynamic relations." *Geochimica et Cosmochimica Acta, 32*, 853–877. New York, New York: Pergamon Press. TIC: 220011.

Helgeson, H. C. 1970. "A chemical and thermodynamic model of ore deposition in hydrothermal systems." *Mineralogical Society of America Special Paper 3*, 155–186. Washington, DC: Mineralogical Society of America. TIC: 248229.

Helgeson, H. C., Brown, T. H., Nigrini, A. and Jones, T. A. 1970. "Calculation of mass transfer in geochemical processes involving aqueous solutions." *Geochimica et Cosmochimica Acta, 34*, 569–592. New York, New York: Pergamon Press. TIC: 225570.

Helgeson, H. C., Garrels, R. M. and Mackenzie, F. T. 1969. "Evaluation of irreversible reactions in geochemical processes involving minerals and aqueous solutions– II. Applications."

Geochimica et Cosmochimica Acta, 33, 455–481. New York, New York: Pergamon Press. TIC: 253168.

INTERA Environmental Consultants, Inc. 1983. "Geochemical Models Suitable for Performance Assessment of Nuclear Waste Storage: Comparison of PHREEQE and EQ3/EQ6." ONWI-473. Columbus, Ohio: Battelle Memorial Institute, Office of Nuclear Waste Isolation. TIC: 223449.

Jarek, R.L. 2002. *Installation Test Plan EQ3/6 Version 8.0.* SDN: 10813-ITP-8.0-00. Las Vegas, Nevada: Bechtel SAIC Company, LLC.

Jove-Colon, C.F. 2002a. *Validation Test Plan EQ3/6 Version 8.0*. SDN: 10813-VTP-8.0-00. Las Vegas, Nevada: Bechtel SAIC Company, LLC.

Jove-Colon, C.F. 2002b. *Validation Test Report EQ3/6 Version 8.0*. SDN: 10813-VTR-8.0-00. Las Vegas, Nevada: Bechtel SAIC Company, LLC.

Langmuir, D. 1997. *Aqueous Environmental Chemistry*. Upper Saddle River, New Jersey: Prentice-Hall. TIC: 237107.

Lasaga, A.C. 1981. "Transition state theory." *Kinetics of Geochemical Processes.* Lasaga, A.C. and Kirkpatrick, R.J., eds. Reviews in Mineralogy Volume 8, 135–169. Washington, D.C.: Mineralogical Society of America. TIC: 239318.

Nordstrom, D.K., Jenne, E.A. and Ball, J.W. 1979. "Redox equilibria of iron in acid mine waters." *Chemical Modeling in Aqueous Systems, Speciation, Sorption, Solubility, and Kinetics*. Jenne, E.A., ed. American Chemical Society Symposium Series 93, 51–79. Washington, D. C.: American Chemical Society. TIC: 235633.

Nordstrom, D.K. and Munoz, J.L. 1985. *Geochemical Thermodynamics*. Menlo Park, California: The Benjamin/Cummings Publishing Co., Inc. TIC: 208228.

Shock, E.L. and Helgeson, H.C., 1990. "Calculation of the thermodynamic and transport properties of aqueous species at high pressures and temperatures: Standard partial molal properties of organic species." *Geochimica et Cosmochimica Acta, 54*, 915–945. New York, New York: Pergamon Press. TIC: 248485

Sposito, G. 1989. *The Chemistry of Soils*. New York, New York: Oxford University Press. TIC: 237211.

Stumm, W. and Morgan, J.J. 1996, *Aquatic Chemistry: Chemical Equilibria and Rates in Natural Waters*, 3rd edition. New York, New York: John Wiley & Sons, Inc. TIC: 246296.

Viani, B.E. and Bruton, C.J. 1992. "Modeling ion exchange in Clinoptilolite Using the EQ3/6 Geochemical Modeling Code" *Water-Rock Interaction, Volume 1, Low Temperature Environments.* Kharaka, Y.K. and Maest, A.S., ed., 73–77. Rotterdam: A.A. Balkema. TIC: 230491.

Wolery, T.J. 1992a. *EQ3/6, A Software Package for Geochemical Modeling of Aqueous Systems: Package Overview and Installation Guide (Version 7.0).* UCRL-MA-110662 PT I. Livermore, California: Lawrence Livermore National Laboratory. TIC: 205087.

Wolery, T.J. 1992b. *EQ3NR, A Computer Program for Geochemical Aqueous Speciation-Solubility Calculations: Theoretical Manual, User's Guide, and Related Documentation*, (*Version 7.0).* UCRL-MA-110662 PT III. Livermore, California: Lawrence Livermore National Laboratory. TIC: 205154.

Wolery, T.J. and Daveler, S.A. 1992. *EQ6, A Computer Program for Reaction Path Modeling of Aqueous Geochemical Systems: Theoretical Manual, User's Guide, and Related Documentation (Version 7.0).* UCRL-MA-110662 PT IV. Livermore, California: Lawrence Livermore National Laboratory. TIC: 205002.

11.2 CODES, STANDARDS, REGULATIONS, AND PROCEDURES

AP-SI.1Q, Rev. 3, ICN 4. *Software Management.* Washington, D.C.: U.S. Department of Energy, Office of Civilian Radioactive Waste Management. ACC: MOL.20020520.0283.

11.3 SOURCE DATA, LISTED BY DATA TRACKING NUMBER

MO0210SPATHDYN.000. Thermodynamic Data Input Files - DATA0.YMP.R2. Submittal date: 10/02/02.

SN0210T0510102.001. Pitzer Thermodynamic Database (DATA0.YPF, Revision 0). Submittal date: 10/22/02.

11.4 SOFTWARE CODES

GETEQDATA V1.0.1. STN: 10809-1.0.1-00.

APPENDIX A

GLOSSARY OF SYMBOLS

INTENTIONALLY LEFT BLANK

a_i	Thermodynamic activity of the i^{th} aqueous solute species.
a_w	Thermodynamic activity of water.
$a_{\sigma\psi}$	Thermodynamic activity of the σ^{th} component of the ψ^{th} solid solution phase.
A_j	Thermodynamic affinity of the j^{th} reaction. The forward direction is implied, which in EQ3/6 is taken to be that in which the associated species is destroyed, for example by dissociation or dissolution.
$A_{+,j}$	Thermodynamic affinity of the j^{th} reaction (forward direction: dissociation, dissolution).
$A_{-,j}$	Thermodynamic affinity of the j^{th} reaction (reverse direction: formation, precipitation).
b_{sr}	Stoichiometric reaction coefficient, the number of moles of the s^{th} aqueous species appearing in the r^{th} aqueous reaction; it is negative for reactants and positive for products.
$b_{s\phi}$	Stoichiometric reaction coefficient, the number of moles of the s^{th} aqueous species appearing in the reaction for the dissolution of the ϕ^{th} pure mineral; it is negative for reactants and positive for products.
b_{sg}	Stoichiometric reaction coefficient, the number of moles of the s^{th} aqueous species appearing in the reaction for the dissolution of the g^{th} gas species; it is negative for reactants and positive for products.
$b_{T,r}$	The quantity $b_{s''r} + \sum_{\substack{s'=1 \\ s' \neq w, s_B}}^{s_Q} b_{s'r}$.
$b_{T,\Phi}$	The quantity $\sum_{\substack{s'=1 \\ s' \neq w, s_B}}^{s_Q} b_{s'\Phi}$.
$b_{T,\sigma\psi}$	The quantity $\sum_{\substack{s'=1 \\ s' \neq w, s_B}}^{s_Q} b_{s'\sigma\psi}$.
$d\xi/dt$	The overall reaction rate; also symbolized by v.
$d\xi_j/d\xi$	Relative rate of the j^{th} irreversible reaction; also symbolized by v_j^{rel} .
$d\xi_j/dt$	Actual rate of the j^{th} irreversible reaction; also symbolized by v_j.
$dt/d\xi$	The inverse rate; also symbolized by $v_{1/t}$.
$\underline{d}$	An array of derivatives of increasing order.

D_{ij}	An element of the matrix $\underline{D}$.
$\underline{D}$	A matrix used to convert an array of finite differences of increasing order to an equivalent array of derivatives.
ε	Subscript indexing a chemical element.
ε_T	Total number of chemical elements in a chemical system.
f_g	Fugacity of the g^{th} gas.
f_j	Ratio of effective to total surface area of the i^{th} irreversibly reacting mineral.
f_{O_2}	Oxygen fugacity.
$\underline{f}$	An array of finite differences of increasing order.
g	Subscript denoting a gas species.
$\tilde{H}_{s'r}$	The factor $\frac{n_s u_{s's''}}{b_{s''r}}$ or $\frac{n_{s''} c_{\varepsilon s''}}{b_{s''r}}$; the $s''^{\text{ th}}$ aqueous species is associated with the r^{th} aqueous reaction, and the ε^{th} chemical element is associated with the $s'^{\text{ th}}$ basis species.
$\tilde{H}_{zr}$	The factor $\frac{n_{s''} z_{s''}}{b_{s''r}}$.
$i_{T,+,j}$	The number of terms appearing in the transition state theory or activity product term net forward rate law for the j^{th} irreversible reaction.
$i_{T,-,j}$	The number of terms appearing in the transition state theory or activity product term net reverse rate law for the j^{th} irreversible reaction.
I	Ionic strength.
IAP	Ion activity product; see Q.
J_{ij}	An element of the Jacobian matrix ($\partial a_i / \partial z_j$).
$\underline{J}$	The Jacobian matrix.
$k_{+,ij}$	Rate constant for the i^{th} term in the net forward rate law for the j^{th} irreversible reaction.
$k_{-,ij}$	Rate constant for the i^{th} term in the net reverse rate law for the j^{th} irreversible reaction.
$k_{T,i}$	The i^{th} coefficient in the polynomial for describing temperature as a function of reaction progress.
K	Thermodynamic equilibrium constant.
m_i	Molal concentration of the i^{th} aqueous solute species.

m_s	Molal concentration of the s^{th} aqueous species.
n_i	Number of moles of the i^{th} aqueous solute species.
n_s	Number of moles of the s^{th} aqueous species.
n_w	Number of moles of water.
$n_{T,+,ij}$	The number of species whose thermodynamic activities appear in the i^{th} term of the net forward rate law for the j^{th} irreversible reaction.
$n_{T,-,ij}$	The number of species whose thermodynamic activities appear in the i^{th} term of the net reverse rate law for the j^{th} irreversible reaction.
n_ϕ	Number of moles of the ϕ^{th} pure mineral.
$n_{\sigma\psi}$	Number of moles of the σ^{th} end member of the ψ^{th} solid solution.
$n_{T,\varepsilon}$	Total number of moles of the ε^{th} chemical element.
$n_{T,s'}$	Total number of moles of the $s'^{\ th}$ (basis) aqueous solute species.
$N_{+,nij}$	Exponent of the activity of the n^{th} species appearing in the i^{th} term of the net forward kinetic rate law for the j^{th} irreversible reaction.
$N_{-,nij}$	Exponent of the activity of the n^{th} species appearing in the i^{th} term of the net reverse kinetic rate law for the j^{th} irreversible reaction.
$O_{2(g)}$	Oxygen gas; in aqueous solution, this refers to a fictive species; also symbolized as O_2.
P	Pressure, bars.
pH	The quantity $-\log\ a_{H+}$
$q_{+,ij}$	Kinetic activity product for the i^{th} term in the net forward rate law for the j^{th} irreversible reaction.
$q_{-,ij}$	Kinetic activity product for the i^{th} term in the net reverse rate law for the j^{th} irreversible reaction.
Q	Activity product of a reaction; *IAP* is used by many others (e.g., Parkhurst et al., 1980) to denote the same quantity.
r	Subscript denoting an aqueous reaction.
r_T	Total number of reactions for the dissociation/destruction of dependent aqueous species.
R	The gas constant, 1.98726 cal/mol-°K.
s	Subscript denoting an aqueous species ($s = w$ implies $H_2O_{(l)}$).
s'	Subscript denoting s in the range from 2 to s_Q, excluding s_B.

s"	Subscript implying the species formally associated with the aqueous reaction designated by r (s" = $r + s_B$).
s_j	Surface area of the mineral destroyed/formed by the j^{th} irreversible reaction.
s_B	Subscript denoting the fictive redox species O_2.
s_Q	The total number of aqueous basis species; depending on the problem at hand, s_Q is equal to or greater than s_B.
s_T	Total number of aqueous species.
SI	Saturation index for a mineral; SI = log (Q/K), where Q and K are the activity product and equilibrium constant, respectively, for the dissolution reaction.
t	Time.
T	Temperature, °K.
T_0	Temperature at $\xi = 0$ (or t =0 if in kinetic mode).
u	Stoichiometric mass balance coefficient calculated from reaction coefficients and certain model constraints; $u_{s's}$is the stoichiometric factor for computing the contribution of the s^{th} aqueous species to the mass balance for the $s'^{\ th}$ basis species.
v	Short for the overall reaction rate, $d\xi/dt$.
v_j^{rel}	Short for the relative rate of the j^{th} irreversible reaction, $d\xi_j/d\xi$.
v_j	Short for the actual rate of the j^{th} irreversible reaction, $d\xi_j/dt$.
$v_{1/t}$	Short for the inverse rate of the overall reaction, $dt/d\xi$.
w	Subscript denoting water (e.g., a_w, the activity of water).
$\underline{w}$	Vector of increasing partial sums of preceding reaction progress steps.
$\underline{W}$	Array of partial derivatives of $\log x_w$ with respect to $\log m_{s'}$, where s' is a basis species. This derivative is zero for s' = w or s_B.
$\underline{\tilde{W}}$	Array of partial derivatives of $\log x_w$ with respect to $\log n_{s'}$, where is a basis species. This derivative is zero for s' = s_B.
x_i	Mole fraction of the i^{th} aqueous solute species.
x_w	Mole fraction of water in aqueous solution.
$x_{\sigma\psi}$	Mole fraction of the σ^{th} end member of the ψ^{th} solid solution.
x	A general algebraic variable.
$\underline{z}$	Vector of algebraic master variables.

2.303	Symbol for and approximation to: ln 10.
$\underline{\alpha}$	Newton-Raphson residual function vector.
α_s	Residual function for mass balance of the s^{th} basis species.
α_ϕ	Residual function for equilibrium with a pure mineral.
$\alpha_{\sigma\psi}$	Residual function for equilibrium with the σ^{th} end member of the ψ^{th} solid solution.
$\underline{\beta}$	Newton-Raphson residual function vector, identical to $\underline{\alpha}$, except that mass balance residual elements are normalized by the corresponding values of total numbers of moles.
β_{max}	The largest absolute value of any element of $\underline{\beta}$.
$\underline{\delta}$	Newton-Raphson correction term vector.
δ_{max}	The largest absolute value of any element of $\underline{\delta}$.
δ_{conv}	Convergence function.
δ'	Under-relaxation parameter.
$\Delta\xi$	Increment of reaction progress.
$\Delta\xi_j$	Increment of reaction progress for an individual reaction.
Δt	Increment of time.
γ_i	Molal activity coefficient of the i^{th} aqueous solute species.
ξ	Overall reaction progress variable.
ξ_j	Reaction progress variable for the j^{th} irreversible reaction.
κ	Under-relaxation parameter in Newton-Raphson iteration.
$\lambda_{\sigma\psi}$	Rational (mole fraction) activity coefficient of the σ^{th} end member of the ψ^{th} solid solution.
σ, σ'	Symbols denoting end member components of a solid solution.
$\sigma_{+,ij}$	Stoichiometric adjustment factor for the i^{th} mechanism in a transition state theory net forward rate law for the j^{th} irreversible reaction.
$\sigma_{-,ij}$	Stoichiometric adjustment factor for the i^{th} mechanism in a transition state theory net reverse rate law for the j^{th} irreversible reaction.
$\xi_{T,\psi}$	Total number of end members in the ψ^{th} solid solution.
ϕ	(a) Subscript denoting a pure mineral; (b) the osmotic coefficient of the aqueous solution.

ϕ_E	Total number of minerals of fixed composition in equilibrium with the aqueous phase.
ψ	Subscript denoting a solid solution.
ψ_E	Total number of solid solutions in equilibrium with the aqueous phase.
Ω	Water constant; 1000 divided by the molecular weight of water; about 55.51.
$+$	Subscript denoting a reaction proceeding in the forward sense; the convention in this report equates this with dissociation, dissolution, or destruction of the associated species.
$-$	Subscript denoting a reaction proceeding in the backward sense; the convention in this report equates this with association, precipitation, or formation of the associated species.

APPENDIX B

EQ3NR SUPPLUMENTARY INFORMATION

INTENTIONALLY LEFT BLANK

CONTENTS

CONTENTS (Continued)

Page

FIGURES

Page

INTENTIONALLY LEFT BLANK

B.1 SPECIATION-SOLUBILITY MODELING OF AQUEOUS SYSTEMS

B.1.1 INTRODUCTION

EQ3NR is a speciation-solubility code for aqueous systems. As such, given sufficient data on a specific aqueous system, it computes a model of the solution that consists of two principal parts: the distribution of species in the solution and a set of saturation indices (SI = log Q/K) for various reactions of interest. The saturation indices are measures of the degree of disequilibrium of the corresponding reactions. They provide a means of searching for solubility controls on natural waters. For example, if a series of related fluids all have calcite SI values close to zero, it is probable that this mineral is present and partial equilibrium with it is maintained as the solutions evolve in composition.

EQ3NR is not a computerized geochemical model, but a code which is capable of evaluating geochemical models which are defined by the contents of a supporting data file (of which there are now five to choose from) and by other assumptions which the user sets on the EQ3NR input file. The supporting data files differ not only in terms of data values, but more importantly in terms of the identities of the components and chemical species represented and in terms of the general approaches to dealing with the problem of activity coefficients. Because of various limitations, some problems may require the use of only certain data files, while others can be treated using any of the available data files. The user must choose the best data file (or files) with which to run a particular problem. The user must also understand both the particular problem and the code capabilities and limitations well enough to construct an adequate input file.

Although speciation-solubility models are commonly used as a means of testing whether or not heterogeneous reactions are in a state of thermodynamic equilibrium, they often just assume that all reactions occurring in aqueous solution are in such a state. Such reactions most likely to be in disequilibrium are redox reactions or reactions for the formation or dissociation of large complexes that are more like small polymers, such as $(UO_2)_3(OH)_7^-$. Speciation-solubility models are better used when they are employed to test the degree of disequilibrium of these kinds of reactions than when they are forced to assume that such reactions are in equilibrium.

A speciation-solubility model cannot by itself predict how aqueous solution composition will change in response to rock/water interactions. Nevertheless, this type of modeling can be a powerful tool for elucidating such interactions when it is applied to a family of related waters. Such a family might be a set of spring waters issuing from the same geologic formation, a sequence of ground water samples taken from along an underground flow path, or a sequence of water samples taken in the course of a rock/water interactions experiment in the laboratory. Jenne (1981) reviews several studies of this kind. Particularly interesting are Nordstrom and Jenne’s (1977) study of fluorite solubility equilibria in geothermal waters and Nordstrom, Jenne, and Ball’s (1979) study of controls on the concentration of iron in acid mine waters.

EQ3NR offers many options for the input file description of the composition of a given water. Consequently, the code can be used in a variety of ways. Many of the descriptive parameters of interest can be either model inputs or outputs. For example, the pH of a buffer solution can be calculated from the buffer recipe by adjusting the hydrogen ion concentration to satisfy charge balance. Alternatively, adjusting the concentration of a buffer component to satisfy the charge balance is a means of computing the complete recipe for a buffer having a desired *pH*. Some of the possible model inputs are assumptions, as of equilibrium with specified minerals. The use of some types of model inputs also pose special problems, some of which occur in particular contexts. The worst of these pertain to *Eh*, alkalinity, and *pH* and will be discussed in some detail later in this section.

B.1.2UNITS OF CONCENTRATION

EQ3NR uses the molal scale as the principal unit of concentration for aqueous species. The molal concentration (molality) of a substance dissolved in water is defined as:

$$m_i = \frac{n_i}{w_w} \tag{B–1}$$

where n_i is the number of moles of the i^{th} solute species and w_w is the number of kilograms of solvent water. Other common measures of aqueous solute concentration are the molarity (moles of substance per liter of aqueous solution), the part-per-million or ppm by volume (mg/L, milligrams of substance per liter of solution), and the ppm by weight (mg/kg, milligrams of substance per kilogram of solution). The EQ3NR code accepts concentration parameters in any of these units (except for solids, see this Appendix section B.2), but converts non-molal concentrations to molalities before computing the aqueous speciation model. Whether or not it does this correctly depends on circumstance and data provided by the user.

The conversion equations in all three cases require a value for the total dissolved salts in mg/kg solution ($C_{T\$, mg/kg}$). The density of the aqueous solution in g/ml ($\rho_{g/ml}$) is also required to convert molarities and mg/L concentrations to molalities. The total dissolved salts in mg/kg may be calculated from the total dissolved salts in mg/L ($C_{T\$, mg/L}$) and the density according to:

$$C_{T\$,mg/kg} = \frac{C_{T\$,mg/L}}{\rho_{g/ml}} \tag{B–2}$$

EQ3NR expects values of $C_{T\$, mg/kg}$ and $\rho_{g/ml}$ on the input file if such conversions are necessary. In place of $C_{T\$, mg/kg}$, one may enter $C_{T\$, mg/L}$ and $\rho_{g/ml}$, and $C_{T\$, mg/kg}$ is calculated from the above equation. If such values are not provided, $C_{T\$, mg/kg}$ is assigned a default value of zero and $\rho_{g/ml}$ is assigned a default value of unity. These values are generally adequate for dilute solutions at temperatures near 25°C. In the case of brines, these values are not adequate, and the user must provide actual values as part of the input in order to obtain accurate conversion. *The code provides no checks or warnings if these are not provided.*

The weight fraction of solvent water is given by:

$$N_w = \frac{1{,}000{,}000 - C_{T\$,mg/kg}}{1{,}000{,}000} \tag{B–3}$$

Letting $C_{i,\,,molar}$ be the molar concentration of the i^{th} solute species, the molality is given by:

$$m_i = \frac{C_{i,molar}}{\rho_{g/ml} N_w} \tag{B–4}$$

Letting $C_{i,\,mg/L}$ be the concentration in mg/L, the conversion is:

$$m_i = \frac{0.001 C_{i,mg/L}}{\rho_{g/ml} M_w N_w} \tag{B–5}$$

where M_w is the molecular weight of the solvent, water ($M_w \approx 18.015$ g/mole). Letting $C_{i,\,mg/kg}$ be the concentration in mg/kg solution, the conversion is:

$$m_i = \frac{0.001 C_{i,mg.L}}{M_w N_w} \tag{B–6}$$

Some dissolved gas analyses are reported in units of ml (STP)/ml solution, where STP refers to standard temperature and pressure (0°C and 1 atm). The conversion equation is:

$$m_i = \frac{1000 C_{i,ml(STP)/ml}}{V_g M_w N_w} \tag{B–7}$$

where V_g is the molar volume of an ideal gas at STP ($V_g = 22{,}413.6$ ml/mole).

The concentration of solvent water is defined as its mole fraction:

$$x_w = \frac{n_w}{n_w + \sum_i n_i} \tag{B–8}$$

where n_w is the number of moles of water. The molality of the i^{th} solute species can also be written as:

$$m_i = \frac{\Omega n_i}{n_w} \tag{B–9}$$

where Ω is the number of moles of water comprising a mass of 1 kg ($\Omega \approx 55.51$; $\Omega w_w = n_w$). Substituting this relation into the one above it gives:

$$x_w = \frac{\Omega}{\Omega + \sum_i m_i} \tag{B–10}$$

EQ3NR uses this relation to calculate the mole fraction of water. This is done in a self-consistent manner in the iteration process. Thus, the user is not required to input a value.

A similar self-consistent treatment could be implemented to handle both $C_{T\$,\ mg/kg}$ and the solution density. However, no such treatment exists in the current version of EQ3NR, nor any other such modeling code known to the present writers. Implementation of a self-consistent treatment of the solution density would require the addition of models for partial molar volumes to the code and incorporation of the corresponding equations in the iteration process. The theoretical and practical aspects of partial molar volumes in solutions extending to high concentration have been addressed for example by Millero (1977), Helgeson, Kirkham and Flowers (1982), Kumar (1986), Connaughton, Millero, and Pitzer (1989), and Monnin (1989).

B.1.3 INPUT CONSTRAINTS, GOVERNING EQUATIONS, AND OUTPUTS

B.1.3.1 Overview

Aqueous speciation models can be constructed to satisfy a wide variety of combinations of possible input constraints and governing equations. The input constraints may include total (analytical) concentrations, an electrical balance requirement, free concentrations, activities, *pH*, *Eh*, *pe*, oxygen fugacity, phase equilibrium requirements, homogeneous equilibria, and run-specific values for equilibrium constants. The governing equations are the corresponding mathematical expressions, such as the mass balance equation and the charge balance equation.

The choice of governing equations in large part depends on which parameters are to be inputs to the model and which are to be outputs. This, in turn, is a function of what data on a given water are available, what form they are in, and what assumptions the modeler would like to use.

Chemical analysis provides mainly a set of values for the so-called total concentrations of dissolved components. The analytical value for an ion such as calcium is an example. It does not discriminate between the various calcium species in solution, but rather estimates the dissolved calcium contributed by all of them. This leads to a mass balance equation of the form:

$$m_{T,\ Ca^{2+}} = m_{Ca^{2+}} + m_{CaOH_{(aq)}} + m_{CaCO_{3(aq)}} + m_{CaHCO_3^+} + \dots \qquad \text{(B–11)}$$

where $m_{T,Ca^{2+}}$ is the total or analytical concentration (on the molal scale) and m_i is the molality of any individual chemical species contributing to the mass balance. The summations must be weighted by the appropriate stoichiometric equivalences; e.g., in the case of F^-, one has:

$$m_{T,\ F^-} = m_{F^-} + m_{HF_{(aq)}} + 2m_{H_2F_{2(aq)}} + 2m_{HF_2^-} + 3m_{AlF_{3(aq)}} + \dots \qquad \text{(B–12)}$$

The total concentration is the most common type of input parameter to an aqueous speciation model. The mass balance constraint, which corresponds to it, is therefore the most common governing equation. As we shall see, there are situations in which a total concentration is replaced by another type of input. In these cases, the mass balance constraint is replaced by a different governing equation, and the total concentration becomes something to be calculated (an output parameter).

From a purely mathematical point of view, there is no reason to discriminate among ion pairs (and ion-triplets, etc.) and complexes. For some investigators, the term "ion pair" implies a species in which an anion is separated from a cation by an unbroken hydration sheath about the latter, whereas the term "complex" implies direct contact and perhaps some degree of covalent bonding. Other investigators use these terms interchangeably. It is a general assumption in cases of geochemical interest that the concentrations of ion-pairs and complexes are governed by thermodynamic equilibrium.

Each case of this equilibrium can be represented by a mass-action equation for the dissociation of the ion-pair or complex. An example will illustrate this. The calcium sulfate ion-pair dissociates according to the reaction:

$$CaSO_{4(aq)} = Ca^{2+} + SO_4^{2-} \qquad \text{(B–13)}$$

where "=" is used as the sign for a reversible chemical reaction. The corresponding mass action equation is:

$$K_{CaSO_{4(aq)}} = \frac{a_{Ca^{2+}} a_{SO_4^{2-}}}{a_{CaSO_{4(aq)}}} \qquad \text{(B–14)}$$

where K is the equilibrium constant and a_i represents the thermodynamic activity of each species. This may also be written in logarithmic form:

$$\log K_{CaSO_{4(aq)}} = \log a_{Ca^{2+}} + \log a_{SO_4^{2-}} - \log a_{CaSO_{4(aq)}} \qquad \text{(B–15)}$$

The thermodynamic activity is related to the molal concentration by the relation:

$$a_i = m_i \gamma_i \qquad \text{(B–16)}$$

where γ_i is the activity coefficient, a function of the composition of the aqueous solution. As the solution approaches infinite dilution, the value of γ_i for each species approaches unity. The set of equations for computing the activity coefficients of aqueous species is chosen by the user on the EQ3/6 input file (by means of the **iopg(1)** option switch, Section 3.3.16 Iopg (Activity Coefficient) Options). The requisite supporting data are on the EQ3/6 data file. The various formulations presently treated by EQ3/6 are discussed in Section B.2.

The following subsections discuss the formulation of aqueous speciation problems in general terms. The rigorous mathematical development is presented in Section B.4.

B.1.3.2 Reference Formulation of the Aqueous Speciation Problem

In general terms, setting up an aqueous speciation model involves choosing n unknowns and n governing equations. The EQ3NR code offers a very wide range of options in this regard. In order to make sense of the different ways of setting up a model, we define a reference formulation for the aqueous speciation problem. This reference formulation serves as a springboard for discussing what goes into speciation models, what comes out, and what the options are. It is also used to compare how the aqueous speciation problem is formulated in EQ3NR (and other speciation-solubility codes in general) with how it is formulated in a reaction-path code like EQ6.

In the reference formulation, we assume that the activity coefficients are known parameters (the numerical treatment of these is discussed in Section B.4). Note that the molal concentration of the solvent is fixed as the number of moles of water in a one-kilogram mass of the pure substance.

We assume that there are ε_T chemical elements in the model. In order to further simplify the reference formulation, we assume that each element is present in only one oxidation state. Suppose that chemical analysis has given us ε_T - 2 total concentration values, each for a chemical species, each of which corresponds to a chemical element other than oxygen and hydrogen (e.g., Na^+ for Na, SO_4^{2-} for S). That gives $\varepsilon_T - 2$ mass balance equations as governing equations.

The charge balance equation plays the role that might have been played by a mass balance equation for hydrogen. The charge balance equation may be written in the general form:

$$\sum_{s=1}^{s_T} z_s m_s = 0 \tag{B–17}$$

where the summation is over all aqueous species, z_s is the electrical charge of a species, and m_s is its molal concentration. The hydrogen mass balance equation cannot be used as a governing equation to calculate the pH from the corresponding analytical data. This is due to the impracticability if not impossibility of ever measuring the total concentration of hydrogen with sufficient accuracy when the solvent contributes nearly all of it. As a practical matter, even the charge balance equation can be used for this purpose only in limited circumstances.

One may associate the solvent, water, with a mass balance for oxygen. However, the mass of water in a speciation-solubility calculation is fixed at 1 kg, and the concentration of water is entirely determined by the concentrations of the other components in the solution. Therefore, no such mass balance is required.

To sum up, the reference formulation consists of $\varepsilon_T - 2$ mass balance equations/total concentrations (one pair for every element except oxygen and hydrogen) and the charge balance equation (to calculate pH). Each element is present in only one oxidation state. Activity coefficients are treated as known parameters.

Before proceeding, we contrast this framework (common to speciation-solubility codes in general) with that employed in the EQ6 code. In the corresponding problem in that code, we would be given ε_T masses, in moles, and the same number of mass balance equations, this time written in terms of masses instead of concentrations. There we have a mass balance equation for oxygen, and we must calculate the mass of the solvent, water. In the case where each element appears in only one oxidation state, as we have temporarily assumed here, the charge balance equation is a linear combination of the mass balance equations, and the governing equation associated with H^+ can be either a hydrogen mass balance equation or the charge balance equation. The speciation-solubility problem has one fewer unknown, hence one less governing equation, than the corresponding EQ6 problem.

In either the EQ3NR or EQ6 type formulation of the problem, we may formally associate one aqueous species with each balance equation; e.g., Na^+ with sodium balance, Al^{3+} with aluminum balance, and H^+ with charge balance. Suppose our model must consider *n* balance equations and *k* aqueous complexes (using the term to include ion-pairs). That gives *k* mass action relationships that are also governing equations. We now have $n + k$ equations in $n + k$ unknowns (the masses/concentrations/activities of the $n + k$ aqueous species).

The number of aqueous complexes is usually much greater than the number of balance equations. This is especially true when the number of balance equations becomes very large. A useful approach is to reduce the number of equations and unknowns by substituting the aqueous mass action equations into the balance equations (see Section B.4.8). This leaves us with *n* equations (modified balance equations) in *n* unknowns (the concentrations or activities of the aqueous species that were chosen to formally correspond to the balance relationships).

This approach leads us to the concept of dealing with a set of master aqueous species. These may also be termed basis species. However, the concept does not arise purely from an attempt to reduce the number of iteration variables. The *k* aqueous complexes give us *k* linearly independent dissociation reactions and *k* linearly independent logarithmic mass action equations. An efficient way to write these reactions and equations is in terms of the associated complex (the species that dissociates) and such a set of master aqueous species. The dissociation reactions are then written as overall dissociation reactions but never as stepwise reactions; e.g., one has:

$$\mathrm{HgCl_3^-} = \mathrm{Hg^{2+}} + 3\mathrm{Cl^-} \qquad \text{(B–18)}$$

not:

$$\mathrm{HgCl_3^-} = \mathrm{HgCl_2^o} + \mathrm{Cl^-} \qquad \text{(B–19)}$$

We will also use this format to write dissolution reactions for minerals and gases and their associated heterogeneous mass action equations.

B.1.3.3 Alternative Constraints

The reference formulation of the aqueous speciation problem consists of:

(1) ε_T - 2 mass balance equations/total concentrations.

(2) the charge balance equation (to calculate *pH*).

We now discuss alternative constraints to the balance equations in the reference formulation. We discuss how to put oxidation-reduction problems into the formulation in the following subsection.

The alternative constraints are:

- Specifying log activity for a species (recall $pH = -\log a_{H^+}$).
- Log activity combination functions (e.g., *pHCl*; Section B.1.3.4).
- Log mean activity of an ion and one of opposite charge (Section B.1.3.4).
- Applying the charge balance constraint to a master species other than H^+.
- Phase equilibrium with a pure mineral.
- Phase equilibrium with an end member of a solid solution (the composition of the solid solution must be specified).
- Phase equilibrium with a gas (the fugacity of the gas must be specified).
- Equilibrium with other aqueous species, without falling under a mass balance constraint.
- Specifying the individual concentration of an aqueous basis species.

When a mass balance constraint is replaced by one of the above, we continue to reduce the number of unknowns to a master set as discussed above. The corresponding total concentrations become parameters to be calculated. We can calculate, for example, the total mass/concentration of hydrogen. This can be done with sufficient relative accuracy to permit the EQ6 code to use it as a constraint to solve for *pH*.

The log activity constraint. The first substitution that we discuss is most often applied to the hydrogen ion. In the course of chemical analysis, the *pH* of an aqueous solution is usually determined by means of a specific-ion electrode. This gives us the activity of the hydrogen ion from the relation:

$$pH = -\log a_{H^+} \tag{B–20}$$

The activities of many other species, including Na^+, Ca^{2+}, S^{2-}, F^-, and Cl^-, to name but a few, may also be measured by specific-ion electrodes.

EQ3NR will accept as an input the logarithm of the activity of a species. Note that this means that the code expects to see –*pH*, not *pH*, on the input file when this option is invoked. The new governing equation is just:

$$m_i = \frac{a_i}{\gamma_i} \tag{B–21}$$

The charge balance constraint. This can be applied to one of the major ions if a charge-balanced speciation model is desired. If EQ3NR does not use the charge balance equation as a constraint, it will calculate the charge imbalance. Otherwise, it will notify the user of the change in total concentration or *pH* that was required to generate a charge-balanced model.

We recommend routinely calculating *pH* from electrical balance only in cases of synthetic solutions for which the ionic totals are exact with respect to charge balance. Such solutions are most likely to be *pH* buffer solutions. In other circumstances, this practice is potentially dangerous because the result is affected by the error in every analytical value that is put in the model and also by every analytical value that should have been put in the model but was not. In general, apart from the case of *pH* buffer solutions, it is only safe to calculate *pH* this way if the *pH* is low (high concentrations of H^+) or high (high concentrations of OH^-).

Equilibrium constraint involving a non-aqueous species. A mass balance constraint may also be replaced by an equilibrium constraint involving a specified pure mineral, solid solution component species, or gas species. Suppose we wanted to know what concentration of dissolved calcium would be required for a water to be in equilibrium with calcite (the stable polymorph of $CaCO_{3(c)}$ at 25°C). The dissolution reaction may be written as:

$$\mathrm{Calcite} + \mathrm{H}^+ = \mathrm{Ca}^{2+} + \mathrm{HCO}_3^- \tag{B–22}$$

and the corresponding governing equation is then:

$$K_{\mathrm{Calcite}} = \frac{a_{Ca^{2+}} a_{HCO_3^-}}{a_{\mathrm{Calcite}} a_{H^+}} \tag{B–23}$$

Because calcite is a pure phase, its activity is fixed at unity.

If the required equilibrium involves an end-member component of a solid solution, the governing equation is slightly modified. Suppose we choose equilibrium with a calcite end-member of a high-magnesium calcite ($(Ca,Mg)CO_{3(c)}$. The governing equation has the same form as above, but the activity of the calcite end-member is no longer unity. Instead, one has:

$$a_{\mathrm{Calcite}} = \lambda_{\mathrm{Calcite}} x_{\mathrm{Calcite}} \tag{B–24}$$

where $\lambda_{Calcite}$ is the activity coefficient and $x_{Calcite}$ is the mole fraction of the calcite component. The mole fraction of the σ^{th} component of the ψ^{th} solid solution is given by:

$$x_{\sigma\psi} = \frac{n_{\sigma\psi}}{\sum_{\sigma'}^{\sigma_{T,\psi}} n_{\sigma'\psi}} \tag{B–25}$$

where $n_{\sigma'\psi}$ is the number of moles of the $\sigma'^{\,th}$ component and $\sigma_{T,\psi}$ is the number of such components. The current version of EQ3NR deals only with solid solutions that are composed of end-member components. The activity coefficients ($\lambda_{\sigma\psi}$) may be computed from a variety of equations. The activity coefficient model for a given solid solution is specified on the EQ3/6 data file, which also contains the requisite supporting parameters. The formulations presently treated and discussed in this EQ3NR supplement in B.3.

Suppose we would like to know how much dissolved carbonate would be in solution if it were in equilibrium with $CO_{2(g)}$. The $CO_{2(g)}$ dissolution reaction may be written as:

$$CO_{2(g)} + H_2O_{(l)} = H^+ + HCO_3^- \qquad \text{(B–26)}$$

The corresponding governing equation is:

$$K_{CO_2(g)} = \frac{a_{H^+} a_{HCO^-}}{f_{CO_2} a_w} \qquad \text{(B–27)}$$

Here f_{CO_2} is the fugacity of CO_2. In order to use this option, the user must provide an input value for it to the speciation model.

Fugacity is a thermodynamic variable for gases that is akin to partial pressure in the same way the thermodynamic activity of an aqueous species is akin to the molal concentration. The formal relationship is given by:

$$f_g = \chi_g p_g \qquad \text{(B–28)}$$

where p_g is the partial pressure and χ_g is the fugacity coefficient of the g^{th} gas. The fugacity coefficient is analogous to the activity coefficient. At low pressures, it approaches unity and hence the fugacity approaches the partial pressure.

Specifying heterogeneous equilibria as inputs to an aqueous speciation model can be a bit dangerous. First, the user must choose which phases, stable or metastable, are controlling solubility equilibria. If a choice is an extremely poor one, the equilibrium concentration of a species so constrained may be very large. Furthermore, the expressions for the logarithm of the ion activity products for all such relations must be a linearly independent set in the corresponding aqueous species. (A corollary to this is that one may not constrain more than one species by the same heterogeneous equilibrium.) Such linear dependence violates the “apparent” or “mineralogic” phase rule (Wolery 1979). This is slightly more restrictive than the phase rule of thermodynamics. Sets of equilibria that satisfy the phase rule, but only because the temperature and pressure happen to fall on a univariant curve, do not satisfy the apparent phase rule.

Equilibrium constraint involving an aqueous species. It is possible to specify equilibrium with other species in a manner in which the species so constrained does not fall under any mass balance constraints. As an example, one might treat dissolved sulfide (represented by HS^-) in this manner, computing it based on equilibrium with sulfate and oxygen gas. The reaction relating sulfide to sulfate is:

$$HS^- + 2O_{2(g)} = H^+ + SO_4^{2-} \qquad \text{(B–29)}$$

The governing equation is the corresponding mass action equation:

$$K_{HS^-} = \frac{a_{H^+} a_{SO_2^-}}{a_{HS^-} f_{O_2}^2} \qquad \text{(B–30)}$$

The sulfide component (HS^- and related species such as $H_2S_{(aq)}$) does not count in the mass balance defined for sulfate. This option is similar to those involving specifying various heterogenous equilibria.

Direct specification of individual molality. EQ3NR allows input of the individual concentrations of master species. The governing equation in this case is just the identity:

$$m_i = m_i \qquad \text{(B–31)}$$

It is largely appropriate only for master species that form no complexes, such as $O_{2(aq)}$ and other dissolved gases.

B.1.3.4 pH in Brines: pHCl and Related Functions as Alternative Constraints

Using standard methods (e.g., Bates 1973a), the *pH* is measured using an ion-specific electrode for the hydrogen ion in combination with a standard reference electrode (usually silver-silver chloride). The electrode pair (commonly marketed as a combination electrode) is calibrated when used by immersion in at least two standard solutions whose *pH* values bracket the expected sample values. This method is appropriate in dilute solutions, but not in brines. The problem is the presence of a liquid junction potential in the reference electrode at the interface between the standard or sample solution and an internal solution composed of concentrated potassium chloride. The idea behind the standard measurement is that for sufficiently dilute sample or standard solutions, the liquid junction potential will remain at an essentially constant value (which can be factored out in the calibration process). As proposed by Bates (1973a), the method should be restricted to solutions of ionic strength no greater than 0.1 molal. However, it is routinely applied to more concentrated solutions, such as seawater (for which the ionic strength is nearly 0.7 molal).

The standard method fails when applied to brines because the liquid junction potential obtained with the sample is significantly changed from that obtained with the relatively dilute calibration buffers. The theory describing liquid junction potentials has been reviewed by Baes and Mesmer (1976). In general, the dependence of the liquid junction potential on the sample solution composition is complex and cannot be solely related to the ionic strength. Thus, one cannot simply make *pH* measurements in the usual way using concentrated calibration standards whose ionic strengths match those of the samples. Furthermore, the theory consists of an ideal and a non-ideal part. Taking only the ideal part and making some approximations leads to the Henderson equation. This has occasionally been put forth as a means of correcting *pH* values in concentrated solutions obtained by going through the mechanics of the standard method. This approach is highly dubious.

Recently Knauss, Wolery, and Jackson (1990, 1991) have proposed a method to quantify *pH* in concentrated solutions that avoids the liquid junction potential problem by eliminating the standard reference electrode. In this method, this electrode is replaced by another specific ion electrode. If this is a chloride electrode, what one measures is *pHCl*, which is the sum of *pH* and *pCl*. As an input to a speciation-solubility code, this is just as adequate as the *pH* as long as there is a separate measurement of dissolved chloride to also input. This maintains a system of *n* equations in *n* unknowns. The code is able to separate *pH* from *pCl* using an activity coefficient model for the dissolved species and a chosen *pH* scale. The subject of aqueous species activity coefficients and *pH* scales is addressed in Section B.2.

Knauss, Wolery, and Jackson (1990) used EQ3NR to compute the *pHCl* and related functions corresponding to different combinations of specific-ion electrodes of various test solutions, such as 0.01 molal HCl with varying concentrations of NaCl. Pitzer's equations were used to compute the activity coefficients in these solutions, using mostly the model of Harvie, Møller, and Weare (1984) and sometimes an alternative data set given by Pitzer (1979). They then measured the corresponding electrical potentials and plotted them against the computed *pHCl* or other function. In most cases, excellent Nernstian responses were obtained, in essence identical to those one would obtain examining the standard *pH* method. This indicated that such solutions could be defined as calibration buffers. Of critical importance to constraining the *pH* in concentrated solutions was the fact that no interference due to sodium was found in the case of the hydrogen ion electrode, even in solutions with very low hydrogen ion concentration and very high sodium ion concentration.

The only observed failures of the method involved cases in which a specific-ion electrode responded to an ion other than the one to which it was supposed to respond. The chloride electrode was found to respond to bromide, for example. In solutions containing both bromide and chloride, however, *pHBr* could be measured without interference by using a bromide electrode. Interferences of this type were no surprise and are in fact well known from the use of the specific ion electrodes in dilute solutions, where they are paired with a standard reference electrode.

The method appears to work, but should receive more study. There are no official recommendations or standards concerning this method, such as those that the National Institute of Standards and Technology (formerly the National Bureau of Standards) has promulgated in the case of the standard *pH* measurement technique. One must currently make up one's own calibration buffers, which ideally should closely resemble the samples. The method has been criticized by Mesmer (1991), who prefers not to obtain *pH* by a method that requires the use of a model for the activity coefficients in the solution. He proposes alternative approaches that involve measuring the concentration of the hydrogen ion. These in turn are criticized by Knauss, Wolery, and Jackson (1991).

Values of *pHCl* and related functions such as *pHBr* and *pH/Na* (= *pH* - *pNa)* can now be input to EQ3NR as alternative constraints. In the case of *pHCl*, the governing equation takes the form:

$$\log m_{H^+} = -\mathrm{pHCl} - \log\gamma_{H^+} - \log m_{Cl^-} - \log\gamma_{Cl^-} \qquad \text{(B–32)}$$

EQ3NR expects to receive input of this type in one of two general forms. The first is the activity combination parameter defined by:

$$\aleph_{ij} = z_j \log a_i - \frac{z_i z_j}{z_j} \log a_j \qquad \text{(B–33)}$$

This is valid for *i* and *j* of any charge combination. Note that = *–pHCl*, so it is actually *–pHCl* that is input to the code, not *pHCl* (analogous to the input of *–pH* instead of *pH*). The more general form of the governing equation is then:

$$\log m_i = \frac{\aleph_{ij}}{z_j} - \log\gamma_i + \frac{z_i}{z_j}\log m_j + \frac{z_i}{z_j}\log\gamma_j \qquad \text{(B–34)}$$

The second general form is to input the mean log activity of the electrolyte composed of ions *i* and *j*:

$$\log a_{\pm,ij} = \frac{z_j \log a_i + z_i \log a_j}{z_i + z_j} \qquad \text{(B–35)}$$

This is not quite as general, because the two ions must have opposite signs of electrical charge. Note that log $a_{+,HCl}$ = –½ *pHCl.* The corresponding governing equation is:

$$\log m_i = \frac{z_i + z_j}{z_j}\log a_{\pm,ij} - \log\gamma_i - \frac{z_i}{z_j}\log m_j - \frac{z_i}{z_j}\log\gamma_j \qquad \text{(B–36)}$$

B.1.3.5 The Carbonate System: Dealing with Alkalinity

To model the carbonate system, EQ3NR expects as normal input an analytical value for total dissolved bicarbonate ($CO_{2(aq)}$ + HCO_3^- + CO_3^{2-}, where these are taken in the sense of components, including any ion pairs or complexes of the corresponding species). The appropriate measurement can be made using ion chromatography or infrared detection of carbon dioxide released from an acidified sample. The results might be expressed on a data sheet as total dissolved CO_2 in mg/L. This must be converted to the equivalent concentration of HCO_3^- for input to EQ3NR, as it is defined on the supporting data files as the basis species corresponding to carbonate mass balance. This can be done by multiplying this quantity by the ratio of the molecular weight of HCO_3^- (61.016 g/mole) to that of $CO_{2(aq)}$ (44.009 g/mole) (the value of this ratio being 1.3864). A data sheet might also list a value for "total free CO_2". This represents only the $CO_{2(aq)}$ component. If this is the only available measure of dissolved bicarbonate, the total dissolved bicarbonate can be computed from this and the *pH* by inverting the relevant equations given later in this section.

Carbonate (in the form of CO_3^{2-} and HCO_3^-, including any ion pairs of these species) makes up nearly all of the alkalinity of many aqueous solutions. Strictly speaking, the alkalinity is a measure of the acid neutralizing capability of an aqueous solution. However, it is also commonly used as a measure of the carbonate system. In fact, alkalinity is only an indirect measure of this system, and its usage as such a measure entails a number of assumptions that are not always valid. In this context, it is also frequently misunderstood and misused. The worst consequence of this usage of alkalinity is that it leads people to think that a direct measurement of total dissolved bicarbonate (in the sense discussed in the above paragraph) is unnecessary. Indeed, it is common to find analytical data sheets on groundwater chemistry that lack such direct measurements.

The purpose of this section is to discuss these problems, and to suggest means by which the total dissolved bicarbonate (in the desired sense) can be estimated, in the case in which direct measurements are lacking. These means are not always entirely satisfactory, and are here suggested mainly for the benefit of those who must work with historical data.

EQ3NR formerly allowed titration alkalinity (A_t) to be input for bicarbonate instead of total concentration. This capability essentially matched that used in the PHREEQE code (Parkhurst, Thorstenson and Plummer 1980). The approach is to define an alkalinity balance equation, which is very similar to a mass balance equation. It may be written in the general form:

$$A_t = \sum_{s=1}^{s_T} \tau_s m_s \qquad \text{(B–37)}$$

where τ_s is the alkalinity factor of the s^{th} species. This is the number of moles of hydrogen ion neutralized by one mole of species in the process of titrating the solution with an acid solution (usually of dilute sulfuric acid) to some end-point, usually in the range of pH 4.0 to 4.5 (See Franson 1976, *Standard Methods for the Examination of Water and Wastewater*, p. 278–293; hereafter referred to as *Standard Methods*). Titration alkalinity defined in this manner is in units of equivalents per kilogram of solvent water, where "equivalent" means hydrogen ion neutralizing equivalent.

Titration alkalinity is usually not reported in these units, however. *Standard Methods* calls for reporting the titration alkalinity in terms of the stoichiometric equivalent of mg/L of $CaCO_3$. We will mark alkalinities in such units with an asterisk (*). Thus, the form of titration alkalinity usually reported must be converted according to:

$$A_t = \frac{A_t^*}{50,000\rho_{g/ml}} \qquad \text{(B–38)}$$

The "50,000" in the above equation is actually the product of 1000 mg/g and the molecular weight of calcium carbonate (taken as 100 g/mole following *Standard Methods*), divided by the alkalinity factor of $CaCO_3$ (2.0 equivalents/mole). The molecular weight of $CaCO_3$ is more accurately 100.088 g/mole, but the 100 g/mole value is used by *Standard Methods* in the formula for calculating A_t^* from the titration data, so retaining it as above is actually more consistent with the titration measurement.

The titration alkalinity (A_t^*) is referred to in *Standard Methods* as T. This quantity may appear on analytical data sheets as "T" or "titration alkalinity" and in units marked "mg/L" or "mg/L

$CaCO_3$". In this context, "mg/L" means "mg/L $CaCO_3$." Users of geochemical modeling codes sometimes mistakenly interpret "mg/L" to mean that the titration alkalinity is given in units of mg/L HCO_3^- (A_{t,HCO_3^-}). It is not unknown for analysts to report the titration alkalinity in such units as well, though this is not a standard practice. It can be obtained by multiplying A_t^* by the molecular weight of HCO_3^- (61.016 g/mole) and the alkalinity factor of $CaCO_3$ (2.0 equivalents/mole) and dividing the result by the product of the molecular weight of $CaCO_3$ and the alkalinity factor of HCO_3^- (1.0 equivalents/mole). In simpler terms, one has:

$$A_{t,\,HCO_3^-} = 1.2192\ A_t^* \tag{B–39}$$

It is very important to note that the titration alkalinity expressed in mg/L HCO_3^- is not equal to the total dissolved bicarbonate (in the sense required for input to EQ3NR) expressed in the same units. Recall that $CO_{2(aq)}$ does not contribute to the titration alkalinity, while it does contribute to the desired total dissolved bicarbonate. Furthermore, CO_3^{2-} contributes differently to the titration alkalinity (by a factor of 2) than it does to the desired total dissolved bicarbonate.

In the determination of alkalinity, *Standard Methods* (Franson 1976) calls for two end points to be determined in the titration. One of these gives *T*, the titration alkalinity, the other *P*, the phenolphthalein alkalinity. The latter corresponds to an end point of *pH* 8.3. If the *pH* of the sample solution is already less than or equal to this, then $P = 0$. The phenolphthalein alkalinity may also appear on an analytical data sheet. *Standard Methods* calls for using *P* to partition the titration alkalinity into components due to bicarbonate, carbonate, and hydroxide; thus, one may write:

$$A_t^* = A_{HCO_3^-}^* + A_{CO_3^{2-}}^* + A_{OH^-}^* \tag{B–40}$$

This scheme implicitly assumes that no other components are present which contribute to the alkalinity. It also ignores ion pairing and complexing as it pertains to these species. Note that each of these component alkalinities is reported in units of equivalent mg/L $CaCO_3$.

These three components, bicarbonate alkalinity, carbonate alkalinity, and hydroxide alkalinity, are determined from *T* and *P* according to the partitioning formula given in *Standard Methods* (Franson 1976, Table 403:I, p. 281). At least one of these three always has a value of zero. Sometimes two are zero. They are supposed to be reported in units of mg/L $CaCO_3$. They are commonly found on analytical data sheets. Since they in essence determine the titration alkalinity, this quantity is sometimes omitted, and if it is desired, it must be computed from then using the above equation.

It is not unknown for analysts to report the bicarbonate alkalinity in units of mg/L HCO_3^-. Users have been known to confuse the bicarbonate alkalinity expressed in such units with the total dissolved bicarbonate (the desired input to EQ3NR), which may be expressed in the same units.

The concentration of the bicarbonate component can be computed from the bicarbonate alkalinity:

$$C_{HCO_3^-,mg/L} = 1.2192 A^*_{HCO_3^-} \tag{B–41}$$

The numerical factor on the right hand side is the same as that appearing in eq (B–39). The molality of the bicarbonate component can be computed directly from the bicarbonate alkalinity:

$$m_{HCO_3^-} = \frac{A^*_{HCO_3^-}}{50{,}044} \tag{B–42}$$

The denominator on the right hand side is the product of 1000 mg/g, the molecular weight of $CaCO_3$, and the alkalinity factor of HCO_3^-, divided by the alkalinity factor of $CaCO_3$. The molecular weight of HCO_3^- is factored out in the derivation of this equation.

The concentration of the carbonate component can be similarly computed from the carbonate alkalinity:

$$C_{CO_3^{2-},mg/L} = 0.5996 A^*_{CO_3^{2-}} \tag{B–43}$$

The numerical factor on the right hand side is the product of the molecular weight of CO_3^{2-} (60.008 g/mole) and the alkalinity factor of $CaCO_3$, divided by the product of the molecular weight of $CaCO_3$ and the alkalinity factor of CO_3^{2-} (also 2.0 equivalent/mole, so the alkalinity factors cancel out). The molality of the carbonate component can be computed directly from the carbonate alkalinity:

$$m_{CO_3^{2-}} = \frac{A^*_{CO_3^{2-}}}{100{,}088} \tag{B–44}$$

It is not unknown for analysts to report the total concentration of bicarbonate as:

$$"C_{T,HCO_3^-,\mathrm{mg/L}}" = C_{HCO_3^-,\mathrm{mg/L}} + 1.0168\, C_{CO_3^{2-},\mathrm{mg/L}} \tag{B–45}$$

where the concentrations on the right hand side are obtained from alkalinities as above and the factor 1.0168 is the ratio of the molecular weight of HCO_3^- to that of CO_3^{2-} and is used to convert the units of carbonate concentration from mg/L CO_3^{2-} to the equivalent mg/L HCO_3^-. In terms of molalities, this is equivalent to taking:

$$"m_{T,HCO_3^-}" = m_{HCO_3^-} + m_{CO_3^{2-}} \tag{B–46}$$

This measure of total bicarbonate, whether reported in mg/L or molality, is not the measure of total bicarbonate which is to be input to EQ3NR, because it does not include the contribution from the component $CO_{2(aq)}$.

Above pH 8.3, the contribution of $CO_{2(aq)}$ to total bicarbonate is negligible (1% or less), and estimates based on the above formulations may be input to EQ3NR with negligible error. At lower pH values, the concentration of $CO_{2(aq)}$ can be estimated from the bicarbonate alkalinity and the pH. *Standard Methods* (Franson 1976, Figure 407:4, p. 297) gives a nomograph for this purpose. The nomograph also takes into account the dependency on the temperature and the ionic strength, using the quantity "total filterable residue" as a proxy for the latter. If this procedure is followed, the total dissolved bicarbonate to be input to EQ3NR can be estimated as:

$$C_{T,HCO_3^-,\mathrm{mg/L}} = 1.3864\, C_{CO_{2(aq)},\mathrm{mg/L}} + C_{HCO_3^-,\mathrm{mg/L}} + 1.0168\, C_{CO_3^{2-},\mathrm{mg/L}} \tag{B–47}$$

where the factor 1.3864 is the ratio of the molecular weight of HCO_3^- to that of $CO_{2(aq)}$ (44.009 g/mole). In terms of molalities, this is equivalent to:

$$m_{T,HCO_3^-} = m_{CO_{2(aq)}} + m_{HCO_3^-} + m_{CO_3^{2-}} \tag{B–48}$$

As an alternative to the nomograph of *Standard Methods*, we note that the molality of $CO_{2(aq)}$ may be estimated from the molality of HCO_3^- and the pH by considering equilibrium for the reaction:

$$CO_{2(g)} + H_2O_{(l)} = H^+ + HCO_3^- \tag{B–49}$$

If the activity of water differs negligibly from a value of unity, the following equation is obtained:

$$m_{CO_{2(aq)}} = 10^{(-\log K_{CO_{2(aq)}} + \log \gamma_{HCO_3^-} - \log \gamma_{CO_{2(aq)}} - pH)}\, m_{HCO_3^-} \tag{B–50}$$

The *log K* for reaction equation (B–49) has values of -6.5804, -6.3447, and -6.2684 at 0, 25, and 60°C (data taken from the **data0.sup.R10** data file). At 25°C, this reduces to:

$$\mathrm{m_{CO_{2(aq)}}} = 10^{6.32 - \mathrm{pH}}\, \mathrm{m_{HCO_3^-}} \tag{B–51}$$

for a dilute solution of ionic strength 0.0024 molal (using the B-dot equation to calculate the activity coefficients; see this Appendix B.2.3). From this, one can see that at pH 4.33, the molality $CO_{2(aq)}$ is 100 times that of HCO_3^-. For seawater (ionic strength of 0.662 molal), the equation becomes:

$$\mathrm{m_{CO_{2(aq)}}} = 10^{6.14 - \mathrm{pH}}\, \mathrm{m_{HCO_3^-}} \tag{B–52}$$

One of the points that may be deduced from these equations is that alkalinity is a poor way to measure the carbonate system in waters of relatively low pH, in which $CO_{2(aq)}$ dominates the total dissolved bicarbonate (defined in the sense desired for input to EQ3NR). The propagated uncertainty in such calculations can become large owing to a contribution from the uncertainty in pH measurement in addition to one from the uncertainty in the measurement in the titration alkalinity (which is interpreted as entirely bicarbonate alkalinity at such low *pH*). The propagated error is also affected to some degree by uncertainty in the values of the activity coefficients, though this is not likely to be of much significance in very dilute solutions. It is probably affected much more by contributions due to uncertainties regarding the contribution to the measured alkalinity of non-carbonate species. This is a potential major problem regarding the use of alkalinity in solutions of any *pH* value.

We have shown above how the total dissolved bicarbonate (in the sense of including aqueous carbon dioxide and carbonate) may be estimated from alkalinity measurements. These methods assume that only bicarbonate, carbonate, and hydroxide contribute to the measured alkalinity. Alkalinity can also be contributed by dissolved organic species such as acetate, by components such as borate, phosphate, silicate, and sulfide, and by some dissolved metals, such as iron and aluminum, in the form of hydroxyl complexes. Of course, if one knows the concentrations of the relevant species, corrections may be attempted. Such corrections could take the form of subtracting the estimated contributions from the measured titration alkalinity. Alternatively, one can make the corrections in a speciation-solubility calculation, using an alkalinity balance equation. It requires assigning alkalinity factors to all the relevant species. Such an approach is available in the PHREEQE code (Plummer, Parkhurst, and Thorstenson, 1980) and previous versions of EQ3NR (Wolery 1983). Either form of correction carries various uncertainties, however, and major problems arise when the corrections are large. In using previous versions of the EQ3NR code in this way, the code has occasionally terminated unsuccessfully because the magnitude of the corrections would have exceeded the value of the reported titration alkalinity.

Many waters of interest to geochemists have substantial amounts of alkalinity due to non-carbonate species. In oil field waters, the titration alkalinity is often heavily dominated (50-100%) by short chain aliphatic anions, chiefly acetate, propionate, butyrate, and valerate, in order of decreasing importance (Willey et al. 1975; Carothers and Kharaka 1978). Organic anions are also present in significant concentrations in the water in and around many landfills and other geologic waste disposal sites. These may be both products of the decomposition of organic wastes and original components of the disposed waste. Waters at disposal sites may also be rich in other components that contribute to alkalinity, such as sulfide, ammonia, phosphate, silicate, and metal hydroxy complexes. Many natural waters of interest are also high in sulfide, and a few are high in borate.

The titration alkalinity input option was removed from EQ3NR for the following reasons:

- To avoid undue propagation of errors inherent in the method, this can be severe in certain cases.

- To avoid possible errors by both analysts and code users concerning the nature, interpretation, and usage of analytical data.

- To avoid the problem of having to assign alkalinity factors to new species added to the supporting data files.

- To encourage the practice of obtaining direct analytical measures of total dissolved bicarbonate.

For cases in which the code user must deal with historical data which include alkalinity measurements but no direct measures of the carbonate system, the following procedure is recommended:

- Using the reported alkalinity and *pH* values, estimate the total dissolved bicarbonate (total dissolved carbon dioxide expressed as bicarbonate) using the methods presented above; make rough corrections if possible for contributions of organics, sulfide, etc.
- Compute a model of the solution.
- Feed the model of the solution to the EQ6 code; simulate the titration process as described by *Standard Methods* (Franson 1976) and compute the corresponding value of the titration alkalinity (an example of this use of EQ6 is given in the EQ6 Theoretical Manual and User's Guide, Wolery and Daveler, 1992).
- Compare the computed value of the titration alkalinity with the reported value; if they match, stop; if not, adjust the estimate of the total dissolved bicarbonate accordingly and repeat the process until a reasonable match if obtained.

This procedure may not always work. For example, if the contribution of bicarbonate/carbonate to the alkalinity is very small compared to that of organics, sulfide, etc., then the available data really offer no constraint on the bicarbonate/carbonate system. In such cases, the user would be wise to recognize the futility of the situation.

B.1.3.6 Redox Constraints

B.1.3.6. There is No "System" *Eh* in Most Real Systems

The high degree of emphasis on trying to understand the geochemistry of natural waters in terms of pure equilibrium thermodynamics has misled many people into believing that the redox state of real aqueous systems can be characterized by a single parameter, usually the *Eh* (a redox potential, given in volts). The related parameter *pe*, the negative of the logarithm of the hypothetical electron, is similarly incapable of describing the overall redox state of a real aqueous system.

The concept of there being such a thing as a "system" *Eh* or a "system" *pe* is based on the assumption that all redox reactions in an aqueous system are in a state of thermodynamic equilibrium. This assumption is a poor one for most real systems (Morris and Stumm, 1967; Jenne, 1981; Hostetler, 1984; Lindberg and Runnells, 1984). In the rush to interpret geochemical data by means of *Eh-pH* and *pe*-activity diagrams, this point is often forgotten or simply ignored. This has had the unfortunate consequence of legitimizing these variables as all-encompassing redox descriptors in the minds of many students.

This misconception has no doubt been reinforced by the use of *Eh* (and sometimes *pe*) as inputs to speciation-solubility codes. Some of these codes require the assumption of a system *Eh*. Most of the better known codes, EQ3NR, WATEQ2 (Ball, Jenne, and Nordstrom, 1979), and PHREEQE (Parkhurst, Plummer, and Thorstenson, 1980) permit the use of such an input but do not require it. With sufficient analytical data, the degree of disequilibrium among various redox couples may be calculated, and the existence of a system *Eh* thus tested. Often, however, the available analytical data are insufficient to do this, and one is forced to assume a system *Eh*.

Redox disequilibrium in natural aqueous systems is created by solar irradiation, radioactive decay, fluid mixing, and transfer of redox components from one phase to another. It is maintained primarily by the strength of covalent bonds, a major factor in the redox disequilibrium of the light elements such as carbon, hydrogen, oxygen, nitrogen, and sulfur. Biological activity literally feeds on redox disequilibrium (e.g., photosynthesis, if one counts the initial disequilibrium due to solar radiation, and chemosynthesis) and catalyzes an overall approach toward redox equilibrium.

Several well known examples of redox disequilibrium in natural aqueous systems can be cited. One is the coexistence of dissolved oxygen and organic carbon in nearly all natural waters, implying disequilibrium between the $O_{2(aq)}/H_2O_{(l)}$ couple and organic/HCO_3^- couples. Another is the disequilibrium between $CH_{4(aq)}/HCO_3^-$ and HS^-/SO_4^{2-} in many marine sediments (Thorstenson, 1970).

As shown below, each such couple can be treated as having have its own redox state. This can be expressed in a variety of ways, including a couple-specific *Eh* or *pe*. These can be calculated using the Nernst equation in conjunction with chemical analyses that are specific with respect to the two oxidation states represented in any redox couple. In the following section, we will discuss the details of this concept.

B.1.3.6. Background: Redox Couples and Half-Reactions

Oxidation-reduction in aqueous systems is commonly treated in terms of redox couples and their associated half-reactions. Common couples in aqueous solution include $O_{2(aq)}/H_2O_{(l)}$, $H_{2(aq)}/H_2O_{(l)}$, Fe^{2+}/Fe^{3+}, HS^{2-}/SO_4^{2-}, SO_3^{2-}/SO_4^{2-}, $S_2O_3^{2-}/SO_4^{2-}$, NH_4^+/NO_3^-, $N_{2(aq)}/NO_3^-$, $CH_{4(aq)}/HCO_3^-$, and a host of organic/HCO_3^- couples. The half-reaction is illustrated in the case of the very important couple $O_{2(aq)}/H_2O_{(l)}$:

$$2H_2O_{(l)} = O_{2(aq)} + 4H^+ + 4e^- \qquad \text{(B–53)}$$

Another very important half-reaction corresponds to the so-called hydrogen electrode:

$$H_{2(g)} = 2H^+ + 2e^- \qquad \text{(B–54)}$$

Multiplying this half-reaction by two and subtracting it from the first yields the following complete redox reaction (which has no electrons among the reactants or products):

$$2H_2O_{(l)} = O_{2(aq)} + 2H_{2(g)} \qquad \text{(B–55)}$$

The thermodynamic convention used to describe the state of electrical potentials of half-reactions in terms of Eh values is to take the electrical potential of the standard hydrogen electrode as zero at all temperatures and pressures. This is consistent with the following additional thermodynamic conventions, where $\Delta G^o_{f,i}$ is the standard state Gibbs energy of the i^{th} species:

- $\Delta G^o_{f,H_{2(g)}} = 0$ at all temperatures and pressures (the standard state fugacity is 1 bar)

- $\Delta G^o_{f,H^+} = 0$ at all temperatures and pressures

- $\Delta G^o_{f,e^-} = 0$ at all temperatures and pressures

The Gibbs energy (ΔG) is related to the electrical potential (E) by the Nernst equation (cf. Garrels and Christ, 1965):

$$\Delta G = +nFE \tag{B–56}$$

where n is the number of electrons in the half-reaction and F is the Faraday constant.

An alternative treatment almost equivalent to that shown above is to write the half-reactions as reduction reactions, so that the electron appears on the left hand side. One then reverses the sign of the right hand side of the Nernst equation as written above. This development is equivalent to the one above, except that the signs of the Gibbs energies and corresponding equilibrium constants and activity products are reversed (Stumm and Morgan, 1981, Chapter 8).

The standard thermodynamic relation describing the Gibbs energy of reaction (ΔG_r),of the r^{th} reaction is:

$$\Delta G_r = \Delta G_r^0 + 2.303RT \log Q_r$$

where ΔG_r^o is the corresponding standard state Gibbs energy of reaction and Q_r is the corresponding activity product. Applying this to half-reactions and using the positive convention version of the Nernst equation, one obtains:

$$E = E^0 + \frac{2.303RT}{nF} \log Q_{+,1/2} \tag{B–58}$$

where E^o is the standard state electrical potential, and $Q_{+,\,1/2}$ is the activity product of the half-reaction. If one uses the negative convention version of the Nernst equation, one obtains instead:

$$E = E^0 - \frac{2.303RT}{nF} \log Q_{-,1/2} \tag{B–59}$$

where $Q_{-,\,1/2}$ is the activity product of the reverse half-reaction. These relations are equivalent because $\log Q_{-,\,1/2} = -\log Q_{+,\,1/2}$. Because the Gibbs energy of the hypothetical electron is always zero, whether it is in the standard state or not, its thermodynamic activity is fixed at unity and it need not explicitly appear in the activity product expressions for half-reactions.

B.1.3.6. Background: *Eh*, *pe*, *Ah*, and Equilibrium Oxygen Fugacity

One can write a modified Nernst equation for any redox couple. In the case of the ferrous-ferric couple, the corresponding half-reaction is:

$$Fe^{2+} = Fe^{3+} + e^{-} \tag{B–60}$$

The corresponding form of the Nernst equation is:

$$E_{Fe^{2+}/Fe^{3+}} = E^{0}_{Fe^{2+}/Fe^{3+}} + \frac{2.303RT}{F}\log\left(\frac{a_{Fe^{3+}}}{a_{Fe^{2+}}}\right) \tag{B–61}$$

Under the thermodynamic conventions adopted above, the potential E on the left hand side of each of the above equations can be taken as the *Eh* for this specific couple. This may or may not equate to the results of an "*Eh*" measurement.

If the two redox couples are in equilibrium with each other, they must have the same *Eh*. Conversely, if they have the same *Eh*, they must be in equilibrium. Any difference in couple-specific *Eh* values is a measure (in volts) of the degree of disequilibrium. This can be shown by relating the Gibbs energy of a combined, complete reaction to the differences in potentials. If the first half-reaction has Eh_1 and n_1 electrons appear in it, and the second half-reaction has Eh_2 and n_2 electrons, one can construct a complete reaction by multiplying the second half-reaction by $-n_1/n_2$ and adding the result to the first half-reaction. Then n_1 electrons are transferred in the complete reaction. The Gibbs energy of this reaction is then given by:

$$\Delta G_r = n_1 F(Eh_1 - Eh_2) \tag{B–62}$$

The condition of zero Gibbs energy of reaction (thermodynamic equilibrium) is met if and only if $Eh_1 = Eh_2$.

The redox parameter *pe*, popularized by Truesdell (1968) and Stumm and Morgan (1981), is defined to be analogous to *pH*:

$$pe = -\log a_e \tag{B–63}$$

where e^- is the hypothetical aqueous electron. It should not be confused with real aqueous electrons, which are extremely scarce in nature. Their thermodynamic properties are not the same. In fact, the hypothetical electron used to define *pe* is not the same as the one used to define *Eh*. The *Eh* conventions require the activity of the hypothetical electron to always be unity. That convention would fix *pe* at a value of zero.

The relation between *pe* and *Eh* is:

$$pe = \left(\frac{F}{2.303RT}\right)Eh \tag{B–64}$$

(Thorstenson, 1970; Stumm and Morgan, 1981). One may derive that this requires the thermodynamic convention

- $\Delta G_{f,e^-} = \frac{1}{2}\Delta G_{H_{2(g)}} - \Delta G_{H^+}$ at all temperatures and pressures

This in turn requires that

- $\Delta G^{o}_{f,e^-} = 0$ at all temperatures and pressures

where as the *Eh* convention for the hypothetical electron was

- $\Delta G^{o}_{f,e^-}\ \Delta G_{f,e^-} = 0$ at all temperatures and pressures

It should be clear that *pe* is not a perfect analog to *pH*, because *pH* is defined with respect to H^+, a real aqueous species, whereas *pe* is defined with respect to a hypothetical species. Each redox couple can have its own *pe*, just as it can have its own *Eh*, the two being related by the equation given above. It follows from the previous development that thermodynamic equilibrium between two redox couples is synonymous with each having the same value of *pe*.

The state of an aqueous redox couple can also be expressed in terms of chemical affinity by the redox affinity, *Ah* (Wolery, 1983). This is a special case of the thermodynamic affinity function (its application to half-reactions). It is related to *Eh* by the relation:

$$Ah = FEh \tag{B–65}$$

The driving force for any kind of complete chemical reaction (meaning to exclude half-reactions) can be expressed by the thermodynamic affinity (A_r), which is related to the equilibrium constant K_r and the activity product Q_r by the equation:

$$A_r = -2.303RT\log\left(\frac{Q_r}{K_r}\right) \tag{B–66}$$

If n_1 electrons appear in one half-reaction and n_2 in another, the two half-reactions can be combined into a complete redox reaction in the manner discussed previously. The thermodynamic affinity of the complete reaction, in which n_1 electrons are transferred, is then related to the *Ah* values (Ah_1 and Ah_2, respectively) of the two half reactions by the equation:

$$A_r = n_1(Ah_1 - Ah_2) \tag{B–67}$$

Thermodynamic equilibrium ($A = 0$) between two redox couples is the case if and only if both couples have the same value of *Ah*.

Alternatively, the state of a redox couple may be expressed in terms of an equilibrium oxygen fugacity (a couple-specific oxygen fugacity). Fugacities are properties of gas species. Gas species do not exist in aqueous solution because, by definition, all species in aqueous solution are aqueous species. Therefore, we can only talk about oxygen fugacities in aqueous solution by reference to hypothetical equilibria with a gas phase. Putting it another way, $O_{2(g)}$ makes a perfectly good hypothetical aqueous species, much like the hypothetical aqueous electron.

Consider the half-reaction:

$$2H_2O_{(l)} = O_{2(g)} + 4H^+ + 4e^- \tag{B–68}$$

where we now take $O_{2(g)}$ to be a hypothetical aqueous species with the thermodynamic properties of the real gas species. We can calculate an equilibrium oxygen fugacity for any half-reaction by coupling it with this one to form a complete redox reaction.

Let K_{Eh} be the equilibrium constant for the $O_{2(g)}/H_2O_{(l)}$ half- reaction given above. Eh and oxygen fugacity are then related by the equation:

$$\log f_{O_2} = \left(\frac{4F}{2.303RT}\right)Eh - 4\log a_{H+} + 2\log a_w + \log K_{Eh} \tag{B–69}$$

This equation can be used to relate the equilibrium oxygen fugacity of any redox couple with its own *Eh*. Two redox couples are in thermodynamic equilibrium with each other if and only if they have the same equilibrium oxygen fugacity.

B.1.3.6. Redox Options: Testing versus Assuming Equilibrium

A commonly used approach in aqueous speciation modeling is to input a total concentration for a dissolved element that occurs in more than one oxidation state and partition it according to a given *Eh*, *pe*, or oxygen fugacity. This, however, requires us to assume that all redox couples in the system are in a state of thermodynamic equilibrium. The EQ3NR code offers this option.

If we constrain the thermodynamic activities of all the aqueous species appearing in a couple's half-reaction without resorting to an input *Eh*, *Ah*, *pe*, or oxygen fugacity, the equations presented above give us a means to calculate its individual redox state expressed as any of the following:

- *Eh* (in terms of an electrical potential, Section B.4.8.13).
- *Ah* (in terms of a chemical potential, Section B.4.8.14).
- *pe* (also Section B.4.8.13).
- oxygen fugacity (Section B.4.8.12).

Analytical techniques do not generally discriminate between a simple species and its ion-pairs and complexes. However, there are techniques in many cases to discriminate between different oxidation states. To calculate the *Eh* of the ferrous-ferric couple, for example, we must have analytical data for both Fe^{2+} and Fe^{3+} (see for example Nordstrom, Jenne, and Ball 1979). If these data are both total concentrations (e.g. total Fe^{2+}, total Fe^{3+}), we simply have two mass balance equations for iron in the aqueous speciation model instead of one.

This is the preferred approach for treating oxidation-reduction in aqueous speciation modeling (see Nordstrom et al. 1979). One may then test whether or not various redox couples are in equilibrium with each other. EQ3NR can treat any redox couple in this fashion. Alternative constraints discussed in the previous subsection could substitute for one or both total concentrations/ mass balances in the usual way. The code will use a redox default to partition an element that appears in more than one oxidation state if insufficient data are input to calculate a couple-specific parameter. The redox default may be an input *Eh*, a *pe*, or log oxygen fugacity. Alternatively, it may be defined by a redox couple for which sufficient data are input to calculate couple-specific parameters. By constraining one or more of the species in the corresponding half-reaction by a heterogeneous equilibrium constraint, it is possible to constrain the default redox state by a heterogeneous equilibrium.

B.1.3.7 Measures of Mineral Saturation

EQ3NR employs two measures of the saturation state of an aqueous solution with respect to minerals. The first is the saturation index defined as:

$$\mathrm{SI} = \log \frac{Q}{K} \tag{B–70}$$

where it is understood that *Q* is the activity product and *K* the equilibrium constant for a dissolution reaction. In the case of the dissolution of calcite, for example, if the reaction is written as:

$$\mathrm{Calcite} + H^{+} = Ca^{2+} + HCO_3^{-} \tag{B–71}$$

the ion activity product is then defined as:

$$Q_{Calcite} = \frac{a_{Ca^{2+}} a_{HCO^{-}}}{a_{Calcite} a_{H^{+}}} \tag{B–72}$$

The second measure of the saturation state is the thermodynamic affinity of the precipitation reaction. The affinity of a reaction (no matter how it is written) is related to its activity product and equilibrium constant by:

$$A = -2.303RT \log \frac{K}{Q} = 2.303RT \log \frac{Q}{K} \tag{B–73}$$

Because log *Q*/*K* reverses sign when the reaction is reversed, the affinity to precipitate is related to the saturation index by:

$$A_{-} = 2.303RT\,SI \tag{B–74}$$

Following these conventions, both SI and A_{-} are positive for supersaturated minerals, zero for saturated ones, and negative for undersaturated minerals.

In the case of solid solution minerals with end-member components, the saturation index of the σ^{th} end member is related to that of the corresponding pure phase ϕ by:

$$\mathrm{SI}_{\sigma\Psi} = \mathrm{SI}_{\phi} - \log a_{\sigma\Psi} = \mathrm{SI}_{\phi} - \log x_{\sigma\Psi} - \log \lambda_{\sigma\Psi} \tag{B–75}$$

where $a_{\sigma\psi}$ is the thermodynamic activity of the end-member, $x_{\sigma\psi}$ is its mole fraction, and $\lambda_{\sigma\psi}$ is its mole fraction activity coefficient. Consideration of an overall dissolution reaction of a solid solution of given composition suggests that the saturation index of the ψ^{th} solid solution should be defined by:

$$\mathrm{SI}_{\Psi} = \sum_{\sigma=1}^{\sigma_{T,\Psi}} x_{\sigma\Psi} \mathrm{SI}_{\sigma\Psi} \tag{B–76}$$

Affinity functions can be defined analogously.

The problem of defining the saturation state of a solid solution for which no composition is given is not so straightforward, because the result is composition-dependent. One approach to this would be to find the compositions that maximize the SI. This is the method presently employed in EQ3/6 (see Bourcier 1985).

B.1.4 USE AND MISUSE OF SPECIATION-SOLUBILITY CODES

There is significant potential to misuse any speciation-solubility code. No such code should be used as a “black box”. As Jenne (1981, p. 36) puts it, “... each application should be viewed as a partial validation.” The geochemical model of each new scenario (e.g., a set of waters in a compositional range not previously studied) may have a different set of important aqueous species, and hence provide a test of some thermodynamic data that have not previously been exercised. In addition, reactions controlled by equilibrium in one situation may be in disequilibrium in another, and vice versa, especially heterogeneous and aqueous redox reactions.

Geochemical modeling with aqueous speciation-solubility codes must actively address three questions. First, are all the significant species in the model? Second, are all the important thermodynamic data sufficiently correct? Do they make sense when compared with the model outputs when working with a set of water samples? Do they make sense in comparison with other knowledge about an aqueous system, such as data on the identities of minerals with which the water is in contact? Third, would disequilibrium constraints be more appropriate than equilibrium constraints for some reactions (especially aqueous redox reactions)? Users should keep in mind the admonition of Nordstrom et al. (1979) that “...no model is better than the assumptions on which it is based.”

If no thermodynamic data are available for species known or suspected to be important in a given application, then such data should be estimated by empirical or semi-empirical methods. EQ3NR has an **input** file option that permits the user to temporarily modify equilibrium constants at run time (see 3.3.14). This makes it convenient to conduct sensitivity studies of the uncertainty in such estimated values. Langmuir (1979) summarizes approaches for estimating thermodynamic properties of aqueous species and reactions. Tardy and Garrels (1974), Wolery (1978), and Helgeson et al. (1978) discuss methods for estimating the thermodynamic properties of minerals.

A common problem faced by novices at speciation-solubility modeling is that their models come out grossly supersaturated with nearly every aluminum and ferric iron-bearing mineral in the data base. This often occurs because analysis is made of inadequately filtered samples, which commonly contain colloidal particles of these two components. These particles then "inflate" the corresponding chemical analyses. Busenberg (1978) showed that large quantities of a colloidal aluminum phase occurred in the size range 0.1–0.45 μm during a set of feldspar dissolution experiments. Laxen and Chandler (1982) did a more detailed study of iron particulate size distribution in fresh waters. Their work shows that a filter finer than 0.1 μm is necessary to effectively remove these particulates from the chemical analysis.

The modeler should be aware that many solubility-controlling phases, especially at low temperature, are metastable (e.g., amorphous $Fe(OH)_3$ may control the level of dissolved iron, not the more stable hematite, Fe_2O_3). In addition, the stability of some controlling phases may be somewhat variable due to such factors as crystallinity (i.e., crystal size), order/disorder, ionic substitution, or, in the case of fresh precipitates, aging. Helgeson et al. (1978) discuss many of these effects.

One approach that may be helpful to users is to estimate the amount of aluminum or iron that would be in solution under the assumption of appropriate solubility equilibria. For example, one might constrain dissolved aluminum to satisfy equilibrium with gibbsite ($Al(OH)_{3(c)}$) or constrain iron to satisfy equilibrium with amorphous $Fe(OH)_3$ or a nontronite (ferric-rich smectite) clay. However, this is not a substitute for analysis of carefully filtered samples.

The state of available analytical data on water compositions is often a limiting factor in the usage of EQ3NR or any other speciation-solubility code. In general, the data must be both accurate and sufficiently complete. Inaccurate data often result when methods suitable for analyzing drinking water are applied to waters very dissimilar to this medium. This can take the form of both positive and negative interferences. Some analytical parameters (e.g., pH, alkalinity, dissolved sulfide) must be measured immediately upon sampling to avoid changes due to mineral precipitation, ingassing, or outgassing. Water samples should be inspected after transportation and storage for the formation of precipitates. Quite often, analytical data are incomplete for geochemical modeling purposes. This may have the effect of completely inhibiting modeling work, or it may result in modeling with assumptions that could have been avoided if the right hard data had been available. In general, analytical work is most useful to modeling if there is interplay between the modeler and the analyst.

Internal consistency can provide useful tests of the quality of aqueous speciation models (see Merino 1979). One such test is to compare the calculated electrical imbalance with the cation/anion subtotals for charge equivalents. EQ3NR makes these calculations, which are a meaningful test if electrical balance is not used as an input constraint. Merino (1979) also recommends the technique of comparing measured and independently calculated values of titration alkalinity. In essence, his recommendation corresponds to the currently recommended procedure for dealing with alkalinity as described earlier in this section.

B.2 ACTIVITY COEFFICIENTS OF AQUEOUS SPECIES

B.2.1INTRODUCTION

The thermodynamic activities (a_i) of aqueous solute species are usually defined on the basis of molalities. Thus, they can be described by the product of their molal concentrations (m_i) and their molal activity coefficients (γ_i):

$$a_i = m_i \gamma_i \tag{B–77}$$

The thermodynamic activity of the water (a_w) is always defined on a mole fraction basis. Thus, it can be described analogously by product of the mole fraction of water (x_w) and its mole fraction activity coefficient (λ_w):

$$a_w = x_w \lambda_w \tag{B–78}$$

It is also possible to describe the thermodynamic activities of aqueous solutes on a mole fraction basis. However, such mole fraction-based activities ($a_i^{(x)}$) are not the same as the more familiar molality-based activities ($a_i^{(m)}$), as they are defined with respect to different choices of standard states. Mole fraction based activities and activity coefficients (λ_i), are occasionally applied to aqueous nonelectrolyte species, such as ethanol in water. In geochemistry, the aqueous solutions of interest almost always contain electrolytes, so mole-fraction based activities and activity coefficients of solute species are little more than theoretical curiosities. In EQ3/6, only molality-based activities and activity coefficients are used for such species, so a_i always implies $a_i^{(m)}$. Because of the nature of molality, it is not possible to define the activity and activity coefficient of water on a molal basis; thus, a_w always means $a_w^{(x)}$.

Solution thermodynamics is a construct designed to approximate reality in terms of deviations from some defined ideal behavior. The complex dependency of the activities on solution composition is thus dealt with by shifting the problem to one of describing the activity coefficients. The usual treatment of aqueous solutions is one which simultaneously employs quantities derived from, and therefore belonging to, two distinct models of ideality (Wolery 1990). All solute activity coefficients are based on molality and have unit value in the corresponding model of ideality, called molality-based ideality. The activity and activity coefficient of water are not constant in an ideal solution of this type, though they do approach unit value at infinite dilution. These solvent properties are derived from mole fraction-based ideality, in which the mole fraction activity coefficients of all species components in solution have unit value. In an ideal solution of this type, the molal activity coefficients of the solutes are not unity, though they approach it at infinite dilution (see Wolery 1990).

Any geochemical modeling code which treats aqueous solutions must provide one or more models by which to compute the activity coefficients of the solute species and the solvent. In many codes, what is computed is the set of γ_i plus a_w. As many of the older such codes were constructed to deal only with dilute solutions in which the activity of water is no less than about 0.98, some of these just take the activity of water to be unity. With the advent of activity coefficient models of practical usage in concentrated solutions (mostly based on Pitzer's 1973, 1975 equations), there has been a movement away from this particular and severe approximation. Nevertheless, it is generally the activity of water, rather than the activity coefficient of water, which is evaluated from the model equations. This is what was previously done in EQ3/6. However, EQ3/6 now evaluates the set of γ_i plus λ_w. This is done to avoid possible computational singularities that may arise, for example if heterogeneous equilibria happen to fix the activity of water (e.g., when a solution is saturated with both gypsum and anhydrite).

Good models for activity coefficients must be accurate. A prerequisite for general accuracy is thermodynamic consistency. The activity coefficient of each aqueous species is not independent of that of any of the others. Each is related to a corresponding partial derivative of the excess Gibbs energy of the solution (G^{EX}). The excess Gibbs energy is the difference between the complete Gibbs energy and the ideal Gibbs energy. Because there are two models of ideality, hence two models for the ideal Gibbs energy, there are two forms of the excess Gibbs energy, G^{EXm} (molality-based) and G^{EXx} (mole fraction-based). The consequences of this are discussed by Wolery (1990). In versions 7.0 and 8.0 of EQ3/6, all activity coefficient models are based on ideality defined in terms of molality. Thus, the excess Gibbs energy of concern is G^{EXm}. The activity of water, which is based on mole-fraction ideality, is imported into this structure as discussed by Wolery (1990). The relevant differential equations are:

$$\ln \gamma_i = \frac{1}{RT}\frac{\partial G^{EXm}}{\partial n_i} \tag{B–79}$$

$$\ln a_w = -\frac{\sum m}{\Omega} + \frac{1}{RT}\frac{\partial G^{EXm}}{\partial n_i} \tag{B–80}$$

where R is the gas constant, T the absolute temperature, Ω the number of moles of solvent water comprising a mass of 1 kg ($\Omega \approx 55.51$),and:

$$\sum m = \sum_i m_j \tag{B–81}$$

the sum of molalities of all solute species. Given an expression for the excess Gibbs energy, such equations give a guaranteed route to thermodynamically consistent results (Pitzer 1984; Wolery 1990). Equations that are derived by other routes may be tested for consistency using other relations, such as the following forms of the cross-differentiation rule (Wolery, 1990):

$$\frac{\partial \ln \gamma_j}{\partial m_i} = \frac{\partial \ln \gamma_i}{\partial m_j} \tag{B–82}$$

$$\frac{\partial \ln a_w}{\partial n_i} = \frac{\partial \ln \gamma_i}{\partial n_w} - \frac{1}{n_w} \tag{B–83}$$

In general, such equations are most easily used to prove that a set of model equations is not thermodynamically consistent. The issue of sufficiency in proving consistency using these and related equations (Gibbs-Duhem equations and sum rules) is addressed by Wolery (1990).

The activity coefficients in reality are complex functions of the composition of the aqueous solution. In electrolyte solutions, the activity coefficients are influenced mainly by electrical interactions. Much of their behavior can be correlated in terms of the ionic strength, defined by:

$$I = \frac{1}{2}\sum_i m_i z_i^2 \tag{B–84}$$

where the summation is over all aqueous solute species and z_i is the electrical charge. However, the use of the ionic strength as a means of correlating and predicting activity coefficients has been taken to unrealistic extremes (e.g., in the mean salt method of Garrels and Christ 1965, p. 58–60). In general, model equations which express the dependence of activity coefficients on solution composition only in terms of the ionic strength are restricted in applicability to dilute solutions.

The three basic options for computing the activity coefficients of aqueous species in EQ3/6 are models based respectively on the Davies (1962) equation, the "B-dot" equation of Helgeson (1969), and Pitzer's (1973, 1975, 1979, 1987) equations. The first two models, owing to limitations on accuracy, are only useful in dilute solutions (up to ionic strengths of 1 molal at most). The third basic model is useful in highly concentrated as well as dilute solutions, but is limited in terms of the components that can be treated.

With regard to temperature and pressure dependence, all of the following models are parameterized along the 1 atm/steam saturation curve. This corresponds to the way in which the temperature and pressure dependence of standard state thermodynamic data are also presently treated in the software. The pressure is thus a function of the temperature rather than an independent variable, being fixed at 1.013 bar from 0–100°C and the pressure for steam/liquid water equilibrium from 100–300°C. However, some of the data files have more limited temperature ranges.

B.2.2THE DAVIES EQUATION

The first activity coefficient model in EQ3/6 is based on the Davies (1962) equation:

$$\log \gamma_i = -A_{\gamma,10} z_i^2 \left(\frac{\sqrt{I}}{1+\sqrt{I}} + 0.2I \right) \tag{B–85}$$

(the constant 0.2 is sometimes also taken as 0.3). This is a simple extended Debye-Hückel model (it reduces to a simple Debye-Hückel model if the "$0.2I$" part is removed). The Davies equation is frequently used in geochemical modeling (e.g., Parkhurst, Thorstenson, and Plummer 1980; Stumm and Morgan 1981). Note that it expresses all dependence on the solution composition through the ionic strength. Also, the activity coefficient is given in terms of the base ten logarithm, instead of the natural logarithm. The Debye-Hückel A_γ parameter bears the additional label "10" to ensure consistency with this. The Davies equation is normally only used for temperatures close to 25°C. It is only accurate up to ionic strengths of a few tenths molal in most solutions. In some solutions, inaccuracy, defined as the condition of model results differing from experimental measurements by more than the experimental error, is apparent at even lower concentrations.

In EQ3/6, the Davies equation option is selected by by choosing the B-dot activity model in the input file. A supporting data file consistent with the use of a simple extended Debye-Hückel model must also be supplied (e.g., **data1** = **data1.com**, **data1.sup**, or **data1.ymp**).

The Davies equation has one great strength: the only species-specific parameter required is the electrical charge. This equation may therefore readily be applied to a wide spectrum of species, both those whose existence is well-established and those whose existence is only hypothetical.

The Davies equation predicts a unit activity coefficient for all neutral solute species. This is known to be inaccurate. In general, the activity coefficients of neutral species that are non-polar (such as $O_{2(aq)}$, $H_{2(aq)}$, and $N_{2(aq)}$) increase with increasing ionic strength (the "salting out effect," so named in reference to the corresponding decreasing solubilities of such species as the salt concentration is increased; cf. Garrels and Christ, 1965, p. 67-70). In addition, Reardon and Langmuir (1976) have shown that the activity coefficients of two polar neutral species (the ion pairs $CaSO_{4(aq)}$ and $MgSO_{4(aq)}$) decrease with increasing ionic strength, presumably as a consequence of dipole-ion interactions.

The Davies equation is thermodynamically consistent. It is easy to show, for example, that it satisfies the solute-solute form of the cross-differentiation equation.

Most computer codes using the Davies equation set the activity of water to one of the following: unity, the mole fraction of water, or a limiting expression for the mole fraction of water. Usage of any of these violates thermodynamic consistency, but this is probably not of great significance as the inconsistency is numerically not significant at the relatively low concentrations at which the Davies equation itself is accurate. For usage in EQ3/6, we have used standard thermodynamic relations to derive the following expression:

where "2.303" is a symbol for and approximation of ln 10 (warning: this is not in general a sufficiently accurate approximation) and:

$$\log a_w = \frac{1}{\Omega}\left(-\frac{\sum m}{2.303} + \frac{2}{3} A_{\gamma,10} I^{\frac{3}{2}} \sigma\left(\sqrt{I}\right) - 2(0.2) A_{\gamma,10} I^2 \right) \tag{B–86}$$

This result is thermodynamically consistent with the Davies equation.

$$\sigma(x) = \frac{3}{x^3}\left(1 + x - \frac{1}{1+x} - 2\ln(1+x) \right) \tag{B–87}$$

B.2.3THE B-DOT EQUATION

The second model for activity coefficients available in EQ3/6 is based on the B-dot equation of Helgeson (1969) for electrically charged species:

$$\log \gamma_i = -\frac{A_{\gamma,10} z_i^2 \sqrt{I}}{1 + \mathring{a}_i B_\gamma \sqrt{I}} + \dot{B} I \tag{B–88}$$

Here $\mathring{a}_i$ is the hard core diameter of the species, B_γ is the Debye-Hückel B parameter, and $\dot{B}$ is the characteristic B-dot parameter. Like the Davies equation, this is a simple extended Debye-Hückel model, the extension being the "$\dot{B} I$" term. The Debye-Hückel part of this equation is equivalent to that of the Davies equation if the product "$\mathring{a}_i B_\gamma$" has a value of unity. In the extended part, these equations differ in that the Davies equation has a coefficient in place of which depends on the electrical charge of the species in question.

In EQ3/6, the B-dot equation option is selected by choosing the B-dot activity model in the input file. A supporting data file consistent with the use of a simple extended Debye-Hückel model must also be supplied (e.g., **data1** = **data1.com**, **data1.sup**, or **data1.ymp**). Note that these data files support the use of the Davies equation as well (the $\mathring{a}_i$ data on these files is simply ignored in that case).

The B-dot equation has about the same level of accuracy as the Davies equation, and almost as much universality (one needs to know $\mathring{a}_i$ in addition to z_i). However, it fails to satisfy the solute-solute form of the cross-differentiation rule. The first term is consistent with this rule only if all hard core diameters have the same value. The second is consistent only if all ions share the same value of the square of the electrical charge. However, the numerical significance of the inconsistency is small in the range of low concentrations in which this equation can be applied with useful accuracy. On the positive side, the B-dot equation has been developed (Helgeson, 1969) to span a wide range of temperature (up to 300°C).

For electrically neutral solute species, the B-dot equation reduces to:

$$\log \gamma_i = \dot{B} I \tag{B–89}$$

As has positive values at all temperatures in the range of application, the equation predicts a salting out effect. However, by tradition (Helgeson et al., 1970), the B-dot equation itself is not used in the case of neutral solute species. The practice, as suggested by Garrels and Thompson (1962) and reiterated by Helgeson (1969), is to assign the value of the activity coefficient of aqueous CO_2 in otherwise pure sodium chloride solutions of the same ionic strength. This function was represented in previous versions of EQ3/6 by a power series in the ionic strength:

$$\log \gamma_i = k_1 I + k_2 I^2 + k_3 I^3 + k_4 I^4 \tag{B–90}$$

The first term on the right hand side dominates the others. The first coefficient is positive, so the activity coefficient of CO_2 increases with increasing ionic strength (consistent with the "salting out" effect). As it was applied in EQ3/6, the coefficients for the power series themselves were represented as similar power series in temperature, and this model was fit to data taken from Table 2 of Helgeson (1969). These data (including extrapolations made by Helgeson) covered the range 25–300°C and 0–3 molal *NaCl*.

The high order power series in eq (B–90) was unfortunately very unstable when extrapolated outside the range of the data to which it was fit. EQ3NR and EQ6 would occasionally run into an unrecoverable problem attempting to evaluate this model for high ionic strength values generated in the process of attempting to find a numerical solution (not necessarily because the solutions in question really had high ionic strength). To eliminate this problem, the high order power series has been replaced by a new expression after Drummond (1981, p. 19):

$$\ln\gamma_i = \left(C + FT + \frac{G}{T}\right)I - (E + HT)\left(\frac{I}{I+1}\right) \qquad \text{(B–91)}$$

where T is the absolute temperature and C = –1.0312, F = 0.0012806, G = 255.9, E = 0.4445, and H = –0.001606. Note that this is presented in terms of the natural logarithm. Conversion is accomplished by using the relation:

$$\log x = \frac{\ln x}{2.303} \qquad \text{(B–92)}$$

This expression is both much simpler (considering the dependencies on both temperature and ionic strength) and is more stable. However, in deriving it, the ionic strength was taken to be equivalent to the sodium chloride molality. In the original model (based on Helgeson 1969), the ionic strength was based on correcting the sodium chloride molality for ion pairing. This correction is numerically insignificant at low temperature. It does become significant at high temperature. However, neither this expression nor the power series formulation it replaced is thermodynamically consistent with the B-dot equation itself, as can be shown by applying the solute-solute cross-differentiation rule.

The more recent previous versions of EQ3/6 only applied the "CO_2" approximation to species that are essentially nonpolar (e.g., $O_{2(aq)}$, $H_{2(aq)}$, $N_{2(aq)}$), for which salting-out would be expected. In the case of polar neutral aqueous species, the activity coefficients were set to unity (following the recommendation of Garrels and Christ, 1965, p. 70); i.e., one has:

$$\log\gamma_i = 0 \qquad \text{(B–93)}$$

This practice is still followed in the present version of the code.

EQ3/6 formerly complemented the B-dot equation with an approximation for the activity of water that was based on assigning values in pure sodium chloride solutions of the same "stoichiometric" ionic strength (Helgeson et al. 1970). This approximation was fairly complex and was, of course, not thermodynamically consistent with the B-dot equation itself. In order to simplify the data requirements, as well as avoid the need to employ a second ionic strength function, this formulation has been replaced by a new one which depends on the parameter and is quasi-consistent with the B-dot equation:

$$\log a_w = \frac{1}{\Omega}\left(-\frac{\sum m}{2.303} + \frac{2}{3} A_{\gamma,10} I^{\frac{3}{2}} \sigma\left(å\mathrm{B}_\gamma \sqrt{I}\right) - \dot{B} I^2 \right) \tag{B–94}$$

The solute hard core diameter (å) is assigned a fixed value of 4.0Å (a reasonable value). This equation is consistent with the B-dot equation if all solute species are ions, have the same fixed value of the hard core diameter, and have the same value of the square of the electrical charge.

B.2.4SCALING OF INDIVIDUAL IONIC ACTIVITY COEFFICIENTS: PH SCALES

Before proceeding to a discussion of Pitzer's (1973, 1975) equations, we will address the problem of scaling associated with the activity coefficients of individual ions. It is not possible to observe (measure) any of the thermodynamic functions of such species, because any real solution must be electrically balanced. Thus, the activity coefficients of aqueous ions can only be measured in electrically neutral combinations. These are usually expressed as the mean activity coefficients of neutral electrolytes. The mean activity coefficient of neutral electrolyte *MX* (*M* denoting the cation, *X* the anion) is given by:

$$\log \gamma_{\pm,\mathrm{MX}} = \frac{\nu_\mathrm{M} \log \gamma_\mathrm{M} + \nu_\mathrm{X} \log \gamma_\mathrm{X}}{\nu_\mathrm{MX}} \tag{B–95}$$

where ν_M is the number of moles of cation produced by dissociation of one mole of the electrolyte, ν_X is the number of moles of anion produced, and:

$$\nu_\mathrm{MX} = \nu_\mathrm{M} + \nu_\mathrm{X} \tag{B–96}$$

Electrical neutrality requires that:

$$z_\mathrm{M} \nu_\mathrm{M} = -z_\mathrm{X} \nu_\mathrm{X} \tag{B–97}$$

Although the activity coefficients of ions can not be individually observed, the corresponding molal concentrations can be. The corresponding products, the thermodynamic activities of the ions, are not individually observable, precisely because of the problem with the activity coefficients. Thus, the problem of obtaining individual activity coefficients of the ions and the problem of obtaining individual activities of the same species is really the same problem.

Individual ionic activity coefficients can be defined on a conventional basis by introducing some arbitrary choice. This is can be made by adopting some expression for the activity coefficient of a single ion. The activity coefficients of all other ions then follow via electroneutrality relations. The activities for all the ions are then also determined (cf. Bates and Alfenaar, 1969). Because this applies to the hydrogen ion, such an arbitrary choice then determines the *pH*. Such conventions are usually made precisely for this purpose, and they are generally known as *pH* scales.The NBS *pH* scale, which is the basis of nearly all modern conventional *pH* measurement, is based on the Bates-Guggenheim equation (Bates, 1973a):

$$\log \gamma_{Cl^-} = \frac{-A_{\gamma,10}\sqrt{I}}{1 + 1.5\sqrt{I}} \tag{B–98}$$

This scale is significant not only to the measurement of *pH*, but of corresponding quantities (e.g., *pCl*, *pBr*, *pNa*) obtained using other specific-ion electrodes (cf. Bates and Alfenaar, 1969; Bates, 1973b; Bates and Robinson, 1974).

The Bates-Guggenheim equation, like the Davies equation and the B-dot equation, is an extended Debye-Hückel formula. However, if one applies the Davies equation or the B-dot equation to the chloride ion, the result is not precisely identical. The difference approaches zero as the ionic strength approaches zero, and is not very significant quantitatively in the low range of ionic strength in which either the Davies equation or the B-dot equation has useful accuracy. Nevertheless, the use of either of these equations in uncorrected form introduces an inconsistency with measured *pH* values, as use of the Davies equation for example would interpret the *pH* as being on an implied "Davies" scale.

Activity coefficients (and activities) of ions can be moved from one scale to another. The general relation for converting from scale (1) to scale (2) is (Knauss, Wolery, and Jackson 1991):

$$\log \gamma_{\mathrm{i}}^{(2)} = \log \gamma_{\mathrm{i}}^{(1)} = \frac{\mathrm{z_i}}{\mathrm{z_j}}(\log \gamma_{\mathrm{j}}^{(2)} - \log \gamma_{\mathrm{j}}^{(1)}) \tag{B–99}$$

For example, if we evaluate the Davies equation for all ions, we may take the results as being on scale (1). To convert these to the NBS scale (here scale (2)), we take the j^{th} ion to be the chloride ion and evaluate the Bates-Guggenheim equation. We then apply the scale conversion equation to every other ion *i*.

In EQ3/6, activity coefficients are first calculated from the "raw" single-ion equations. They are then immediately rescaled, unless no rescaling is to be done. Thus, rescaling occurs during the iteration process; it is not deferred until convergence has been achieved. The user can control r rescaling via the options denoted in the input file. If option to rescale is chosen, all single-ion activity coefficients and activities are put on a scale which is defined by the relation:

$$\log \gamma_{H^+} = 0 \tag{B–100}$$

This has the effect of making the activity and the molality of the hydrogen ion numerically equal. This may have some advantages in comparing with experimental measurements of the hydrogen ion molality. Such measurement techniques have recently been discussed by Mesmer (1991).

The problem of scaling the activity coefficients of ions is more acute in concentrated solutions, and the need to discriminate among different scales in geochemical modeling codes has only been addressed as such codes have been written or modified to treat such solutions (e.g., Harvie, Møller, and Weare 1984; Plummer et al. 1988).

B.2.5 PITZER'S EQUATIONS

B.2.5.1 Introduction

Pitzer (1973, 1975) proposed a set of semi-empirical equations to describe activity coefficients in aqueous electrolytes. These equations have proven to be highly successful as a means of dealing with the thermodynamics of concentrated solutions (e.g., Pitzer and Kim 1974). Models based on these equations have been developed to describe not only solution properties, but also equilibrium between such solutions and salt minerals (e.g., Harvie and Weare 1980; Harvie, Møller, and Weare 1984). The utility of these models in geochemical studies has been well established. For example, such models have been shown to account for the mineral sequences produced by evaporation of seawater (Harvie and Weare 1980), the process of trona deposition in Lake Magadi, Kenya (Monnin and Schott 1984), and the formation of the borate-rich evaporite deposits at Searles Lake, California (Felmy and Weare 1986).

Pitzer's equations are based on a semi-theoretical (see Pitzer 1973) interpretation of ionic interactions, and are written in terms of interaction coefficients (and parameters from which such coefficients are calculated). There are two main categories of such coefficients, "primitive" ones which appear in the original theoretical equations, but most of which are only observable in certain combinations, and others which are "observable" by virtue of corresponding to observable combinations of the primitive coefficients or by virtue of certain arbitrary conventions. Only the observable coefficients are reported in the literature.

There is a very extensive literature dealing with Pitzer's equations and their application in both interpretation of experimental data and calculational modeling. A complete review is beyond the scope of the present manual. Discussion here will be limited to the equations themselves, how to use them in EQ3/6, and certain salient points that are necessary in order to use them in an informed manner. Readers who wish to pursue the subject further are referred to reviews given by Pitzer (1979, 1987, 1991). Jackson (1988) has addressed the verification of the addition of Pitzer's equations to EQ3/6.

In EQ3/6, the Pitzer's equations option is selected by setting the option in the input file. A supporting data file consistent with this option must also be supplied (e.g., **data1** = **data1.hmw** or **data1.ypf**). If the option for a Pitzer activity is selected and the supporting data file is not of the appropriate type, the software terminates with an error message.

Pitzer's equations are based on the following virial expansion for the excess Gibbs energy:

$$G^{EXm} = RT\left(w_w f(I) + \left(\frac{1}{w_w}\right)\sum_{ij} \lambda_{ij}(I) n_i n_j + \left(\frac{1}{w_w^2}\right)\sum_{ijk} \mu_{ijk} n_i n_j n_k \right) \tag{B–101}$$

where w_w is the number of kilograms of solvent water, $f(I)$ is a Debye-Hückel function describing the long-range electrical interactions to first order, the subscripts i, j, and k denote aqueous solute species, and n_i is the number of moles of the i^{th} solute species. The equation also contains two kinds of interaction or virial coefficients: the λ_{ij} are second order interaction coefficients, and the μ_{ijk} are third order interaction coefficients. A key element in the success of Pitzer's equations is the treatment of the second order interaction coefficients as functions of ionic strength. As will be discussed later in more detail, the λ_{ij} consist of both theoretically defined and empirical parts, while the μ_{ijk} are completely empirical. As the term is used here, "empirical" means that at least some of the parameter values required to evaluate a quantity must be obtained by fitting experimental data.

The sums in the interaction coefficient terms are actually double and triple sums. As the number of components in a system increases, the number of interaction coefficients of the type shown above becomes very large. It turns out there are many more of the λ_{ij} and μ_{ijk} than can actually be observed, other than in combination. For example, in the equation for the excess Gibbs energy, it is quickly obvious that λ_{12} and λ_{21} can only be seen in the combination $(\lambda_{12} + \lambda_{21})$, and a similar situation holds in the case of the μ_{ijk}. This leads to the first simplification in dealing with these coefficients, which is that those with the same subscripts (regardless of order or permutation) are required to be equal (Pitzer 1973). This is not the end of the story, as other constraints (mostly related to electrical neutrality) force even more simplifications (which will be addressed later).

A set of thermodynamically consistent equations for the activity coefficients follows by application of the partial differential equations given previously. In the case of solute species, this leads to:

$$\ln \gamma_i = \left(\frac{z_i^2}{2}\right) f'(I) + 2\sum_i \lambda_{ij}(I) m_j + \sum_{ik} \left(\left(\frac{z_i^2}{2}\right) \lambda'_{jk}(I) + 3\mu_{ijk}\right) m_j m_k \qquad \text{(B–102)}$$

where $f'(I)$ is the derivative df/dI and $\lambda'_{ij}(I)$ is similarly $d\lambda_{ij}/dI$. For water, the corresponding result is:

$$\ln a_w = -\frac{\sum m}{\Omega} - \frac{1}{\Omega}\left(If'(I) - f(I)\right) + \frac{1}{\Omega}\left(\sum_{ij} \left(\lambda_{ij}(I) + \lambda'_{ij}(I)\right) m_i m_j + 2\sum_{ijk} \mu_{ijk} m_i m_j m_k\right) \text{(B–103)}$$

The activity of water is closely related to the osmotic coefficient (ϕ):

$$\ln a_w = -\left(\frac{\sum m}{\Omega}\right)\phi \qquad \text{(B–104)}$$

The thermodynamic properties of water are often discussed in the physical chemistry literature in terms of the osmotic coefficient instead of the activity of water (or the mole fraction activity coefficient of water).

The Debye-Hückel model used in Pitzer's equations is not the usual Debye-Hückel-charging formulation exemplified in the Davies or B-dot equations, but a different one derived by Pitzer (1973) and called the Debye-Hückel-osmotic model. The relevant equations are:

$$f(I) = -\left(\frac{4A_\phi I}{b}\right)\ln(1 + b\sqrt{I}) \qquad \text{(B–105)}$$

$$f'(I) = -2A_\phi\left(\frac{2}{b}\ln(1 + b\sqrt{I}) + \frac{I}{(1 + b\sqrt{I})}\right) \qquad \text{(B–106)}$$

The Debye-Hückel parameter A_ϕ is related to the more familiar $A_{\gamma,10}$ by:

$$A_\phi = \frac{2.303\,A_{\gamma,10}}{3} \qquad \text{(B–107)}$$

The parameter b is assigned a constant value of 1.2 (Pitzer 1973). Theoretically, this is the product $\mathring{a}B_\gamma$; thus the hard core diameter at 25°C is effectively fixed at a value of about 3.65Å (and somewhat different values at other temperatures). Differences in the hard core diameters of various ions in solution are not explicitly accounted for (this is the case also in the Davies equation). However, the interaction coefficient terms of the equation effectively compensate for this. A very important feature of the Debye-Hückel-osmotic model is that it, like the Debye-Hückel-charging model, is consistent with the Debye-Hückel limiting law:

$$\log\gamma_i \rightarrow -A_{\gamma,10}z_i^2\sqrt{I} \quad \text{as} \quad I \rightarrow 0 \qquad \text{(B–108)}$$

B.2.5.2 Solutions of Electrolytes

In a pure solution of aqueous neutral electrolyte *MX*, the following combinations of interaction coefficients are observable:

$$B_{MX}(I) = \lambda_{MX}(I) + \frac{z_X}{2z_M}\lambda_{MM}(I) + \frac{z_M}{2z_X}\lambda_{XX}(I) \qquad \text{(B–109)}$$

$$C_{MX}^{\phi} = 3\left(\left|\frac{z_x}{z_M}\right|^{\frac{1}{2}}\mu_{MMX} + \left|\frac{z_M}{z_X}\right|^{\frac{1}{2}}\mu_{MXX}\right) \qquad \text{(B–110)}$$

For example, the osmotic coefficient for such a solution can be written in the form (Pitzer 1973):

$$\phi - 1 = \frac{|z_M z_X|}{2}\left(If'(I) - f(I)\right) + \left(\frac{2v_M v_X}{v_{MX}}\right) B^{\phi}_{MX}(I) m_{MX} + \left(\frac{2(v_M v_X)^{\frac{3}{2}}}{v_{MX}}\right) C^{\phi}_{MX} m^2_{MX} \quad \text{(B–111)}$$

Appearing in this equation is β^{ϕ}_{MX}, which is given by:

$$B^{\phi}_{MX}(I) = B_{MX}(I) + IB'_{MX}(I) \quad \text{(B–112)}$$

Here $B'_{MX}(I)$ is the derivative of $B_{MX}(I)$ with respect to the ionic strength.

The ionic strength dependence of B^{ϕ}_{mx} was defined by Pitzer (1973) to take the following form:

$$B^{\phi}_{MX}(I) = \beta^{(0)}_{MX} + \beta^{(1)}_{MX} \mathrm{e}^{-\alpha\sqrt{I}} \quad \text{(B–113)}$$

where α was assigned a constant value of 2.0. $\beta^{(0)}_{MX}$ and $\beta^{(1)}_{MX}$, along with C^{ϕ}_{MX}, are parameters whose values are determined by fitting experimental data, such as for the osmotic coefficient. Corresponding to the above equation is:

$$B_{MX}(I) = \beta^{(0)}_{MX} + \beta^{(1)}_{MX} g\left(\alpha\sqrt{I}\right) \quad \text{(B–114)}$$

where:

$$g(x) = \left(\frac{2}{x^2}\right)\left(1 - (1+x)\mathrm{e}^{-x}\right) \quad \text{(B–115)}$$

Pitzer and Mayorga (1974) proposed a description for β^{ϕ}_{MX} in the case of 2:2 electrolytes that is based on an additional fitting parameter:

$$\beta^{\phi}_{MX}(1) = \beta^{(0)}_{mx} + \beta^{(1)}_{mx^e}{}^{-a_1\sqrt{1}} + \beta^{(2)}_{MX^e}{}^{-a_2\sqrt{1}} \quad \text{(B–116)}$$

Here α_1 is assigned a value of 1.4 and α_2 one of 12.0 and $\beta^{(2)}_{MX}$ is the additional fitting parameter. Corresponding to this is:

$$\beta_{MX}(1) = \beta^{(0)}_{MX} + \beta^{(1)}_{MX} g(a_1\sqrt{I} + \beta^{(2)}_{MX} g(a_2\sqrt{I} \quad \text{(B–117)}$$

We consider first the exponential function in eqs (B–113) and (B–117). This is shown in Figure B–1 for the three commonly used values of α. At zero ionic strength, this function has a value of unity. Thus, $\beta^{\phi}_{MX} = \beta^{(0)}_{MX} + \beta^{(1)}_{MX}$ or $\beta^{\phi}_{MX} = \beta^{(0)}_{MX} + \beta^{(1)}_{MX} + \beta^{(2)}_{MX}$. The magnitude of each term containing $\beta^{(1)}_{MX}$ or $\beta^{(2)}_{MX}$ decreases exponentially as the ionic strength increases, approaching zero as the ionic strength approaches infinity (a limit which is not of physical interest). Most of the decay takes place in the very low ionic strength range. Thus, the terms in $\beta^{(1)}_{MX}$ and $\beta^{(1)}_{MX}$ are important parts of the model, even in dilute solutions.

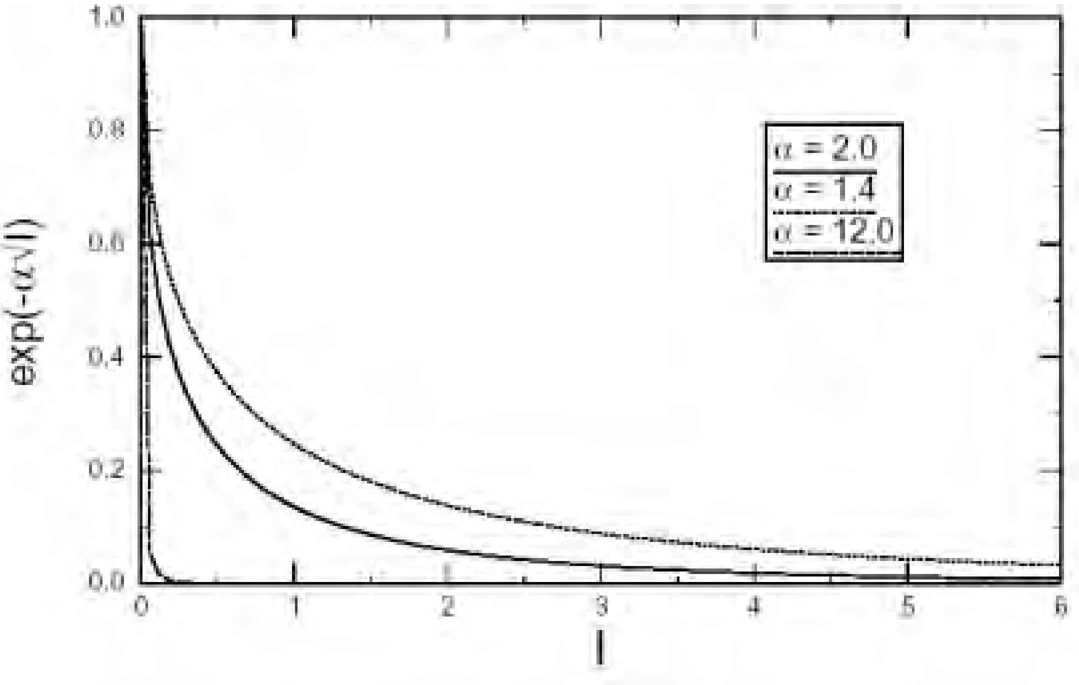

Figure B–1. Behavior of the exponential function governing the ionic strength dependence of second-order interactions among cations and anions.

The function *g(x)* is shown in Figure B–2 for the three commonly used α values. It resembles the above exponential function, though it does not decay quite so rapidly. This function may be expanded as follows:

$$g(x) = 1 - 2\left(\frac{2x}{3!} - \frac{3x}{4!} + \frac{4x^2}{5!} - \frac{5x^3}{6!} + \ldots\right) \qquad \text{(B–118)}$$

This shows that *g(x)* = 1 at *x* = 0 (*I* = 0). Thus, at zero ionic strength, $\beta_{MX} = \beta^{(0)}_{MX} + \beta^{(1)}_{MX}$ or $\beta_{MX} = \beta^{(0)}_{MX} + \beta^{(1)}_{MX} + \beta^{(2)}_{MX}$ It can be shown that *g(x)* approaches zero as *x* (and *I*) approach infinity.

The development thus far shows that there are two major categories of interaction coefficients. The λ_{ij} and the μ_{ijk} in terms of which the theoretical equations were originally derived are what we will call the primitive interaction coefficients. The observable combinations of these, such as

$\beta^{(0)}_{MX}$, $\beta^{(1)}_{MX}$, $\beta^{(2)}_{MX}$, and C^{ϕ}_{MX}, are what we will call the observable interaction coefficients. This latter kind of interaction coefficient represents the model data that are reported for the various systems for which Pitzer's equations have been fit to experimental data.

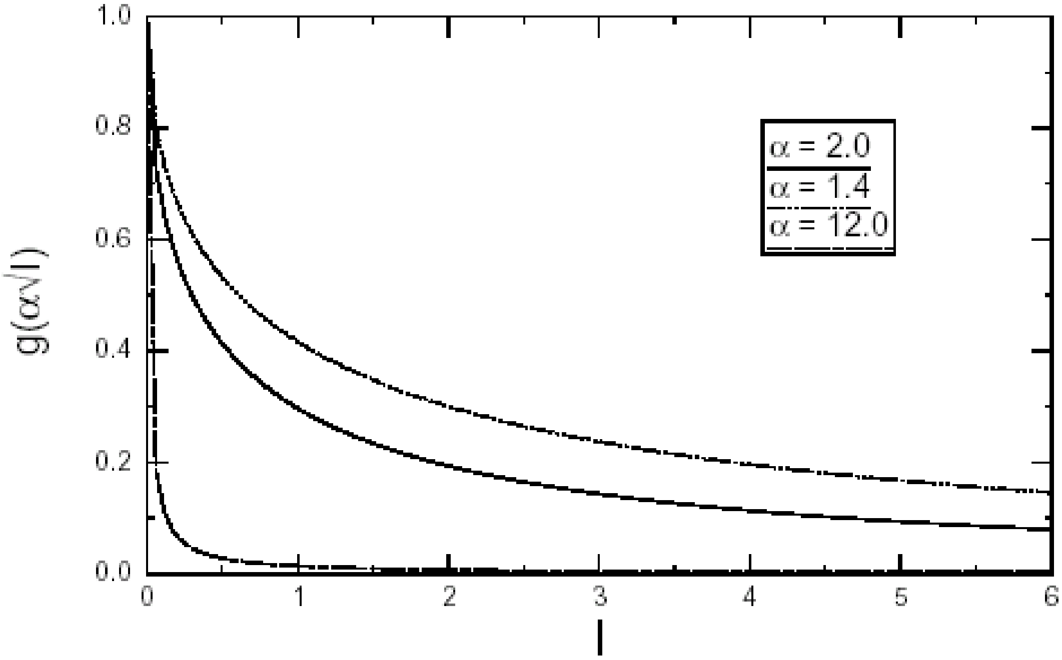

Figure B–2. Behavior of the g(x) function governing the ionic strength dependence of second-order interactions among cations and anions.

It is possible to rewrite the equations for *ln* γ_i and *ln* a_w in complex mixtures in terms of the observable interaction coefficients. An example of such equations was suggested by Pitzer (1979) and adopted with changes in notation by Harvie, Møller, and Weare (1984). These equations are much more complex than the original form written in terms of the primitive interaction coefficients. They have been incorporated into computer codes, such as that of Harvie, Møller, and Weare (1984), PHRQPITZ (Plummer et al. 1988), and SOLMINEQ.88 (Kharaka et al. 1988). As noted in the previous section, there is no unique way to construct equations for single-ion activity coefficients. Furthermore, direct usage of such equations constitutes implicit adoption of a corresponding *pH* scale. In the case of the single-ion activity coefficient equation suggested by Pitzer, this could be termed the "Pitzer" scale.

The equations for *ln* γ_i and *ln* a_w which are evaluated in EQ3/6 are those written in terms of the primitive interaction coefficients. The set of these which is used is not the generalized theoretical set, which is not obtainable for the reasons discussed previously, but a practical set that is obtained by mapping the set of reported observable interaction coefficients using a set of equations that contain arbitrary conventions. These mapping equations imply a *pH* scale. We will show that the conventions chosen here match those suggested by Pitzer (1979), so this implied *pH* scale is identical to his.

The basic guides to choosing such mapping conventions are pleasing symmetries and the desirability of minimizing the number of conventional primitive interaction coefficients with non-zero values. In the case of the second order coefficients, both of these considerations suggest the following definitions:

$$\lambda_{MM}(I) = 0 \qquad \text{(B–119)}$$

$$\lambda_{XX}(I) = 0 \qquad \text{(B–120)}$$

$$\lambda_{MX}(I) = B_{MX}(I) \qquad \text{(B–121)}$$

Analogous to the formulas used to describe β_{MX}, one may write:

$$\lambda_{MX}(I) = \lambda_{MX}^{(0)} + \lambda_{MX}^{(1)} g(\alpha \quad I) \qquad \text{(B–122)}$$

or:

$$\lambda_{MX}(I) = \lambda_{MX}^{(0)} + \lambda_{MX}^{(1)} g(\alpha_1 \quad I) + \lambda_{MX}^{(2)} g(\alpha_2 \quad I) \qquad \text{(B–123)}$$

From the principle of corresponding terms, it follows that the corresponding mapping equations are:

$$\lambda_{MM}^{(n)} = 0 \quad \text{for } n = 0,\ 2 \qquad \text{(B–124)}$$

$$\lambda_{XX}^{(n)} = 0 \quad \text{for } n = 0,\ 2 \qquad \text{(B–125)}$$

$$\lambda_{MX}^{(n)} = \beta_{MX}^{(n)} \quad \text{for } n = 0,\ 2 \qquad \text{(B–126)}$$

Evaluation of the equations for ln γ_i and ln a_w also requires the ionic strength derivatives of the λ_{ij} coefficients. These are given by:

$$\lambda'_{MX}(I) = \lambda_{MX}^{(1)} g'(x)\left(\frac{\alpha}{2 \quad I}\right) \qquad \text{(B–127)}$$

or:

$$\lambda'_{MX}(I) = \lambda_{MX}^{(1)} g'(x)\left(\frac{\alpha_1}{2 \quad I}\right) + \lambda_{MX}^{(2)} g'(x)\left(\frac{\alpha_2}{2 \quad I}\right) \qquad \text{(B–128)}$$

where $g'(x)$ is the derivative of $g(x)$ (with respect to x, not I), given by:

$$g'(x) = -\left(\frac{4}{x^3}\right)\left(1 - e^{-x}\left(1 + x + \frac{x^2}{2}\right)\right) \qquad \text{(B–129)}$$

The principle of pleasing symmetry suggests the following mapping equations for dealing with the C^{ϕ}_{MX} parameter:

$$\mu_{MMX} = \frac{1}{6}\left(\frac{z_M}{z_X}\right)^{\frac{1}{2}} C^{\phi}_{MX} \qquad \text{(B–130)}$$

$$\mu_{MXX} = \frac{1}{6}\left(\frac{z_X}{z_M}\right)^{\frac{1}{2}} C^{\phi}_{MX} \qquad \text{(B–131)}$$

The two μ coefficients are then related by:

$$\frac{\mu_{MMX}}{z_M} = \frac{\mu_{MXX}}{z_X} \qquad \text{(B–132)}$$

These are in fact the mapping equations used in EQ3/6. However, the principle of minimizing the number of conventional primitive interaction coefficients would suggest instead mapping relations such as:

$$\mu_{MMX} = \frac{1}{3}\left(\frac{z_M}{z_X}\right)^{\frac{1}{2}} C^{\phi}_{MX} \qquad \text{(B–133)}$$

$$\mu_{MXX} = 0 \qquad \text{(B–134)}$$

Note that with this set of mapping relations, a different pH scale would be implied.

In mixtures of aqueous electrolytes with a common ion, two additional observable combinations of interaction coefficients appear (Pitzer 1973; Pitzer and Kim 1974):

$$\theta_{MM'}(I) = \lambda_{MM'}(I) - \left(\frac{z_{M'}}{2z_M}\right)\lambda_{MM}(I) - \left(\frac{z_M}{2z_{M'}}\right)\lambda_{M'M'}(I) \qquad \text{(B–135)}$$

and:

$$\psi_{MM'X} = 6\mu_{MM'X} - \left(\frac{3z_{M'}}{z_M}\right)\mu_{MMX} - \left(\frac{3z_M}{z_{M'}}\right)\mu_{M'M'X} \tag{B–136}$$

Here *M* and *M'* are two cations and *X* is the anion, or *M* and *M'* are two anions and *X* is the cation. From previously adopted mapping conventions, it immediately follows that the corresponding mappings are given by:

$$\lambda_{MM'}(I) = \theta_{MM'}(I) \tag{B–137}$$

$$\mu_{MM'X} = \frac{1}{6}\left(\psi_{MM'X} + \left(\frac{3z_{M'}}{z_M}\right)\mu_{MMX} + \left(\frac{3z_M}{z_{M'}}\right)\mu_{M'M'X}\right) \tag{B–138}$$

In the original formulation of Pitzer's equations (Pitzer, 1973), the $\theta_{MM'}$ coefficient is treated as a constant. It was later modified by Pitzer (1975) to take the following form:

$$\theta_{MM'}(I) = {}^{S}\theta_{MM'} + {}^{E}\theta_{MM'}(I) \tag{B–139}$$

$\theta_{MM'}(I)$ corresponds to the Φ_{ij} of Harvie, Møller, and Weare (1984). The first term is a constant and accounts for short-range effects (this is the Φ_{ij} of Harvie, Møller, and Weare). The second term, which is the newer part, is entirely theoretical in nature and accounts for higher-order electrostatic effects. Only the $^{S}\theta_{MM'}$ part is obtained by fitting. Corresponding to this is the equation:

$$\lambda_{MM'}(I) = {}^{S}\lambda_{MM'} + {}^{E}\lambda_{MM'}(I) \tag{B–140}$$

The relevant mapping relation is then:

$$^{S}\lambda_{MM'} = {}^{S}\theta_{MM'} \tag{B–141}$$

The $^{E}\lambda_{MM'}(I)$ part is obtainable directly from theory (Pitzer, 1975):

$$^{E}\theta_{MM'} = \left(\frac{z_M z_{M'}}{4I}\right)\left(J(x_{MM'}) - \frac{J(x_{MM})}{2} - \frac{J(x_{M'M'})}{2}\right) \tag{B–142}$$

where:

$$J(x) = \frac{1}{x}\int_0^{\infty}\left(1 + q + \frac{q^2}{2} - e^q\right)y^2\,dy \tag{B–143}$$

in which:

$$q = -\left(\frac{x}{y}\right)e^{-y} \tag{B–144}$$

and:

$$x_{ij} = 6z_iz_jA^{\phi}\ \ I \tag{B–145}$$

The derivative of ${}^{E}\lambda_{MM'}(I)$ is given by:

$$\begin{aligned} {}^{E}\lambda'_{MM'}(I) &= -\left(\frac{{}^{E}\lambda(I)}{I}\right) \\ &+ \left(\frac{z_Mz_{M'}}{8I^2}\right)\left(x_{MM'}J'(x_{MM'}) - \frac{x_{MM}J'(x_{MM})}{2} - \frac{x_{M'M}J'(x_{M'M'})}{2}\right) \end{aligned} \tag{B–146}$$

Expansion of *J(x)* gives (Pitzer, 1975):

$$J(x) = -\left(\frac{x^2}{6}\right)(\ln x + 0.419711) + ... \tag{B–147}$$

Application of L'Hospital's rule shows that *J(x)* goes to zero as *x* goes to zero (hence also as the ionic strength goes to zero). *J(x)* is a monotonically increasing function. So is *J'(x)*, which approaches a limiting value of 0.25 as *x* goes to infinity. The function *J(x)* and its derivative are approximated in EQ3/6 by a Chebyshev polynomial method suggested by Harvie and Weare (1980). This method is also described in the review by Pitzer (1987, p. 131–132).

Pitzer (1979) showed that substitution of the observable interaction coefficients into the single-ion activity coefficient equation gives the following result for cation *M*:

$$\ln \gamma_M = z_M^2 f^\gamma + 2\sum_a m_a[B_{Ma} + (\Sigma mz)C_{Ma}] +$$

$$2\sum_c m_c\theta_{Mc} + \sum_c\sum_a m_c m_a[z_M^2 B'_{ca} + z_M C_{ca} + \psi_{Mca}] +$$

$$\frac{1}{2}\sum_a\sum_{a'} m_a m_{a'}[z_M^2\theta'_{aa'} + \psi_{Maa'}] + \frac{z_M^2}{2}\sum_c\sum_{c'} m_c m_{c'}\theta'_{cc'} +$$

$$z_M\left\{\sum_c \frac{m_c\lambda_{cc}}{z_c} - \sum_a \frac{m_a\lambda_{aa}}{z_a} + \frac{3}{2}\sum_c\sum_a m_c m_a\left(\frac{\mu_{cca}}{z_c} - \frac{\mu_{caa}}{z_a}\right)\right\} \tag{B–148}$$

Here *a* denotes anions, *c* denotes cations, and:

$$f^\gamma = \frac{f'}{2} \tag{B–149}$$

$$C_{MX} = \frac{C_{MX}^{\phi}}{2\sqrt{\left|{}^{z}M^{Z}fX\right|}} \tag{B–150}$$

$$\sum mz = \sum_c m_c m_c \tag{B–151}$$

(The single-ion equation for an anion is analogous). As pointed out by Pitzer, the unobservability of single-ion activity coefficients in his model lies entirely in the last term (the fourth line) of the equation and involves the primitive interaction coefficients λ_{cc}, λ_{aa}, μ_{cca}, and μ_{caa}. His suggested conventional single-ion activity coefficient equation is obtained by omitting this part. This requires the affected primitive interaction coefficients to be treated exactly as in the previously adopted mapping equations. This approach could in fact have been used to derive them.

In theory, the relevant data required to evaluate Pitzer's equations for complex mixtures of relatively strong aqueous electrolytes can all be obtained from measurements of the properties of pure aqueous electrolytes (giving the observable interaction coefficients $\beta_{MX}^{(0)}$, $\beta_{MX}^{(1)}$, $\beta_{MX}^{(2)}$, and C_{MX}^{ϕ}) and mixtures of two aqueous electrolytes having a common ion (${}^{S}\theta_{MM'}$ and $\psi_{MM'X}$). There is one peculiarity in this fitting scheme in that ${}^{S}\theta_{MM'}$ is obtainable from more than one mixture of two electrolytes having a common ion, because this parameter does not in theory depend on that ion. Thus, the value adopted may have to be arrived at by simultaneously considering the experimental data for a suite of such mixtures.

B.2.5.3 Solutions of Electrolytes and Nonelectrolytes

In general, it is necessary to consider the case of solutions containing nonelectrolyte solute species in addition to ionic species. Examples of such uncharged species include molecular species such as $O_{2(aq)}$, $CO_{2(aq)}$, $CH_{4(aq)}$, $H_2S_{(aq)}$, $C_2H_5OH_{(aq)}$, and $SiO_{2(aq)}$; strongly bound complexes, such as $HgCl_{3(aq)}$ and $UO_2CO_{3(aq)}$; and weakly bound ion pairs such as $CaCO_{3(aq)}$ and $CaSO_{4(aq)}$. The theoretical treatment of these kinds of uncharged species is basically the same. There are practical differences, however, in fitting the models to experimental data. This is simplest for the case of molecular neutral species. In the case of complexes or ion pairs, the models are complicated by the addition of corresponding mass action equations.

The treatment of solutions of electrolytes using Pitzer's equations is quite standardized. In such solutions, there is one generally accepted relation for describing single-ion activity coefficients, though it may be expressed in various equivalent forms. Thus, in such solutions there is only one implied "Pitzer" *pH* scale. Also, the set of parameters to be obtained by regressing experimental measurements is well established. Unfortunately, this is not the case for the treatment of solutions containing both electrolytes and nonelectrolytes.

Harvie, Møller, and Weare (1984) used Pitzer's equations to construct a model of all of the major components of seawater at 25°C. They modified the equations for electrolyte systems to include some provision for neutral species-ion interactions. Additional modification was made by Felmy and Weare (1986), who extended the Harvie, Møller, and Weare model to include borate as a component. The Felmy and Weare equation for the activity of water (obtained from their equation for the osmotic coefficient) is:

$$\begin{aligned} In\, a_w = -\frac{\sum m}{\Omega} - \frac{2}{\Omega}\Bigg(& \frac{If - f}{2} + \sum_c \sum_a m_c m_a \left(B_{ca}^{\phi} + ZC_{ca}\right) \\ & + \sum_c \sum_{c' > c} m_c m_{c'} \left(\Phi_{cc'}^{\phi} + \sum_a m_a \varphi_{adc}\right) \\ & + \sum_a \sum_{a' > a} m_a m_d \left(\Phi_{aa'}^{\phi} + \sum_c m_c \varphi_{cc'a}\right) \\ & + \sum_n \sum_c m_n m_c \lambda_{nc} + \sum_n \sum_a m_n m_a \lambda_{na} + \sum_n \sum_c \sum_a m_n m_c m_a \zeta_{nca} \Bigg) \end{aligned} \qquad \text{(B–152)}$$

In this equation, *c* denotes a cation and *a* an anion, and the following definitions are introduced:

$$Z = \sum_i z_i m_i \qquad \text{(B–153)}$$

$$\Phi_{ij}^{\phi} = {}^{S}\theta_{ij} + {}^{E}\theta_{ij}(I) + I\, {}^{E}\theta'_{ij}(I) \qquad \text{(B–154)}$$

The first three lines are equivalent to the mixture formulation given by Pitzer (1979). The fourth line (last three terms) is the new part. Here *n* denotes a neutral species, λ_{nc} and λ_{na} are second order interaction coefficients describing neutral species-ion interactions, and ζ_{nca} is an observable third order coefficient. These new interaction coefficients are treated as constants. The terms in λ_{nc} and λ_{na} were introduced by Harvie, Møller, and Weare (1984) to treat the species $CO_{2(aq)}$. The term in ζ_{nca} was put in by Felmy and Weare (1986) and is a third order interaction coefficient. It was necessary to include it in the equations to account for interactions involving the species $B(OH)_{3(aq)}$.

The corresponding single-ion equation for cation *M* takes the following form:

$$\begin{aligned} \ln \gamma_M = {} & z_M^2 F + \sum_a m_a [2B_{Ma} + ZC_{Ma}] \\ & + \sum_c m_c \left(2\Phi_{Mc} + \sum_a m_a \psi_{Mca} \right) \\ & + \sum_a \sum_{a'>a} m_a m_{a'} \psi_{Maa'} + z_M \sum_c \sum_a m_c m_a C_{ca} \\ & + 2\sum_n m_n \lambda_{nM} + \sum_n \sum_a m_n m_a \zeta_{naM} \end{aligned} \qquad \text{(B–155)}$$

Here Φ_{ij} is the θ_{ij} of the earlier notation and:

$$F = \frac{f'}{2} + \sum_c \sum_a m_c m_a B'_{ca} + \sum_c \sum_{c'>c} m_c m_{c'} \Phi'_{cc'} + \sum_a \sum_{a'>a} m_a m_{a'} \Phi'_{aa'} \qquad \text{(B–156)}$$

The first three lines are equivalent to Pitzer's suggested single-ion activity coefficient equation. The fourth line (last two terms) is the new part. The corresponding equation for anions is analogous. The corresponding equation for the N^{th} neutral species is:

$$\ln \gamma_N = 2\sum_c m_c \lambda_{Nc} + 2\sum_a m_a \lambda_{Na} + \sum_c \sum_a m_c m_a \zeta_{Nca} \qquad \text{(B–157)}$$

To deal with the fact that the λ_{nc} and λ_{na} are only observable in combination, Harvie, Møller, and Weare (1984) adopted the convention that:

$$\lambda_{N,\,H^+} = 0 \qquad \text{(B–158)}$$

These equations were presented for the modeling of specific systems, and are not completely general. They are missing some terms describing interactions involving neutral species. A set of complete equations is given by Clegg and Brimblecombe (1990). Their equation for the activity coefficient of a neutral solute species is:

$$
\begin{aligned}
\ln \gamma_N &= 2\sum_n m_n \lambda_{Nn} + 2\sum_c m_c \lambda_{Nc} + 2\sum_a m_a \lambda_{Na} \\
&+ 6\sum_n \sum_c m_n m_c \mu_{Nnc} + 6\sum_n \sum_a m_n m_a \mu_{Nna} \\
&+ 3\sum_c m_c^2 \mu_{Ncc} + 3\sum_a m_a^2 \mu_{Naa} + 6\sum_c \sum_a m_c m_a \mu_{Nca} \\
&+ 6\sum_n m_n^2 \mu_{Nnn} + 6\sum_{n \neq N} m_N m_n \mu_{NNn} + 6\sum_{n \neq N} \sum_{n' \neq N} m_n m_{n'} \mu_{Nnn'} \\
&+ 3\sum_n m_n^2 \mu_{Nnn} + 6\sum_{n \neq N} m_N m_n \mu_{NNn} + 6\sum_{n \neq N} \sum_{n' \neq N} m_n m_{n'} \mu_{Nnn'}
\end{aligned}
\tag{B–159}
$$

This is a complete and general representation of the activity coefficient of a neutral species in terms of all possible second order and third order primitive interaction coefficients. The first line of this equation contains the same terms in λ_{nc} and λ_{na} as appear in the Felmy-Weare equation. This line is augmented by an addition term which describes second order interactions among neutral species (and which was also pointed out by Pitzer 1987). The third line in this equation is equivalent to the term in ζ_{Nca} that appears in the Felmy-Weare equation. Clegg and Brimblecombe (1990) have pointed out that this observable interaction coefficient is related to the corresponding primitive interaction coefficients by the relation:

$$
\zeta_{\mathrm{NMX}} = 6\mu_{\mathrm{NMX}} + \frac{3\,z_{\mathrm{X}}}{z_{\mathrm{M}}}\mu_{\mathrm{NMM}} + \frac{3\,z_{\mathrm{M}}}{z_{\mathrm{X}}}\mu_{\mathrm{NXX}}
\tag{B–160}
$$

The second, fourth, and fifth lines consist of terms not found in the Felmy-Weare equation.

In a solution of a pure aqueous nonelectrolyte, the activity coefficient of the neutral species takes the form:

$$
\ln \gamma_{\mathrm{N}} = 2\mathrm{m}_{\mathrm{N}}\lambda_{\mathrm{NN}} + 3\mathrm{m}_{\mathrm{N}}^2\mu_{\mathrm{NNN}}
\tag{B–161}
$$

This activity coefficient is directly observable. Hence the two interaction coefficients on the right hand side are also observable. In a study of the solubility of aqueous ammonia, Clegg and Brimblecombe (1989) found that the term including μ_{NNN} was significant only for concentrations greater than 25 molal (a solution containing more ammonia than water). They therefore dropped this term and reported model results only in terms of λ_{NN}. Similarly, Barta and Bradley (1985) found no need for a μ_{NNN} term to explain the data for pure solutions of $CO_{2(aq)}$, $H_2S_{(aq)}$, and $CH_{4(aq)}$, and no such term was apparently required by Felmy and Weare (1986) to explain the data for $B(OH)_{3(aq)}$. Pitzer and Silvester (1976) report a significant μ_{NNN} term for undissociated phosphoric acid. This result now appears somewhat anomalous and has not been explained. The bulk of the available data, however, suggest that the μ_{NNN} term is generally insignificant in most systems of geochemical interest and can be ignored without loss of accuracy.

This result suggests that in more complex solutions, terms in $\lambda_{NN'}$, $\mu_{NNN'}$, $\mu_{N'N'N}$, and $\mu_{NN'N''}$ can also often be ignored. While there may be solutions in which the full complement of these terms are significant, one could argue that they must be so concentrated in nonelectrolyte components that they have little relevance to the study of surface waters and shallow crustal fluids (though some deep crustal fluids are rich in CO_2). Furthermore, one could argue that to address such solutions, it would be more appropriate to use a formalism based on a different kind of expansion than the one used in the present treatment (see Pabalan and Pitzer 1990).

In an aqueous solution consisting of one nonelectrolyte and one electrolyte, the activity coefficient of the neutral species takes the form:

$$\ln \gamma_N = 2m_N\lambda_{NN} + 2(m_M\lambda_{NM} + m_X\lambda_{NX})$$

$$+ 6m_N(m_M\mu_{NNM} + m_X\mu_{NNX}) + m_M m_X \zeta_{NMX} + 3m_N^2\mu_{NNN} \quad \text{(B–162)}$$

Three new terms appear. The resemblance of the term in λ_{NM} and λ_{NX} to a traditional Setchenow term has been pointed out by various workers (e.g., Felmy and Weare 1986; Pitzer 1987). Work reported by Clegg and Brimblecombe (1989, 1990) for a number of such systems containing ammonia showed that the most important of the three new terms were the second (λ_{NM}, λ_{NX}) term and the third (μ_{NNM}, μ_{NNX}) term. They defined these using the following conventions:

$$\lambda_{N,Cl^-} = 0 \quad \text{(B–163)}$$

$$\mu_{N,N,Cl^-} = 0 \quad \text{(B–164)}$$

Note that the first of these conventions conflicts with the corresponding convention adopted by Harvie, Møller, and Weare (1984), though it matches that proposed by Pitzer and Silvester (1976) in a study of the dissociation of phosphoric acid. a weak electrolyte. Clegg and Brimblecombe found that in one system, the use of the fourth (ζ_{NMX}) term was also required, though the contribution was relatively small. No use was required of the last (μ_{NNN}) term, as was shown by fitting the data for pure aqueous ammonia.

There seems to be some disagreement in the literature regarding the above picture of the relative significance of the (μ_{NNM}, μ_{NNX}) term versus that of the ζ_{NMX} term, although the seemingly contradictory results involve nonelectrolytes other than ammonia. We have noted above that Felmy and Weare (1986) used a ζ_{NMX} term to explain the behavior of boric acid-electrolyte mixtures. It is not clear if they considered the possibility of a (μ_{NNM}, μ_{NNX}) term. Pitzer and Silvester (1976) found no apparent need to include a (μ_{NNM}, μ_{NNX}) term or a ζ_{NMX} term to explain the thermodynamics of phosphoric acid dissociation in electrolyte solutions. The data on aqueous silica in electrolyte solutions of Chen and Marshall (1981), discussed by Pitzer (1987), require a ζ_{NMX} term, but no (μ_{NNM}, μ_{NNX}) term. A similar result was obtained by Barta and Bradley (1985) for mixtures of electrolytes with $CO_{2(aq)}$, $H_2S_{(aq)}$, and $CH_{4(aq)}$. Simonson et al. (1987) interpret data for mixtures of boric acid with sodium borate and sodium chloride and of boric acid with potassium borate and potassium chloride exclusively in terms of the first (λ_{NN}) and second (λ_{NM}, λ_{NX}) terms, using neither of the third order terms for nonelectrolyte-electrolyte interactions.

The (μ_{NNM}, μ_{NNX}) term can only be observed (and hence is only significant) when the concentrations of both the nonelectrolyte and the electrolyte are sufficiently high. In contrast, evaluating the ζ_{NMX} term requires data for high concentrations of the electrolyte, but low concentrations of the nonelectrolyte will suffice. Some nonelectrolytes, such as aqueous silica, are limited to low concentrations by solubility constraints. Thus, the results of Chen and Marshall (1981) noted by Pitzer (1987) are not surprising. In the case of more soluble nonelectrolytes, the range of the available experimental data could preclude the evaluation of the (μ_{NNM}, μ_{NNX}) term. This may be why Pitzer and Silvester (1976) reported no need for such a term to describe the data for mixtures of electrolytes with phosphoric acid and why Barta and Bradley (1985) found no need for such a term for similar mixtures of electrolytes with $CO_{2(aq)}$, $H_2S_{(aq)}$, and $CH_{4(aq)}$. The data analyzed by Felmy and Weare (1986) correspond to boric acid concentrations of about one molal, which may not be high to observe this term (or require its use). In the case of Simonson et al. (1987), who also looked at mixtures of electrolytes and boric acid, the need for no third order terms describing nonelectrolyte-electrolyte interactions is clearly due to the fact that the concentrations of boric acid were kept low to avoid the formation of polyborate species.

The equations for solutions containing nonelectrolytes can be considerably simplified if the model parameters are restricted to those pertaining to solutions of pure aqueous nonelectrolytes and mixtures of one nonelectrolyte and one electrolyte. This is analogous to the usual restriction in treating electrolyte solutions, in which the parameters are restricted to those pertaining to solutions of two electrolytes with a common ion. Furthermore, it seems appropriate as well to drop the terms in μ_{NNN}. The equation for the activity coefficient of a neutral electrolyte in electrolyte-nonelectrolyte mixtures then becomes:

$$\begin{aligned}\ln \gamma_N = {} & 2m_N \lambda_{NN} + 2\left(\sum_c m_c \lambda_{Nc} + \sum_a m_a \lambda_{Na}\right) \\ & + 6m_N\left(\sum_c m_c \mu_{NNc} + \sum_a m_a \mu_{NNa}\right) + \sum_c \sum_a m_c m_a \zeta_{Nca}\end{aligned} \qquad \text{(B–165)}$$

The reduction in complexity is substantial. In the context of using Pitzer’s equations in geochemical modeling codes, this level of complexity is probably quite adequate for dealing with nonelectrolytes in a wide range of application.

If a higher level of complexity is required, the next step is probably to add in terms in μ_{NNN} and $\lambda_{NN'}$. The first of these has been discussed previously and is obtained from data on pure aqueous nonelectrolytes. The second must be obtained from mixtures of two aqueous electrolytes (one could argue that this is also analogous to the treatment of electrolytes). This higher level of complexity may suffice to deal with at least some CO_2-rich deep crustal fluids and perhaps other fluids of interest in chemical engineering. However, an even higher level of complexity would probably be best addressed by a formalism based on an alternate expansion, as noted earlier.

The observability and mapping issues pertaining to the remaining parameters may be dealt with as follows. In the case of λ_{NN}, no mapping relation is required because this parameter is directly observable. The same is true of μ_{NNN} and $\lambda_{NN'}$, if the higher level of complexity is required.

The λ_{NM} and λ_{NX}, and μ_{NNM} and μ_{NNX}, are only observable in combinations, but can be dealt with by adopting the following respective conventions:

$$\lambda_{N,J} = 0 \tag{B–166}$$

$$\mu_{N,N,J} = 0 \tag{B–167}$$

where J is a reference ion ($J = H^+$ as suggested by Felmy and Weare 1986; $J = Cl^-$ as suggested by Pitzer and Silvester 1976, and Clegg and Brimblecombe 1989, 1990). In any data file used to support code calculations, the choice of reference ion must be consistent. This may require the recalculation of some published data.

The ζ_{NMX} parameter is observable and can be mapped into primitive form by adopting the following conventions:

$$\mu_{NMM} = 0 \tag{B–168}$$

$$\mu_{NXX} = 0 \tag{B–169}$$

$$\mu_{NMX} = \frac{\zeta_{NMX}}{6} \tag{B–170}$$

These relations are analogous to those defined for the C^{ϕ}_{MX} parameter.

The above conventions correspond well with the current literature on the subject. However, the treatment of the λ_{NM} and λ_{NX}, and μ_{NNM} and μ_{NNX}, though valid and functional, still stands out in that it is not analogous to, or a natural extension of, the conventions which have been universally adopted in the treatment of electrolyte solutions. The logical extension, of course, is to define observable interaction coefficients to represent the primitive coefficients which can only be observed in combination, and to then follow Pitzer (1979) in determining exactly which parts of the theoretical equations constitute the non-observable part. The conventions would then be defined so as to make these parts have zero value.

The suggested process can be shown to be consistent with the above mapping conventions for all the other coefficients treated above, including ζ_{NMX}. However, the process which worked so nicely for electrolytes fails to work for λ_{NM}-λ_{NX}, and μ_{NNM}-μ_{NNX}. We will demonstrate this for the case of the λ_{NM}-λ_{NX}. Application of the above equation to the case of an aqueous mixture of a neutral species (N) and a neutral electrolyte (MX) immediately shows that the corresponding observable combination of primitive interaction coefficients is given by:

$$\mathrm{L_{NMX}} = \mathrm{z_X}\,\lambda_{\mathrm{NM}} + \mathrm{z_M}\lambda_{\mathrm{NX}} \tag{B–171}$$

In such a system, the activity coefficient of the neutral species can be written as:

$$\ln \gamma_N = \frac{2}{z_M + z_X} \nu_{MX} L_{NMX} m_{MX} \tag{B–172}$$

In the manner of Pitzer (1979), one can show that the relevant term in the single-ion activity coefficient for cation *M* expands in the following manner:

$$2\sum_n m_n \lambda_{nM} = 2\sum_n m_n \frac{L_{nMX'}}{z_{X'}} - 2\sum_n m_n \frac{z_M}{z_{X'}} \lambda_{nX'} \tag{B–173}$$

where *X'* is some reference anion. When *X'* is Cl^-, we have the convention proposed by Pitzer and Silvester (1976) and followed by Clegg and Brimblecombe (1989, 1990). The first term on the right hand side is the relevant observable part; the second term is the non-observable part. Following the logic of Pitzer (1979), we could set the second term to zero. This would have the effect of defining the following mapping relations:

$$\lambda_{NX'} = 0 \tag{B–174}$$

$$\lambda_{NM} = \frac{L_{NMX'}}{z_{X'}} \tag{B–175}$$

Although this makes the relevant non-observable part vanish in the single-ion activity coefficient equation for all cations, it forces the complementary part in the corresponding equation for anions to not vanish, as we will now show. The relevant part of the anion equation gives the following analogous result:

$$2\sum_n m_n \lambda_{nX} = 2\sum_n m_n \frac{L_{nM'X}}{z_{M'}} - 2\sum_n m_n \frac{z_X}{z_{M'}} \lambda_{nM'} \tag{B–176}$$

where *M'* is some reference cation. As before, the second term on the right hand side is the non-observable part. Using the above mapping equation for λ_{NM}, this can be transformed to:

$$2\sum_n m_n \lambda_{nX} = 2\sum_n m_n \frac{L_{nM'X}}{z_{M'}} - 2\sum_n m_n \frac{z_X}{z_{M'} z_{X'}} L_{nM'X'} \tag{B–177}$$

Thus, under the conventions defined above, the non-observable part of the single-ion activity coefficient equation for anions does not vanish.

There are alternatives, but none are particularly outstanding. For example, one could reverse the situation and make analogous conventions so that the non-observable part of the anion equation vanishes, but then the non-observable part of the cation equation would not vanish. When *M'* is

H^+, we have the convention proposed by Felmy and Weare (1986). One could also try a symmetrical mapping, based on the following relation:

$$\lambda_{NM} = \frac{z_M}{z_X}\lambda_{NX} \tag{B–178}$$

This would lead to the following mapping relations:

$$\lambda_{NM} = \frac{z_M}{\left(z_M + |z_X|\right)} L_{NMX} \tag{B–179}$$

$$\lambda_{NX} = \frac{z_X}{\left(z_M + |z_X|\right)} L_{NMX} \tag{B–180}$$

Unfortunately, this would lead to a non-vanishing non-observable part in the equations for both cations and anions.

B.2.5.4 Temperature and Pressure Dependence

Pitzer's equations were originally developed and applied to conditions of 25 °C and atmospheric pressure (e.g., Pitzer and Kim, 1974). The formalism was subsequently applied both to activity coefficients under other conditions and also to related thermodynamic properties which reflect the temperature and pressure dependence of the activity coefficients (see the review by Pitzer 1987).

The first effort to extend the Pitzer formalism to high temperature was a detailed study of the properties of aqueous sodium chloride (Silvester and Pitzer 1977). In this study, the data were fit to a complex temperature function with up to 21 parameters per observable interaction coefficient and which appears not to have been applied to any other system. In general, the early efforts concerning the temperature dependence of the activity coefficients focused mainly on estimating the first derivatives of the observable interaction coefficient parameters with respect to temperature (e.g., Silvester and Pitzer 1978). The results of the more detailed study of sodium chloride by Silvester and Pitzer (1977; see their Figures 4, 5, and 6) suggest that these first derivatives provide an extrapolation that is reasonably accurate up to about 100°C.

In more recent work, the temperature dependence has been expressed in various studies by a variety of different temperature functions, most of which require only 5–7 parameters per observable interaction coefficient. Pabalan and Pitzer (1987) used such equations to develop a model for the system $Na\text{-}K\text{-}Mg\text{-}Cl\text{-}SO_4\text{-}OH\text{-}H_2O$ which appears to be generally valid up to about 200°C. Pabalan and Pitzer (1988) used equations of this type to built a model for the system $Na\text{-}Cl\text{-}SO_4\text{-}OH\text{-}H_2O$ that extends to 300°C. Greenberg and Møller (1989), using an elaborate compound temperature function, have constructed a model for the $Na\text{-}K\text{-}Ca\text{-}Cl\text{-}SO_4\text{-}H_2O$ system that is valid from 0-250°C. More recently, Spencer, Møller, and Weare (1990) have used a more compact equation to develop a model for the $Na\text{-}K\text{-}Ca\text{-}Mg\text{-}Cl\text{-}SO_4\text{-}H_2O$ system at temperatures in the range –60 to 25°C.

The pressure dependence of activity coefficients has also been looked at in the context of the Pitzer formalism. For descriptions of more recent work, see Kumar (1986), Connaughton, Millero and Pitzer (1989), and Monnin (1989).

B.2.5.5 Practical Aspects

In practice, the matter of obtaining values for the observable interaction coefficients is more complicated. Not all models based on Pitzer's equations are mutually consistent. Mixing reported data can lead to inconsistencies. For the most part, differences in reported values for the same coefficient are functions of the exact data chosen for use in the fitting process, not just whose data, but what kind or kinds of data as well. Some older reported values for the mixture parameters (e.g., Pitzer 1979) are based on fits not employing the $^{E}\theta_{MM'}$ formalism, which has become firmly entrenched in more recent work.

Some differences in the values of reported Pitzer parameters are due to minor differences in the values used for the A^{ϕ} Debye-Hückel parameter (e.g., 0.39 versus 0.392; see Plummer et al. 1988, p. 3, or Plummer and Parkhurst 1990). The general problem of minor discrepancies in this and other limiting law slope parameters has been looked at in some detail by Ananthaswamy and Atkinson (1984). Recently, Archer (1990) has also looked at this problem and proposed a method for adjusting reported Pitzer coefficients for minor changes in Debye-Hückel parameters without resorting to refitting the original experimental data.

There has also been some occasional modification of the basic activity coefficient equations themselves. For example, in treating the activity coefficients of alkali sulfate salts at high temperature, Holmes and Mesmer (1986ab) changed the recommended value of the α parameter from 2.0 to 1.4. Also Kodytek and Dolejs (1986) have proposed a more widespread usage of the $\beta_{MX}^{(2)}$ parameter, based on the empirical grounds that better fits can be obtained for some systems. The usage of this parameter was originally restricted to the treatment of 2:2 electrolytes (Pitzer and Mayorga 1973).

The formal treatment of speciation in the solutions (assumptions of which species are present) can also lead to different models. Association phenomena were first recognized in the Pitzer formalism in order to deal with phosphoric acid (Pitzer and Silvester, 1976) and sulfuric acid (Pitzer, Roy, and Silvester 1977). In general, ion pairs have been treated formally as non-existent. An exception is in the model of Harvie, Møller, and Weare (1984), who employ three ion pair species: $CaCO_{3(aq)}$, $MgCO_{3(aq)}$, and $MgOH^{+}$.

Components which form strong complexes have received relatively little attention in the Pitzer formalism, presumably because of the much greater experimental data requirements necessary to evaluate the greater number of parameters associated with the greater number of species. However, Millero and Byrne (1984) have used Pitzer's equations to develop a model of activity coefficients and the formation of lead chloro complexes in some concentrated electrolyte solutions. Huang (1989) has also recently looked at some examples of complex formation in the context of the Pitzer formalism. However, because strong complexing can not be represented even mathematically by the interaction coefficient formalism without taking explicit account of the associated chemical equilibria, and because such models are more difficult to develop, the practical application of the Pitzer formalism remains limited mostly to systems of relatively strong electrolytes, molecular nonelectrolytes, and a few weak nonelectrolytes.

B.2.5.6 Pitzer's Equations in EQ3/6: Current Status

The present treatment of Pitzer's equations in EQ3/6 is somewhat limited, particularly in regard to some of the advances that have been made with these equations in the past few years. These limitations have to do with the state of the existing data files which support the use of Pitzer's equations, the treatment of the temperature dependence of the interaction coefficients, and the treatment of neutral solute species.

The **hmw** data file is an implementation of the model of Harvie, Møller, and Weare (1984). This model is restricted to 25°C. The **pit** data file is based mostly on the data summarized by Pitzer (1979). These data include the first order temperature derivatives of the interaction coefficients. The nominal temperature range of this data file is 0–100°C. These data are not based on the currently universally accepted ${}^{E}\theta$ formalism introduced by Pitzer (1975).

EQ3/6 uses or ignores the ${}^{E}\theta$ formalism, depending on the value of a flag parameter on the data file. The temperature dependence, if any, is handled by using first and second order temperature derivatives of the interaction coefficients, which are expected for use at temperatures other than 25°C. The code permits a $\beta_{MX}^{(2)}$ parameter to be specified on the data file for any electrolyte. The α parameters are also provided on the data file for each electrolyte. Thus, non-standard values can be employed if desired.

The temperature dependence is presently limited to a representation in terms of a second-order Taylor's series in temperature. This requires the presence on the supporting data file of first and second temperature derivatives. No provision has yet been made for the more sophisticated representations proposed for example by Pabalan and Pitzer (1987) or Spencer, Møller, and Weare (1990).

EQ3/6 is presently quite limited in terms of the treatment of nonelectrolyte components by means of Pitzer's equations. This limitation is expressed in the structure of the data files and the mapping relations presently built into the EQPT data file preprocessor. These are presently set up to deal only with electrolyte parameters. However, it is possible to enter λ_{NN}, $\lambda_{NN'}$, λ_{NM}, and λ_{NX} parameters as though they were $\beta_{MX}^{(0)}$ parameters. The λ_{NM} and λ_{NX} parameters that are part of the model of Harvie, Møller, and Weare (1984) are included on the **hmw** data file in this manner.

The present version of EQPT can not handle the ζ_{MNX} interaction coefficient; however.

The means of storing and representing interaction coefficient data in EQ3/6 deserves some comment. There is a natural tendency to represent λ_{ij} by a two-dimensional array, and μ_{ijk} by a three-dimensional array. However, arrays of this type would be sparse (for example, $\lambda_{ij} = 0$ for many i, j). and many of the entries would be duplicates of others ($\lambda_I = \lambda_{ji}$, etc.). Therefore, the λ_{ij} are represented instead by three parallel one-dimensional arrays. The first contains the λ_{ij} values themselves, the second contains indices identifying the i-th species, and the third identifies the j-th species. The treatment is analogous for μ_{ijk}, which only requires an additional array to identify the k-th species. These arrays are constructed from data listed on the data0 data files.

Coefficients which must be zero by virtue of the mapping relations or other conventions are not included in the constructed arrays. Also, the storage scheme treats for example λ_{ij} and λ_{ji} as one coefficient, not two.

B.3 ACTIVITY COEFFICIENTS OF SOLID SOLUTION COMPONENTS

B.3.1 INTRODUCTION

The thermodynamic activities (a_i) of solid solution components are always defined on the basis of mole fractions. Thus, they can be described by the product of their mole fractions (x_i) and their rational (mole fraction) activity coefficients (λ_i):

$$a_i = x_i \lambda_i \tag{B–181}$$

The same treatment is typically applied to all components in non-aqueous liquid phases. It is also applied to water in aqueous solutions (cf. B.2).

Mole fraction ideality is the reference ideality when dealing with solid solutions. Therefore, the corresponding excess Gibbs energy is G^{EXx} (see B.2). The relevant differential equation linking this with the mole fraction activity coefficients is:

$$\ln \lambda_i = \frac{1}{RT} \frac{\partial G^{EXx}}{\partial n_i} \tag{B–182}$$

where R is the gas constant and T the absolute temperature. Given an expression for the excess Gibbs energy, this equation gives a guaranteed route to thermodynamically consistent results (cf. Wolery 1990).

Problems involving the thermodynamic consistency of activity coefficients in non-aqueous phases seem to be uncommon. However, consistency may be tested using various relations, such as the following form of the cross-differentiation rule (cf. Wolery 1990):

$$\frac{\partial \ln \lambda_j}{\partial n_i} = \frac{\partial \ln \lambda_i}{\partial n_j} \tag{B–183}$$

The issue of sufficiency in proving consistency using this and related equations (Gibbs-Duhem equations and sum rules) is addressed by Wolery (1990).

In most speciation-solubility calculations, the activity coefficients of solid solution components only affect the corresponding calculated saturation indices; they do not change the model of the aqueous solution itself (i.e., the speciation). However, if an equilibrium relation involving a solid solution phase is used as a constraint in defining a speciation-solubility problem, all of the model results may be affected by the choice of activity coefficient model. The results may similarly affected when such a constraint is used in mass transfer calculations, including reaction path calculations.

Mixing tends to stabilize a solid solution relative to its end-member components. Thus, an aqueous solution may be supersaturated with respect to a solid solution, yet undersaturated with respect to each of the pure end members. Consequently, a solid solution may form in a system in which some or none of the pure end members would form. This effect is true in the ideal case, in which the activity coefficient has a value of unity. If the activity coefficient is less than unity, this stabilizing effect is increased. If it is greater than unity, it is lessened.

In general, the activity coefficient of a solid solution component depends on the composition of the solid solution. This is in turn is normally expressed in terms of the mole fractions of the components. In order to use an equilibrium constraint involving a solid solution component in a speciation-solubility calculation, the user must provide this composition in order to allow calculation of the activity coefficient of the component involved. In mass transfer calculations involving solid solutions in equilibrium with an aqueous solution, the solid solution composition is itself a subset of the unknowns to be calculated. In speciation-solubility and other kinds of equilibrium calculations, it is necessary to calculate a saturation index for a solid solution which is not presumed to be in equilibrium with the aqueous phase. This presents a problem, because no composition is defined. This is solved in EQ3/6 by finding the composition which maximizes the computed saturation index (Bourcier 1985, 1989).

In liquid solutions, the solutes may mix over the whole volume of the solution. This type of mixing is sometimes referred to as molecular mixing. It is commonly applied to non-aqueous liquids, such as a solution composed of hydrocarbons. In the ideal case, the activity coefficient of each component is unity. In aqueous solutions and other solutions involving a solvent with a high dielectric constant, this concept is modified to account for ionic dissociation. The concept of molecular mixing has been applied many times to solid solutions (cf. the examples presented later in this section), and is predicated on the use of end-member components (for example, calcite [$CaCO_3$] and magnesite [$MgCO_3$] in magnesian calcite [$(Ca,Mg)CO_3$]). The activity coefficients of these components in the non-ideal case are then described using interaction coefficients more or less resembling those used in Pitzer's equations to describe the activity coefficients of aqueous species.

In crystalline solids, mixing tends rather strongly to occur over well-defined sites in the crystal structure (see for example Nordstrom and Munoz 1985). Some ions may mix over more than one kind of site. Vacancies may be involved in the mixing process. They may be created or destroyed by substitutions of one ion for another of different electrical charge. Mixing which takes account of such effects is referred to as site mixing. In site-mixing models, the concept of ideality is modified from that appropriate to molecular mixing, though still based on the mole fractions of components. It is possible to utilize as the components species such as ions, vacancies, and framework moieties instead of end members. However, the more common practice is to continue using end-member components. This is followed in the present version of EQ3/6. However, the activity coefficient of an end-member component in an ideal site-mixing model may have a value other than unity. A site-mixing model will appear to be ideal in this sense only if there is only one site, an ion substitutes for others of the same charge type, and vacancies are not present on the site. Site mixing then effectively reduces to molecular mixing.

Nearly all of the site-mixing models that have been proposed for the various solid solutions are ideal in the site-mixing sense (see for example Viani and Bruton 1992). The only parameters of such models are site-mixing parameters. It is possible to consider site-mixing models that are non-ideal even in the site-mixing sense. These would be described by both site-mixing parameters and interaction coefficients. No models of this type are presently treated in EQ3/6.

In EQ3/6, all solid solution models are defined on the supporting data file. At present, only the **com** file contains any solid solutions. All of these are treated with ideal site-mixing models (the exception being olivine, which is treated according to a binary regular solution model). The actual types of models used on the data file are defined by the **jsol** flag array. The corresponding parameters (site-mixing parameters, interaction coefficients, and parameters used to compute interaction coefficients) are stored on the data file as an array. The elements of this array are represented below as $p_{k\psi}$. In EQ3NR and EQ6, solid solutions are presently ignored unless the option switch **iopt(4)** (Section 3.3.15) is set to a value greater than or equal to 1.

The various models presently treated in EQ3/6 are discussed in the following sections. To avoid confusion, we will write the activity, mole fraction, activity coefficient, and related parameters of a solid solution component with a "$\sigma\psi$" subscript in place of "*i*". Here σ denotes the component itself (takes the place of "*i*"), and ψ the solid solution (in order to be completely explicit about which solid solution is being addressed).

B.3.2 IDEAL SOLUTION, WITH ONE OPTIONAL SITE-MIXING PARAMETER

The first activity coefficient model for solid solutions in EQ3/6 is for an ideal solution in either the molecular-mixing sense or a limited site-mixing model in which mixing is confined to one site and vacancies are ignored. The former is a special case of the latter. This model corresponds to **jsol** = 1 and is characterized by the equation (Viani and Bruton 1992):

$$a_{\sigma\Psi} = x_{\sigma\Psi}^{N_\Psi} \qquad \text{(B–184)}$$

where N_ψ is the site mixing parameter. This formulation is equivalent to:

$$\log \lambda_{\sigma\Psi} = (N_\Psi - 1) \log x_{\sigma\Psi} \qquad \text{(B–185)}$$

If N_ψ = 1, the above model is mathematically equivalent to an ideal molecular-mixing model ($\log \lambda_{\sigma\varphi} = 0$).

The N_ψ parameter is stoichiometric in nature. In essence, it is the number of formula units of the site on which mixing occurs per formula unit of the solid solution framework. In principle, the formula for all the end-member components of a solid solution can be written so as to yield N_ψ = 1, hence $\log \lambda_{\sigma\varphi} = 0$.

In the case of heterovalent single-site solid solutions such as clays and zeolites, vacancies are involved. In order to simplify the solution model and preserve the simple relationship defined by eqs (B–184) and (B–185), Viani and Bruton (1992) have chosen to treat such solid solutions according to a model in which the mixing entities are ions or ion-vacancy complexes. Thus, two sodium ion entities might mix with a calcium ion-vacancy entity.

The N_ψ parameter is obtained from the parameters read from the data file according to:

$$N_\Psi = p_{7\Psi} \tag{B–186}$$

B.3.3 THIRD-ORDER MACLAURIN MODEL FOR A BINARY SOLUTION

The third-order Maclaurin model for a binary solution corresponds to **jsol** = 2. It is taken from Helgeson et al. (1970). The activity coefficients of the two end-member components are given by:

$$\log \lambda_{1\psi} = \frac{1}{2.303RT}\left[-\left(\frac{W_{2\psi}}{2}\right)x_2^2 - \left(\frac{W_{3\psi}}{3}\right)x_2^3\right] \tag{B–187}$$

$$\log \lambda_{2\psi} = \frac{1}{2.303RT}\left[-\left(\frac{W_{2\psi}+W_{3\psi}}{2}\right)x_1^2 - \left(\frac{W_{3\psi}}{3}\right)x_1^3 + \left(W_{1\psi} + \frac{W_{2\psi}}{2} + \frac{W_{3\psi}}{6}\right)\right] \tag{B–188}$$

Here $W_{1\psi}$, $W_{2\psi}$, and $W_{3\psi}$ are interaction coefficients. There are no site-mixing parameters.
The formulation represented by eqs (B–187) and (B–188) is highly unsymmetrical. In order to satisfy the condition that $\log \lambda_{2\psi} \rightarrow 0$ as $x_1 \rightarrow 0$, the interaction coefficients are required to satisfy the relation:

$$W_{1\psi} = -\frac{W_{2\psi}}{2} - \frac{W_{3\psi}}{6} \tag{B–189}$$

The interaction coefficients are obtained from the parameters read from the data file according to:

$$W_{1\psi} = p_{1\psi} \tag{B–190}$$

$$W_{2\psi} = p_{2\psi} \tag{B–192}$$

$$W_{3\psi} = p_{3\psi} \tag{B–191}$$

However, $W_{1\psi}$ is actually recalculated using eq (B–189).

B.3.4 REGULAR SOLUTION MODEL FOR A BINARY SOLUTION

The regular solution model for a binary solution corresponds to **jsol** = 3. It is also called a parabolic Maclaurin model. For a discussion of this model, see Saxena (1973, p. 11–12). The activity coefficients of the two end-member components are given by:

$$\log \lambda_{1\psi} = \frac{1}{2.303RT} W_\psi x_2^2 \tag{B–193}$$

$$\log \lambda_{2\psi} = \frac{1}{2.303RT} W_{\psi} x_1^2 \qquad \text{(B–194)}$$

Here W_{ψ} is the single interaction coefficient. There are no site-mixing parameters. This formulation is symmetrical.

The interaction coefficients are obtained from the parameters read from the data file according to:

$$W_{\psi} = p_{1\psi} + p_{2\psi}T + p_{3\psi}P \qquad \text{(B–195)}$$

Thus, the interaction coefficient in this model can be treated as a function of temperature and pressure. On the **com** data file in the R10 and R16 sets, there is a regular solution model for the solid solution olivine. The $p_{2\psi}$ and $p_{3\psi}$ parameters are set to zero, so the interaction coefficient is actually treated as a constant. A non-unit site-mixing parameter is also given in the $p_{7\psi}$ parameter, but this is not used.

B.3.5 CUBIC MACLAURIN MODEL FOR A BINARY SOLUTION

The cubic Maclaurin model for a binary solution corresponds to **jsol** = 4. For a discussion of this model, see Saxena (1973, p. 16). The activity coefficients of the two end-member components are given by:

$$\log\lambda_{1\psi} = \frac{1}{2.303\mathrm{RT}}[(2\mathrm{W}_{2\psi} - \mathrm{W}_{1\psi})\mathrm{x}_2^2 + 2(\mathrm{W}_{1\psi} - \mathrm{W}_{2\psi})\mathrm{x}_2^3] \qquad \text{(B–196)}$$

$$\log\lambda_{2\psi} = \frac{1}{2.303\mathrm{RT}}[(2\mathrm{W}_{1\psi} - \mathrm{W}_{2\psi})\mathrm{x}_1^2 + 2(\mathrm{W}_{2\psi} - \mathrm{W}_{1\psi})\mathrm{x}_1^3] \qquad \text{(B–197)}$$

Here $W_{1\psi}$, and $W_{2\psi}$ are interaction coefficients. There are no site-mixing parameters. This formulation is asymmetrical.

The interaction coefficients are obtained from the parameters read from the data file according to:

$$\mathrm{W}_{1\psi} = \mathrm{p}_{1\psi} + \mathrm{p}_{2\psi}\mathrm{T} + \mathrm{p}_{3\psi}\mathrm{P} \qquad \text{(B–198)}$$

$$\mathrm{W}_{2\psi} = \mathrm{p}_{4\psi} + \mathrm{p}_{5\psi}\mathrm{T} + \mathrm{p}_{6\psi}\mathrm{P} \qquad \text{(B–199)}$$

B.3.6 GUGGENHEIM POLYNOMIAL MODEL FOR A BINARY SOLUTION

The Guggenheim polynomial model for a binary solution corresponds to **jsol** = 5. For a discussion of this model, see Saxena (1973, p. 14–15). The activity coefficients of the two end-member components are given by:

$$\log \lambda_{1\psi} = \frac{1}{2.303RT}[(W_{1\psi} + 3W_{2\psi} + 5W_{3\psi})x_2^2 + (-4W_{2\psi} - 16W_{3\psi})x_2^3 + 12W_{3\psi}x_2^4] \quad \text{(B–200)}$$

$$\log \lambda_{2\psi} = \frac{1}{2.303RT}[(W_{1\psi} - 3W_{2\psi} + 5W_{3\psi})x_1^2 + (4W_{2\psi} - 16W_{3\psi})x_1^3 + 12W_{3\psi}x_1^4] \quad \text{(B–201)}$$

Here $W_{1\psi}$, $W_{2\psi}$, and $W_{3\psi}$ are interaction coefficients. There are no site-mixing parameters. This formulation is asymmetrical.

The interaction coefficients are obtained from the parameters read from the data file according to:

$$W_{1\psi} = p_{1\psi} + p_{2\psi}T + p_{3\psi}T^2 \quad \text{(B–202)}$$

$$W_{2\psi} = p_{4\psi} + p_{5\psi}T + p_{6\psi}T^2 \quad \text{(B–203)}$$

$$W_{3\psi} = p_{7\psi} + p_{8\psi}T + p_{9\psi}T^2 \quad \text{(B–204)}$$

The full form of this model can be used in the present version of EQ3/6, although the parameters $p_{k\psi}$ for $k \geq 7$ are now intended to be reserved for site-mixing parameters.

B.3.7REGULAR SOLUTION MODEL FOR A TERNARY SOLUTION

The regular solution model for a ternary solution corresponds to **jsol** = 6. The activity coefficients of the three end-member components are given by:

$$\log \lambda_{1_\psi} = \frac{1}{2.303RT}[W_{12_\psi} x_2^2 + W_{13_\psi} x_3^2 + (W_{12_\psi} - W_{23_\psi} + W_{13_\psi}) x_2 x_3] \qquad \text{(B–205)}$$

$$\log \lambda_{2_\psi} = \frac{1}{2.303RT}[W_{12_\psi} x_1^2 + W_{23_\psi} x_3^2 + (W_{12_\psi} - W_{13_\psi} + W_{23_\psi}) x_1 x_3]$$

$$\log \lambda_{3_\psi} = \frac{1}{2.303RT}[W_{13_\psi} x_1^2 + W_{23_\psi} x_2^2 + (W_{13_\psi} - W_{12_\psi} + W_{23_\psi}) x_1 x_2] \qquad \text{(B–207)}$$

Here $W_{1\psi}$, $W_{2\psi}$, and $W_{3\psi}$ are interaction coefficients. There are no site-mixing parameters. This formulation is symmetrical.

The interaction coefficients are obtained from the parameters read from the data file according to:

$$W_{12_\psi} = p_{1_\psi} \qquad \text{(B–208)}$$

$$W_{13_\psi} = p_{2_\psi} \qquad \text{(B–209)}$$

$$W_{23_\psi} = p_{3_\psi} \qquad \text{(B–210)}$$

B.4 SOLVING THE GOVERNING EQUATIONS

B.4.1INTRODUCTION

The governing equations that apply to speciation-solubility modeling were introduced in Sections B.1, B.2, and B.3. The purpose of this section is to derive the mathematics necessary to solve them. The approach is to set up the problem in terms of *n* equations in *n* unknowns (or "iteration variables") and solve them. Technically, there are a large number of equations and corresponding unknowns to deal with. The unknowns include the concentrations of the all the species appearing in the model and their thermodynamic activity coefficients. The corresponding equations are algebraic, and these must be solved using appropriate methods.

In EQ3NR, the set of unknowns is first reduced to a relatively small set of unknowns, from which the remaining unknowns can be calculated. These are the primary iteration variables. They are defined in this code as the log concentrations of the species in the active basis set. The algebraic equations are solved by a combination of two iterative methods which are applied in sequence. The first method, called pre-Newton-Raphson optimization, has the characteristic of rapid convergence far from the solution, and slow (limiting first order) convergence near the solution. It is used primarily to get all of the primary iteration variables within an order of magnitude of the solution. The second method, a hybrid Newton-Raphson method, has the characteristic of poor convergence behavior far from the solution, and very fast (limiting second order) convergence near the solution. These methods thus complement one another. We will discuss these, as well as supplementary methods designed to aid convergence. Lastly, we will briefly discuss the subject of crash diagnostics.

B.4.2THE SET OF MASTER ITERATION VARIABLES

In the EQ3NR code, the number of equations and unknowns is reduced by substituting all governing aqueous mass action equations into the mass balance and electrical balance equations. The remaining aqueous species giving rise to unknowns then comprise the relatively small active basis set. These master iteration variables reside in the vector $\underline{z}$, which has the following structure:

$$\underline{z} = \begin{bmatrix} \log x_w \\ . \\ . \\ \log m_{s'}, \; s' = 1 . s \neq w \\ . \\ . \\ \log f_{O_2} \\ . \\ . \\ \log m_{s'}, s' = s_B + 1, s_Q \end{bmatrix} \qquad \text{(B–211)}$$

The first part of this vector contains entries for the s_B strict basis species appearing in a given problem. The second part contains any auxiliary basis variables which appear in the problem and for which the **jflag** string is not set to *Make non-basis* (refer to Section 3.3.9). The structure is further simplified if $s_Q = s_B$ (no active auxiliary basis set). For a problem in which this is the case, it is implied that the aqueous solution is in a state of complete internal (homogeneous) equilibrium. If $s_Q = s_B + 1$, complete internal equilibrium is also implied if the $s_Q{}^{\text{th}}$ species is part of a redox couple used to define the redox state of the fluid (the **irdxc3** = 1, *Couple (aux. sp.)* option, Section 3.3.8). In all other cases, partial internal disequilibrium is implied.

We will first look at the aqueous mass action equations that are to be eliminated. Here s'' denotes a non-basis species and r denotes the corresponding reaction. As index labels, these are related according to:

$$r = s'' - s_B \tag{B–212}$$

This reflects the fact that strict basis species appear first in the list of all aqueous species and that they have no associated reactions. A mass action equations takes the following logarithmic form:

$$\log K_r = b_{wr}(\log x_w + \log \lambda_w) = b_{s_B r} \log f_{O_2} + b_{s''r}\left(\log m_{s''} + \log \gamma_{s''}\right)$$
$$\sum_{\substack{s'\ 1 \\ s' \neq w, s_B}}^{s_Q} b_{s'r}\left(\log m_{s'} = \log \gamma_{s'}\right) \tag{B–213}$$

where K_r is the thermodynamic equilibrium constant for the reaction b_{sr} is the reaction coefficient for the s-th species, λ_w is the activity coefficient of water, and γ_s ($s \neq w$) is the molal activity coefficient of the s-th species. Note that s' implies a basis species, and that s'' denotes the only non-basis species appearing in the reaction. This equation can be rearranged to give:

$$\log m_{s''} = \frac{\log Kr}{b_{s''r}} - \log \gamma_{s''} - \frac{b_{wr}}{b_{s''r}}\left(\log x_w + \log \lambda_w\right) - \frac{b_{s_B r}}{b_{s''r}} \log f_{O_2}$$
$$- \sum_{\substack{s'=1 \\ s' \neq w, s_B}}^{s_Q} \frac{b_{s'r}}{b_{s''r}}\left(\log m_{s'} + \log \gamma_{s'}\right) \tag{B–214}$$

Recall the relation:

$$\frac{dx}{d \log x} = 2.303\, x \tag{B–215}$$

It follows that:

$$\frac{dm_{s''}}{dx} = 2.303 m_{s''} \frac{d \log m_{s''}}{dx} \tag{B–216}$$

Treating the activity coefficients as constants, this can be used to show that:

$$\frac{\partial m_{s''}}{\partial \log x_w} = -\, 2.303\, m_{s''} \frac{b_{wr}}{b_{s'r}} \tag{B–217}$$

$$\frac{\partial m_{s''}}{\partial \log f_{O_2}} = -\, 2.303\, m_{s''} \frac{b_{s_B r}}{b_{s''r}} \tag{B–218}$$

$$\frac{\partial m_{s''}}{\partial \log m_{s'}} = -\, 2.303\, m_{s''} \frac{b_{s'r}}{b_{s''r}}, s' = 1, s_Q\ , s' \neq w \tag{B–219}$$

These relations will be used later in this section in deriving the Jacobian matrix elements corresponding to the mass and charge balance residual functions. This matrix is used in Newton-Raphson iteration.

Certain factors will appear repeatedly in some of the derivations below and will be given special symbols. In the EQ3NR code, these parameters are themselves evaluated before the calculation of the Jacobian matrix elements in which they appear. This is done to avoid repetitive arithmetic in the code. These are defined as follows:

$$H_{s'r} = \frac{m_{s''} u_{s''s'}}{b_{s''r}}\ ,\ s' = 1, s_Q\ , s' \neq w \tag{B–220}$$

$$H_{zr} = \frac{m_{s''} z_{s''}}{b_{s''r}} \tag{B–221}$$

All of the so-called alternative constraints involve equations which are written only in terms of the concentrations (or activities) of species in the active basis set. Therefore, it is not necessary to make any substitutions of the sort noted above in dealing with the mass balance equations and the charge balance equation.

B.4.3 EXPANDING THE SYSTEM FROM THE SET OF MASTER ITERATION VARIABLES

If one knows the vector $\underline{z}$, one may “expand the system” by computing the concentrations of all non-basis species (all species not in the active basis set; this includes any auxiliary basis species with **jflag** defined as *Make non-basis;* see Section 3.3.9) and the activity coefficients of all species in solution. The $\underline{z}$ vector uniquely defines all system properties. However, the process of expanding the system is not exactly straightforward. In order to calculate the concentrations of the non-basis species, one must evaluate the corresponding mass action equations. The activity coefficients appear in these equations. Hence, the activity coefficients must be evaluated first. However, the activity coefficients depend in general on the concentrations of all solute species, both basis and non-basis. So to deal with these, one must compute the concentrations of the non-basis species first.

This creates a problem analogous to the old puzzle, “Which came first, the chicken or the egg?” In older versions of EQ3/6 (e.g., Wolery 1983), this problem was overcome by treating the ionic strength as a master iteration variable. The equation defining the ionic strength has the same form as a mass balance equation, and this equation was treated in like form. However, this only works if the activity coefficient model depends only on the ionic strength, not the specific composition of the solution. Thus, this approach works for the Davies equation and the B-dot equation, but not for Pitzer’s equations or any other set of equations likely to be valid in concentrated solutions.

The concentrations of non-basis species may vary over many orders of magnitude. The activity coefficients of aqueous species generally vary over about two orders of magnitude or less. The approach taken in EQ3/6 is to start by computing a set of reasonable values for the activity coefficients, then hold these constant until the concentrations of the non-basis species become reasonably stable. The activity coefficients are then updated. In the pre-Newton-Raphson optimization algorithm, the computed concentrations of the basis and non-basis species are typically adjusted 3–7 times before the activity coefficients are recalculated. In the hybrid Newton-Raphson method, they are recalculated between each Newton-Raphson step (it is because of this treatment that we refer to our usage of the Newton-Raphson method as a hybrid). In either case, the system is expanded by first calculating the new concentrations of the non-basis species, using the existing values of the activity coefficients. The activity coefficients are then recalculated. An exception to this order occurs when starting values are constructed (see below).

One could view the expansion itself as an iterative process. One could recalculate the concentrations of the non-basis species, recalculate the activity coefficients, and then repeat the process one or more times. In the context of the pre-Newton-Raphson optimization method, this makes little sense because this method is only used to get in the neighborhood of the solution. The merits of a single update, a double update, and a multiple update (repeating the process until a convergence tolerance is satisfied) were examined in the context of the hybrid Newton-Raphson method. Significant differences in performance were only observed in the case of highly concentrated electrolyte solutions. Thus, in the case of less concentrated solutions, the single update method was best because it gave the same performance for the lowest cost. In the case of the more concentrated solutions, it was found that the single update also gave the best performance, followed by the multiple update method. In such solutions, the double update method often led to failure to converge. The method which has been adopted, therefore, is the single update method.

B.4.4BEGINNING THE PROCESS: COMPUTING STARTING VALUES

The whole process must begin by assigning starting values. Initially, this is done as follows. For every basis species, their **jflag** string is assigned such that either the total or free molality, or the log activity is known (explicitly or by conversion). The available **jflag** options are reiterated here from the input files and Section 3.3.9:

```
* Valid jflag strings (ujf3(jflgi(n))) are:                                *
*    Suppressed          Molality            Molarity                      *
*    mg/L                mg/kg.sol           Alk., eq/kg.H2O               *
*    Alk., eq/L          Alk., eq/kg.sol     Alk., mg/L CaCO3              *
*    Alk., mg/L HCO3-    Log activity        Log act combo                 *
*    Log mean act        pX                  pH                            *
*    pHCl                Hetero. equil.      Homo. equil.                  *
*    Make non-basis                                                        *
```

The concentration of each such individual basis species is assigned this corresponding value. This value is an upper bound if the corresponding concentration is the total molality, and the actual value if it is a free molality. For species having a **jflag** string related to a an activity, the concentration of each such species is assigned a value equal to the thermodynamic activity, thus assuming that the activity coefficient has unit value. All other basis species are initially assigned a value of 1 x 10^{-7} molal. All non-basis species at this point have assigned concentrations of zero.

The charge imbalance is computed. The functions Σm and the ionic strength are then estimated. The estimate of the ionic strength at this point includes a term in which the computed charge imbalance is treated as though it were due to an unmeasured monovalent ion. The mole fraction of water is computed from the value of Σm. Then the activity coefficients are computed. The concentrations of non-basis species are taken as zero until the code enters the pre-Newton-Raphson optimization stage.

Assuming that the concentration of a basis species is equal to its total concentration may or may not be a good approximation. If it turns out to be a very poor approximation, then the first estimate of the concentration of at least one non-basis species will have to be large, typically a few to a few tens of orders of magnitude greater than the limit on its actual concentration imposed by the corresponding mass balance constraint. This first estimate is very often quite large in an absolute as well as a relative sense, often on the order of 10^{+10} to 10^{+60} molal. It is critical not to compute functions such Σm, the ionic strength, and the activity coefficients until the concentrations for such species have been brought down to physically realistic values.

B.4.5 METHODS TO AID CONVERGENCE

Several techniques are used in EQ3NR to aid convergence, both in pre-Newton-Raphson optimization and hybrid Newton-Raphson iteration. These are:

- Use of logarithmic iteration variables.
- Under-relaxation techniques.
- Automatic and user-specified basis switching.

We have not found it necessary to employ other methods, such as the "curve-crawler" technique discussed by Crerar (1975).

The physical quantities that correspond to the iteration variables are intrinsically positive. Use of logarithmic iteration variables restricts the generated values to the physically reasonable range.

Also, logarithmic corrections are effectively relative corrections to the corresponding physical quantities. Recall that *d log x*/*dx* = 1/(2.303 *x*). It follows that:

$$\Delta \log x \approx \frac{\Delta x}{2.303\, x} \tag{B–222}$$

Because of this, effective under-relaxation techniques are especially easy to implement when using logarithmic iteration variables.

Under-relaxation is the technique of judiciously reducing the magnitude of the computed correction terms. Assume that the unmodified method involves adding a correction term vector (δ_k), where *k* is the iteration number. This is typical in Newton-Raphson iteration. The new vector of master iteration variables is obtained thusly:

$$\underline{z}_{k=1} = \underline{z}_k = \underset{-k}{\delta} \tag{B–223}$$

If the new vector of master iteration variables is obtained instead by evaluating some set of corresponding equations not in this format, one can still utilize under-relaxation by defining a correction term vector as follows:

$$\underset{-k}{\delta} = \underline{z}_{k+1} - \underline{z}_k \tag{B–224}$$

Global under-relaxation is effected by replacing the correction equation given above by:

$$\underline{z}_{i+1} = \underline{z}_i + \kappa\, \underset{-i}{\delta} \tag{B–225}$$

where κ is a positive number less than one. Non-global under-relaxation is also possible. This does not involve the use of an under-relaxation factor. Rather it involves truncating the magnitudes of individual correction terms to satisfy specified limits, which may be different depending on the species involved and on the direction of change.

There are several methods of applying global under-relaxation. EQ3NR uses two relatively simple ones in making Newton-Raphson steps. The first of these places a limit on the element of the correction term vector having the largest magnitude:

$$\kappa = \frac{\delta'}{\delta_{\max}} \tag{B–226}$$

where δ' is the imposed limit and δ_{max} is the max norm of δ. In a Newton-Raphson iteration step this imposed limit is represented by the variable **screwd**. In EQ3NR, this is set in the main program at a value of 2.0. Besides aiding convergence, this method causes divergence, when it does occur, to occur more slowly. In such cases, it helps to yield useful information about the cause of divergence.

The other global under-relaxation method is applied for only the first 8 iterations. The under-relaxation factor is cut in half if the residual vector max norm β_{max} exceeds the value of the variable **screwn**. Initially, κ is set to a value of unity; when the current method of under-relaxation is applied, this factor may have been reduced as a result of applying the method described above. In EQ3NR, **screwn** is set to 0.5 in the main program.

Some degree of non-global under-relaxation is also employed in pre-Newton-Raphson optimization. Here under-relaxation is effected by imposing truncation limits on changes for individual master variables. The master variables for species constrained by mass balance equations are not permitted to decrease by more than 20.0 log units in a given step. A master variable constrained by the charge balance equation may not change by more than 2.0 log units.

Some truncation limits also apply to the activity coefficients and the functions Σm and the ionic strength. These limits are applied during both pre-Newton-Raphson optimization and hybrid Newton-Raphson iteration. These limits are defined in the variable **chgfac**. The value of this variable is set in the calling modules, and is usually scaled inversely with the value of Σm. Values range from 1.3 to 100.

If automatic basis switching is turned on (**iopt(11)** = 1, see Section 3.3.15), EQ3NR will attempt to improve the starting values by means of basis switching. The methodology here is quite simple. Consider the case of dissolved aluminum. The data file basis species is Al^{3+}. At low temperature, typically low values of dissolved aluminum, and moderate to high *pH*, the mass balance is typically very strongly dominated by the species $Al(OH)_4^-$. The concentration of Al^{3+} is many orders of magnitude below the concentration of this species. If one assumes that the concentration of this species is instead essentially equal to the total concentration, the computed concentration of $Al(OH)_4^-$ may be something on the order of 10^{20}–10^{40} molal. The value of the corresponding residual function will be similarly extremely large. On the other hand, when $Al(OH)_4^-$ is in the basis set, the initial assumption is that its concentration is equal to the total concentration, and the computed concentration of Al^{3+} is an appropriately much smaller number.

In the present version of EQ3NR, the total concentration quantity associated with a mass balance is redefined in terms of the new basis species. Assuming this quantity is expressed as molality, this change has no numerical significance in the above example. However, if a species to be switched into the basis set contributes to the original total concentration by a factor different from that of the original basis species, the difference is quite significant. For example, if $Al_{13}O_4(OH)_{24}^{7+}$ is switched into the basis set in place of Al^{3+}, then the associated total concentration must be redefined as:

$$m_{T,Al_{13}O_4(OH)_{24}^{7+}} = \frac{m_{T,Al^{3+}}}{13} \qquad \text{(B–227)}$$

Automatic basis switching is accomplished in a loop structure. More than one switch may be done each time through the loop. After this, the activity coefficients are recomputed (again assuming that the concentrations of non-basis species are zero), the residual functions are recomputed and more switches may be made. In the process, some switches may be undone by later ones. For example, $Al_{13}O_4(OH)_{24}^{7+}$ may first be switched into the basis set in place of Al^{3+}, and then $Al(OH)_4^-$ in place of, $Al_{13}O_4(OH)_{24}^{7+}$. When one switch replaces another, the original switch is first undone. In this example, Al^{3+} is switched back into the basis in place of $Al_{13}O_4(OH)_{24}^{7+}$; $Al(OH)_4^-$ is then switched into the basis in place of Al^{3+}. This loop continues until there are no candidates for basis switching or the loop has been passed through **nlopmx** times; this variable is currently set to 12.

To be a candidate for automatic basis switching, a species must have a computed concentration ten times that of the corresponding basis species. Furthermore, it can not already be in the basis set. A data file basis species which has been switched out of the basis set can only be brought back into the basis set by undoing an earlier switch. The involvement of a basis species in the input constraint associated with another basis species may prevent switching it out of the basis set. For example, if the input constraint for H^+ is a value for the *pHCl* function, then Cl^- is locked into the basis set. Conflicts may arise in candidate basis switches. For example, the same species could dominate more than one mass balance. It is then switched into the basis so as to reduce the affected mass balance residual which has the highest value.

The user may specify certain basis switches on the input file (see 3.3.2). If certain switches really need to be made, it is more efficient to make them in this manner.

If automatic basis switching is turned off (**iopt(11)** = 0, Section 3.3.15), the code will proceed directly from having made the initial starting estimates into pre-Newton-Raphson optimization. If it is turned on and the user has directed certain switches to be made on the input file, it may undo one or more of those switches, as well as make additional switches.

B.4.6THE PRE-NEWTON-RAPHSON OPTIMIZATION ALGORITHM

After any automatic basis switching is completed, an optimization algorithm is run. This process occurs in a loop structure, the times through which are known as *passes*. At the end of a pass, the activity coefficients are recomputed. If the concentration of an ion is to be adjusted to satisfy electrical balance, this adjustment is also recomputed at this point. Within each pass is another loop structure, the times through which are called *cycles*. Here, adjustments are made to the concentrations of the basis species (other than one which is constrained to satisfy electrical balance). A pass is completed after some number of cycles. The cycles within a pass terminate if some rather rough convergence criteria are satisfied, or if the maximum number of cycles in a pass have been completed. This is determined by the variable **ncylim**, which is currently set to 15. The passes terminate if rough convergence criteria applying to both the cycles and passes are satisfied, or if the maximum number of passes has been completed. This is determined by the variable **nplim**, which is currently set to 7.

The cycle algorithm is applied only to basis species which are constrained by mass balances. It is an example of what is sometimes called a “continued fraction” method. A variation on this approach (Wolery and Walters, 1975) was in fact the principal method used to solve speciation-solubility problems in a an early version of what is now EQ3NR. The derivation to be given here is different than that previously given elsewhere, and includes an important modification that has apparently not been previously noted.

Consider the case of dissolved aluminum. The total dissolved aluminum is expressed as total Al^{3+}. The normalized mass balance residual is:

$$\beta_{Al^{3+}} = \frac{m_{T,calc,Al^{3+}} - m_{T,Al}{}^{3+}}{m_{T,Al^{3+}}} \tag{B–228}$$

where: $m_{T,calc,Al^{3+}}$ is the total concentration of Al^{3+} as calculated from a mass balance expression, using the current estimated values of the concentrations of the basis species and estimates of the concentrations of non-basis species as calculated from the associated mass action equations, using in these the current estimated values of the concentrations of the basis species. In contrast, $m_{T,Al^{3+}}$ is one of the model constraints.

We will assume, for the moment, that Al^{3+} dominates this calculated mass balance. We may express this by writing:

$$m_{Al^{3+}} \approx m_{T,calc,Al^{3+}} \tag{B–229}$$

We may then write:

$$\beta_{Al^{3+}} \approx \frac{m_{Al^{3+}} - m_{T,Al^{3+}}}{m_{T,Al^{3+}}} \tag{B–230}$$

We can rearrange this to:

$$\frac{m_{Al^{3+}}}{\beta_{Al^{3+}} + 1} \approx m_{T,Al^{3+}} \tag{B–231}$$

We take the current iteration in the cycle to be the k^{th}. Applying the above equation to this iteration, we may write:

$$\frac{m_{Al^{3+},k}}{\beta_{Al^{3+},k} + 1} \approx m_{T,Al^{3+}} \tag{B–232}$$

Similar, applying it to the next iteration (the $k+1^{\text{th}}$) gives:

$$\frac{m_{Al^{3+},k+1}}{\beta_{Al^{3+},k+1} + 1} \approx m_{T,Al^{3+}} \tag{B–233}$$

Combining these equations then gives:

$$m_{Al^{3+},k+1} \approx \frac{m_{Al^{3+},k}\left(\beta_{Al^{3+},k+1}+1\right)}{\beta_{Al^{3+},k}+1} \qquad \text{(B–234)}$$

We would like the residual function to approach zero quickly. This desire can be written as:

$$\beta_{Al^{3+},k+1} \approx 0 \qquad \text{(B–235)}$$

Substitution of this into the above equation gives the following the iteration equation:

$$m_{Al^{3+},k+1} = \frac{m_{Al^{3+},k}}{\beta_{Al^{3+},k}+1} \qquad \text{(B–236)}$$

In a dilute, acid solution, the species Al^{3+} will indeed dominate its own mass balance. The starting assumption based on this should be a good one. Also, this species should dominate the calculated mass balances during the iteration process. We would expect the above iteration equation to work well. However, in solutions of moderate to high *pH*, the species $Al(OH)_4^-$ dominates the mass balance of Al^{3+}. What happens then? Note that we could do a basis switch, replacing Al^{3+} with $Al(OH)_4^-$. The above equations would then apply to $Al(OH)_4^-$ and all should again work well.

If we do not make this basis switch, we would still assume that $Al(OH)_4^-$ dominates the calculated mass balances. Following the previous approach, we are led to an equation of the form:

$$m_{Al(OH)_{4,}k=1} = \frac{m_{Al(OH)_{4,}k}}{\beta_{Al^{3+},k}+1} \qquad \text{(B–237)}$$

This is almost what we would have if we had made the basis switch. However, in this case, the normalized residual is still defined in terms of Al^{3+}, not $Al(OH)_4^-$. However, we can not directly use this equation, because it is written in terms of molalities of $Al(OH)_4^-$, not Al^{3+}. However, we can convert it into a form in terms of molalities of Al^{3+}. The two species are related by the reaction:

$$Al(OH)_4^- + 4H^+ = Al^{3+} + 4H_2O_{(l)} \qquad \text{(B–238)}$$

The corresponding mass action equation is:

$$K_{Al(OH)_4^-} = \frac{m_{Al^{3+}}\gamma_{Al^{3+}}x_w^{\Delta}\lambda_w^4}{m_{(OH)_4^-}\gamma_{Al(OH)_4^-}m_{H^+}^4\gamma_{H^+}^4} \qquad \text{(B–239)}$$

Let us consider all activity coefficients to be fixed, as well as the concentrations of other basis species appearing in this relation. Then the concentrations of the two aluminum species of interest satisfy the following proportionality:

$$m_{Al^{3+}} \propto m_{Al(OH)_4^-} \tag{B–240}$$

Presuming that the proportionality constant does not change significantly allows us to use this relation to obtain the same iteration equation that we had before:

$$m_{Al^{3+},k+1} = \frac{m_{Al^{3+},k}}{\beta_{Al^{3+},k} + 1} \tag{B–241}$$

The only significant difference in the iteration process in this case (versus either that in which Al^{3+} is dominant or that in which $Al(OH)_4^-$ is dominant but switched into the basis set) is that the starting estimate is not so good. In fact, it might be very bad, off by many orders of magnitude. However, it turns out that this algorithm is very good for quickly getting to about the right order of magnitude, even if the starting estimate is off (high) by several tens of orders of magnitude. On the other hand, it is not so efficient in a close neighborhood of the solution. This makes it a good complement to Newton-Raphson iteration, which is very efficient near the solution, but which often fails to converge at all if the starting estimates are far from the solution.

Now suppose the complex $Al_{13}O_4(OH)_{24}^{7+}$ dominates the calculated mass balance of Al^{3+}. Again, we do not make a basis switch. This then leads us to a result of the form:

$$m_{Al_{13}O_4(OH)_{24}^{7+},k+1} = \frac{13 m_{Al_{13}O_4(OH)_{24}^{7+},k}}{\beta_{\mathrm{Al}^{3+},k} + 1} \tag{B–242}$$

The species $Al_{13}O_4(OH)_{24}^{7+}$ and $Al^{3+}$are related by the reaction:

$$Al_{13}O_4(OH)_{24}^{7+} + 32H^+ = 13Al^{3+} + 28H_2O_{(l)} \tag{B–243}$$

The corresponding mass action equation is:

$$K_{Al_{13}O_4(OH)_{24}^{7+}} = \frac{m_{Al^{3+}}^{13}\gamma_{Al^{3+}}^{13}x_w^{28}\lambda_w^{28}}{m_{Al_{13}O_4(OH)_{24}^{7+}}\gamma_{Al_{13}O_4(OH)_{24}^{7+}}m_{H^+}^{32}\gamma_{H^+}^{32}} \tag{B–244}$$

Using the same assumptions as before leads to the following proportionality:

$$m_{Al^{3+}}^{13} \propto m_{Al_{13}O_4(OH)_{24}^{7+}} \tag{B–245}$$

Using this as before leads to the following iteration equation:

$$m_{Al^{3+},k+1} = \frac{13^{\frac{1}{13}} m_{Al^{3+},k}}{\left(\beta_{Al^{3+},k} + 1\right)^{\frac{1}{13}}} \tag{B–246}$$

This differs from the previous results in two ways. First, an exponent now appears on the (β + 1) term. This exponent is the factor expressing the stoichiometric equivalence of the basis species corresponding to a mass balance expression to the species which dominates the calculated mass balance. Also, a factor appears in the numerator on the right hand side which is the inverse of the same exponent with a matching exponent This is the general case. These "new" elements of the equation did not explicitly appear in the previous results because the stoichiometric equivalence happened to be unity.

The exponent on the (β + 1) term is critical to the success of this method. This is because it has a large effect on the exponent of the resultant calculated concentration of the basis species. When one starts the iteration process, the value of β may initially be something like 10^{+60}. Use of the iteration equation ignoring this exponent would result in the calculated concentration of the basis species being lowered by 60 orders of magnitude. The effect of the exponent on the (β + 1) is to cause it to be lowered only about 4.6 orders of magnitude. The effect of ignoring this exponent on the remainder of the calculation would be wild oscillation.

The effect of the factor in the numerator is less extreme. It does not affect the resulting order of magnitude. For example, $1^{\frac{1}{1}} = 1,\ 2^{\frac{1}{2}} \approx 1.41, 3^{\frac{1}{3}} \approx 1.44, 4^{\frac{1}{4}} \approx 1.41, and\ 5^{\frac{1}{5}} \approx 1.38$. In the case shown above, $13^{\frac{1}{13}} \approx 1.22$. The absence of this factor doesn't have much effect if all one is trying to do is get within about an order of magnitude of the solution before switching to another algorithm. However, it would cause convergence to fail in a close neighborhood of the solution if one attempted to use this algorithm to obtain a final solution.

Not all basis species are constrained by mass balance relations. If a species is constrained by a value for its log activity, the concentration is simply estimated from this value using the current value for the corresponding activity coefficient:

$$\log m_{i,k+1} = \log a_i - \log \gamma_i \tag{B–247}$$

The concentration of such a species can not change during a pass, because activity coefficients are only recalculated as the end of a pass.

A basis species whose concentration is to be adjusted to satisfy electrical balance has its concentration recalculated in a cycle after the concentrations of all other basis species have been recalculated. This will be discussed below.

The concentrations of basis species which are constrained by any other types of constraints are calculated simultaneously by solving a matrix equation. This is the case for $O_{2(g)}$ when an *Eh* or *pe* value is input, for cases in which heterogeneous or homogeneous equilibria must be satisfied, and cases in which a combination activity function such as *pHCl* is utilized. It is not always strictly necessary to make these calculations simultaneously, but it is more convenient to always do it this way than to deal otherwise with those cases which would allow complete or partial solution by a sequence of individual calculations (whose order would have to be determined in each individual case).

For example, suppose dissolved calcium is constrained to satisfy equilibrium with calcite and bicarbonate is constrained to satisfy a specified fugacity of carbon dioxide. We will assume that chloride is constrained by a total concentration and that the hydrogen is constrained by an input value of *pH*. The respective relevant governing equations for Ca^{2+} and H^{+} are:

$$\log K_{calcite} = \log m_{Ca^{2+}} + \log\gamma_{Ca^{2+}} + \log m_{HCO_3^-} + \log\gamma_{HCO_3^-} - \log m_{H^+} - \log\gamma_{H^+} \quad \text{(B–248)}$$

$$\log K_{CO_{2(g)}} = \log m_{HCO_3^-} + \log\gamma_{HCO_3^-} + \log m_{H^+} + \log\gamma_{H^+} - \log f_{CO_2} - \log x_w - \log\lambda_w \quad \text{(B–249)}$$

These may be rearranged to give:

$$\log m_{Ca^{2+}} + \log m_{HCO_3^-} = \log K_{calcite} - \log\gamma_{Ca^{2+}} - \log\gamma_{HCO_3^-} + \log m_{H^+} + \log\gamma_{H^+} \quad \text{(B–250)}$$

$$\log m_{HCO_3^-} - \log x_w = \log K_{CO_{2(g)}} - \log\gamma_{HCO_3^-} - \log m_{H^+} - \log\gamma_{H^+} + \log f_{CO_2} + \log\lambda_w \quad \text{(B–251)}$$

where the variables treated as the unknowns are on the left hand sides. As all these concentration values pertain to the $k + 1^{th}$ iteration, they could be so marked in the rearranged equations (but we leave them out for clarity). The mole fraction of water may also be adjusted as part of the process, using the approximate relation:

$$\log x_{w,k+1} = \log x_{w,k} + \sum_{s' \in S} \frac{\partial \log x_w}{\partial \log m_{s'}} \left(\log m_{s',k+1} - \log m_{s',k}\right) \quad \text{(B–252)}$$

where *S* is the set of solute basis species whose concentrations must be solved for in this manner. This is a Taylor's series truncated to first order. For notational convenience, we make the following definition:

$$W_{s'} = \frac{\partial \log x_w}{\partial \log m_{s'}}, \qquad s' \neq w,\ s_B \quad \text{(B–253)}$$

Then the equation in the present example may be written as:

$$\log x_w - W_{Ca^{2+}} \log m_{Ca^{2+}} - W_{HCO_3^-} \log m_{HCO_3^-}$$

$$= \log x_{w,\,k} - W_{Ca^{2+}} \log m_{Ca^{2+},\,k} - W_{HCO_3^-} \log m_{HCO_3^-,\,k} \quad \text{(B–254)}$$

where again all the variables treated as unknowns are on the left hand side. For notational consistency, we have dropped the "$k + 1$" subscripts.

We now evaluate the necessary partial derivatives. The mole fraction of water must be expressed as a function of the concentrations of the basis species other than water. We first write this in the form:

$$\log x_w = \log \left(\frac{\Omega}{\Omega + \sum_{\substack{s' = 1 \\ s' \neq w}}^{s_Q} m_{s'} + \sum_{r=1}^{r_T} m_{s''}} \right) \tag{B–255}$$

where s'' denotes the non-basis species associated with the r^{th} aqueous reaction. Partial differentiation leads to the following intermediate result:

$$W_{s'} = -\frac{x_w}{\Omega}\left(m_{s'} - \sum_{r}^{r_T} \frac{b_{s'r} m_{s''}}{b_{s''r}} - W_{s'} \sum_{r}^{r_T} \frac{b_{wr} m_{s''}}{b_{s''r}} \right) \tag{B–256}$$

where s' is a basis species other than water. Rearranging then gives the final result:

$$W_{s'} = \frac{-\frac{x_w}{\Omega}\left(m_{s'} - \sum_{r}^{r_T} \frac{b_{s'r} m_{s''}}{b_{s''r}} \right)}{\left(1 - \frac{x_w}{\Omega} \sum_{r}^{r_T} \frac{b_{wr} m_{s''}}{b_{s''r}} \right)} \tag{B–257}$$

Technically, these partial derivatives should be evaluated using data corresponding to the k-th iteration. However, as the method is not exact anyway and the mole fraction of water does not really vary much in the process, this is not critical.

In the above example, we have three equations in three unknowns. These equations are linear in the four log concentration variables; hence, they can be solved simultaneously by solving a corresponding matrix equation. This takes the form:

$$\begin{matrix} 1 & 1 & 0 \\ 0 & 1 & -1 \\ -W_{Ca^{2+}} & -W_{HCO_3^-} & 1 \end{matrix} \quad \begin{matrix} \log m_{Ca^{2+}} \\ \log m_{HCO_3^-} \\ \log x_w \end{matrix} = \begin{matrix} R_{Ca^{2+}} \\ R_{HCO_3^-} \\ R_w \end{matrix} \tag{B–258}$$

where the elements in the right hand side vector are equal to the expressions on the right hand side of eqs (B–250), (B–251), and (B–254). Matrix equations of this form are evaluated after new values have been calculated for the concentrations of the other basis species.

At the end of a cycle, a full set of residual functions is computed. This includes the β array and its max norm, β_{max}. The only non-zero elements of β at the end of a cycle are those which pertain to mass and charge balance constraints. The elements of this vector corresponding to the former have been previously introduced. In the case of the latter, the relevant equation is:

$$\beta_{s_z} = \frac{\alpha_{s_z}}{\sum_c z_c m_c + \sum_a |z_a| m_a} \tag{B–259}$$

where s_z is the basis species so constrained, c denotes cations and a anions, and:

$$\alpha_{s_z} = \sum_c z_c m_c + \sum_a z_a m_a \tag{B–260}$$

The concentration of this species is then adjusted so as to make these residuals zero:

$$m_{s_z,k+1} = m_{s_z,k} - \frac{\alpha_{s_z,k}}{|z_{s_z}|} \tag{B–261}$$

A pass ends when one of the following occurs:

- The non-zero (mass balance) elements of β satisfy a loose convergence test (all fall in the range -10% to +50%).
- The maximum number of cycles per pass have been completed.
- A convergence function β_{func} (**betfnc**) indicates that iteration in the present cycle is diverging. This convergence function will be discussed later in this section.

At the end of a pass, the Σm function and the ionic strength are recalculated, the mole fraction of water is recalculated, and the activity coefficients are recalculated. The code defines residual functions based on the magnitude of the changes in Σm, the ionic strength, and the activity coefficients from the values pertaining to the previous pass. The sequence of passes is stops when one of the following occurs:

- The residuals defined for Σm, the ionic strength, and the activity coefficients satisfy a loose convergence test (all less than or equal to 0.1).

- The maximum number of passes have been completed.

The optimization is deemed successful if both sets of loose convergence tolerances are satisfied. The code will then execute hybrid Newton-Raphson iteration. If optimization is not successful, the code checks to see if any equilibrium constraints appear to imply unrealistically high solute concentrations. If this is the case, the code will not attempt hybrid Newton-Raphson iteration and a message to this effect is written to the screen and output files. Otherwise, the code will execute hybrid Newton-Raphson iteration, which often succeeds even when the optimization step fails to satisfy its own convergence tolerances.

The cycle algorithm discussed in this section is an example of a first order method. This means that in a close neighborhood of the solution, one has that:

$$\delta_{i,\,k+1} = \sum_{j=1}^{n} p_{ij}\delta_{jk} \tag{B–262}$$

where n is the number of iteration variables and the p_{ij} are constants.

B.4.7THE NEWTON-RAPHSON METHOD

The Newton-Raphson method is a well-known iterative technique for solving non-linear systems of algebraic equations (see for example Van Zeggeren and Storey, 1970; Carnahan, Luther, and Wilkes, 1969; or any introductory text on numerical methods). We will not discuss the derivation of the method here, only its application. Given a set n governing equations and n unknowns (represented by a vector $\underline{z}$ of iteration variables), one may construct a set of residual functions (represented by the vector α), each member of which has a value of zero when the n equations are satisfied. Both $\underline{z}$ and α are of length n.

A simple example will illustrate this. Suppose we wish to solve the general quadratic equation:

$$ax^2 + bx + c = 0 \tag{B–263}$$

This is a case in which $n = 1$. Here a, b, and c are given coefficients, and x is the only unknown. The residual function can be defined as:

$$\alpha = ax^2 + bx + c \tag{B–264}$$

Other definitions are possible, however. The only requirement is that they take on a value of zero when the governing equation is satisfied. We note here that the choice of definition may affect the convergence behavior.

Let *k* be the number of iterations, such that $\underline{z}_k$ and α_k are the iteration variable and residual function vectors on the *k*-th iteration. Let $\underline{z}_0$ represent the set of starting estimates. An iteration step is made by calculating $\underline{z}_{k+1}$ from $\underline{z}_k$. The Newton-Raphson method does this by computing a vector of correction terms, δ, by solving the matrix equation:

$$\underline{J}\underline{\delta} = -\underline{\alpha} \tag{B–265}$$

Here $\underline{J}$ is the Jacobian matrix, defined by:

$$\underline{J} = \left(\frac{\partial \alpha_i}{\partial z_j} \right) \tag{B–266}$$

where *i* and *j* are the matrix coordinates. In our example, this becomes:

$$J = 2ax + b \tag{B–267}$$

The correction term is then applied:

$$\underline{z}_{k+1} = \underline{z}_k + \delta_{-k} \tag{B–268}$$

If the iteration converges, all elements of both α and δ approach zero. It is useful to define another residual function vector β which is identical to α, except that some elements may be normalized to provide a better measure of convergence. It is then convenient to define β_{max} and δ_{max} as the largest absolute values of the elements of β and δ, respectively. Both β_{max} and δ_{max} may then be used in tests to determine if the iteration has converged satisfactorily.

A useful measure of how well convergence is proceeding may also be constructed. The Newton-Raphson method is a second order method. This means that in a close neighborhood of the solution, one has that:

$$\delta_{i,\,k+1} = \sum_{j=1}^{n} p_{ij} \delta_{jk}^2 \tag{B–269}$$

where the p_{ij} are constants. In practice, there is usually no attempt to actually evaluate them. The significance of this is that in a close neighborhood of the solution, $\delta_{max,\,k+1}$ should be much less than $\delta_{max,\,k}$. The function δ_{func} (the variable **delfnc**) is defined:

$$\delta_{func,k+1} = 1 - \left(\frac{\delta_{max,\,k+1}}{\delta_{max,\,k}} \right) \tag{B–270}$$

may therefore be expected to approach (from below) a value of unity if the iteration is converging very rapidly (as theory suggests when the cross terms describing the evolution of the correction vector are small; i. e., the p_{ij} are small for the case $i \neq j$). Convergence to a lesser value, say ≈ 0.72 instead of ≈0.99 is not unknown. This may imply non-negligible cross terms or an error in writing the Jacobian matrix. It also may result from modifications to the basic Newton-Raphson method, such as we have introduced by updating the activity coefficients between Newton-Raphson steps. The function, β_{func} (the variable **betfnc**) is defined similarly:

$$\beta_{\mathrm{func},k+1} = 1 - \left(\frac{\beta_{\mathrm{max},k+1}}{\beta_{\mathrm{max},k}}\right) \tag{B–271}$$

and has essentially the same properties.

The use of a pure Newton-Raphson method would require the activity coefficients and their associated model equations to be brought directly into the set of *n* equations and *n* unknowns solved by the method, either directly or by substitution. In a previous section in this section, we noted that there was a problem in expanding the current set of master iteration variables in that the activity coefficients would have to be calculated before the concentrations of the non-basis species, and vice versa. This problem precludes taking care of the activity coefficients by a substitution mechanism that leaves the current set of master iteration variables unchanged. One would have to instead treat the activity coefficients themselves as master iteration variables. These are many, so this would not be very convenient. Alternatively, one could treat the ionic strength as a master iteration variable, but this would only suffice for simple extended Debye-Hückel formalisms that are only valid in dilute solutions. We have instead chosen to hybridize the Newton-Raphson method by simply updating the activity coefficients between Newton-Raphson steps. In practice, this seems to work quite well, except in some extremely concentrated solutions.

The maximum number of iterations in a Newton-Raphson calculation is determined by the input file variable **itermx**. This has a default value of 30 in EQ3NR. Convergence is achieved when β_{max} is less than the tolerance parameter **tolbt**, δ_{max} is less than the tolerance parameter **toldl**, and max norms on the changes in the Σm function, the ionic strength, and the activity coefficients are all less than **tolbt**. The tolerance parameters **tolbt** and **toldl** both appear on the input file, and both have a default value of $1\text{x}10^{-6}$.

B.4.8 DERIVATION OF RESIDUAL FUNCTIONS AND THE JACOBIAN MATRIX

In this section, we shall derive the residual functions and the Jacobian matrix for the Newton-Raphson iteration procedures used by the EQ3NR code. Given a set of governing equations and an equal number of unknowns, there is no unique way to formulate residuals and Jacobians. The number of equations and unknowns may be reduced by means of substitutions. Furthermore, one may then construct the residual functions in any number of ways. Once the residual functions have been chosen, the form of the Jacobian is determined according to the partial derivatives of these functions.

We will now take each remaining governing equation, construct a corresponding pair of residual functions (α and β), and derive the corresponding row of elements in the Jacobian matrix by partial differentiation. The α residuals are the true Newton-Raphson residual functions and are the subject of partial differentiation to define the Jacobian matrix. The β residuals are better measures of satisfactory convergence.

B.4.8. Mass Balance

This may be applied to any aqueous species in the basis set (s = 1 through s_Q) except water (w), H^+, and $O_{2(g)}$ (the s_Bth species). Mass balance is specified as the governing equation by setting the corresponding **jflag** value *Molality* and entering a total concentration on the molal scale ($m_{T,s}$). Alternatively, one may enter total concentration in other units using other **jflag** string values (*Molarity, mg/L, mg/kg.sol*), which EQ3NR will then recompute into molality and set **jflag** to *Molality*. The governing equation can be written as:

$$m_{T,s} = \sum_{s'=1}^{s_Q} u_{s's} m_s + \sum_{s''=s_Q+1}^{s_T} u_{s''s} m_{s''} \tag{B–272}$$

The residual functions are defined by:

$$\alpha_s = -m_{T,s} + \sum_{s'=1}^{s_Q} u_{s's} m_s + \sum_{s''=s_Q+1}^{s_T} u_{s''s} m_{s''} \tag{B–273}$$

$$\beta_s = \frac{\alpha_s}{m_{T,s}} \tag{B–274}$$

where in the last part of eqs (B–272) and (B–273) it is implied that $s'' = s_r$. From this point, we may use the relations developed above to derive the following Jacobian elements:

$$\begin{aligned} J_{sw} &= \frac{\partial \alpha_s}{\partial \log x_w} \\ &= \sum_{r=1}^{r_T} u_{s''s} \frac{\partial m_{s''}}{\partial \log x_w} \\ &= -2.303 \sum_{r=1}^{r_T} \frac{u_{s''s} m_{s''} b_{wr}}{b_{s''r}} \\ &= -2.303 \sum_{r=1}^{r_T} b_{wr} H_{sr} \end{aligned} \tag{B–275}$$

$$\begin{aligned} J_{ss_B} &= \frac{\partial \alpha_s}{\partial \log fO_2} \\ &= \sum_{r=1}^{r_T} u_{s''s} \frac{\partial m_{s''}}{\partial \log fO_2} \\ &= 2.303 \sum_{r=1}^{r_T} \frac{u_{s''s} m_{s''} b_{s_B r}}{b_{s''r}} \\ &= 2.303 \sum_{r=1}^{r_T} b_{s_B r} H_{sr} \end{aligned} \tag{B–276}$$

and for $s \neq w$, s_B:

$$J_{ss'} = \frac{\partial \alpha_s}{\partial \log m_{s'}}$$

$$= \sum_{s'''=1}^{s_Q} u_{s'''s} \frac{\partial m_{s'''}}{\partial \log m_{s'}} + \sum_{s''=s_Q+1}^{s_T} u_{s''s} \frac{\partial m_{s''}}{\partial \log m_{s'}}$$

$$= -2.303 \left(u_{s's} m_{s'} - \sum_{r=1}^{r_T} \frac{u_{s''s} m_{s''} b_{s'r}}{b_{s''r}} \right)$$

$$= -2.303 \left(u_{s's} m_{s'} - \sum_{r=1}^{r_T} b_{s'r} H_{sr} \right) \qquad \text{(B–277)}$$

Here $u_{s'''s'} = 1.0$ if $s' = s'''$, otherwise $u_{s'''s'} = 0$.

B.4.8. Electrical Balance

This governing equation may be applied to one of the ions in the aqueous species basis set, here denoted by *s*. Apart from the definition of the β residual, the treatment is exactly analogous to that for mass balance. The governing equation can be written as:

$$\sum_{s'=1}^{s_Q} z_{s'} m_s + \sum_{s''=s_Q+1}^{s_T} z_{s''} m_{s''} = 0 \qquad \text{(B–278)}$$

The residual functions are defined by:

$$\alpha_s = \sum_{s'=1}^{s_Q} z_{s'} m_s + \sum_{s''=s_Q+1}^{s_T} z_{s''} m_{s''} \qquad \text{(B–279)}$$

$$\beta_s = \frac{\alpha_s}{\sum_{s'=1}^{s_Q} z_{s'} m_s + \sum_{s''=s_Q+1}^{s_T} z_{s''} m_{s''}} \qquad \text{(B–280)}$$

The corresponding Jacobian elements are as follows:

$$J_{sw} = \frac{\partial \alpha_s}{\partial \log x_w}$$

$$= -2.303 \sum_{r=1}^{r_T} b_{wr} H_{zr} \qquad \text{(B–281)}$$

$$J_{ss_B} = \frac{\partial \alpha_s}{\partial \log f_{O_2}}$$

$$= 2.303 \sum_{r=1}^{r_T} b_{s_B r} H_{zr} \qquad \text{(B–282)}$$

and for $s \neq w$, s_B:

$$J_{ss'} = \frac{\partial \alpha_s}{\partial \log m_{s'}}$$

$$= -2.303 \left(z_{s'} m_{s'} - \sum_{r=1}^{r_T} b_{s'r} H_{zr} \right) \qquad \text{(B–283)}$$

B.4.8.3 The Mole Fraction Of Water

The governing equation can be written as:

$$x_w = \frac{\Omega}{\Omega + \sum_{s=1}^{s_T} m_s} \qquad \text{(B–284)}$$

where Ω is the number of moles of solvent water comprising a mass of 1 kg (Ω ≈ 55.51) and s_T is the number of aqueous species in the solution.

The corresponding residual functions are defined as:

$$\alpha_w = \log \left| \frac{\Omega}{\Omega + \sum_{s=1}^{s_T} m_s} \right| - \log x_w \tag{B–285}$$

$$\beta_s = \alpha_s \tag{B–286}$$

Because it is necessary to distinguish between basis species and non-basis species, it is helpful to write the equation for the first residual function in the slightly expanded form:

$$\alpha_w = \log \left| \frac{\Omega}{\Omega + \sum_{s'=1}^{s_Q} m_{s'} + \sum_{r=1}^{r_T} m_{s''}} \right| - \log x_w \tag{B–287}$$

In the following equations, we will take *s*' and s''' to be basis species other than water. The following Jacobian elements are then obtained:

$$J_{ww} = \left(\frac{1}{\Omega} \right) \left| \frac{\Omega}{\Omega + \sum_{s'=1}^{s_Q} m_{s'} + \sum_{r=1}^{r_T} m_{s''}} \right| \left(\sum_{r=1}^{r_T} \frac{b_{wr} m_{s''}}{b_{s''r}} \right) - 1.0 \tag{B–288}$$

$$J_{ws'''} = \left(-\frac{1}{\Omega} \right) \left| \frac{\Omega}{\Omega + \sum_{s'=1}^{s_Q} m_{s'} + \sum_{r=1}^{r_T} m_{s''}} \right| \left(m_{s'''} - \sum_{r=1}^{r_T} \frac{b_{s'''r} m_{s''}}{b_{s''r}} \right) \tag{B–289}$$

Note that these Jacobian elements differ from the corresponding set of partial derivatives used in the truncated Taylor's expansion as part of the pre-Newton-Raphson optimization. In the present case, $\log x_w$ is treated as an independent variable. In the previous case, it was not.

B.4.8.4 Specified Free Concentration

This represents the **jflag** *Molality* option for $s = 1$ through s_Q, except when *s* is water (*w*) or $O_{2(g)}$ (s_B). The free concentration is in molality. If it is initially entered in molarity (**jflag** = *Molarity*), EQ3NR converts it to molality and resets **jflag** to *Molality* before beginning Newton-Raphson iteration. The governing equation is just the identity:

$$m_s = m_s \tag{B–290}$$

Hence, the corresponding residual functions are given by:

$$\alpha_s = 0 \tag{B–291}$$

$$\beta_s = 0 \tag{B–292}$$

In order to prevent singularity in the Jacobian, we set:

$$J_{ss} = 1.0 \tag{B–293}$$

B.4.8.5 Specified Thermodynamic Activity

This represents the **jflag** = *Log activity* option for s =1 through s_Q, except when s is water (w) or $O_{2(g)}$ (s_B). This option is most frequently employed with H^+ in order to specify a pH value ($pH = -log\ a_{H^+}$). The governing equation is:

$$\log m_s + \log \gamma_s = \log a_s \tag{B–294}$$

The residual functions are:

$$\alpha_s = -\log a_s + \log m_s + \log \gamma_s \tag{B–295}$$

$$\beta_s = \alpha_s \tag{B–296}$$

The only non-zero Jacobian element is:

$$J_{ss} = 1.0 \tag{B–297}$$

B.4.8.6 Log Activity Combination

For this set **jflag** = *Log act combo*. Recall that the activity combination parameter is defined by:

$$\aleph_{ij} = \left|z_j\right| \log a_i - \frac{z_i z_j}{\left|z_j\right|} \log a_j \tag{B–298}$$

We will identify i as the basis species s to which this constraint is applied, and s^* as the other basis species involved. The governing equation can then be written as:

$$\log m_s = \frac{\aleph_{ss^*}}{\left|z_{s^*}\right|} - \log y_s + \frac{z_s}{z_{s^*}} \log m_{s^*} = \frac{z_s}{z_{s^*}} = \frac{z_s}{z_{s^*}} \log \gamma_{s^*} \tag{B–299}$$

Hence, the corresponding residual functions are given by:

$$\alpha_s = \frac{\aleph_{ss*}}{|z_{s*}|} - \log\gamma_s + \frac{z_s}{z_{s*}}\log m_{s*} + \frac{z_s}{z_{s*}}\log\gamma_{s*} - \log m_s \tag{B–300}$$

$$\beta_s = \alpha_s \tag{B–301}$$

The corresponding non-zero elements of the Jacobian matrix are then:

$$J_{ss} = -1.0 \tag{B–302}$$

$$J_{ss*} = \frac{z_s}{z_{s*}} \tag{B–303}$$

B.4.8.7 Mean Log Activity

Use this option with **jflag** = *Log mean act.* The mean log activity of two oppositely charge ions *i* and *j* can be written as:

$$\log a_{\pm,ij} = \frac{|z_j|\log a_i + |z_i|\log a_j}{|z_i| + |z_j|} \tag{B–304}$$

We will identify *i* as the basis species *s* to which this constraint is applied, and *s** as the other basis species involved. The governing equation can then be written as:

$$\log m_s = \frac{|z_s| + |z_{s*}|}{|z_{s*}|}\log a_{\pm,ss*} - \log\gamma_s - \left|\frac{z_s}{z_{s*}}\right|\log m_{s*} - \left|\frac{z_s}{z_{s*}}\right|\log\gamma_{s*} \tag{B–305}$$

Hence, the corresponding residual functions are given by:

$$\alpha_s = \frac{|z_s| + |z_{s*}|}{|z_{s*}|}\log a_{\pm,ss*} - \log\gamma_s - \left|\frac{z_s}{z_{s*}}\right|\log m_{s*} - \left|\frac{z_s}{z_{s*}}\right|\log\gamma_{s*} - \log m_s \tag{B–306}$$

$$\beta_s = \alpha_s \tag{B–307}$$

The corresponding non-zero elements of the Jacobian matrix are then:

$$J_{ss} = -1.0 \tag{B–308}$$

$$J_{ss*} = -\left|\frac{z_s}{z_{s*}}\right| \tag{B–309}$$

B.4.8.8 Equilibrium With A Pure Mineral

This option (**jflag** = *Hetero. equil.*) may be specified for any aqueous species denoted by $s = 1$ through s_Q, except when s is water (w). Let ϕ denote the mineral in question. The governing equation is:

$$\log K_{\phi} = b_{w\phi}\left(\log x_w + \log \lambda_w\right) + b_{s_B\phi} \log f_{O_2} + \sum_{\substack{s'=1 \\ s' \neq w, s_B}}^{s_Q} b_{s'\phi}\left(\log m_s + \log \gamma_{s'}\right) \tag{B–310}$$

The residual functions are defined as:

$$\alpha_s = \frac{1}{b_{s\phi}} \begin{pmatrix} \log K_{\phi} - b_{w\phi}\left(\log x_w + \log \lambda_w\right) - b_{s_B\phi} \log f_{O_2} \\ - \displaystyle\sum_{\substack{s'=1 \\ s' \neq w, s_B}}^{s_Q} b_{s'\phi}\left(\log m_s + \log \gamma_{s'}\right) \end{pmatrix} \tag{B–311}$$

$$\beta_s = \alpha_s \tag{B–312}$$

The corresponding Jacobian elements are then:

$$J_{ss'} = -\frac{b_{s'\phi}}{b_{s\phi}} \tag{B–313}$$

The residual function α_s defined in eq (B–311) has in a sense been normalized by dividing by the stoichiometric reaction coefficient $b_{s\phi}$. This makes the residual equivalent to the difference between the calculated and current values of log m_s, independent of how the reaction has been written. This avoids some potential numerical scaling problems. Other options involving mass action equations are treated in the same manner.

B.4.8.9 Equilibrium With A Solid Solution End-member Component

This option (also set by **jflag** = *Hetero. equil.*) may be specified for any aqueous species denoted by $s = 1$ through s_Q, except when s is water (w). The treatment is closely analogous to that for equilibrium with a pure mineral. Let σ and ψ denote the end-member and solid solution phase, respectively. The governing equation contains an additional term in the mole fraction and activity coefficient of the solid solution end-member and is given by:

$$\begin{aligned} \log K_{\sigma\psi} = {} & b_{\sigma\psi\sigma\psi}\left(\log x_{\sigma\psi} + \log \lambda_{\sigma\psi}\right) + b_{w\sigma\psi}\left(\log x_w + \log \lambda_w\right) + b_{s_B\sigma\psi} \log f_{O_2} \\ & + \sum_{\substack{s'=1 \\ s' \neq w, s_B}}^{s_Q} b_{s'\sigma\psi}\left(\log m_{s'} + \log \gamma_{s'}\right) \end{aligned} \tag{B–314}$$

The residual functions are defined as:

$$\alpha_s = \frac{1}{b_{s\sigma\psi}} \left(\log K_{\sigma\psi} - b_{\sigma\psi\sigma\psi} \left(\log x_{\sigma\psi} + \log \lambda_{\sigma\psi} \right) - b_{w\sigma\psi} (\log x_w + \log \lambda_w \right) + \\ - b_{s_B\sigma\psi} \log f_{O_2} - \sum_{\substack{s' = 1 \\ s' \neq w, s_B}}^{s_Q} b_{s'\sigma\psi} \left(\log m_{s'} + \log \gamma_{s'} \right) \tag{B–315}$$

$$\beta_s = \alpha_s \tag{B–316}$$

The corresponding Jacobian elements are then:

$$J_{ss'} = -\frac{b_{s'\sigma\psi}}{b_{s\sigma\psi}} \tag{B–317}$$

B.4.8. Equilibrium With A Gas

This option (again by **jflag** = *Hetero. equil.*) may be specified for any aqueous species denoted by s = 1 through s_Q, except when s is water (w). The treatment is closely analogous to that for equilibrium with a pure mineral. Let g denote the gas in question. The governing equation contains an additional term in the fugacity of this gas and is given by:

$$\log Kg = b_{gg} \log f_g + b_{wg} \left(\log x_w + \log \lambda_w \right) + b_{s_Bg} \log f_{O_2} + \sum_{\substack{s'=1 \\ s' \neq w, s_B}}^{s_Q} b_{s'g} \left(\log m_{s'} + \log \gamma_{s'} \right) \tag{B–318}$$

The residual functions are defined as:

$$\alpha_s = \frac{1}{b_{sg}} \left(\begin{array}{l} \log K_g - b_{gg} \log f_g - b_{wg} \left(\log x_w + \log \lambda_w \right) \\ - b_{s_Bg} \log f_{O_2} - \displaystyle\sum_{\substack{s'=1 \\ s' \neq w, s_B}}^{s_Q} b_{s'g} \left(\log m_{s'} + \log \gamma_{s'} \right) \end{array} \right) \tag{B–319}$$

$$\beta_s = \alpha_s \tag{B–320}$$

The corresponding Jacobian elements are then:

$$J_{ss'} = \frac{b_{s'g}}{b_{sg}} \tag{B–321}$$

B.4.8.11 Concentration Fixed By Internal Equilibrium

This option (**jflag** =*Homo. equil.*) excludes the species to which it is applied, which must be in the auxiliary basis set, and its ion-pairs and complexes from the mass balance of the corresponding basis species to which it is linked by its own associated reaction (usually a strict basis species). This is a good choice for dissolved gas species such as $O_{2(aq)}$ and $H_{2(aq)}$. If HS^- is an auxiliary basis species with **jflag** = *Homo. equil.* and it is linked to SO_4^{2-}, then HS^- and its "complexes" (other species whose reactions link them to this species) are not included in calculating the SO_4^{2-} mass balance.

Let *s* be the auxiliary basis species constrained by *Homo. equil.*, and let *r* denote its associated reaction. The governing is then:

$$\begin{aligned}\log K_r &= b_{sr}\left(\log m_s + \log \gamma_s\right) + b_{wr}\left(\log x_w + \log \lambda_w\right) \\ &+ b_{s_B r} \log f_{O_2} + \sum_{\substack{s'=1 \\ s' \neq w, s_B, s}}^{s_Q} b_{s'r}\left(\log m_{s'} + \log \gamma_{s'}\right)\end{aligned} \tag{B–322}$$

The residual functions are defined as:

$$\alpha_s = \frac{1}{b_{sr}} \left(\begin{aligned} &\log K_r - b_{sr}\left(\log m_s + \log \gamma_s\right) - b_{wr}\left(\log x_w + \log \lambda_w\right) \\ &- b_{s_B r} \log f_{O_2} - \sum_{\substack{s'=1 \\ s' \neq w, s_B, s}}^{s_Q} b_{s'r}\left(\log m_{s'} + \log \gamma_{s'}\right) \end{aligned} \right) \tag{B–323}$$

$$\beta_s = \alpha_s \tag{B–324}$$

The corresponding Jacobian elements are then:

$$J_{ss'} = -\frac{b_{s'r}}{b_{sr}} \tag{B–325}$$

B.4.8.12 Specified Log Oxygen Fugacity

This option (**irdxc3** = 0, see Section 3.3.8) allows direct specification of the log oxygen fugacity. The governing equation is just the identity:

$$\log f_{O_2} = \log f_{O_2} \tag{B–326}$$

The residuals are:

$$\alpha_s = 0 \tag{B–327}$$

$$\beta_s = 0 \tag{B–328}$$

The only non-zero Jacobian element in the corresponding row is:

$$J_{ss} = 1.0 \tag{B–329}$$

B.4.8.13 Specified *Eh* or *pe*

This option (**irdxc3** = –1) allows indirect specification of the log oxygen fugacity. If *pe* is specified (**irdxc3** = –2), EQ3NR converts it to *Eh* before it does the Newton-Raphson iteration. Letting *F* be the Faraday constant, *R* the gas constant, and *T* the absolute temperature, the governing equation can be written as:

$$\log f_{O_2} = \frac{4FEh}{2.303RT} + \log K_{Eh} + 2\left(\log x_w + \log \lambda_w\right) - 4\left(\log m_{H^+} + \log \gamma_{H^+}\right) \tag{B–330}$$

The residual functions are:

$$\alpha_{s_B} = \frac{4FEh}{2.303RT} + \log K_{Eh} + 2\left(\log x_w + \log \lambda_w\right) - 4\left(\log m_{H^+} + \log \gamma_{H^+}\right) - \log f_{O} \tag{B–331}$$

$$\beta_s = \alpha_s \tag{B–332}$$

The non-zero Jacobian elements in the corresponding row are:

$$J_{s_B w} = 2 \tag{B–333}$$

$$J_{s_B H^+} = -4 \tag{B–334}$$

$$J_{s_B s_B} = -1 \tag{B–335}$$

B.4.8.14 Oxygen Fugacity Fixed by An Aqueous Redox Couple

This represents the **irdxc3** = 1 option. Here s is restricted to s_B. The couple is specified on the input file by setting **uredox** to the name of the auxiliary basis species which comprises half of the couple (the other half is automatically the corresponding strict basis species; again, refer to Section 3.3.8). Let r denote the reaction associated with the auxiliary basis species (s_r) in the desired couple. The governing is then:

$$\log K_r = b_{s_r r} + \left(\log m_{s_r} + \log \gamma_{s_r}\right) + b_{wr}\left(\log x_w + \log \lambda_w\right) + b_{s_B r} \log f_{O_2} + \sum_{\substack{s'=1 \\ s' \neq w, s_B, s_r}}^{s_Q} b_{s'r}\left(\log m_{s'} + \log \gamma_{s'}\right) \tag{B–336}$$

The residual functions are defined as:

$$\alpha_s = \frac{1}{b_{s_r r}} \begin{pmatrix} \log Kr - b_{s_r r}\left(\log m_{s_r} + \log \gamma_{s_r} \ \log \lambda_w\right) \\ - b_{s_B r} \ \log f_{O_2} - \sum\limits_{\substack{s'=1 \\ s' \neq w, s_B, s_r}}^{s_Q} b_{s' r}\left(\log m_{s'} + \log \gamma_s\right) \end{pmatrix} \tag{B–337}$$

$$\beta_s = \alpha_s \tag{B–338}$$

The corresponding Jacobian elements are then:

$$J_{ss'} = -\frac{b_{s'r}}{b_{sr}} \tag{B–339}$$

B.4.9METHODS FOR CRASH DIAGNOSTICS

The iteration procedure is good enough that divergence almost always results from the input of a bad set of input constraints. EQ3NR screens the input prior to Newton-Raphson iteration, but some cases are not sufficiently obvious to be caught at this stage. When the iteration diverges, EQ3NR examines final state of the iteration process in an attempt to generate diagnostics to write on the screen and output files. Barring the generation of a useful diagnostic, the user should examine the iteration summary on the output file for clues to the cause of the situation.

An iteration crash is generally the result of an iteration variable “blowing up” or, more commonly, “blowing down.” “Blowing up” means that the value of an iteration variable or its corresponding residual function, usually a mass balance residual, is increasing to the point where the matrix equation can no longer be solved by the computer and iteration must terminate. Sometimes in this situation the operating system may terminate the code execution due to an overflow condition. If a variable “blows down,” it probably means that no physical solution exists for the problem as posed. Because the primary iteration variables are logarithmic, underflow can only occur when these variables are exponentiated. When this happens, the code is usually stopped by encountering what appears to be a singular Jacobian matrix.

“Blow down” occurs for example when the problem calls for the concentration of a species to be adjusted to satisfy electrical balance, but this balance can only be achieved if the resulting concentration takes a negative value. An iteration process which only allows adjustments to log concentrations will never allow the generation of concentration value which is non-positive. What happens instead is that the calculated adjustments to the log concentration become large and negative. These are truncated to a value of $-\delta'$ (–**screwd**). As this adjustment truncation parameter has a magnitude of 2.0, the largest adjustment (element of the **del** array) takes on a value of –2.0. This can be seen in the iteration summary that is printed on the output file as the code executes Newton-Raphson iteration.

An iteration block printed during this iteration is exemplified by the following:

```
 iter=   2
        del(conc    so4--   )= -1.77931E-03, delfnc= 9.34442E-01
       beta(conc    so4--   )= 8.87762E-05, betfnc= 9.79484E-01
         bbig= 8.87762E-05, ubbig= so4--
         bneg= 0.00000E+00, ubneg= none
         bgamx= 1.61604E-04, ubgamx= mg4(oh)4++++
         bsigmm= -1.18619E-04
         bxi= -3.19106E-04
         btfcnr= 9.97838E-01
```

The **del** element with the largest magnitude and the **beta** element with the largest magnitude are printed first. We see that in both cases, these are the elements corresponding to the concentration of the basis species SO_4^{2-}. “Blow down” would be evident if the **del** output in this block were to take on a value of –2.0 (usually for the last several iterations before the process is terminated). “Blow up” would be evident if the **beta** output in this block were to take on very high positive values. The variables **delfnc** and **betfnc** are convergence functions for the **del** and **beta** arrays, respectively. They usually start at small positive values less than unity and then approach unity if the iteration process converges. If one or the other takes on sustained negative values, the iteration process usually diverges. Here **bbig** is the largest positive mass balance residual (**ubbig** identifying the corresponding species) and **bneg** is the negative mass balance residual (if any) with the largest magnitude (**ubneg** identifying the corresponding species). The variable **bgamx** is an aqueous species activity coefficient residual function (the max norm on the absolute values of the differences between current and previous values of the activity coefficients of the aqueous species); **ubgamx** identifies the corresponding species. Also, **bsigm** is a residual on the Σm function (difference between the current and previous values), and **bxi** is a similar residual on the ionic strength. The variable **btfncr** is similar to **betfnc**, but measures the convergence in a pure Newton-Raphson step. Usually **betfnc** has a smaller value than **btfncr** because any adjustment of activity coefficient values in between Newton-Raphson steps tends to reduce the overall reduction in the residual functions.

B.5 EXAMPLE EQ3NR OUTPUT FILE

```
EQ3/6, Version 8.0 (EQ3/6-V8-REL-V8.0-PC)
EQ3NR Speciation-Solubility Code (EQ3/6-V8-EQ3NR-EXE-R43-PC)
Supported by the following EQ3/6 libraries:
  EQLIB (EQ3/6-V8-EQLIB-LIB-R43-PC)
  EQLIBG (EQ3/6-V8-EQLIBG-LIB-R43-PC)
  EQLIBU (EQ3/6-V8-EQLIBU-LIB-R43-PC)

Copyright (c) 1987, 1990-1993, 1995, 1997, 2002 The Regents of the
University of California, Lawrence Livermore National Laboratory.
All rights reserved.

This work is subject to additional statements and
disclaimers which may be found in the README.txt file
included in the EQ3/6 software transmittal package.


Run  17:17:26  27Aug2002


Reading the data1 file header section ...

Reading the rest of the DATA1 file ...

  The data file title is:

    data0.ymp.R0
    CII: GEMBOCHS.V2-EQ8-data0.ymp.R0
    THERMODYNAMIC DATABASE
    Data0.ymp.R0A was generated by GEMBOCHS.V2-Jewel.src.R9 24-mar-2000
    16:56:16
    Data0.ymp.R0 was modified from that file by a technical assessment
    in accordance with the development plan, TDP-EBS-MD-000044. Results
    are available in Data Qualification Report for Thermodynamic Data File,
    Data0.ymp.R0 for Geochemical Code, EQ3/6, TDR-EBS-MD-00012 REV 00.
    The user is referred to this report for details, especially limitations
    on the qualification effort, as noted particularly in Section 6.

    If for individual species (aqueous, solid, or gaseous) Data0.ymp.R0A
    values are the same as in Data0.com.R2, which was widely distributed
    and used, the assessment team performed little further examination.
    The assessment team checked all changes to species in Data0.com.R2 and
    all newly added species that include elements thought to be important
    for YMP applications, but not including elements of lesser importance,
    for accuracy compared to data sources. Namely, checks were made for
    species containing only one or more of the following elements:  Ag, Al,
    Am, B, Ba, C, Ca, Cd, Cl, Cr, Cs, Cu, F, Fe, Gd, H, Hf, I, In, K, Li,
    Mg, Mn, Mo, N, Na, Nb, Ni, Nd, Np, O, P, Pb, Pu, Ra, Rn, S, Si, Sm, Sn,
    Sr, Tc, Th, Ti, U, and Zr. This resulted in several corrections.

    The user is also cautioned to check on the temperature limitations for
    individual species. If a temperature other than 25 C is used for
    calculations and if equilibrium constants exist only for 25 C for a
    species involved in the calculation, EQ3/6 will use the 25 C data.
    This can result in errors. Users are advised to examine outputs
    carefully to determine whether such species have had a significant
    effect on the output.

    Output package:  eq3
    Data set:        ymp
```

```
     Last modified by the Data Qualification Assessment Team 11-sep-2000
     as described in the Data Qualification Report TDR-EBS-MD-000012 REV 00.

     YMP Qualification Status is indicated in data blocks (definition for
     Q data taken from YMP AP-SIII.2Q):

     Qualified (Q): Data acquired or developed under an approved
         Quality Assurance (QA) program that meets the
         requirements of 10 CFR 60 Subpart G (i.e., qualified
         from origin), or data that have undergone the
         qualification process (QARD).


   Continuing to read the DATA1 file ...

   Done reading the DATA1 file.


 Reading problem   1 from the input file ...

|------------------------------------------------------------------------------|
| Title                  | (utitl(n))                                          |
|------------------------------------------------------------------------------|
|EQ3NR input file name= sio2.3i                                                |
|Description= "Solution containing 101.0 mg/L SiO2"                            |
|Version level= 8.0                                                            |
|Revised 02/14/97    Revisor= T.J. Wolery                                      |
|This is part of the EQ3/6 Test Case Library                                   |
|                                                                              |
|  Solution containing 101.0 mg/L of dissolved SiO2. This test case is         |
|otherwise nearly identical to that in the EQ3NR test case input file          |
|deion.3i.                                                                    |
|                                                                              |
|  Purpose: to initialize the EQ6 test case input files pptqtz.6i and          |
|pptqtza.6i, both of which simulate the precipitation of quartz from           |
|supersaturated solution at 105C. The former EQ6 test case is test problem 1   |
|from Delany, Puigdomenech, and Wolery (1986, p. 19-21, 31-33), which          |
|simulates an experiment (Run 2E) reported by Rimstidt and Barnes (1980,       |
|p. 1687-1688, Figure 2) and uses their rate law model, which is independent  |
|of pH. The latter is the same problem, but uses a rate law model from         |
|Knauss and Wolery (1988) that does include a dependence on pH.               |
|                                                                              |
|  Note that the dissolved gases O2 and H2 have been suppressed. This is       |
|because this problem has no redox aspect. The EQ6 option switch iopt(15)      |
|should be set to 1 in the EQ6 test case input files so that EQ6 knows this.  |
|                                                                              |
|  Note that precipitation of the silica polymorphs tridymite, chalcedony,     |
|SiO2(am), and "Cristobalite(alpha)" has been prevented by means of nxmod      |
|suppress options. These options have no effect in EQ3NR, but will be passed   |
|on to EQ6 via the pickup file used to construct the EQ6 test case input       |
|files pptqtz.6i and pptqtza.6i.                                              |
|                                                                              |
|                                                                              |
|                                References                                    |
|                                                                              |
|Delany, J.M., Puigdomenech, I., and Wolery, T.J., 1986, Precipitation         |
|  Kinetics Option for the EQ6 Geochemical Reaction Path Code: UCRL-53642,     |
|  Lawrence Livermore National Laboratory, Livermore, California, 44 p.       |
|                                                                              |
|Knauss, K.G., and Wolery, T.J., 1988, The dissolution kinetics of quartz as   |
|  a function of pH and time at 70C: Geochimica et Cosmochimica Acta, v. 52,   |
|  p. 43-53.                                                                  |
|                                                                              |
```

```
|Rimstidt, J.D., and Barnes, H.L., 1980, The kinetics of silica-water          |
|  reactions: Geochimica et Cosmochimica Acta, v. 44, p. 1683-1699.           |
|                                                                              |
|------------------------------------------------------------------------------|
|Special Basis Switches (for model definition only)       | (nsbswt)           |
|------------------------------------------------------------------------------|
|Replace |None                                            | (usbsw(1,n))       |
|   with |None                                            | (usbsw(2,n))       |
|------------------------------------------------------------------------------|
|Temperature (C)         | 2.50000E+01| (tempc)                                |
|------------------------------------------------------------------------------|
|Pressure option (jpres3):                                                     |
|  [x] ( 0) Data file reference curve value                                    |
|  [ ] ( 1) 1.013-bar/steam-saturation curve value                             |
|  [ ] ( 2) Value (bars) | 0.00000E+00| (press)                                |
|------------------------------------------------------------------------------|
|Density (g/cm3)         | 1.00000E+00| (rho)                                  |
|------------------------------------------------------------------------------|
|Total dissolved solutes option (itdsf3):                                      |
|  [x] ( 0) Value (mg/kg.sol) | 0.00000E+00| (tdspkg)                          |
|  [ ] ( 1) Value (mg/L)      | 0.00000E+00| (tdspl)                           |
|------------------------------------------------------------------------------|
|Electrical balancing option (iebal3):                                         |
|  [ ] ( 0) No balancing is done                                               |
|  [x] ( 1) Balance on species |H+                      | (uebal)              |
|------------------------------------------------------------------------------|
|Default redox constraint (irdxc3):                                            |
|  [ ] (-3) Use O2(g) line in the aqueous basis species block                  |
|  [ ] (-2) pe (pe units)      | 0.00000E+00| (pei)                            |
|  [ ] (-1) Eh (volts)         | 0.00000E+00| (ehi)                            |
|  [x] ( 0) Log fO2 (log bars) | 0.00000E+00| (fo2lgi)                         |
|  [ ] ( 1) Couple (aux. sp.)  |None                    | (uredox)             |
|------------------------------------------------------------------------------|
|Aqueous Basis Species/Constraint Species        |Conc., etc. |Units/Constraint|
| (uspeci(n)/ucospi(n))                          | (covali(n))|(ujf3(jflgi(n)))|
|------------------------------------------------------------------------------|
|H+                                              | 5.50000E+00|pH              |
|SiO2(aq)                                        | 1.01000E+02|mg/L            |
|HCO3-                                           |-3.50000E+00|Hetero. equil. |
|->|CO2(g)                                          | (ucospi(n))              |
|O2(aq)                                          | 0.00000E+00|Suppressed      |
|H2(aq)                                          | 0.00000E+00|Suppressed      |
|------------------------------------------------------------------------------|
* Valid jflag strings (ujf3(jflgi(n))) are:                                    *
*    Suppressed          Molality            Molarity                          *
*    mg/L                mg/kg.sol           Alk., eq/kg.H2O                   *
*    Alk., eq/L          Alk., eq/kg.sol     Alk., mg/L CaCO3                  *
*    Alk., mg/L HCO3-    Log activity        Log act combo                     *
*    Log mean act        pX                  pH                                *
*    pHCl                Hetero. equil.     Homo. equil.                     *
*    Make non-basis                                                            *
*------------------------------------------------------------------------------*
|Create Ion Exchangers  | (net)                                                |
|------------------------------------------------------------------------------|
|Advisory: no exchanger creation blocks follow on this file.                  |
|Option: on further processing (writing a PICKUP file or running XCON3 on the  |
|present file), force the inclusion of at least one such block (qgexsh):       |
|  [ ] (.true.)                                                                |
|------------------------------------------------------------------------------|
|Ion Exchanger Compositions      | (neti)                                      |
|------------------------------------------------------------------------------|
|Exchanger phase |None                    | (ugexpi(n))                        |
|------------------------------------------------------------------------------|
|->|Moles/kg.H2O    |  0.0000    | (cgexpi(n))                                 |
```

```
|------------------------------------------------------------------------------|
|->|Exchange site   |None    | (ugexji(j,n))                                   |
|------------------------------------------------------------------------------|
|--->|Exchange species        |Mole frac. | (this is a table header)          |
|------------------------------------------------------------------------------|
|--->|None                    | 0.00000E+00| (ugexsi(i,j,n), xgexsi(i,j,n))    |
|------------------------------------------------------------------------------|
|Solid Solution Compositions     | (nxti)                                      |
|------------------------------------------------------------------------------|
|Solid Solution          |None                    | (usoli(n))                 |
|------------------------------------------------------------------------------|
|->|Component               |Mole frac. | (this is a table header)           |
|------------------------------------------------------------------------------|
|->|None                    | 0.00000E+00| (umemi(i,n), xbari(i,n))            |
|------------------------------------------------------------------------------|
|Alter/Suppress Options  | (nxmod)                                             |
|------------------------------------------------------------------------------|
|Species                                         |Option          |Alter value |
| (uxmod(n))                                     |(ukxm(kxmod(n)))| (xlkmod(n))|
|------------------------------------------------------------------------------|
|Tridymite                                       |Suppress        | 0.00000E+00|
|Chalcedony                                      |Suppress        | 0.00000E+00|
|SiO2(am)                                        |Suppress        | 0.00000E+00|
|Cristobalite(alpha)                             |Suppress        | 0.00000E+00|
|------------------------------------------------------------------------------|
* Valid alter/suppress strings (ukxm(kxmod(n))) are:                           *
*    Suppress            Replace             AugmentLogK                       *
*    AugmentG                                                                  *
*------------------------------------------------------------------------------*
|Iopt Model Option Switches ("( 0)" marks default choices)                     |
|------------------------------------------------------------------------------|
|iopt(4) - Solid Solutions:                                                    |
|  [x] ( 0) Ignore                                                             |
|  [ ] ( 1) Permit                                                             |
|------------------------------------------------------------------------------|
|iopt(11) - Auto Basis Switching in pre-N-R Optimization:                      |
|  [x] ( 0) Turn off                                                           |
|  [ ] ( 1) Turn on                                                            |
|------------------------------------------------------------------------------|
|iopt(17) - PICKUP File Options:                                               |
|  [ ] (-1) Don't write a PICKUP file                                          |
|  [x] ( 0) Write a PICKUP file                                                |
|------------------------------------------------------------------------------|
|iopt(19) - Advanced EQ3NR PICKUP File Options:                                |
|  [x] ( 0) Write a normal EQ3NR PICKUP file                                   |
|  [ ] ( 1) Write an EQ6 INPUT file with Quartz dissolving, relative rate law  |
|  [ ] ( 2) Write an EQ6 INPUT file with Albite dissolving, TST rate law       |
|  [ ] ( 3) Write an EQ6 INPUT file with Fluid 1 set up for fluid mixing       |
|------------------------------------------------------------------------------|
|Iopg Activity Coefficient Option Switches ("( 0)" marks default choices)      |
|------------------------------------------------------------------------------|
|iopg(1) - Aqueous Species Activity Coefficient Model:                         |
|  [ ] (-1) The Davies equation                                                |
|  [x] ( 0) The B-dot equation                                                 |
|  [ ] ( 1) Pitzer's equations                                                 |
|  [ ] ( 2) HC + DH equations                                                  |
|------------------------------------------------------------------------------|
|iopg(2) - Choice of pH Scale (Rescales Activity Coefficients):                |
|  [ ] (-1) "Internal" pH scale (no rescaling)                                 |
|  [x] ( 0) NBS pH scale (uses the Bates-Guggenheim equation)                  |
|  [ ] ( 1) Mesmer pH scale (numerically, pH = -log m(H+))                     |
|------------------------------------------------------------------------------|
|Iopr Print Option Switches ("( 0)" marks default choices)                     |
|------------------------------------------------------------------------------|
```

```
|iopr(1) - Print All Species Read from the Data File:                          |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print                                                              |
|------------------------------------------------------------------------------|
|iopr(2) - Print All Reactions:                                                |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print the reactions                                                |
|  [ ] ( 2) Print the reactions and log K values                               |
|  [ ] ( 3) Print the reactions, log K values, and associated data             |
|------------------------------------------------------------------------------|
|iopr(3) - Print the Aqueous Species Hard Core Diameters:                      |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print                                                              |
|------------------------------------------------------------------------------|
|iopr(4) - Print a Table of Aqueous Species Concentrations, Activities, etc.:  |
|  [ ] (-3) Omit species with molalities < 1.e-8                               |
|  [ ] (-2) Omit species with molalities < 1.e-12                              |
|  [ ] (-1) Omit species with molalities < 1.e-20                              |
|  [x] ( 0) Omit species with molalities < 1.e-100                             |
|  [ ] ( 1) Include all species                                                |
|------------------------------------------------------------------------------|
|iopr(5) - Print a Table of Aqueous Species/H+ Activity Ratios:                |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print cation/H+ activity ratios only                               |
|  [ ] ( 2) Print cation/H+ and anion/H+ activity ratios                       |
|  [ ] ( 3) Print ion/H+ activity ratios and neutral species activities        |
|------------------------------------------------------------------------------|
|iopr(6) - Print a Table of Aqueous Mass Balance Percentages:                  |
|  [ ] (-1) Don't print                                                        |
|  [x] ( 0) Print those species comprising at least 99% of each mass balance   |
|  [ ] ( 1) Print all contributing species                                     |
|------------------------------------------------------------------------------|
|iopr(7) - Print Tables of Saturation Indices and Affinities:                  |
|  [ ] (-1) Don't print                                                        |
|  [x] ( 0) Print, omitting those phases undersaturated by more than 10 kcal   |
|  [ ] ( 1) Print for all phases                                               |
|------------------------------------------------------------------------------|
|iopr(8) - Print a Table of Fugacities:                                        |
|  [ ] (-1) Don't print                                                        |
|  [x] ( 0) Print                                                              |
|------------------------------------------------------------------------------|
|iopr(9) - Print a Table of Mean Molal Activity Coefficients:                  |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print                                                              |
|------------------------------------------------------------------------------|
|iopr(10) - Print a Tabulation of the Pitzer Interaction Coefficients:         |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print a summary tabulation                                         |
|  [ ] ( 2) Print a more detailed tabulation                                   |
|------------------------------------------------------------------------------|
|iopr(17) - PICKUP file format ("W" or "D"):                                   |
|  [x] ( 0) Use the format of the INPUT file                                   |
|  [ ] ( 1) Use "W" format                                                     |
|  [ ] ( 2) Use "D" format                                                     |
|------------------------------------------------------------------------------|
|Iodb Debugging Print Option Switches ("( 0)" marks default choices)           |
|------------------------------------------------------------------------------|
|iodb(1) - Print General Diagnostic Messages:                                  |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print Level 1 diagnostic messages                                  |
|  [ ] ( 2) Print Level 1 and Level 2 diagnostic messages                      |
|------------------------------------------------------------------------------|
|iodb(3) - Print Pre-Newton-Raphson Optimization Information:                  |
|  [x] ( 0) Don't print                                                        |
```

```
|  [ ] ( 1) Print summary information                                          |
|  [ ] ( 2) Print detailed information (including the beta and del vectors)    |
|  [ ] ( 3) Print more detailed information (including matrix equations)       |
|  [ ] ( 4) Print most detailed information (including activity coefficients)  |
|------------------------------------------------------------------------------|
|iodb(4) - Print Newton-Raphson Iteration Information:                         |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print summary information                                          |
|  [ ] ( 2) Print detailed information (including the beta and del vectors)    |
|  [ ] ( 3) Print more detailed information (including the Jacobian)           |
|  [ ] ( 4) Print most detailed information (including activity coefficients)  |
|------------------------------------------------------------------------------|
|iodb(6) - Print Details of Hypothetical Affinity Calculations:                |
|  [x] ( 0) Don't print                                                        |
|  [ ] ( 1) Print summary information                                          |
|  [ ] ( 2) Print detailed information                                         |
|------------------------------------------------------------------------------|
|Numerical Parameters                                                          |
|------------------------------------------------------------------------------|
| Beta convergence tolerance      | 0.00000E+00| (tolbt)                       |
| Del convergence tolerance       | 0.00000E+00| (toldl)                       |
| Max. Number of N-R Iterations   |   0        | (itermx)                      |
|------------------------------------------------------------------------------|
|Ordinary Basis Switches (for numerical purposes only)    | (nobswt)           |
|------------------------------------------------------------------------------|
|Replace |None                                            | (uobsw(1,n))       |
|   with |None                                            | (uobsw(2,n))       |
|------------------------------------------------------------------------------|
|Sat. flag tolerance     | 0.00000E+00| (tolspf)                               |
|------------------------------------------------------------------------------|
|Aq. Phase Scale Factor  | 1.00000E+00| (scamas)                               |
|------------------------------------------------------------------------------|
|End of problem                                                                |
|------------------------------------------------------------------------------|

   Done reading problem   1.


 The phase Tridymite has been user-suppressed.

 The phase Chalcedony has been user-suppressed.

 The phase SiO2(am) has been user-suppressed.

 The phase Cristobalite(alpha) has been user-suppressed.

 The activity coefficients of aqueous species will be
 calculated using the B-dot equation.


 Temperature=  25.00 C


 jpres3=    0 (Pressure option switch)

   Pressure=   1.0132     bars (data file reference curve value)


       --- Numbers of Phases, Species, and Groups Thereof---

       Entity               Date Base    Dimension   Current Problem

   Chemical Elements             86          86           5
   Basis Species                185         195          12
```

```
  Phases                      1046        1070          13
  Species                     2320        2812          37
  Aqueous Species             1160        1160          19
  Pure Minerals               1033        1033          10
  Pure Liquids                   1           3           1
  Gas Species                   95          95           7
  Solid Soutions                10          10           0


iopt(1)=   0 (Used only by EQ6)
iopt(2)=   0 (Used only by EQ6)
iopt(3)=   0 (Used only by EQ6)
iopt(4)=   0 (Solid solutions)
iopt(5)=   0 (Used only by EQ6)
iopt(6)=   0 (Used only by EQ6)
iopt(7)=   0 (Not used)
iopt(8)=   0 (Not used)
iopt(9)=   0 (Not used)
iopt(10)=  0 (Not used)
iopt(11)=  0 (Auto basis switching, in pre-Newton-Raphson optimization)
iopt(12)=  0 (Used only by EQ6)
iopt(13)=  0 (Not used)
iopt(14)=  0 (Not used)
iopt(15)=  0 (Used only by EQ6)
iopt(16)=  0 (Not used)
iopt(17)=  0 (pickup file options)
iopt(18)=  0 (Used only by EQ6)
iopt(19)=

  iopg(1)=   0 (Aqueous species activity coefficient model)
  iopg(2)=   0 (pH scale)


iopr(1)=   0 (List all species)
iopr(2)=   0 (List all reactions)
iopr(3)=   0 (List HC diameters)
iopr(4)=   0 (Aqueous species concentration print cut-off)
iopr(5)=   0 (Ion/H+ activity ratios)
iopr(6)=   0 (Mass balance percentages)
iopr(7)=   0 (Affinity print cut-off)
iopr(8)=   0 (Fugacities)
iopr(9)=   0 (Mean molal activity coefficients)
iopr(10)=  0 (Pitzer coefficients tabulation)
iopr(11)=  0 (Not used)
iopr(12)=  0 (Not used)
iopr(13)=  0 (Not used)
iopr(14)=  0 (Not used)
iopr(15)=  0 (Not used)
iopr(16)=  0 (Not used)
iopr(17)=  0 (pickup file format)


  iodb(1)=   0 (General diagnostics)
  iodb(2)=   0 (Used only by EQ6)
  iodb(3)=   0 (pre-Newton-Raphson optimization iterations)
  iodb(4)=   0 (Newton-Raphson iterations)
  iodb(5)=   0 (Used only by EQ6)
  iodb(6)=   0 (Hypothetical affinity iterations)
  iodb(7)=   0 (Used only by EQ6)


irdxc3=    0 (Default redox constraint switch)

  The default redox state is constrained by Log fO2 =   0.0000     (log bars).
```

```
iebal3=    1 (Electrical balancing option switch)

  The species H+ will be adjusted to
  achieve electrical balance.


Solution density =  1.00000 g/ml


itdsf3=    0 (Total dissolved solutes option switch)

  Total dissolved salts =       0.00 mg/kg.sol


tolbt  =  1.00000E-06 (convergence tolerance on residual functions)
toldl  =  1.00000E-06 (convergence tolerance on correction terms)
tolspf =  5.00000E-05 (saturation print flag tolerance, does not affect
                        convergence)


itermx = 200 (maximum number of iterations)


scamas =  1.00000E+00 (scale factor for aqueous solution
                        mass written on the pickup file)


                    --- Original Input Constraints ---

    Species                    coval   jflag   Type of Input

 H+                         5.50000E+00  20  pH
 SiO2(aq)                   1.01000E+02   2  Total mg/L
 HCO3-                     -3.50000E+00  25  Heterogenous equilibrium
                                             Species= CO2(g)
                                               Phase= CO2(g)

     1.000  CO2(g) (Gas)
 +   1.000  H2O
         ==
     1.000  H+
 +   1.000  HCO3-



                    --- Modified Input Constraints ---

    Species                    coval   jflag   Type of Input

 H2O                        0.00000E+00   0  Total molality
 H+                         5.50000E+00  20  pH
 HCO3-                                   25  Heterogenous equilibrium
                                             Species= CO2(g)
                                               Phase= Gas

     1.000  CO2(g) (Gas)
 +   1.000  H2O
         ==
     1.000  H+
 +   1.000  HCO3-

 SiO2(aq)                   1.68097E-03   0  Total molality
```

```
 Acetic acid(aq)                        30  Make non-basis
 Formic_acid(aq)                        30  Make non-basis
 Ethane(aq)                             30  Make non-basis
 Formaldehyde(aq)                       30  Make non-basis
 Methane(aq)                            30  Make non-basis
 Methanol(aq)                           30  Make non-basis


Electrical balance will be achieved by adjusting
  the concentration of H+.


                    --- Inactive Species ---

   Chalcedony
   Cristobalite(alpha)
   SiO2(am)
   Tridymite

- - BEGIN ITERATIVE CALCULATIONS  - - - - - - - - - - - - - - - - - - - - - - - -


Starting Pre-Newton-Raphson Optimization.

            Completed pass   1 in   1 cycles.
            Completed pass   2 in   1 cycles.

  Done. Optimization ended within requested limits.


Starting hybrid Newton-Raphson iteration.

  Done. Hybrid Newton-Raphson iteration converged in   3 iterations.


- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -


          --- Elemental Composition of the Aqueous Solution ---

  Element         mg/L         mg/kg.sol        Moles/kg.H2O

    O          0.88816E+06    0.88816E+06    5.5511825023E+01
    Cl          0.0000         0.0000        0.0000000000E+00
    H          0.11190E+06    0.11190E+06    1.1101687474E+02
    C          0.15440        0.15440        1.2855161492E-05
    Si          47.211         47.211        1.6809716481E-03



          --- Numerical Composition of the Aqueous Solution ---

  Species                     mg/L         mg/kg.sol         Molality

H2O                         0.10000E+07    0.10000E+07    5.5508424514E+01
H+                          0.12957E-01    0.12957E-01    1.2855161492E-05
HCO3-                       0.78438        0.78438        1.2855161492E-05
SiO2(aq)                     101.00         101.00        1.6809716481E-03
O2(g)                       0.13492E-13    0.13492E-13    4.2165487379E-19

  Some of the above data may not be physically significant.


          --- Sensible Composition of the Aqueous Solution ---
```

```
  Species                    mg/L         mg/kg.sol        Molality

H+                         0.12957E-01    0.12957E-01    1.2855161492E-05
HCO3-                      0.78438        0.78438        1.2855161492E-05
SiO2(aq)                    101.00         101.00        1.6809716481E-03

  The above data have physical significance, but some may be
  inconsistent with certain analytical methods or reporting schemes.



                  Oxygen fugacity=  1.00000E+00 bars
              Log oxygen fugacity=   0.0000

                Activity of water=  9.99969E-01
            Log activity of water= -1.32705E-05

           Mole fraction of water=  9.99969E-01
       Log mole fraction of water= -1.32703E-05

    Activity coefficient of water=  1.00000E+00
 Log activity coefficient of water= -1.81288E-10

              Osmotic coefficient=   1.0000
Stoichiometric osmotic coefficient=  0.99382

                Sum of molalities=  1.69614E-03
  Sum of stoichiometric molalities=  1.70668E-03

               Ionic strength (I)=  2.30725E-06 molal
    Stoichiometric ionic strength=  1.28552E-05 molal

              Ionic asymmetry (J)= -4.32576E-11 molal
   Stoichiometric ionic asymmetry=  5.08514E-18 molal

                 Solvent fraction=   1.0000     kg.H2O/kg.sol
                  Solute fraction= -2.08167E-17 kg.solutes/kg.sol


          --- The pH, Eh, pe-, and Ah on various pH scales ---

                            pH     Eh, volts        pe-       Ah, kcal

NBS pH scale              5.6377     0.8955     1.5139E+01     20.6533
Mesmer pH scale           5.6369     0.8956     1.5139E+01     20.6543

  The single ion activities and activity coefficients listed below
  are consistent with the NBS pH scale.

    The pHCl is undefined because no Cl- is present.


          --- HCO3-CO3-OH Total Alkalinity ---

                0.211724E-05 eq/kg.H2O
                0.105862     mg/kg.sol CaCO3
                0.129067     mg/kg.sol HCO3-
                0.105862     mg/L CaCO3
                0.129067     mg/L HCO3-


          --- Extended Total Alkalinity ---
```

```
                0.211724E-05 eq/kg.H2O
                0.105862     mg/kg.sol CaCO3
                0.129067     mg/kg.sol HCO3-
                0.105862     mg/L CaCO3
                0.129067     mg/L HCO3-


          --- Electrical Balance Totals ---

                              eq/kg.H2O

       Sigma(mz) cations=     2.3072017945E-06
        Sigma(mz) anions=    -2.3072017944E-06
            Total charge=     4.6144035889E-06
             Mean charge=     2.3072017945E-06
        Charge imbalance=     3.0511434946E-17


        The electrical imbalance is:

             0.0000 per cent of the total charge
             0.0000 per cent of the mean charge


     --- Electrical Balancing on H+ ---

               pH

    Input        5.5000
    Final        5.6377
    Adj          0.1377



               --- Distribution of Aqueous Solute Species ---

   Species                    Molality    Log Molality   Log Gamma  Log Activity

SiO2(aq)                     1.6808E-03      -2.7745       0.0000      -2.7745
CO2(aq)                      1.0742E-05      -4.9689       0.0000      -4.9689
H+                           2.3072E-06      -5.6369      -0.0008      -5.6377
HCO3-                        2.1128E-06      -5.6752      -0.0008      -5.6759
HSiO3-                       1.8996E-07      -6.7213      -0.0008      -6.7221
OH-                          4.3990E-09      -8.3567      -0.0008      -8.3574
CO3--                        4.3258E-11     -10.3639      -0.0031     -10.3670
H2O2(aq)                     8.4331E-19     -18.0740       0.0000     -18.0740
HO2-                         7.5866E-25     -24.1200      -0.0008     -24.1207
Formate                      9.3036E-48     -47.0314      -0.0008     -47.0321
Formic_acid(aq)              1.2111E-49     -48.9168       0.0000     -48.9168
Formaldehyde(aq)             1.0287E-95     -94.9877       0.0000     -94.9877

   Species with molalities less than  1.000-100 are not listed.



     --- Major Species by Contribution to Aqueous Mass Balances ---


Species Accounting for 99% or More of Aqueous HCO3-

    Species                     Factor    Molality     Per Cent

  CO2(aq)                        1.00    1.0742E-05     83.56
  HCO3-                          1.00    2.1128E-06     16.44
```

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  Subtotal                             1.2855E-05    100.00


Species Accounting for 99% or More of Aqueous SiO2(aq)

    Species                   Factor    Molality     Per Cent

  SiO2(aq)                     1.00    1.6808E-03     99.99
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  Subtotal                             1.6808E-03     99.99



               --- Aqueous Redox Reactions ---

  Couple                            Eh, volts      pe-      log fO2   Ah, kcal

DEFAULT                               0.896    1.5139E+01     0.000    20.653

   Couples required to satisfy the default redox constraint are not listed.



     --- Saturation States of Aqueous Reactions Not Fixed at Equilibrium ---

  Reaction                            Log Q/K    Affinity, kcal

None



          --- Saturation States of Pure Solids ---

      Phase                        Log Q/K    Affinity, kcal

    Chalcedony                      0.95361      1.30100     SSATD
    Coesite                         0.41481      0.56592     SSATD
    Cristobalite(alpha)             0.67431      0.91995     SSATD
    Cristobalite(beta)              0.23081      0.31489     SSATD
    Ice                            -0.13871     -0.18924
    Quartz                          1.22481      1.67099     SSATD
    SiO2(am)                       -0.06089     -0.08307
    Tridymite                       1.05331      1.43702     SSATD

       Phases with affinities less than -10 kcal are not listed.



          --- Saturation States of Pure Liquids ---

      Phase                        Log Q/K    Affinity, kcal

    H2O                            -0.00001     -0.00002     SATD

       Phases with affinities less than -10 kcal are not listed.


          --- Summary of Saturated and Supersaturated Phases ---

               There is 1 saturated phase.
               There are    6 supersaturated phases.
```

```
                    --- Fugacities ---

    Gas                    Log Fugacity    Fugacity

  O2(g)                        0.00000    1.00000E+00
  H2O(g)                      -1.58541    2.59769E-02
  CO2(g)                      -3.50000    3.16228E-04
  H2(g)                      -41.55251    2.80212E-42
  CO(g)                      -48.55790    2.76758E-49
  C(g)                      -190.18760    6.49232-191
  Si(g)                     -219.82709    1.48906-220


- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -


The pickup file has been written.

No further input found.


         Start time = 17:17:26  27Aug2002
           End time = 17:17:27  27Aug2002

          Run time =  0.620     seconds

Normal exit
```

B.6 REFERENCES

Ananthaswamy, J. and Atkinson, G., 1984. "Thermodynamics of concentrated electrolyte mixtures. 4. Pitzer-Debye-Hückel limiting slopes for water from 0 to 100°C and from 1 atm to 1 kbar." *Journal of Chemical and Engineering Data, 29*, 81–87.

Archer, D. G., 1990. "Effect of revisions of Debye-Hückel limiting law coefficients on the thermodynamic parameters for strong-electrolyte solutions." *Journal of Chemical and Engineering Data, 35*, 340–344. TIC: 253628.

Baes, C.F., Jr. and Mesmer, R.E. 1976. *The Hydrolysis of Cations.* New York, New York: John Wiley & Sons. TIC: 217440.

Ball, J.W., Jenne, E.A., and Nordstrom, D.K. 1979. "WATEQ2: A computerized chemical model for trace and major element speciation and mineral equilibria for natural water." Jenne, E.A., editor. *Chemical Modeling in Aqueous Systems, Speciation, Sorption, Solubility, and Kinetics.* ACS Symposium Series 93. Washington, D.C.: American Chemical Society. TIC: 208922.

Barta, L. and Bradley, D. J., 1985. "Extension of the specific interaction model to include gas solubilities in high temperature brines." *Geochimica et Cosmochimica Acta, 49*, 195–203. New York, NY: Pergamon Press. TIC: 253629.

Bates, R.G. 1973a. *Determination of pH: Theory and Practice*, Second Edition. New York, NY: John Wiley & Sons. TIC: 245388.

Bates, R. G., 1973b. "Ion activity scales for use with ion-selective electrodes." *Pure and Applied Chemistry, 36*, 407–420.

Bates, R.G. and Alfenaar, M. 1969. "Activity Standards for Ion-Selective Electrodes." *Ion-Selective Electrodes.* Durst, R.A., ed. Proceedings of an NBS Symposium, NBS Special Publication 314, 191–214. Washington, D.C.: U.S. Government Printing Office. TIC: 235627.

Bates, R.G. and Robinson, R.A. 1974. "An Approach to Conventional Scales of Ionic Activity for the Standardization of Ion-Selective Electrodes." *Pure and Applied Chemistry, 37*, 575–577. London, England: Butterworths. TIC: 253630.

Bourcier, W.L. 1985. *Improvements in the Solid Solution Modeling Capabilities of the EQ3/6 Geochemical Code*. UCID-20587. Livermore, CA: Lawrence Livermore National Laboratory. TIC: 209067.

Bourcier, W.L. 1986. "Improvements in the Solid Solution Modeling Capabilities of EQ3/6." *Proceedings of the Workshop on Geochemial Modeling CONF-8609134,* Editors Jackson, K.J. and Bourcier, W.L. Livermore, CA: Lawrence Livermore National Laboratory, 41–48. TIC: 202125.

Busenberg, E. 1978. "The Products of the Interaction of Feldspars with Aqueous Solutions at 25ºC." *Geochimica et Cosmochimica Acta, 42*, 1679–1686. New York, NY: Pergamon Press. TIC: 253600.

Carnahan, B., Luther, H.A., and Wilkes, J.O. 1969. "Approximation of the Solution of Partial Differential Equations." *Applied Numerical Methods*, 429–530. New York, NY: John Wiley & Sons. TIC: 245078.

Carothers, W.W. and Kharaka, Y.K. 1978. "Aliphatic Acid Anions in Oil-Field Waters-Implications for Origin of Natural Gas." *The American Association of Petroleum Geologists Bulletin, 62*, 2441–2453. Tulsa, OK: American Association of Petroleum Geologists. TIC: 253596.

Chen, C.-T. A., and Marshall, W. L., 1981. "Amorphous silica solubilities IV. Behavior in pure water and aqueous sodium chloride, sodium sulfate, magnesium chloride, and magnesium sulfate solutions up to 350°C." *Geochimica et Cosmochimica Acta, 46*, 279–287. New York, NY: Pergamon Press. TIC: 235346.

Clegg, S. L., and Brimblecombe, P., 1989. "Solubility of ammonia in pure aqueous and multicomponent solutions." *Journal of Physical Chemistry, 93*, 7237–7248. Washington, D.C.: American Chemical Society. TIC: 253631.

Clegg, S. L., and Brimblecombe, P., 1990. "Solubility of volatile electrolytes in multicomponent solutions with atmospheric applications." Editors: Melchior, D. C., and Bassett, R. L. *Chemical Modeling of Aqueous Systems II. American Chemical Society Symposium Series 416*, 58–73. Washington, D. C.: American Chemical Society. TIC: 241139.

Connaughton, L.M., Millero, F.J. and Pitzer, K.S. 1989. "Volume Changes for Mixing the Major Sea Salts: Equations Valid to Ionic Strenght 3.0 and Temperature 95ºC." *Journal of Solution Chemistry*, *18*, 1007–1017. New York, NY: Plenum Press. TIC: 253590.

Crerar, D.A. 1975. "A Method for Computing Multicomponent Equilibria Based on Equilibrium Constants." *Geochimica et Cosmochimica Acta, 39*, 1375–1384. New York, NY: Pergamon Press. TIC: 241122.

Daveler, S.A. and Wolery, T.J. 1992. *EQPT, A Data File Preprocessor for the EQ3/6 Software Package: User's Guide and Related Documentation (Version 7.0).* UCRL-MA-110662 PT II. Livermore, CA: Lawrence Livermore National Laboratory. TIC: 205240**.**

Davies, C. W., 1962. *Ion Association.* Washington, D.C.: Butterworths. TIC: 244840.

Drummond, S.E., Jr. 1981. *Boiling and Mixing of Hydrothermal Fluids: Chemical Effects on Mineral Precipitation.* Ph.D. dissertation. The Pennsylvania State University, University Park, PA. TIC: 246545.

Felmy, A. R., and Weare, J. H., 1986. "The prediction of borate mineral equilibria in natural waters: Application to Searles Lake, California." *Geochimica et Cosmochimica Acta, 50*, 2771–2783. New York, NY: Pergamon Press. TIC: 248987.

Franson, M.A. 1976. *Standard Methods for the Examination of Water and Wastewater,* 14th edition. Washington, D.C: American Public Health Association. TIC: 217558.

Garrels, R. M., and Christ, C. L. 1965. *Solutions, Minerals and Equilibria.* San Francisco, CA: Freeman, Cooper and Company. TIC: 217443.

Garrels, R. M., and Thompson, M. E. 1962. "A chemical model for sea water at 25°C and one atmosphere total pressure." *American Journal of Science, 260*, 57–66. New Haven, CT: Kline Geology Laboratory, Yale University. TIC: 239179.

Greenberg, J. P., and Møller, N. 1989. "The prediction of mineral solubilities in natural waters: A chemical equilibrium model for the Na-K-Ca-Cl-SO_4-H_2O system to high concentration from 0 to 250°C." *Geochimica et Cosmochimica Acta, 53*, 2503–2518. New York, NY: Pergamon Press. TIC: 249020.

Harvie, C.E., Møller, N. and Weare J.H. 1984. "The Prediction of Mineral solubilities in Natural Waters: The Na-K-Mg-Ca-H-Cl-SO_4-OH-HCO_3-CO_3-$CO_2$$H_2O$ System to High Ionic Strengths at 25ºC." *Geochimica et Cosmochimica Acta*, *48*, 723–751. New York, NY: Pergamon Press. TIC: 239849.

Harvie, C.E. and Weare, J.H. 1980. "The Prediction of Mineral Solubilities in Natural Waters: The Na-K-Mg-Ca-Cl-SO_4-H_2O System from Zero to High Concentrations at 25ºC." *Geochimica et Cosmochimica Acta, 34*, 569-592. New York, NY: Pergamon Press. TIC: 235515.

Helgeson, H. C., 1969. "Thermodynamics of hydrothermal systems at elevated temperatures and pressures." *American Journal of Science, 267*, 729–804. New Haven, CT: Scholarly Publications. TIC: 241817.

Helgeson, H.C., Brown, T.H., Nigrini, A. and Jones, T.A. 1970. "Calculation of Mass Transfer in Geochemical Processes Involving Aqueous Solutions." *Geochimica et Cosmochimica Acta, 34*, 569-592. New York, NY: Pergamon Press. TIC: 225570.

Helgeson, H.C., Delany, J.M., Nesbitt, H.W. and Bird, D.K. 1978. "Summary and critique of the Thermodynamic Properties of Rock-Forming Minerals." *American Journal of Science, 278-A*, 1–229. New Haven, CT: Kline Geology Laboratory, Yale University. TIC: 220013.

Helgeson, H.C.; Kirkham, D.H.; Flowers, G.C. 1982. "Theoretical prediction of the thermodynamic behavior of aqueous electrolytes at high pressures and temperatures: Parts I through IV." New Haven, CT: Kline Geology Laboratory, Yale University. TIC: 243502.

Holmes, H. F., and Mesmer, R. E. 1986a. "Isopiestic studies of aqueous solutions at elevated temperatures VIII. The alkali-metal sulfates." *Journal of Chemical Thermodynamics, 18*, 263–275.

Holmes, H. F., and Mesmer, R. E. 1986b. "Thermodynamics of aqueous solutions of the alkali metal sulfates." *Journal of Solution Chemistry, 15*, 495–518. New York, NY: Plenum Publishing Corporation. TIC: 253633.

Hostettler, J.D. 1984. "Electrode Electrons, Aqueous Electrons, and Redox Potentials in Natural Waters." *American Journal of Science*, *284*, 734–759. New Haven, CT: Kline Geology Laboratory, Yale University. TIC: 248082.

Huang, H.-H., 1989. “Estimation of Pitzer’s ion interaction parameters for electrolytes involved in complex formation using a chemical equilibrium model.” *Journal of Solution Chemistry, 18*, 1069–1084. New York, NY: Plenum Publishing Corporation. TIC: 253634.

Jackson, K. J., 1988. “Verification and Validation Studies of the Addition of Pitzer’s Equations to the EQ3/6 Brine Model.” UCRL-53841. Livermore, California: Lawrence Livermore National Laboratory. TIC: 212964.

Jenne, E.A. 1981. “Geochemical Modeling: A Review.” PNL-3574. Richland, WA: Battelle Pacific Northwest Laboratory. TIC: 229656.

Knauss, K.G., Wolery, T.J. and Jackson K.J. 1990. “A New Approach to Measuring pH in Brines and Other Concentrated Electrolytes.” *Geochimica et Cosmochimica Acta*, *54*, 1519–1523. New York, NY: Pergamon Press. TIC: 253592.

Knauss, K.G., Wolery, T.J. and Jackson K.J. 1991. “Reply to comment by R. E. Mesmer on ‘A New Approach to Measuring pH in Brines and Other Concentrated Electrolytes.’ ” *Geochimica et Cosmochimica Acta*, *55*, 1177–1179. New York, NY: Pergamon Press. TIC: 253593.

Kodytek, V., and Dolejs, V. 1986. “On the Pitzer equation for the osmotic coefficient of aqueous solutions of 3:1 rare earth electrolytes.” *Zeitschrift für Physikalische Chemie, 267*, 743–746. Leipzig, Germany: Akademische Verlagsgesellschaft Geest & Portig. TIC: 253635.

Kumar, A. (1986). “Prediction of Densities of Concentrated Brines by Pitzer Theory.” *Journal of Chemical and Engineering Data*, *31*, 19–20. Washington, D.C.: American Chemical Society. TIC: 253589.

Langmuir, D. 1979. “Techniques of Estimating Thermodynamic Properties for Some Aqueous Complexes of Geochemical Interest.” Chapter 18 of *Chemical Modeling in Aqueous Systems, Speciation, Sorption, Solubility, and Kinetics.* Jenne, E.A., ed. ACS Symposium Series 93. Pages 353–387. Washington, D.C.: American Chemical Society. TIC: 208922.

Laxen, D.P.H. and Chandler, I.M. 1982. “Comparison of Filtration Techniques for Size Distribution in Freshwaters.” *Analytical Chemistry*, *54,* 1350–1355. Washington, D.C.: American Chemical Society. TIC: 253601.

Lindberg, R.D. and Runnells, D.D. 1984. "Ground Water Redox Reactions: An Analysis of Equilibrium State Applied to Eh Measurements and Geochemical Modeling." *Science, 225*, 925–927. Washington, D.C.: American Association for the Advancement of Science. TIC: 224111.

Merino, E. 1979. “Internal Consistency of a Water Analysis and Uncertainty of the Calculated Distribution of Aqueous Species at 25ºC.” *Geochimica et Cosmochimica Acta, 43,* 1533–1542. New York, NY: Pergamon Press. TIC: 253602.

Mesmer, R.E. 1991. “Comments on ‘A New Approach to Measuring pH in Brines and Other Concentrated Electrolytes’ by K. G. Knauss, T. J. Wolery, and K. J. Jackson.” *Geochimica et Cosmochimica Acta, 55*, 1175–1176. New York, NY: Pergamon Press. TIC: 253594.

Millero, F.J. 1977. "The use of the Specific Interaction Model to Estimate the Partial Molal Volumes of Electrolytes in Seawater." *Geochimica et Cosmochimica Acta*, *41*, 215–223. New York, NY: Pergamon Press. TIC: 253588.

Millero, F. J. and Byrne, R. H. 1984. "Use of Pitzer's equations to determine the media effect on the formation of lead chloro complexes." *Geochimica et Cosmochimica Acta, 48*, 1145–1150. New York, NY: Pergamon Press. TIC: 253636.

Monnin, C. 1989. "An Ion Interaction Model for the Volumetric Properties of Natural Waters: Density of the Solution and Partial Molal Volumes of Electrolytes to High Concentrations at 25ºC." *Geochimica et Cosmochimica Acta*, *53*, 1177–1188. New York, NY: Pergamon Press. TIC: 253591.

Monnin, C. and Schott, J. 1984. "Determination of the solubility products of sodium carbonate minerals and an application to trona deposition in Lake Magadi (Kenya)." *Geochimica et Cosmochimica Acta, 48*, 571–581. New York, NY: Pergamon Press. TIC: 253637.

Morris, J.C. and Stumm, W. 1967. "Redox Equilibria and Measurements of Potentials in the Aquatic Environment." Stumm, W. editor. *Equilibrium Concepts in Natural Water Systems.* American Chemical Society Advances in Chemistry Series 67. Pages 270–285. Washington, D.C.: American Chemical Society. TIC: 246519.

Nordstrom, D.K. and Jenne, E.A. 1977. "Fluorite Solubility Equilibria in Selected Geothermal Waters." *Geochimica et Cosmochimica Acta, 41*, 175–188. New York, NY: Pergamon Press. TIC: 253587.

Nordstrom, D.K., Jenne, E.A. and Ball, J.W. 1979. "Redox equilibria of iron in acid mine waters." Jenne, E.A., ed. 1979. *Chemical Modeling in Aqueous Systems, Speciation, Sorption, Solubility, and Kinetics.* ACS Symposium Series 93. Washington, D.C.: American Chemical Society. TIC: 235633.

Nordstrom, D. K. and Munoz, J. L. 1985. *Geochemical Thermodynamics*. Menlo Park, California: Benjamin/Cummings Publishing Company, Inc. TIC: 208228.

Nordstrom, D.K., Plummer, L.N., Wigley, T.M.L., Wolery, T.J., Ball, J.W., Jenne, E.A., Bassett, R.L., Crerar, D.A., Florence, T.M., Fritz, B., Hoffman, M., Holdren, G.R., Jr., Lafon, G.M., Mattigod, S.V., McDuff, R.E. 1979. "A Comparison of Computerized Chemical Models for Equilibrium Calculations in Aqueous Systems." Jenne, E.A., ed. 1979. *Chemical Modeling in Aqueous Systems, Speciation, Sorption, Solubility, and Kinetics.* ACS Symposium Series 93. Washington, D.C.: American Chemical Society. TIC: 224155.

Pabalan, R. T. and Pitzer, K. S. 1987. "Thermodynamics of concentrated electrolyte mixtures and the prediction of mineral solubilities to high temperatures for mixtures in the system Na-K-Ca-Mg-Cl-SO_4-OH-H_2O." *Geochimica et Cosmochimica Acta, 51*, 2429–2443. New York, NY: Pergamon Press. TIC: 253508.

Pabalan, R. T. and Pitzer, K. S. 1988. "Heat capacity and other thermodynamic properties of $Na_2SO_{4(aq)}$ in hydrothermal solutions and the solubilities of sodium sulfate minerals in the system

Na-Cl-SO_4-OH-H_2O." *Geochimica et Cosmochimica Acta, 52*, 2393–2904. New York, NY: Pergamon Press. TIC: 253638.

Pabalan, R. T. and Pitzer, K. S. 1990. "Models for aqueous electrolyte mixtures for systems extending from dilute solutions to fused salts." Editors: Melchior, D. C. and Bassett, R. L. *Chemical Modeling of Aqueous Systems II, American Chemical Society Symposium Series, 416,* 44–57. Washington, D. C.: American Chemical Society. TIC: 241139.

Parkhurst, D.L., Thorstenson, D.C. and Plummer, L.N. 1980. "PHREEQE - A Computer Program for Geochemical Calculations." USGS WRI 80-96. Reston, VA: U.S. Geological Survey. TIC: 223154.

Kharaka, Y.K., Gunter, W.D., Aggarwal, P.D., Perkins, E.H., Debraal, J.D. 1988. "SOLMINEQ-88: A Computer Program for Geochemical Modeling of Water-Rock Interactions." USGS-WRI-88-4227. Menlo Park, CA: U.S. Geological Survey. TIC: 235463.

Pitzer, K.S. 1973. "Thermodynamics of Electrolytes. I. Theoretical Basis and General Equations." *Journal of Physical Chemistry, 77,* 268–277. Washington, D.C.: American Chemical Society. TIC: 239503.

Pitzer, K. S. 1975. "Thermodynamics of electrolytes. V. Effects of higher-order electrostatic terms." *Journal of Solution Chemistry, 4*, 249–265. New York, NY: Plenum Publishing. TIC: 248997.

Pitzer, K.S. 1979. "Theory: Ion Interaction Approach." Pytkowicz, R.M., editor. *Activity Coefficients in Electrolyte Solutions.* Boca Raton, FL: CRC Press. TIC: 239527.

Pitzer, K. S. 1984. "A consideration of Pitzer's equations for activity and osmotic coefficients in mixed electrolytes." *Journal of the Chemical Society, Faraday Transactions I, 80*, 3451–3454. London, England: Royal Society of Chemistry. TIC: 253639.

Pitzer, K. S. 1987. "Thermodynamic model for aqueous solutions of liquid-like density." Editors: Carmichael, I. S. E. and Eugster, H. P. *Thermodynamic Modeling of Geological Materials: Minerals, Fluids and Melts, Reviews in Mineralogy, 17*, 97–142. Washington, D. C.: Mineralogical Society of America. TIC: 253653

Pitzer, K. S. 1991. "Ion interaction approach: Theory and Data Correlation." Editor: Pitzer, K. S. *Activity Coefficients in Electrolyte Solutions,* 2nd edition, Chapter 3, 75–153. Boca Raton, Florida: CRC Press. TIC: 251799.

Pitzer, K. S. and Kim J. J. 1974. "Thermodynamics of electrolytes. IV. Activity and osmotic coefficients for mixed electrolytes." *Journal of the American Chemical Society, 96*, 5701–5707. Washington D.C.: American Chemical Society. TIC: 246223.

Pitzer, K. S. and Mayorga, G. 1974. "Thermodynamics of electrolytes. III. Activity and osmotic coefficients for 2-2 electrolytes." *Journal of Solution Chemistry, 3*, 539–546. New York, NY: Plenum Publishing. TIC: 249017.

Pitzer, K. S., Roy, R. N. and Silvester, L. F. 1977. “Thermodynamics of electrolytes. 7. Sulfuric acid.” *Journal of the American Chemical Society, 99*, 4930–4936. Washington D.C.: American Chemical Society. TIC: 249015.

Pitzer, K. S. and Silvester, L. F., 1976. “Thermodynamics of electrolytes. VI. Weak electrolytes including H_3PO_4.” *Journal of Solution Chemistry, 5*, 269–278. New York, NY: Plenum Publishing. TIC: 249014.

Plummer, L. N. and Parkhurst, D. L. 1990. “Application of the Pitzer equations to the PHREEQE geochemical model.” Editors: Melchior, D. C., and Bassett, R. L. *Chemical Modeling of Aqueous Systems II. American Chemical Society Symposium Series, 416,* 128–137. Washington, D. C.: American Chemical Society. TIC: 241139.

Plummer, L. N., Parkhurst, D. L., Fleming, G. W. and Dunkle, S. A. 1988. “A Computer Program Incorporating Pitzer’s Equations for Calculation of Geochemical Reactions in Brines.” *United States Geological Survey Water-Resources Investigations Report 88-4153*. Reston, VA: U.S. Geological Survey. TIC: 248053.

Reardon, E.J. and Langmuir, D. 1976. “Activity Coefficients of $MgSO_4^0$ and $CaSO_4^0$ Ion Pairs as a Function of Ionic Strength.” *Geochimica et Cosmochimica Acta, 40*, 549–554. New York, NY: Pergamon Press. TIC: 253654.

Rimstidt, J.D. and Barnes, H.L. 1980. “The Kinetics of Silica-Water Reactions.” *Geochimica et Cosmochimica Acta, 44*, 1683–1699. New York, NY: Pergamon Press. TIC: 219975.

Saxena, S.R. 1973. *Thermodynamics of Rock-Forming Crystalline Solutions.* New York, NY: Springer-Verlag. TIC: 253603.

Silvester, L. F. and Pitzer, K. S. 1977. “Thermodynamics of electrolytes. 8. High-temperature properties, including enthalpy an heat capacity, with application to sodium chloride.” *Journal of Solution Chemistry, 7*, 327–337. New York, NY: Plenum Publishing. TIC:248979.

Silvester, L. F. and Pitzer, K. S., 1978. “Thermodynamics of electrolytes. X. Enthalpy and the effect of temperature on the activity coefficients.” *Journal of Physical Chemistry, 7*, 327–337. TIC: 249012.

Simonson, J. M., Roy, R. N., Roy L. N. and Johnson, D. A., 1987. “The thermodynamics of aqueous borate solutions I. Mixtures of boric acid with sodium or potassium borate and chloride.” *Journal of Solution Chemistry, 16*, 791–803. New York, NY: Plenum Publishing Corporation. TIC: 253640.

Spencer, R. J., Møller, N. and Weare, J. H. 1990. “The prediction of mineral solubilities in natural waters: A chemical equilibrium model for the Na-K-Ca-Mg-Cl-SO_4-H_2O system at temperatures below 25°C.” *Geochimica et Cosmochimica Acta, 54*, 575–590. New York, NY: Pergamon Press. TIC: 248999.

Stumm, W. and Morgan, J.J. 1981. *Aquatic Chemistry: An Introduction Emphasizing Chemical Equilibria in Natural Waters*, Second Edition. New York, NY: John Wiley and Sons. TIC: 217460.

Tardy Y. and Garrels, R.M. 1974. "A Method of Estimating the Gibbs Energies of Formation of Layer Silicates." *Geochimica et Cosmochimica Acta, 38*, 1101–1116. Oxford, England; New York, NY: Pergamon. TIC: 252611.

Thorstenson, D.C. 1970. "Equilibrium Distribution of Small Organic Molecules in Natural Waters." *Geochimica et Cosmochimica Acta, 34*, 745–770. New York, NY: Pergamon Press. TIC: 253597.

Truesdell, A.H. 1968. "The Advantage of Using pe Rather than Eh in Redox Equilibrium Calculations." *Journal of Geological Education, 16*, 17–20. Appleton, WI: National Association of Geology Teachers. TIC: 253599.

Van Zeggeren, F. and Storey, S.H. 1970. *The Computation of Chemical Equilibrium Analysis.* London, England: Cambridge University Press. TIC: 252891.

Viani, B.E. and Bruton, C.J. 1992. *Modeling Ion Exchange in Clinoptilolite Using the EQ3/6 Geochemical Modeling Code.* UCRL-JC-109952. Livermore, CA: Lawrence Livermore National Laboratory. TIC: 230491**.**

Willey, L.M., Kharaka, Y.K., Presser, T.S., Rapp, J.B. and Barnes, I. 1975. "Short Chain Aliphatic Acid Anions in Oil Field Waters and Their Contribution to the Measured Alkalinity." *Geochimica et Cosmochimica Acta, 39*, 1707–1711. New York, NY: Pergamon Press. TIC: 253595.

Wolery, T.J. 1978. *Some Chemical Aspects of Hydrothermal Processes at Mid-Oceanic Ridges - A Theoretical Study. I. Basalt-Sea Water Reaction and Chemical Cycling Between the Oceanic Crust and the Oceans. II. Calculation of Chemical Equilibrium Between Aqueous Solutions and Minerals.* Ph.D. dissertation. Evanston, Illinois: Northwestern University. TIC: 219640.

Wolery, T.J. 1979. *Calculation of Chemical Equilibrium Between Aqueous Solution and Minerals: The EQ3/6 Software Package.* UCRL-52658. Livermore, California: Lawrence Livermore Laboratory. ACC: HQS.19880517.2586.

Wolery, T.J. 1983. *EQ3NR, A Computer Program for Geochemical Aqueous Speciation-Solubility Calculations: User's Guide and Documentation.* UCRL-53414. Livermore, California: Lawrence Livermore National Laboratory. TIC: 204035.

Wolery, T.J. 1990. "On the Thermodynamic Framework of Solutions." *American Journal of Science, 290*, 296–320. New Haven, CT: Yale University, Kline Geology Laboratory. TIC: 252904.

Wolery, T.J. and Walters, L.J., Jr. 1975. "Calculation of Equilibrium Distributions of Chemical Species in Aqueous Solutions by Means of Monotone Sequences." *Journal of the International Association for Mathematical Geology, 7,* 99–115. New York, NY: Plenum. TIC: 253656.

INTENTIONALLY LEFT BLANK

APPENDIX C

EQ3/6 VERSION 8.0 SAMPLE PICKUP FILE

INTENTIONALLY LEFT BLANK

```
* Start of the bottom half of the input file                                   *
*------------------------------------------------------------------------------*
| Secondary Title        | (utitl2(n))                                         |
|------------------------------------------------------------------------------|
|EQ3NR input file name= sio2.3i                                                |
|Description= "Solution containing 101.0 mg/L SiO2"                            |
|Version level= 8.0                                                            |
|Revised 02/14/97    Revisor= T.J. Wolery                                      |
|This is part of the EQ3/6 Test Case Library                                   |
|                                                                              |
|  Solution containing 101.0 mg/L of dissolved SiO2. This test case is         |
|otherwise nearly identical to that in the EQ3NR test case input file          |
|deion.3i.                                                                     |
|                                                                              |
|  Purpose: to initialize the EQ6 test case input files pptqtz.6i and          |
|pptqtza.6i, both of which simulate the precipitation of quartz from           |
|supersaturated solution at 105C. The former EQ6 test case is test problem 1   |
|from Delany, Puigdomenech, and Wolery (1986, p. 19-21, 31-33), which          |
|simulates an experiment (Run 2E) reported by Rimstidt and Barnes (1980,       |
|p. 1687-1688, Figure 2) and uses their rate law model, which is independent   |
|of pH. The latter is the same problem, but uses a rate law model from         |
|Knauss and Wolery (1988) that does include a dependence on pH.                |
|                                                                              |
|  Note that the dissolved gases O2 and H2 have been suppressed. This is       |
|because this problem has no redox aspect. The EQ6 option switch iopt(15)      |
|should be set to 1 in the EQ6 test case input files so that EQ6 knows this.   |
|                                                                              |
|  Note that precipitation of the silica polymorphs tridymite, chalcedony,     |
|SiO2(am), and "Cristobalite(alpha)" has been prevented by means of nxmod      |
|suppress options. These options have no effect in EQ3NR, but will be passed   |
|on to EQ6 via the pickup file used to construct the EQ6 test case input       |
|files pptqtz.6i and pptqtza.6i.                                               |
|                                                                              |
|                                                                              |
|                            References                                        |
|                                                                              |
|Delany, J.M., Puigdomenech, I., and Wolery, T.J., 1986, Precipitation         |
|  Kinetics Option for the EQ6 Geochemical Reaction Path Code: UCRL-53642,     |
|  Lawrence Livermore National Laboratory, Livermore, California, 44 p.        |
|                                                                              |
|Knauss, K.G., and Wolery, T.J., 1988, The dissolution kinetics of quartz as   |
|  a function of pH and time at 70C: Geochimica et Cosmochimica Acta, v. 52,   |
|  p. 43-53.                                                                   |
|                                                                              |
|Rimstidt, J.D., and Barnes, H.L., 1980, The kinetics of silica-water          |
|  reactions: Geochimica et Cosmochimica Acta, v. 44, p. 1683-1699.            |
|                                                                              |
|------------------------------------------------------------------------------|
|Special Basis Switches (for model definition only)       | (nsbswt)           |
|------------------------------------------------------------------------------|
|Replace |None                                            | (usbsw(1,n))       |
|   with |None                                            | (usbsw(2,n))       |
|------------------------------------------------------------------------------|
|Original temperature (C) | 2.50000E+01| (tempci)                              |
|------------------------------------------------------------------------------|
|Original pressure (bars) | 1.01320E+00| (pressi)                              |
|------------------------------------------------------------------------------|
|Create Ion Exchangers  | (net)                                                |
|------------------------------------------------------------------------------|
|Advisory: no exchanger creation blocks follow on this file.                   |
|Option: on further processing (writing a pickup file or running XCON6 on the  |
|present file), force the inclusion of at least one such block (qgexsh):       |
```

```
|  [ ] (.true.)                                                                |
|------------------------------------------------------------------------------|
|Alter/Suppress Options  | (nxmod)                                             |
|------------------------------------------------------------------------------|
|Species                                         |Option          |Alter value |
| (uxmod(n))                                     |(ukxm(kxmod(n)))| (xlkmod(n))|
|------------------------------------------------------------------------------|
|Tridymite                                       |Suppress        | 0.00000E+00|
|Chalcedony                                      |Suppress        | 0.00000E+00|
|SiO2(am)                                        |Suppress        | 0.00000E+00|
|Cristobalite(alpha)                             |Suppress        | 0.00000E+00|
|------------------------------------------------------------------------------|
* Valid alter/suppress strings (ukxm(kxmod(n))) are:                           *
*    Suppress            Replace             AugmentLogK                       *
*    AugmentG                                                                  *
*------------------------------------------------------------------------------*
|Iopg Activity Coefficient Option Switches ("( 0)" marks default choices)      |
|------------------------------------------------------------------------------|
|iopg(1) - Aqueous Species Activity Coefficient Model:                         |
|  [ ] (-1) The Davies equation                                                |
|  [x] ( 0) The B-dot equation                                                 |
|  [ ] ( 1) Pitzer's equations                                                 |
|  [ ] ( 2) HC + DH equations                                                  |
|------------------------------------------------------------------------------|
|iopg(2) - Choice of pH Scale (Rescales Activity Coefficients):                |
|  [ ] (-1) "Internal" pH scale (no rescaling)                                 |
|  [x] ( 0) NBS pH scale (uses the Bates-Guggenheim equation)                  |
|  [ ] ( 1) Mesmer pH scale (numerically, pH = -log m(H+))                     |
|------------------------------------------------------------------------------|
|Matrix Index Limits                                                           |
|------------------------------------------------------------------------------|
|No. of chem. elements   |    4| (kct)                                         |
|No. of basis species    |    5| (kbt)                                         |
|Index of last pure min. |    5| (kmt)                                         |
|Index of last sol-sol. |    5| (kxt)                                          |
|Matrix size             |    5| (kdim)                                        |
|PRS data flag           |    0| (kprs)                                        |
|------------------------------------------------------------------------------|
|Mass Balance Species (Matrix Row Variables)     |Units/Constraint| --         |
| (ubmtbi(n))                                    |(ujf6(jflgi(n)))| --         |
|------------------------------------------------------------------------------|
|H2O                     Aqueous solution        |Moles           | --         |
|H+                      Aqueous solution        |Moles           | --         |
|HCO3-                   Aqueous solution        |Moles           | --         |
|SiO2(aq)                Aqueous solution        |Moles           | --         |
|O2(g)                   Aqueous solution        |Moles           | --         |
|------------------------------------------------------------------------------|
* Valid jflag strings (ujf6(jflgi(n))) are:                                    *
*    Moles               Make non-basis                                        *
*------------------------------------------------------------------------------*
|Mass Balance Totals (moles)                                                   |
|------------------------------------------------------------------------------|
|Basis species (info. only)      |Equilibrium System    |Aqueous Solution      |
| (ubmtbi(n))                    | (mtbi(n))            | (mtbaqi(n))          |
|------------------------------------------------------------------------------|
|H2O                     Aqueous | 5.550842451378904E+01| 5.550842451378904E+01|
|H+                      Aqueous | 1.285516149223014E-05| 1.285516149223014E-05|
|HCO3-                   Aqueous | 1.285516149219963E-05| 1.285516149219963E-05|
|SiO2(aq)                Aqueous | 1.680971648121736E-03| 1.680971648121736E-03|
|O2(g)                   Aqueous | 4.216548737912356E-19| 4.216548737912356E-19|
|Electrical imbalance            | 3.051143494623865E-17| 3.051143494623865E-17|
|------------------------------------------------------------------------------|
|Ordinary Basis Switches (for numerical purposes only)    | (nobswt)           |
```

```
|------------------------------------------------------------------------------|
|Replace |None                                            | (uobsw(1,n))      |
|   with |None                                            | (uobsw(2,n))      |
|------------------------------------------------------------------------------|
|Matrix Column Variables and Values                                            |
|------------------------------------------------------------------------------|
|Basis species (uzveci(n))                       |Log moles (zvclgi(n)) | --   |
|------------------------------------------------------------------------------|
|H2O                     Aqueous solution        | 1.744358983526984E+00| --   |
|H+                      Aqueous solution        |-5.636914419180132E+00| --   |
|HCO3-                   Aqueous solution        |-5.675150470992916E+00| --   |
|SiO2(aq)                Aqueous solution        |-2.774488692003520E+00| --   |
|O2(g)                   Aqueous solution        | 0.000000000000000E+00| --   |
|------------------------------------------------------------------------------|
|Phases and Species in the PRS                                                 |
|------------------------------------------------------------------------------|
|Phase           |None                    | (uprphi(n))                        |
|------------------------------------------------------------------------------|
|->|No. of Moles    | 0.000000000000000E+00| (mprphi(n))                       |
|------------------------------------------------------------------------------|
|--->|Species                 |No. of Moles          | --                      |
|--->| (uprspi(i,n))          | (mprspi(i,n))        | --                      |
|------------------------------------------------------------------------------|
|--->|None                    | 0.000000000000000E+00| --                      |
|------------------------------------------------------------------------------|
|End of problem                                                                |
|------------------------------------------------------------------------------|
```

INTENTIONALLY LEFT BLANK

APPENDIX D

EQ6 SUPPLIMENTARY INFORMATION

INTENTIONALLY LEFT BLANK

CONTENTS

FIGURES

Page

D.1. GOVERNING EQUATIONS

D.1.1. GENERAL DISCUSSION

As noted earlier, in EQ6 there is a separation in the methodology of treating equations that are intrinsically algebraic from those that are intrinsically differential. The former govern thermodynamic calculations, and the latter comprise rate laws for irreversible processes. This numerical decoupling makes it possible to perform thermodynamic calculations, given the necessary inputs of total number of moles of components, the temperature, and the pressure, independently of the integration of rate equations. This contrasts to the treatment used in the PATHI code (Helgeson 1968, Helgeson, et al. 1970), in which the algebraic relations were differentiated with respect to ξ and the resulting Ordinary Differential Equations (ODEs; they are discussed extensively in Appendix E) were integrated numerically. This decoupling permits making "single point" thermodynamic calculations, such as the temperature jump, problems in which rate equations do not even appear.

In EQ6 reaction path models, the two types of equations are coupled in the mathematically formal sense, but the solution of each is performed semi-independently. Each type of calculation is performed alternately, the output of one becoming the input to the next execution of the other. For example, in moving a step forward in reaction progress (ξ), the rate equations are integrated. This defines new values for the temperature, the pressure, and the total number of moles of the components, which are inputs to the following thermodynamic calculation. This in turn gives a new distribution of species, from which may be calculated values for the rates of the irreversible processes at the new point. If accuracy tests on the ODE integration are satisfied, these rate values are then used in making the next integration step. Otherwise, the step size may be cut until those tests are satisfied.

When the rate chosen to constrain an irreversible process is a relative rate ($d\xi_j/d\xi$), the rate function is either a constant or a simple function of the overall reaction progress variable (ξ). When EQ6 operates in the mode of arbitrary kinetics (all irreversible processes constrained by relative rate expressions, no time variable in the model), these rates can be integrated by simple closed-form expressions. It is therefore possible to take arbitrarily large step sizes, subject only to the following conditions.

In the case of closed and open system calculations, the rate of an irreversible reaction is set to zero when the corresponding thermodynamic driving force, the affinity, is no longer positive. Affinities are outputs of the thermodynamic calculations. EQ6 locates the point of reaction progress where the affinity goes to zero. If the corresponding reactant is a mineral, then this means that the aqueous solution has reached saturation. The code then changes the status of the reactant to inactive (meaning it is effectively removed from the set of reactants). Any remaining mass of the reactant is then moved into the *equilibrium system*. Titration calculations are very similar to closed system calculations, but the rate of an irreversible reaction is not set to zero when saturation is reached, and the remaining reactant mass continues to be added to the equilibrium system according to the rate law.

The rate of an irreversible reaction rate also becomes zero when the associated "reactant" becomes exhausted, no matter what kind of system model the code is dealing with. The user specifies how much of a reactant is available at the start of the run. The code then finds the point of reaction progress at which exhaustion occurs.

The purpose of this section is to review the governing equations pertaining to calculations in EQ6. The following section reviews the algebraic equations that govern thermodynamic equilibrium calculations. Many of these equations have been discussed in Appendix B – EQ3NR Supplementary Information, although some of the equations have some slight differences in the way they pertain to EQ6. The final section discusses the equations dealing with reaction progress, reaction rates, temperature, and pressure. For the details of the ODE Integration Methods refer to Appendix E.

D.1.2. CONSTRAINTS ON THERMODYNAMIC CALCULATIONS

D.1.2.1. Mass Balance

Mass balance in EQ6 takes on a slightly different form than in EQ3NR. All species appearing in the equilibrium system contribute to mass balance. In EQ3NR, this system consisted only of the aqueous phase. However, in EQ6 it also includes minerals and other substances in equilibrium with the aqueous solution. A second difference is that in EQ3NR, mass balance was treated in terms of molal concentrations. In EQ6, it is treated in terms of numbers of moles. The molal concentration of the s^{th} aqueous species (m_s) is related to the corresponding number of moles (n_s) by the equation:

$$m_s = \frac{\Omega n_s}{n_w} \qquad \text{(D–1)}$$

where Ω is the number of moles of water (≈ 55.51) comprising a mass of one kilogram and n_w is the number of moles of water.

A general expression of mass balance for the $^{\text{th}}$ basis species in an equilibrium system composed of aqueous solution, pure minerals, and solid solutions, is:

$$\sum_{s=1}^{s_T} u_{s's} n_s + \sum_{\phi=1}^{\phi_T} u_{s'\phi} n_\phi + \sum_{\psi=1}^{\psi_T} \sum_{\sigma=1}^{\sigma_{T,\psi}} u_{s'\sigma\psi} n_{\sigma\psi} = n_{T,\,s'} \qquad \text{(D–2)}$$

This is an extension of the mass balance equation used in EQ3NR (Appendix B.4.8.1), in addition to the conversion from molalities to numbers of moles. Here s_T is the number of aqueous species in the system, ϕ_T is the number of minerals of fixed composition present and in equilibrium with the water, ψ_T is the number of such solid solution phases, and $\sigma_{T,\psi}$ is the number of components in the ψ^{th} solid solution. Also, n_ϕ is the mass of the ϕ^{th} pure mineral present, $n_{\sigma\psi}$ is the mass of the σ^{th} component of the ψ^{th} solid solution, and $n_{s'}$ is the total number of moles of the s' th basis species. The stoichiometric factors ($u_{s's}$, $u_{s'\phi}$, and $u_{s'\sigma\psi}$) relate the contributions of the aqueous, pure mineral, and solid solution species to the mass balance. For non-basis species, these factors are computed from reaction coefficients; the factor for a basis species in its own mass balance is unity. There is such a mass balance equation for each of the basis species present in the model except the s_Bth, the fictive aqueous species O_2.

The present version of EQ6 actually uses mass balances defined in terms of chemical elements. Thus, for the ε th element, the appropriate expression is:

$$\sum_{s=1}^{s_T} c_{\varepsilon s} n_s + \sum_{\phi=1}^{\phi_T} c_{\varepsilon\phi} n_\phi + \sum_{\psi=1}^{\psi_T} \sum_{\sigma=1}^{\sigma_{T,\psi}} c_{\varepsilon\sigma\psi} n_{\sigma\psi} = n_{T,\varepsilon} \tag{D–3}$$

The factors $c_{\varepsilon s}$, $c_{\varepsilon\phi}$, and $c_{\varepsilon\sigma\psi}$ are the coefficients of this element in the chemical formulas for the sth aqueous species, the φth pure mineral, and the σth component of the ψth solid solution, respectively. This equation is equivalent to eq (D–2) if the set of active basis species contains no auxiliary basis species.

If an auxiliary basis is present, the number of basis species minus one is greater than the number of chemical elements. Hence, there is then a greater number of corresponding mass balance equations in the model constraints.

D.1.2.2. Charge Balance

The charge balance equation comes into the calculations if a redox parameter is needed to complete the model, which is generally the case. The fictive aqueous species is the corresponding basis species, and the primary redox parameter is the oxygen fugacity. From this, other redox parameters can be calculated, such as *Eh*, *pe*, or *Ah* (see Appendix B.1.3.6). The charge balance equation can be written as:

$$\sum_{s}^{s_T} z_s n_s = 0 \tag{D–4}$$

where z_s is the electrical charge. This is analogous to eq (D–3), except that the weighting factors are different and there are no terms for non-aqueous species.

In practice, the initial model computed by EQ3NR may be either charge balanced or out of charge balance. EQ6 can handle either case. EQ3NR calculates the charge imbalance (Δ_z) at the end of it speciation calculation and writes this on its pickup file with the header *Electrical* *imbalance*. EQ6 then adds this term into the electrical balance, so that the initial charge imbalance is maintained throughout the run. In other words, eq (D–4) is replaced by:

$$-\Delta_z + \sum_{s}^{s_T} z_s n_s = 0 \tag{D–5}$$

D.1.2.3. Mass Action

A mass action equation describes thermodynamic equilibrium for a given chemical reaction. The reactions on the EQ3/6 data file are always written so as to destroy the associated non-basis species (e.g., the dissociation reaction of an aqueous complex, the dissolution reaction of a mineral). All other species appearing in the reaction are basis species. The mass action expression for j^{th} reaction for the destruction of the associated species, denoted here as the i^{th}, takes the following form when the associated species is not a gas:

$$b_{ij} \log a_i + b_{s_B j} \log f_{O_2} + \sum_{\substack{s' = 1 \\ s' \neq s_B}}^{s_Q} b_{s'j} \log a_{s'} = \log K_j \tag{D–6}$$

Here the reaction coefficients *b* are negative for reactants and positive for products, following the usual EQ3/6 convention. The symbol *a* represents thermodynamic activity of the species denoted by the subscript. K_j is the equilibrium constant. The fictive aqueous species O_2 is the s_B^{th} species. The symbol *s'* implies a basis species. If the associated species is a gas, the corresponding mass action equation is:

$$b_{ij} \log f_i + b_{s_B j} \log f_{O_2} + \sum_{\substack{s' = 1 \\ s' \neq s_B}}^{s_Q} b_{s'j} \log a_{s'} = \log K_j \tag{D–7}$$

Here the fugacity (*f*) appears in place of the activity of the i^{th} species. Currently EQ6 does not treat the possible existence of an explicit gas phase in the equilibrium system. This would require pressure and volume to be defined for the gas phase. Hence, eq (D–7) is used only to compute equilibrium fixed-fugacities.

The following familiar relations treat thermodynamic activities. In the case of solvent water, the activity is given by:

$$\log a_w = \log x_w + \log \lambda_w \tag{D–8}$$

where x_w is the mole fraction of water in the aqueous solution and λ_w is the corresponding mole fraction activity coefficient. The mole fraction of water is given by:

$$x_w = \frac{\Omega}{\Omega + \sum m} \tag{D–9}$$

where Σm is the sum of the molalities of the solute species. For the i^{th} aqueous solute species, the activity is given by:

$$\log a_i = \log m_i + \log \gamma_i \tag{D–10}$$

where γ_i is the corresponding the molal activity coefficient. For the ϕ^{th} pure mineral, the activity is a constant:

$$\log a_\phi = 0 \tag{D–11}$$

For the σ^{th} component of the ψ^{th} solid solution, the activity is defined analogously to that of solvent water:

$$\log a_{\sigma\psi} = \log x_{\sigma\psi} + \log \lambda_{\sigma\psi} \tag{D–12}$$

Here $x_{\sigma\psi}$ is the mole fraction of the component and $\lambda\sigma\psi$ is the corresponding mole fraction activity coefficient. The mole fraction of the σ^{th} component of the ψ^{th} phase is given by:

$$x_{\sigma\psi} = \frac{n_{\sigma\psi}}{\sum_{\sigma'=1}^{\sigma_{T,\psi}} n_{\sigma'\psi}} \tag{D–13}$$

where $\sigma_{T,\psi}$ is the number of components in the components in the ψ^{th} phase.

D.1.2.4. Activity Coefficients of Aqueous Species

Activity coefficients are functions of solution composition, temperature, and pressure. However, there is no recognized single universal set of equations for calculating them in aqueous solutions. Instead, there are various equations that each have their own strengths and weaknesses. Each must provide not only the molal activity coefficients of the solute species, but also the mole fraction activity coefficient of water (or equivalently, its activity). Several different models have been programmed into EQ3/6. These are discussed in detail in EQ3NR Supplementary Information, Appendix B.2.

The activity coefficient option is specified on the input file by the option switch **iopg(1)**. The existing options are:

–1	The Davies (1962) equation.
0	The B-dot equation (Helgeson, 1969).
1	Pitzer's (1973, 1975, 1979, 1987) equations.

This selection of options is made in the input file in the section reproduced here:

```
|iopg(1) - Aqueous Species Activity Coefficient Model:                        |
|  [ ] (-1) The Davies equation                                               |
|  [x] ( 0) The B-dot equation                                                |
|  [ ] ( 1) Pitzer's equations                                                |
|  [ ] ( 2) HC + DH equations                                                 |
```

Note that option **iopg(1)** = 2, *HC + DH equations*, is not functional nor available at this time in EQ3/6 Version 8.0. The Davies equation and the B-dot equation are simple extended Debye-Hückel models. The activity coefficients are treated as functions of the ionic strength:

$$I = \sum_{s=1}^{s_T} m_s z_s^2 \tag{D–14}$$

These models ignore any dependence on the specific composition of the aqueous solution. This is realistic only in dilute solutions, and these options should not be used in calculations involving solutions in which the ionic strength exceeds 1.0 molal. In comparison with the best physical chemistry measurements, inaccuracy is apparent at values less than 0.1 molal. The **com**, **sup**, and **nea** data files may be used with the Davies and B-dot equations. A large number of chemical components may be treated using these models.

Pitzer's (1973, 1975, 1979, 1987) equations treat the activity coefficients as functions of both the ionic strength and the specific composition of the aqueous solution. strength. These equations may be applied to concentrated brines (e.g., Harvie, Møller, and Weare, 1984). The **hmw** and **pit** data files may be used with Pitzer's equations. In general, a more restricted set of chemical components may be treating using Pitzer's equations.

D.1.2.5. Activity Coefficients of Solid Solution Components

The activities and activity coefficients of solid solution components are defined on the usual mole fraction scale. The treatment of solid solutions in EQ3/6 is currently limited to simple molecular mixing models. These are models in which the components are end members, equivalent to pure minerals. The existing capabilities are described elsewhere (Appendix B.3; Wolery 1979, Table 3, p. 12–13; Bourcier 1985, 1989) and require coding not only for evaluating the activity coefficient (λ), but also the derivative function $\Lambda_{ij} = (d \log \lambda_i / d \log x_j)$, where both *i* and *j* denote components.

D.1.2.6. Saturation Indices and Affinities

Saturation indices and affinities are important because they are measures of the thermodynamic driving forces behind the progress of irreversible reactions. These quantities form an important bridge between calculations that are purely thermodynamic and those that pertain to dynamic evolution. If there is no positive driving force for a reaction, it will not proceed in the forward direction. Unless the user has specified titration mode (**iopt(1)** = 1), EQ6 sets a calculated rate to zero when the corresponding driving force is zero or opposed to further progress. The affinity function also appears directly as a parameter in some important kinetic rate laws, as will be noted later in this section.

The saturation index (SI) is a commonly used measure of the state of an aqueous solution with respect to the progress of some reaction, usually the dissolution of a mineral. It is defined by:

$$\mathrm{SI} = \log \frac{Q_+}{K_+} \tag{D–15}$$

where Q is the ion activity product (*IAP*), K is the equilibrium constant, and the subscript "+" denotes that these quantities pertain to the reaction as written for the dissolution/ dissociation/destruction of the associated species (the forward direction in EQ3/6). Saturation indices are zero for the case of exact saturation, negative for undersaturation, and positive for supersaturation.

In expanded EQ3/6 nomenclature, Q is evaluated for the j^{th} reaction according to:

$$\log Q_{+,j} = b_{ij} \log a_i + b_{s_B j} \log f_{O_2} + \sum_{\substack{s'=1 \\ s' \neq s_B}}^{s_Q} b_{s'j} \log a_{s'} \tag{D–16}$$

Here must f_i be substituted for a_i if the i^{th} species is a gas. Because the present version of EQ6 does not treat the possible presence of a gas phase in the equilibrium system, ion activity products and saturation indices for reactions for destruction of gas species are not currently evaluated.

The saturation index of the ψ^{th} solid solution phase can be constructed from the saturation indices of its components according to:

$$\mathrm{SI}_{\psi} = \sum_{\sigma = 1}^{\sigma_{T,\psi}} x_{\sigma} \mathrm{SI}_{\sigma\psi} \tag{D–17}$$

This can be easily derived by creating a reaction for the dissolution of the phase by making a linear combination of the reactions for the individual components.

If a solid solution is present in the equilibrium system, the thermodynamic activity of each of its components that is also present is defined and eq (D–16) and hence eq (D–17) can be evaluated. Of course, the saturation index of each of the components, hence that of the phase, will of necessity have a value of zero, so this is not very interesting. However, if a solid solution is not present, the thermodynamic activity of each of its components is not directly available, and eqs (D–16) and (D–17) by themselves are inadequate to determine a saturation index. The scheme presently used in EQ3/6 is to define a hypothetical saturation index as that corresponding to the composition that maximizes the value of this index (Bourcier, 1985, 1989).

The affinity is closely related to the saturation index. Affinities may be defined for both the forward ("+") and reverse ("–") directions of a reaction. In EQ3/6, "forward" means in the direction of dissolution/ dissociation/destruction of the associated species. Hence, "–" signifies precipitation/association/production of the associated species.

Following these conventions, the affinity to dissolve/dissociate/disappear is related to the saturation index by:

$$A_+ = -2.303RT\,\mathrm{SI} \qquad \text{(D–18)}$$

The affinity to precipitate/associate/form is related to the SI by:

$$A_- = 2.303RT\,\mathrm{SI} = -A_+ \qquad \text{(D–19)}$$

When the affinity is positive, it means that the reaction is thermodynamically favored to move in the corresponding direction. Note that affinity has the dimensions of energy per mole (of reaction progress). Also, the affinity to precipitate/associate/form follows the same sign conventions with respect to supersaturation/undersaturation as the saturation index.

D.1.3. CONSTRAINTS RELATED TO REACTION PROGRESS

D.1.3.1. The Reaction Progress Variable

The reaction progress variable (ξ) is a measure of the extent to which a reaction has proceeded. Any chemical reaction can be written in the following format:

$$aA + bB + \ldots = cC + dD + \ldots \qquad \text{(D–20)}$$

Here *A*, *B*, etc. are reactants, *C*, *D*, etc., are products, and *a*, *b*, *c*, *d*, etc., are the reaction coefficients (all positive when written in this form). There are two equivalent ways to describe a complex chemical process, such as rock/water interaction. The first is to consider it as an array of simple reactions. Here the reaction coefficients are constants and are usually chosen to be small integers or common rational numbers such as 1/2, 1/4, etc. The second is to view the whole process as a single reaction, whose reaction coefficients are generally neither "nice" numbers nor constants along a reaction path.

In EQ6, reaction progress variables are used to describe only the reactions that are irreversible (not at equilibrium). The irreversible process is defined by an array of simple irreversible reactions, corresponding to the first case described above. Each of these reactions has its own associated progress variable (ξ_i). There is in addition an overall reaction progress variable (ξ) for the process as a whole, which corresponds to the second case described above.

It is possible to define progress variables for reversible reactions as well. This is done in some methods for making thermodynamic calculations (See Van Zeggeren and Storey, 1970), in which case these parameters are used as iteration variables in place of masses or concentrations of component species (in this usage, reaction progress variables are generally symbolized by λ instead of ξ.) This type of treatment is not used in EQ6, however.

Reaction progress variables are commonly defined in the following differential form (e.g., Helgeson, 1968; Lasaga, 1981a):

$$d\xi_j = -\frac{dn_{A,j}}{a_j} = \ldots = +\frac{dn_{C,j}}{c_j} = \ldots \tag{D–21}$$

Here *j* denotes an irreversible reaction. The total change in the mass of a component must be the sum of the changes due to the individual reactions. Hence,

$$d\xi = -\frac{dn_A}{a} = \ldots = +\frac{dn_C}{c} = \ldots \tag{D–22}$$

$$dn_A = \sum_{j=1}^{j_T} dn_{A,j} \tag{D–23}$$

and so forth for *B*, *C*, *D*, where j_T is the total number of individual irreversible reactions. Then coefficients of the overall reaction are related to those of the individual reactions by

$$a = \sum_{j=1}^{j_T} a_j\left(\frac{d\xi_j}{d\xi}\right) \tag{D–24}$$

and so forth for *b*, *c*, and *d*. The quantity $d\xi_j/d\xi$ (also symbolized as v_j^{rel}) is called the relative rate of the j^{th} irreversible reaction.

The motivation behind the reaction progress variable is to normalize the effect of the reaction coefficients that appear in a given reaction. Recasting eqs (D–21) and (D–22) in terms of derivatives leads to the following results:

$$-\frac{dn_A}{d\xi_j} = a_j \quad \text{(D–25)}$$

$$+\frac{dn_C}{d\xi_j} = c_j \quad \text{(D–26)}$$

$$-\frac{dn_A}{d\xi} = a \quad \text{(D–27)}$$

$$+\frac{dn_C}{d\xi} = c \quad \text{(D–28)}$$

These equations lead to the derivative equivalent of eq (D–23):

$$\frac{dn_A}{d\xi} = \sum_{j=1}^{j_T} \frac{dn_A}{d\xi_j} \quad \text{(D–29)}$$

and so forth for *B*, *C*, and *D*.

Following the usual EQ3/6 convention for writing reactions on the data file, the reaction coefficients (*b*) are intrinsically negative for reactants. Switching to this notation, eqs (D–25) and (D–26) can be written for the i^{th} component without regard to whether it is a reactant or a product:

$$\frac{dn_i}{d\xi_j} = b_{ij} \quad \text{(D–30)}$$

Similarly, eqs (D–27) and (D–28) can be written as:

$$\frac{dn_i}{d\xi} = b_i \quad \text{(D–31)}$$

where b_i is the coefficient of this component in the overall reaction. Eq (D–24) can be written as:

$$b_i = \sum_{j=1}^{j_T} b_{ij}\left(\frac{d\xi_j}{d\xi}\right) \quad \text{(D–32)}$$

It should be clear that changes in the quantities of the components *A*, *B*, *C*, *D*, etc., can be obtained by integrating with respect to the appropriate progress variable. For the case of an individual reaction, this gives:

$$\Delta n_{i,j} = \int_{\xi_{j,0}}^{\xi_{j,1}} b_{ij} d\xi_j \tag{D–33}$$

where the subscripts 0 and 1 signify two successive points along the reaction path. Because the coefficients in this equation are constants, integration of this equation yields the following result:

$$\Delta n_{i,j} = b_{ij} \Delta\xi_j \tag{D–34}$$

where $\Delta\xi_j = \xi_{j,1} - \xi_{j,0}$. This result leads to a more immediate understanding of the reaction progress variable. Basically, it says that the change in the number of moles of component *i* due to reaction *j* is proportional to the change in ξ_j. Thus, when $\Delta\xi_j = 1$, $\Delta n_{i,j} = b_{ij}$

The overall reaction equivalent of eq (D–33) is:

$$\Delta n_i = \int_{\xi_0}^{\xi_1} b_i d\xi \tag{D–35}$$

However, the overall reaction coefficient b_i is not in general a constant. Substituting eq (D–32) into eq (D–35) and integrating gives:

$$\Delta n_i = \sum_{j=1}^{j_T} b_{ij} \Delta\xi_j \tag{D–36}$$

This is the central equation used to calculate irreversible mass transfer in EQ6. Note that it depends on integrating the relative rates ($d\xi_j / d\xi$) with respect to the overall reaction progress variable.

D.1.3.2. Reaction Rates and Time

The user of EQ6 specifies for each irreversible reaction a function for either the relative rate or the absolute rate ($d\xi_j / dt$, also symbolized as v_j). If a relative rate function is specified, then the evaluation of eq (D–36) is straightforward. If an actual rate is specified, then a relative rate function is derived from this as shown below. Time is not defined unless at least one actual rate specification is included in the model to be calculated. The scheme used here to handle relative rates and time when calculating a kinetic reaction path is very similar to that described by Helgeson and Murphy (1983). It differs in two ways (Delany, Puigdomenech, and Wolery, 1986). First, the approach is extended to allow mixing of relative rates with absolute rates in defining a problem. Second, it allows consideration of the kinetics of reactions proceeding in the backward direction.

In reaction progress mode (no time frame), ξ is simply the integration variable and no further definition is required. In time mode, this is not the case. A convenient definition is then given by:

$$\xi = \sum_{j=1}^{j_K} \xi_j \tag{D–37}$$

where j_K is the number of irreversible reactions constrained by actual rate laws (j_K is usually equal to j_T, but may be lesser if relative rates are directly specified for some of the irreversible reactions). In the treatment of Helgeson and Murphy (1983), the overall reaction progress variable is defined as the sum of the individual reaction progress variables instead of as the sum of their absolute values. The motivation behind the definition given in eq (D–37) becomes clearer by differentiating it with respect to time. Letting *v* represent the overall rate ($d\xi/dt$), the result can be written as

$$v = \sum_{j=1}^{j_K} v_j \tag{D–38}$$

Having defined *v*, one can now calculate relative rates for the irreversible reactions constrained by absolute rates:

$$v_j^{rel} = \frac{v_j}{v} \tag{D–39}$$

The inverse rate ($dt/d\xi$ or $v_{1/t}$) can be similarly calculated:

$$v_{1/t} = \frac{1}{v} \tag{D–40}$$

It is then possible to calculate absolute rates for irreversible reactions that are constrained by relative rates:

$$v_j = v_j^{rel} v \tag{D–41}$$

The progress increment for the j^{th} irreversible reaction can be calculated for a reaction progress step by integrating the relative rate (Helgeson and Murphy 1983; Delany, Puigdomenech, and Wolery 1986):

$$\Delta\xi_j = \int_{\xi_0}^{\xi_1} v_j d\xi \qquad \text{(D–42)}$$

Similarly, a time increment can be calculated for a reaction progress step by integrating the inverse rate:

$$\Delta t = \int_{\xi_1}^{\xi_2} v_{1/t} d\xi \qquad \text{(D–43)}$$

This has the advantage of allowing one to use ξ as the independent variable in calculations either reaction progress mode or time mode. Furthermore, in the latter mode, it allows the calculations to extend to the case in which time reaches infinity (Delany, Puigdomenech, and Wolery 1986). This would not be possible if time were used as the independent variable.

D.1.3.3. Rate Laws Programmed into EQ6

The rate laws must be written in terms of parameters that currently appear in the code. For example, Lasaga (1981a) proposed a rate law for feldspar dissolution that is explicitly linked to the sorption of hydrogen ion on the feldspar surface. Such a rate law cannot be dealt with in the present version of EQ6, because there is no provision for treatment of sorption (surface chemistry) phenomenon . Also, at the present time, the use of kinetic rate laws is restricted to the dissolution and precipitation growth kinetics of solids.

A more detailed discussion of rate law modeling in EQ6, including a discussion of the underlying scientific foundation and the presentation of several examples, is given by Delany, Puigdomenech, and Wolery (1986). The discussion here will be somewhat abbreviated. All rate laws evaluated by EQ6 describe net rates. However, the code permits the user to apply to each irreversible reaction one rate law for the net forward (e.g., dissolution) reaction and another for the net backward (e.g., precipitation) reaction. For some types of rate laws, one of these forms can be specified to apply to both net forward and backward rates, the actual net direction being determined by the sign of the calculated rate.

The forward rate law for the **nrk**th irreversible reactant is determined by **nrk(1,n)**, which is read from the input file (see Section 3.4.4). Currently available options for the **nrk** string are:

- Use backward rate law

- Relative rate equation
- TST rate equation
- Linear rate equation

The backward rate law is specified by **nrk(2,n)**. Currently available options for this **nrk** string are:

- Use forward rate law
- Partial equilibrium
- Relative rate equation
- TST rate equation
- Linear rate equation

If both forward and backward forms are given for an irreversible reaction, the sign of the affinity determines which form is evaluated. Some rate laws yield rates that are always consistent with the reaction affinities; that is, the sign of the rate always matches that of the thermodynamic driving force, and the rate is zero when that driving force is zero. The transition state theory and activity term rate laws have this property. For such rate laws, one form (forward or backward) can be used to describe the net reaction rate regardless of the sign of the affinity (the user, however, must decide whether or not this is really appropriate). For other rate laws, it is important to remember that EQ6 truncates the calculated rate (sets it to zero) in the absence of a favoring thermodynamic driving force.

The following discussion is patterned after Delany, Puigdomenech, and Wolery (1986). In general, net rate expressions can be written for both the forward (+) and backward (–) directions, thus separately defining v_j and $-v_j$ (or v_j^{rel} and $-v_j^{rel}$), in terms of forward and backward rate constants, respectively. In the code, which form is evaluated depends on the sign of the affinity. If $A_{+,j}$ is positive or zero, the forward form is evaluated; otherwise, the backward form is used.

There is no justification for the use of elaborate expressions for relative rates. Only one expression for relative rates is programmed into EQ6, a truncated second-order Taylor's series in overall reaction progress. The two forms are:

$$v_j^{rel} = k_{+,1j} + k_{+,2j}\xi + \frac{1}{2}k_{+,3j}\xi^2 \tag{D–44}$$

$$v_j^{rel} = k_{+,1j} + k_{-,2j}\xi + \frac{1}{2}k_{+,3j}\xi^2 \tag{D–45}$$

Here $k_{+,ij}$ and $k_{-,ij}$ are forward and backward rate constants. Values for these are entered on the input file when this rate law is selected. Normally, relative rates are specified as constants; i.e., only the first term on the right hand side is used. Picking reasonable values for relative rates is not easy even then. One useful rule of thumb when modeling the interaction of a rock with a water is to pick the relative rates of dissolution of the minerals in the rock such that they are proportional to their abundances.

The transition-state theory (TST) rate law takes the forms:

$$v_j = f_j s_j \sum_{i=1}^{i_{T,+,j}} k_{+,ij} \left(\prod_{n=1}^{n_{T,+,ij}} a_n^{-N_{+}} nij \right) \left(1 - e^{\frac{A_{+,j}}{\sigma_{+,ij} RT}} \right) \tag{D–46}$$

$$v_j = f_j s_j \sum_{i=1}^{i_{T,-,j}} k_{-,ij} \left(\prod_{n=1}^{n_{T,-,ij}} a_n^{-N_{-}} nij \right) \left(1 - e^{\frac{A_{-,j}}{\sigma_{+,ij} RT}} \right) \tag{D–47}$$

Here s_j is the total surface area of the phase dissolving in the j^{th} irreversible reaction. The factor f_j is a fudge factor representing the proportion of effective to total surface area; normally, it is taken as unity. The net forward form provides for treating $i_{T,+,j}$ parallel mechanisms. For each, there is a rate constant ($k_{+,ij}$) a *kinetic activity product*, and a term that depends on the affinity ($A_{+,j}$). The kinetic activity product (which can be symbolized as $q_{+,ij}$; Delany, Puigdomenech, and Wolery 1986) depends on the thermodynamic activities of $n_{T,+,ij}$ species, each raised to a non-zero power characteristic of the mechanism ($-N_{+,nij}$). The kinetic activity product has a value of unity if $n_{T,+,ij} = 0$. The kinetic activity product most often reflects the dependence of the reaction mechanism on *pH*, and usually consists of just the activity of the hydrogen ion raised to some power (See Delany, Puigdomenech, and Wolery 1986). The affinity factor goes to zero when the affinity goes to zero, forcing the reaction rate to do likewise. This factor also depends on the gas constant (R), the absolute temperature (T), and a stoichiometric factor ($\sigma_{+,ij}$) that relates the affinity of a macroscopic reaction ($A_{+,j}$) to that of the corresponding microscopic or elementary reaction ($A_{+,j}/\sigma_{+,ij}$). The stoichiometric factor is usually taken as having a value of unity. The net reverse form parallels the net forward form. If one desires to use TST to describe the kinetics of both the net forward and reverse directions, one ought ordinarily input to the code only one of the above forms and direct the code to use it regardless of the sign of the affinity (set **nrk(1,n)** or **nrk(2,n)**, as appropriate, to *TST rate equation*).

The transition state theory rate law is probably the most important of the kinetic rate laws used in EQ6. For an introduction to the theory suitable for geochemists, see Aagaard and Helgeson (1982) or Lasaga (1981b). Note that the TST rate law is a function of the chemistry of the aqueous solution, but is not an explicit function of time. The rate law for quartz dissolution and growth proposed by Rimstidt and Barnes (1980), though written in a slightly different form, is equivalent to a one-term form of the above equation (See Delany, Puigdomenech, and Wolery 1986). More recent work on quartz dissolution by Knauss and Wolery (1988) suggests a two-term form, reflecting a region of *pH* in which the rate is independent of *pH* and essentially follows the behavior observed by Rimstidt and Barnes (1980), and a region of higher *pH* in which the rate increases with increasing *pH*. The transition state theory rate law form has also been applied to the dissolution kinetics of many other minerals, especially feldspars and other silicates (e.g., Helgeson, Murphy, and Aagaard 1984; Chou and Wollast 1984; Knauss and Wolery 1986, 1988; Murphy and Helgeson 1987, 1989). This is currently a very active field, and it is not possible here to summarize all the progress to date or even to name all of the investigators who have made important contributions to the field.

Transition state theory has the strongest theoretical foundation among the various rate laws that have been applied to the kinetics of mineral dissolution and growth (see for example Lasaga 1981b; Aagaard and Helgeson 1982; Delany, Puigdomenech, and Wolery 1986). Nevertheless, the reader should be aware that in aqueous geochemistry it has been used primarily merely to provide a mathematical form that can be used to explain some measurements. Such explanations may not be unique, even within the scope of the theory itself. The actual mechanisms, and even the number of mechanisms required to explain available data, may be open to debate.

As eqs (D–46) and (D–47) used in the present version of EQ6, the species whose activities may appear in the kinetic activity product must all be aqueous species. Several investigators (e.g., Lasaga 1981a; Chou and Wollast 1984; Murphy and Helgeson 1987; Carroll-Webb and Walther 1988) have developed transition state theory models for mineral dissolution in which speciation on the surface of the mineral is treated in the rate model. Two approaches to incorporating such models into a code such as EQ6 are possible. One is to try to treat the surface speciation implicitly in the rate expression, so that the rate expression is still evaluated in terms of the activities of aqueous species instead of those of surface species with which they are in equilibrium (see Lasaga 1981a, for an example). The other approach is to treat surface speciation in an explicit manner, just as aqueous species are normally treated. The number of moles of surface species would then be included, for example, in calculating mass balances. Kinetic activity products could then include contributions from surface species in the usual manner. This is a more rigorous and general approach, and may be implemented in future versions of EQ6.

Another problem concerns the issue of rate dependence on particle size when ultra-fine particles are present (e.g., Holdren and Berner 1979; Petrovich 1981ab). Attempts have been made to quantify this (e.g., Helgeson, Murphy, and Aagaard 1984; Talman and Nesbitt 1988). No explicit treatment of this is provided in the present version of EQ6. It is possible, however, to use the *special reactant* feature to create a second form of a reactant mineral, with a higher dissolution rate, in order to make a first-order simulation of the presence of a component of ultra-fines.

The constant rate "rate law" forms are:

$$v_j = f_j s_j k_{+,1j} \tag{D–50}$$

$$-v_j = f_j s_j k_{-,1j} \tag{D–51}$$

The rate is not actually constant unless the surface area is constant. There is no dependence of this rate law on chemistry, which is unrealistic. However, this rate law is still of some usefulness.

D.2. TYPES OF EQ6 CALCULATIONAL MODELS

D.2.1. INTRODUCTION

EQ6 calculational models can first be broken down into “single-point” thermodynamic equilibrium calculations and reaction path calculations. A single point thermodynamic calculation is essentially just the special case of a reaction path with no steps. Reaction paths may be calculated for titrations, irreversible reaction in closed systems, and irreversible reaction in certain well defined types of open systems. Such calculations may be in reaction progress mode or time mode, depending on the absence or presence of actual rate laws.

The purpose of this section is to discuss the various kinds of calculations EQ6 is capable of making, to note the assumptions peculiar to the different kinds of calculational scenarios, and to introduce some of the corresponding key input file parameters. A comprehensive discussion of the input file is presented in Section 3.4.

D.2.2. “SINGLE-POINT” THERMODYNAMIC CALCULATIONS

D.2.2.1. General Discussion

Single point thermodynamic calculations are useful for several purposes. For example, suppose an EQ3NR calculation shows that an aqueous solution is supersaturated with several minerals. A single point calculation can find which of possibly many such minerals would actually be present if the system were to reach thermodynamic equilibrium. A single point calculation with a temperature jump can be used to calculate the high temperature, *in situ pH* and other properties of an aqueous fluid in an autoclave experiment from measurements made on quenched fluid samples. A single point calculation is specified on the EQ6 input file by setting *Maximum number of steps* (**kstpmx**) = 0, this located at the bottom of Run Parameters Superblock, Section 3.4.5. This variable is the maximum number of steps of reaction progress that will be taken before the program terminates execution of the current problem.

D.2.2.2. Precipitating Supersaturated Phases

An aqueous solution may be supersaturated with respect to a large number of minerals. EQ6 can be used to calculate the assemblage of aqueous solution plus minerals that would result if the aqueous system reached thermodynamic equilibrium. In general, only a few of the phases that supersaturate the solution prior to precipitation appear in the final assemblage. This is a result of the common ion effect. For example, if the water is supersaturated with respect to more than one aluminum-bearing mineral, precipitation of any one of these minerals reduces the concentration and thermodynamic activity of Al^{3+}, thereby lowering the saturation index of all aluminum-bearing minerals. EQ6 finds the correct phase assemblage through a trial and error process. As part of the calculation, EQ6 determines any changes in the amount of solvent, the *pH*, the *Eh*, etc., that result from the precipitation of the mineral phases.

This type of calculation typically occurs at the beginning of every EQ6 calculation, whether or not it is to be made for just one point. Precipitation of supersaturated phases (so that no solubility is exceeded) is a default condition. Any mineral loaded into memory is eligible to be precipitated unless the user suppresses it. This may be done using either the **nxopt** subset selection suppression options or the **nxmod** suppression option (see Section 3.4.11 for discussion of these options). Exceptions to the subset selection suppression options may be specified (**nxopex**; see again Section 3.4.11). By using the subset selection suppression option **all** in conjunction with the exceptions option, the user may specify just those minerals that are not to be suppressed.

D.2.2.3. The Temperature Jump

A temperature jump occurs when the temperature at the start of an EQ6 calculation does not match that of the initializing calculation done by EQ3NR or a previous run of EQ6. Temperature jumps may occur at the start of a reaction path calculation, as well as in a single point calculation. At the present time, pressure is constrained to be a function of temperature (a limitation of the current thermodynamic data files), so a corresponding *pressure jump* may occur as well. When generalized pressure corrections are available in EQ6, independent pressure jumps may also be possible.

EQ6 detects a temperature jump by calculating the starting temperature from the relevant parameters specified in the top part of the input file (see Section 3.4.2) and comparing it with the temperature of the previous calculation (**tempci**) which appears on the bottom part of the input file.

The temperature jump is useful for calculating the thermodynamic properties of an aqueous solution when the measurements (as of *pH*, *Eh*, etc.) pertain to samples of the fluid that have been heated or cooled from the temperature of interest (such as the *in situ* temperature of an autoclave). Such sampling and measurement are normally carried out so as to avoid mineral precipitation. Thus, all mineral precipitation should be suppressed in the calculations. The **nxopt** subset selection suppression *all* (Section 3.4.11) is convenient for this purpose. Thus, one can use EQ6 to estimate the *in situ* or *at temperature pH* in an autoclave experiment from measurements including the quench (room temperature) *pH*. Other properties, such as the *in situ* oxygen fugacity, *Eh*, *pe*, saturation indices, reaction affinities, and equilibrium gas fugacities, are also calculated.

It is also possible to use a single point temperature jump calculation to find the quench properties that correspond to a fluid generated in an elevated temperature reaction path simulation. For purposes of comparing experimental results with theoretical simulations, it is better to estimate the *in situ* properties from the measured quench properties than to calculate the quench properties corresponding to the simulations. One may have to conduct a series of simulations in order to arrive at a satisfactory model for a single experiment. One might therefore have to perform a larger number of calculations to obtain the quench properties from the simulations than would be required to obtain the *in situ* properties from the experiment.

D.2.3. REACTION PATH CALCULATIONS

There are three major system models for of reaction path calculation in EQ6:

- Closed system (**iopt(1)** = 0).
- Titration (**iopt(1)** = 1).
- Fluid-centered flow-through open system (**iopt(1)** = 2).

Here **iopt(1)** is a parameter that appears on the input file, as discussed in Section 3.4.8 and reproduced here:

```
|iopt(1) - Physical System Model Selection:                              |
|  [x] ( 0) Closed system                                                |
|  [ ] ( 1) Titration system                                             |
|  [ ] ( 2) Fluid-centered flow-through open system                      |
```

. Another possible system model, not currently implemented in EQ6, would be a solid-centered flow-through open system. Any of the above system models can be modified so that the system behaves as though open to a large external reservoir of gas. This results in models with fixed fugacities for the relevant gas species. The “closed” system model is therefore partially open under this option.

Reaction path calculations may be long and complex. The step size control mechanism is a major factor in determining the run time. In its normal calculational mode, EQ6 generates finite-difference approximations of the iteration variables used in the thermodynamic calculations, using data from the most recent points of reaction progress. It uses these to generate starting estimates at a new point. The step size is normally constrained to a range in which these approximations are accurate. Furthermore, EQ6 also normally constrains the step size in the process of locating phase boundaries (where a phase either appears or disappears). A consequence of these step size control factors is that the state of the reacting system is usually calculated at numerous points of reaction progress between those for which printed or plotted output is generated.

When operating in time mode, the finite-difference predictor functions (here generated for both algebraic master variables and rate functions) must be kept accurate in order to get an accurate integration. The normal phase boundary location constraints must also be operative. Full accuracy constraints are also required for any type of flow-through model, in time mode or not. All of these conditions require the use of normal calculation mode. However, in reaction progress mode with **iopt(1)** = 0 or 1, significant reduction in run time may be achievable by modes employing more relaxed constraints on the step size.

EQ6 therefore offers two special calculational modes for obtaining shorter run times. *Economy mode* is selected by setting **iopt(13)** = 1 (see Section 3.4.8). This loosens the general step size constraints associated with maintaining accuracy in the finite-difference predictor functions. It causes EQ6 to operate in a calculational mode more like that of PHREEQE (Parkhurst, Thorstenson, and Plummer 1980). Economy mode does not compromise the code’s abilities to locate phase boundaries. *Super economy mode* may be selected by setting **iopt(13)** = 2. In this mode, the step size is typically larger, usually matching the desired print interval. In super economy mode, the code does not locate phase boundaries. Selection of economy mode or super economy mode is disallowed when other model constraints (such as **iopt(1)** = 2) require normal calculational mode.

D.2.3.1. Simulating a Titration Process

D.2.3.1.1 General Discussion

A titration process involves the addition of reactants to a system. This option is selected in EQ6 by setting **iopt(1)** = 1 on the input file. The titration mode of EQ6 is strictly non-kinetic, so only relative rates may be specified. The concept of relative rates was explained in this Appendix, Section D.1.3. In the titration model, if there is only one reactant, its relative rate is usually set to 1.0. The number of moles of reactant reacted is then normally numerically equivalent to the reaction progress variable ξ.

The titration concept is illustrated in Figure D-1. At each step of reaction progress, a small quantity of reactant (represented by the small cubes) is added to the contents of the flask, which is a system comprised of the aqueous solution and any product phases. After each addition, the increment of reactant dissolves and any product phases re-equilibrate with the aqueous solution. Usually some secondary phases are transients- that is, phases that disappear later on in the process. The growth and dissolution of such phases proceeds at whatever rate is required to maintain solubility equilibrium.

If the aqueous solution becomes saturated with a reactant (in equilibrium with it), then the substance continues to be titrated into the aqueous system according to the governing relative rate law. In the titration model, the calculated rate functions are not truncated by the absence of a thermodynamic driving force, as they are under the closed system (**iopt(1)** = 0) and fluid-centered flow-through open system (**iopt(1)** = 2) options. However, the mass increments under the condition of saturation effectively do not dissolve, but merely accumulate as secondary mass of the same substance. It is as though the small cube in Figure D-1 simply fell to the bottom of the flask without dissolving.

A reactant that is titrated into the system can be any kind of substance: a mineral, non-mineral crystalline solid, a glass phase, a “whole” rock, a gas, or another aqueous solution. The key point is that each reactant must have some stoichiometry. It may or may not have any defined thermodynamic stability (if not, then it is always unstable, and there is no possible saturation condition). A more detailed discussion about reactants and how the user of EQ6 deals with them is presented in Section 3.4.4.

REACTANT

Figure D-1. Conceptual model of a titration process. A reactant is added in small increments (depicted as small cubes) to a system consisting of an aqueous solution and various product minerals. Dissolution of each increment of reactant changes the fluid chemistry, driving the formation of product phase (depicted by the diamond-shaped and tabular crystals on the bottom of the flask).

D.2.3.1.2 Fluid Mixing

A good example of a titration model is fluid mixing. The reactant in this case is a second aqueous solution. It is entered on the input file as a *special reactant*. The composition of one mole of this substance is given that file. See Section 3.4.4 for details. For aqueous solution reactants, the recommended procedure is to equate one "mole" with the mass of solution containing 1 kg of solvent water. The composition of such a solution must be determined by a separate EQ3NR calculation. The special reactant composition of this solution can be conveniently extracted from the EQ3NR pickup file.

One aspect of fluid mixing calculations in EQ6 is that the mass of the system being titrated, and the contained mass of solvent water, increases dramatically. At $\xi = 0$, the mass of solution is around 1 kg and the solvent mass = 1 kg (barring any equilibration at this point). At $\xi = 1$, using the above recommended definition of one “mole” for an aqueous solution, the mass of solution is around 2 kg and the mass of solvent water is also around 2 kg.

D.2.3.1.3 Evaporation

Evaporation can be thought of as a sort of negative titration, in which the titrating substance, solvent water, is removed from the system rather than added to it. An evaporation run is easily set up by declaring $H_2O_{(l)}$ a reactant and specifying a negative relative rate. If v_j^{rel} is set to –1, then each unit advancement of ξ corresponds to the removal of 1 mole of solvent. Recall that the number of moles of water in 1 kg is about 55.51. Assuming the usual case in which there is an amount of aqueous solution containing 1 kg of solvent, the upper practical limit on ξ (specified by the parameter *Maximum Xi Value*, see Section 3.4.5) would be about 55.51. If the extent of evaporation would increase the ionic strength above about one molal, the user should specify an activity coefficient option that is valid for the case of highly concentrated salt solutions. In the present version of EQ3/6, the only option suitable for this is Pitzer’s equations (**iopg(1)** = 1, Section 3.3.16).

D.2.3.2. Calculating the Reaction Path in a Closed System

Calculating the reaction path in a closed system is conceptually and calculationally very similar to the case of computing a simulation of a titration process. This option is specified on the input file by **iopt(1)** = 0. The conceptual process is illustrated in Figure D-2. The chief difference between the closed system model is that the unreacted masses of reactants are considered to be *in the system*, instead of outside it. A small increment of the remaining mass of each reactant dissolves at each step of reaction progress. The effect is the same as in the titration model, with one exception.

In the closed system, a reactant that saturates may be treated in one of two ways. If a precipitation kinetics rate law is specified for such a reactant, then the unreacted mass is treated according to the rate law specifications, as is the case under the titration option. However, if no such rate law is specified, then all of the unreacted mass is immediately transferred to secondary mineral status. The governing rate function is then set to zero. This is primarily a consequence of there being no material left to react, although it is also true that some rate laws (see this Appendix D.1.3) evaluate to zero when the driving affinity is zero. In this case, the rate description of the formation/dissolution of the phase shifts from the specified rate law to the rate that is consistent with keeping the reaction at equilibrium.

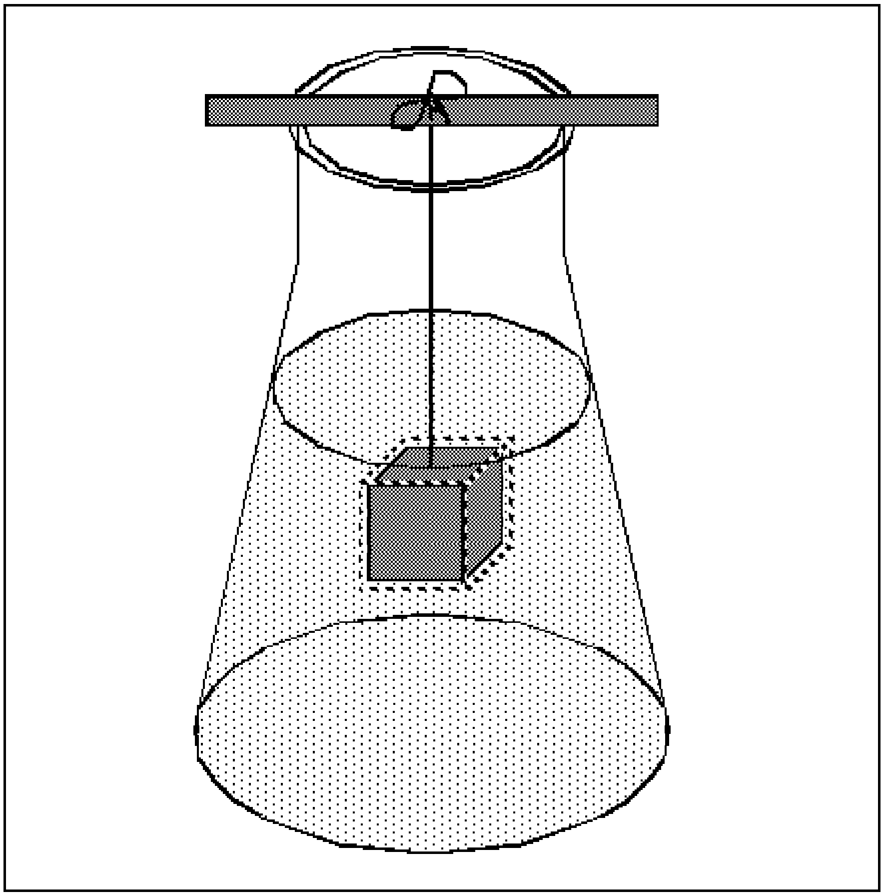

Figure D-2. Conceptual model of irreversible reaction in a closed system. The reactant is represented by the large cube, which dissolves in increments represented by layers. Product minerals are represented by the diamond-shaped and tabular crystals on the bottom of the flask.

D.2.3.3. A Fluid-Centered Flow-Through Open System

The fluid-centered flow-through system is a special type of open system that follows the evolution of a particular packet of water as it flows through a medium. This option is specified on the input file by **iopt(1)** = 2. The medium could be a fracture, a pipe, or a porous medium. The concept is illustrated in Figure D-3. Reactants are presumed to line the medium in homogeneous fashion and interact with the fluid packet as it passes by. Alternatively, there may be no reactants, only a change in temperature or pressure. Either way, secondary phases form as a result. As the packet moves on, it physically separates from the masses of secondary phases produced. The result is that transiently formed product phases do not have the opportunity to redissolve in that particular packet of fluid. A consequence is that overall equilibrium can be achieved sooner in terms of reaction progress than is the case in the closed system.

Calculationally, it is not necessary to separate the masses of secondary pure minerals at each step of reaction. It is only necessary to make sure that no mass (actually there is a small but finite limit) of any such phase is destroyed. More of a problem results when solid solution products are involved. Because their composition is in general continually changing, they must be removed as they form.

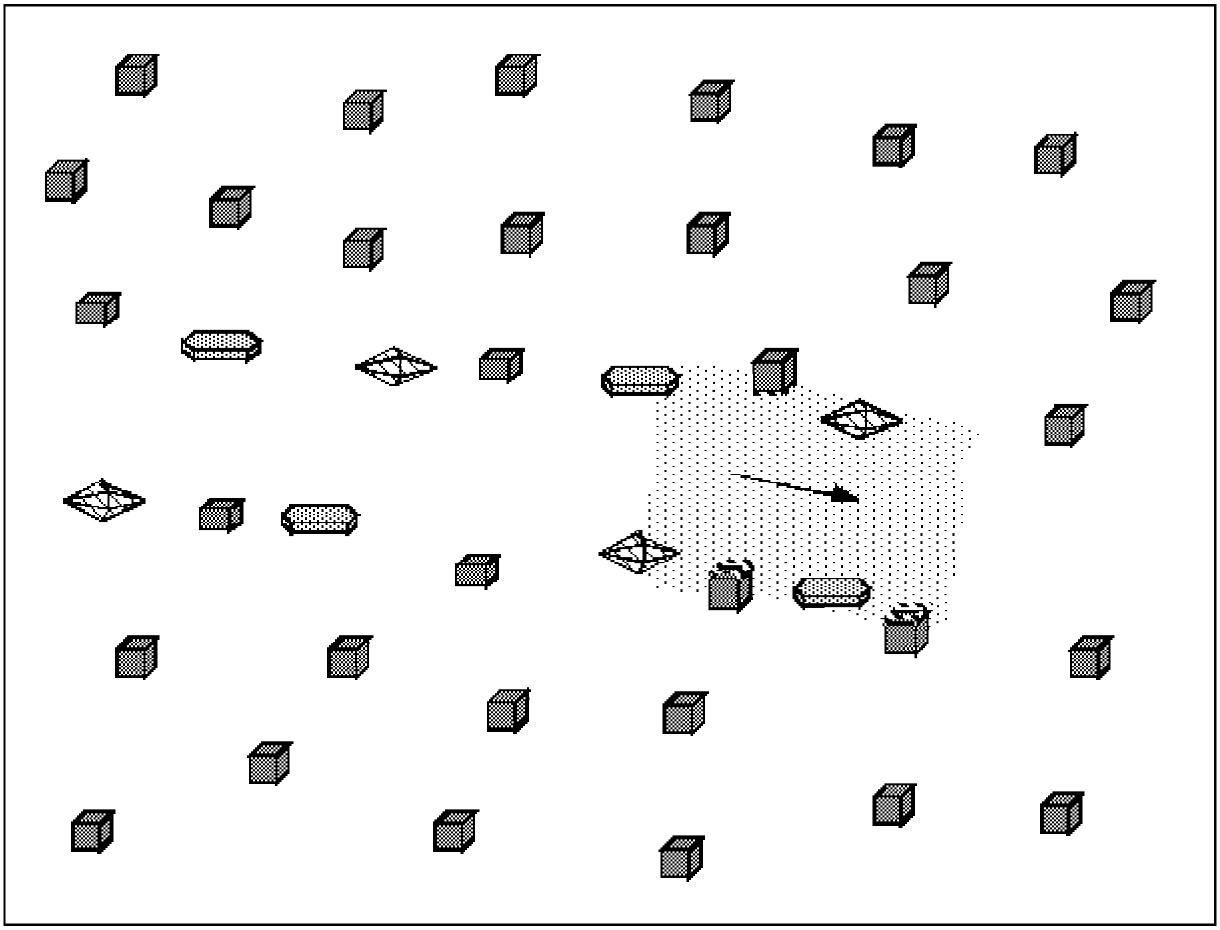

Figure D-3. Conceptual model of irreversible reaction in a fluid-centered flow-through open system. The packet of water is shown in light gray. The arrow marks the direction of flow. The reactant is represented by the small cubes. The product minerals, depicted by diamond-shaped and tabular crystals, stick to the medium and become physically separated from the packet of water. Note that no product minerals appear ahead of the package of water, which is the first packet.

D.2.3.4. A Solid-Centered Flow-Through Open System

A fundamentally different type of flow-through system would focus on the evolution of solids interacting with a mass of fluid that is either continuously or discretely recharged by a fresh supply of aqueous solution of fixed composition. This concept is illustrated in Figure D-4. Such an option is not currently programmed into EQ6. This system closely matches the scenario in many flow-through interactions experiments (for example, the feldspar dissolution rate experiments reported by Knauss and Wolery, 1986). The system of primary and secondary solids and the mass of aqueous phase about them is equivalent to the contents of a leaching cell. The replacement water must be specified as a reactant, and appropriate specification of the replacement process must also be made. The input of fresh aqueous solution is balanced by the output of an equivalent mass.

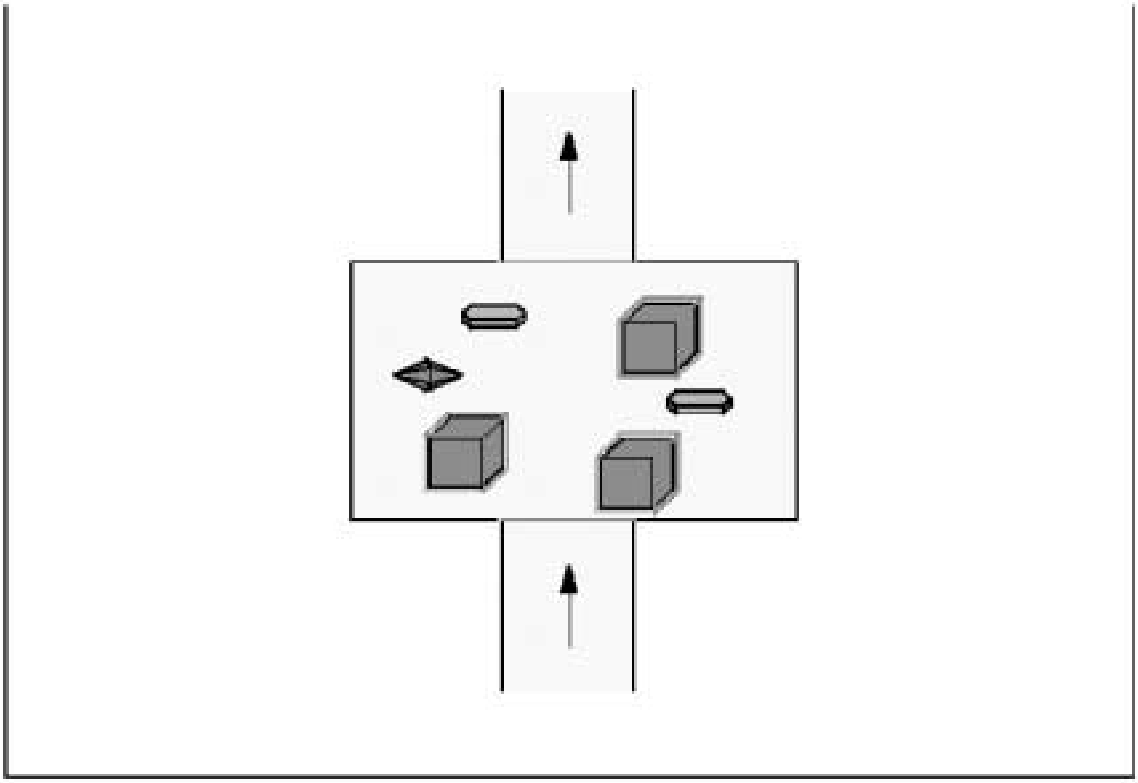

Figure D-4 Conceptual model of irreversible reaction in a solid-centered flow-through open system. Water flows in at a constant composition and flows out at another, variable composition. The direction of flow is shown by the arrows. The reactant is represented by the cubes. The product minerals are depicted by diamond-shaped and tabular crystals. All solids are constrained to remain physically in the system.

D.2.3.5. Systems Open to External Gas Reservoirs

This option has been described elsewhere (Delany and Wolery, 1984). It assumes that the reacting system (of any of the above discussed types) is in contact with a large external gas reservoir, such that specified gas species move to or from that reservoir in order to maintain specified fixed fugacities in the reacting system. Models of this type may be appropriate for describing weathering at the earth's surface, reactions in soils, geochemical interactions in partially saturated (in the hydrologic sense) rock formations, and reactions in certain kinds of experimental configurations. The gas species most likely to be appropriately treated by this option are O_2 and CO_2.

This option is invoked by specifying each gas to be so treated, the desired fugacity, and an optional mass of the gas to be added to the system at the start of the run. Calculationally, the "external reservoir" is treated as part of the reacting system, and the desired fugacity is maintained by the solubility equilibrium for a fictive mineral whose composition is identical to that of the gas. Without the presence of such a phase, the desired fugacity is really only an upper limit. That is the reason for the option to add mass of the gas component to the system. This permits the user to saturate the system. The mass of such a fictive mineral is finite and may be exhausted. The user may wish to choose the initial addition of mass of gas to the system in order define a point at which the external reservoir of the gas is exhausted.

The idea behind using a fictive mineral is as follows. Reactions for the dissolution of gas species are maintained on the data file in order to permit the calculation of equilibrium fugacities. Such a reaction is illustrated by the case for $CO_{2(g)}$:

$$CO_{2(g)} + H_2O_{(l)} = H^+ + HCO_3^- \tag{D–52}$$

The corresponding mass action equation is:

$$\log a_{H^+} + \log a_{HCO_3^-} - \log f_{CO_2} - \log a_w = \log K_{CO_{2(g)}} \tag{D–53}$$

Now consider the parallel reaction for the dissolution of the corresponding fictive mineral ($CO_{2(s)}$):

$$CO_{2(s)} + H_2O_{(l)} = H^+ + HCO_3^- \tag{D–54}$$

The corresponding mass action equation is:

$$\log a_{H^+} + \log a_{HCO_3^-} - \log a_w = \log K_{CO_{2(s)}} \tag{D–55}$$

(the thermodynamic activity of a pure solid is unity). Subtracting eq (D–52) from eq (D–54) and eq (D–53) from eq (D–55) respectively yields:

$$CO_{2(s)} = CO_{2(g)} \tag{D–56}$$

and

$$\log f_{CO_2} = \log K_{CO_{2(s)}} - \log K_{CO_{2(g)}} \tag{D–57}$$

This shows how a fixed fugacity model can be imposed by equilibrium with a fictive solid. A slight rearrangement of eq (D–53) shows how to choose the equilibrium constant for the fictive mineral:

$$\log K_{CO_{2(s)}} = \log K_{CO_{2(g)}} + \log f_{CO_2} \tag{D–58}$$

Invocation of this option causes the corresponding gas reaction and its thermodynamic data to be copied into the corresponding mineral data arrays. The log K function is then modified according to eq (D–53).

D.3. THERMODYNAMIC CALCULATIONAL METHODS

D.3.1. INTRODUCTION

The governing equations for reaction path calculations were introduced here in D.1. This section continues the development begun there, to show how EQ6 actually solves the system composed of these equations. As was pointed out in D.1, the governing equations fall into two categories. The first deals with the equations that present themselves in the calculation of thermodynamic equilibrium. These equations are fundamentally algebraic (e.g., mass balance, mass action). The second problem deals with rate equations, which are by nature ordinary differential equations (ODEs). The calculational problem involving these is integration, generally numerical integration.

This section discusses the means by which EQ6 carries out thermodynamic equilibrium calculations. The methodology is very similar to that used in EQ3NR (see Appendix B.4). The reader of the present section is presumed to be familiar with methodology used in EQ3NR. The following section in the present report deals with the integration of rate equations.

D.3.2. THE SET OF MASTER ITERATION VARIABLES

In the EQ6 code, the set of master iteration variables is similar to that used in EQ3NR, with two key differences. First, concentration variables are replaced by number of moles variables. In the case of the solvent, water, the concentration is expressed by the mole fraction (x_w); the number of moles is represented by n_w. In the case of aqueous solute species, the concentration is expressed by the molality (m_i); the corresponding number of moles is represented by n_i. When no product minerals are present, the vector of master iteration variables ($\underline{z}$) has the following structure:

$$z = \begin{array}{c} \log n_w \\ . \\ . \\ \log n_{s'}\ ,\ s' = 1, s_B - 1\ ,\ s \neq w \\ . \\ . \\ \log f_{O_2} \\ . \\ . \\ \log n_{s'}\ ,\ s' = s_B + 1, s_Q \end{array} \qquad \text{(D–60)}$$

This is exactly analogous to that used in EQ3NR. All of these variables formally correspond to mass balance relations, except the fictive redox species $O_{2(g)}$, which formally corresponds to the charge balance equation. The first part of this vector contains entries for the s_B strict basis species. The last of these is $O_{2(g)}$, which continues to be represented in the set of master iteration variables by the corresponding fugacity. The second part contains entries for any auxiliary basis variables for which **jflag** ≠ *Make non-basis* (those with **jflag** = *Make non-basis* are treated as non-basis species, see the input file example in Section 3.3.9). The structure is further simplified if $s_Q = s_B$ (no active auxiliary basis set), which implies that the aqueous solution is in a state of complete internal (homogeneous) equilibrium. Otherwise, partial internal disequilibrium is implied. In the present version of EQ6, the code's internal data structure differs from that of EQ3NR and does not permit the use of an auxiliary basis set. For conceptual reasons, however, we include it in the present discussion. Some EQ6 problems are redox indifferent. In this case, the redox variable is automatically dropped from the basis set (the option switch **iopt(15)** can be set to *"Do it"* to suppress all aqueous redox reactions to avoid problems caused by the presence of insignificant amounts of the species $O_{2(aq)}$ and $H_{2(aq)}$).

If product minerals are present in the equilibrium system, the vector of master iteration variables is expanded to include the number of moles of the relevant species:

$$z = \begin{array}{c} \log n_w \\ \cdot \\ \cdot \\ \log n_{s'} \ , \ s' = 1, s_B - 1 \ , \ s \neq w \\ \cdot \\ \cdot \\ \log f_{O_2} \\ \cdot \\ \cdot \\ \log n_{s'} \ , \ s' = s_B + 1, s_Q \\ \cdot \\ \cdot \\ \log n_\phi \ , \ \phi = 1, \phi_T \\ \cdot \\ \cdot \\ \log n_{\sigma\psi} \ , \ \sigma = 1, \sigma_{T,\psi} \ , \ \psi = 1, \psi_T \end{array} \qquad \text{(D–61)}$$

These additional master iteration variables formally correspond to the associated mass action equations. Here ϕ denotes a pure mineral in the equilibrium system, n_ϕ is the corresponding number of moles, and ϕ_T is the number of such pure minerals. If ϕ_T is zero, this block is simply deleted. Here also ψ denotes a solid solution in the same system, and ψ_T is the number of such phases. If ψ_T is zero, this block is also simply deleted. Here σ denotes an end member component, $n_{\sigma\psi}$ is the number of moles of the σ^{th} component of the ψ^{th} solid solution, and $\sigma_{T,\psi}$ is the number of such components belonging to the ψ^{th} solid solution present in the equilibrium system.

D.3.3. EXPANDING THE SYSTEM FROM THE SET OF MASTER ITERATION VARIABLES

One may "expand the system" from the vector of master iteration variables $\underline{z}$ by computing the number of moles, concentrations, and activity coefficients of all species present in the equilibrium system. As was pointed in the case of EQ3NR (Appendix B.4), this is not a straightforward process when non-basis species are present in the model. The problem is that for any phase containing such species, the concentrations of the non-basis species are generally required to compute the activity coefficients; one must evaluate the corresponding mass action equations. The activity coefficients appear in these equations. Hence, the activity coefficients must be evaluated first. However, the activity coefficients depend in general on the concentrations of all solute species, both basis and non-basis. So to deal with these, one must compute the concentrations of the non-basis species first.

The approach to resolving this conundrum is nearly the same as that taken in the case of EQ3NR (see Appendix B.4). The system is expanded by first calculating the new concentrations of the non-basis species, using the existing values of the activity coefficients. The activity coefficients are then recalculated. In the pre-Newton-Raphson optimization algorithm, the computed concentrations of the basis and non-basis species may be adjusted several times before the activity coefficients are recalculated. In the hybrid Newton-Raphson method, they are recalculated between each Newton-Raphson step. The expansion itself technically calls for a process of repeated steps, each consisting of recalculation of the concentrations of non-basis species, followed by recalculation of the activity coefficients. However, EQ6 uses a one step or single update method in hybrid Newton-Raphson iteration, because there seems to be no definite advantage to the use of repeated steps.

D.3.4. BEGINNING THE PROCESS: COMPUTING STARTING VALUES

The problem of assigning starting values is very different in EQ6 than it is in EQ3NR. Starting values at the initial point of reaction progress are read from the input file. These originate from either an EQ3NR pickup file or an EQ6 pickup file. Starting values at subsequent points are calculated using finite-difference based predictor functions. If a calculation then requires precipitation of a new product phase to eliminate a corresponding supersaturation, the starting value for the number of moles of this phase is taken as 5 percent of the possible maximum value, as computed from the aqueous phase composition. If the phase is a solid solution, the starting value for the number of moles of each end member component is assigned by taking the product of 5 percent of the maximum number of moles of the phase and the mole fraction of the component corresponding to the composition which maximizes the affinity function for the phase.

D.3.5. METHODS TO AID CONVERGENCE

Several techniques are used in EQ6 to aid convergence of the thermodynamic equilibrium calculations. They include:

- Use of logarithmic iteration variables
- Under-relaxation techniques
- Automatic and user-specified basis switching

The physical quantities that correspond to the iteration variables (number of moles) used by EQ6 are intrinsically positive. Use of logarithmic iteration variables prevents iteration from producing negative numbers. Other than the fact that EQ6 uses numbers of moles rather than concentrations as master variables, the usage of this technique is the same as in EQ3NR.

Under-relaxation is the technique of judiciously reducing the magnitude of the computed correction terms. Assume that the unmodified method involves adding a correction term vector (δ_k), where *k* is the iteration number. This is typical in Newton-Raphson iteration. The new vector of master iteration variables is obtained thusly:

$$\underline{z}_{k+1} = \underline{z}_k + \underline{\delta}_k \qquad \text{(D–62)}$$

If the new vector of master iteration variables is obtained instead by evaluating some set of corresponding equations not in this format, one can still utilize under-relaxation by defining a correction term vector as follows:

$$\underline{\delta}_k = \underline{z}_{k+1} - \underline{z}_k \qquad \text{(D–63)}$$

Global under-relaxation is effected by replacing the correction equation given above by:

$$\underline{z}_{i+1} = \underline{z}_i + \kappa\underline{\delta}_i \qquad \text{(D–64)}$$

where κ is a positive number less than one. Non-global under-relaxation is also possible. This does not involve the use of an under-relaxation factor. Rather it involves truncating the magnitudes of individual correction terms to satisfy specified limits, which may be different depending on the species involved and on the direction of change.

There are several methods of applying global under-relaxation, distinguished by different methods of choosing a value for the under-relaxation factor. EQ6 uses two simple ones in making Newton-Raphson steps.

The first of these limits the size of the largest correction term:

$$\kappa = \frac{\delta'}{\delta_{\max}} \qquad \text{(D–65)}$$

where δ' is the imposed limit and δ_{max} is the max norm of δ. In a Newton-Raphson iteration step, this limit is represented by the variable **screwd**. In EQ3NR, this is set at a value of 2.0. In EQ6, it is set on the input file through the variable **screw5**. This has a default value of 4.0. This method of under-relaxation not only aids convergence, but is very helpful in inducing iteration to provide helpful information about the nature of the problem when it diverges. This information may be used by EQ6 to pick a phase to delete from the equilibrium phase assemblage or to enter the redox scan mode.

The other global under-relaxation method is applied for only the first 8 iterations. The under-relaxation factor is cut in half if the residual vector max norm β_{max} exceeds the value of the variable **screwn**. Initially, κ is set to a value of unity; when the current method of under-relaxation is applied, this factor may have been reduced as a result of applying the method described above. Initially, it is set to a value of 0.10. If the phase assemblage is changed, the value of **screwn** is temporarily reduced to 0.005. In practice, this under-relaxation method comes into play relatively infrequently in EQ6 compared to the one described above.

Some degree of non-global under-relaxation is also employed in pre-Newton-Raphson optimization. Here under-relaxation is effected by imposing truncation limits on changes for individual master variables. The master variables for water and the hydrogen ion may not change in a given step by more than certain limits, which are specific to each species (0.05 and *2*.0 log units, respectively) Other master variables are not subject to truncation limits.

Some truncation limits also apply to the activity coefficients and the functions Σm and the ionic strength. These limits are applied during both pre-Newton-Raphson optimization and hybrid Newton-Raphson iteration, and, in EQ6, during calculations involving the use of predictor functions. These limits are defined in the variable **chgfac**. The value of this variable is set in the calling modules, and is usually scaled inversely with the value of Σm. Values range from 1.3 to 100.

Basis switching, the practice of changing the aqueous species in the basis set during execution of EQ3NR or EQ6, is often a very beneficial and sometimes also necessary device in order to achieve convergence. The general rule is that it is best to choose a basis species that makes up a significant fraction of the corresponding mass balance. For example, if $UO_2(CO_3)_2^{2-}$ makes up most of the total balance of uranium, then it is a better choice for the corresponding basis species than the data file master species, UO_2^{2+}. Experience with the Newton-Raphson method in EQ3NR and EQ6 has shown that basis-switching is occasionally critical to achieving convergence. It is not necessary for the chosen basis species to dominate the corresponding mass balance. However, it is critical that it not compose an extremely scarce fraction of that mass balance. The present version of EQ6 does not allow water, the hydrogen ion, or O_2 (the fictive redox species) to be switched out of the active basis set. Also, it does not allow non-aqueous species(e.g., pure minerals and solid solution end member components) to be switched into the active basis set.

The user may specify certain basis switches on the input file. These switches are executed prior to calculation of the state of the system at the initial point of reaction progress. This appears on the part of the input file that corresponds to the EQ3NR pickup file. If the user specifies certain basis switches on the EQ3NR input file, these switches will be listed on the corresponding pickup file. It is probably best to make such switches at this point. However, it is possible to direct such switches later by modifying the appropriate part of the EQ6 input file.

There is also an automatic basis switching mode, specified by setting **iopt(11) or iopt(12)** = 1 (**iopt(11)** is also available in EQ3NR). It is not necessary to invoke automatic basis switching for most reaction path runs. Automatic basis switching can operate in EQ6 as part of the pre-Newton-Raphson optimization method by **iopt(11)**, or after when the automatic basis switching is made after a successful calculation of the state of the system at a given point of reaction progress **iopt(12)**.

D.3.6.THE PRE-NEWTON-RAPHSON OPTIMIZATION ALGORITHM

The pre-Newton-Raphson optimization algorithm is simplified in comparison to that used in EQ3NR in that there is a much smaller number of types of constraints corresponding to the set of master iteration variables.

The optimization process begins by recalculating the Σm, the ionic strength, and the activity coefficients. It then re-expands the system and computes a full suite of residual functions. It then utilizes a loop structure similar to that employed in EQ3NR. The primary loop structure consists of *passes*. At the end of a pass, Σm, the ionic strength, and the activity coefficients are recomputed. Within each pass is another loop structure, the times through which are called *cycles*. Here, adjustments are made to the number of moles of the basis species corresponding to balance equations. A pass is completed after some number of cycles. The cycles within a pass terminate if some rather rough convergence criteria are satisfied, or if the maximum number of cycles in a pass have been completed. The passes terminate if rough convergence criteria applying to both the cycles and passes are satisfied, or if the maximum number of passes has been completed. This is determined by the variable **nplim**, which is currently set to 5.

To illustrate the cycle algorithm, we again consider the case of aluminum. The total aluminum in EQ6 is presently expressed as total elemental aluminum. This includes aluminum in both the aqueous solution and any other phases present in the equilibrium system. The normalized mass balance residual is:

$$\beta_{Al} = \frac{n_{T,calc,Al} - n_{T,Al}}{n_{T,Al}} \qquad \text{(D–66)}$$

where: $n_{T,calc,Al}$ is the total number of moles of aluminum as calculated from a mass balance expression, using the current values of the master iteration variables. Here $n_{T,Al}$ is the total number of moles of aluminum in the equilibrium system.

The basic cycle algorithm is based on that employed in EQ3NR and is modified principally only in that numbers of moles replace molalities. As applied to basis species other than water or O_2 (the fictive redox species), this assumption assumes that there is little change in the number of moles of water. Assuming that the basis species is that Al^{3+} and that it dominates the calculated mass balance, the cycle algorithm is illustrated by the equation:

$$n_{Al^{3+},k+1} = \frac{n_{Al^{3+},k}}{\beta_{Al,k} + 1} \tag{D–67}$$

If the basis species is Al^{3+} and the aluminum mass balance is dominated by the complex $Al_{13}O_4(OH)_{24}^{7+}$, the equation is modified to the form:

$$n_{Al^{3+},k+1} = \frac{13^{\frac{1}{13}}\, n_{Al^{3+},k}}{\left(\beta_{Al,k} + 1\right)^{\frac{1}{13}}} \tag{D–68}$$

where the “13” is the ratio of the reaction coefficients of Al^{3+} and $Al_{13}O_4(OH)_{24}^{7+}$ in the reaction for the dissociation of $Al_{13}O_4(OH)_{24}^{7+}$:

$$Al_{13}O_4(OH)_{24}^{7+} + 32H^+ = 13Al^{3+} + 28H_2O_{(l)} \tag{D–69}$$

This algorithm is applied to all basis species corresponding to mass balance equations, including water and the hydrogen ion. Special under-relaxation truncation limits apply to these latter two species, as was discussed earlier in this section.

There are two principle deficiencies to the cycle algorithm described above. First, it is not complete. The master variable corresponding to O_2 (the fictive redox species) is not optimized. Neither are any master variables defined for pure minerals and solid solutions. Second, the optimization equations illustrated above were derived on the assumption that the dominant species is an aqueous species. They do not work well when all aqueous species contribute insignificantly to a mass balance. Thus, on both counts, the optimization algorithm tends to perform poorly in rock dominated systems.

At the end of a cycle, a full set of residual functions is computed. This includes the β array and its max norm, β_{max}. A pass (sequence of cycles) ends when one of the following occurs:

The non-zero (mass balance) elements of β satisfy a loose convergence test (all fall in the range –10 percent to +10 percent).

The maximum number of cycles per pass have been completed.

A convergence function β_{func} (**betfnc**) indicates that iteration in the present cycle is diverging. This convergence function will be discussed later in this section.

At the end of a pass, the Σm function, the ionic strength, and the activity coefficients are recalculated. The code defines residual functions based on the magnitude of the changes in Σm, the ionic strength, and the activity coefficients from the values pertaining to the previous pass. The sequence of passes is stops when one of the following occurs:

The residuals defined for Σm, the ionic strength, and the activity coefficients satisfy a loose convergence test (–10 percent to +10 percent).

The maximum number of passes have been completed.

The optimization is deemed successful if both sets of loose convergence tolerances are satisfied. Following optimization, the code executes hybrid Newton-Raphson iteration, whether or not the optimization was successful.

The incomplete nature of the cycle algorithm in EQ6 is a major weakness in the present version of this code. This makes it difficult for the code to successfully deal with large steps in reaction progress. In particular, it causes the code some difficulty in dealing with large changes in temperature, such as are encountered in a temperature jump. This problem is basically responsible for the very limited usefulness of "economy mode" and "super economy mode," as the code in these modes tries to take large steps. The deficiencies of the present cycle algorithm sometimes also cause the code to spend a lot of run time trying to optimize starting estimates with little result, thus causing long run times.

D.3.7. THE NEWTON-RAPHSON METHOD

The Newton-Raphson methods for implementing it in EQ6 are very similar and in many aspects exactly the same as those in EQ3NR. The differences that do exist come about because the Given a set n governing equations and n unknowns (represented by a vector $\underline{z}$ of iteration variables), one may construct a set of residual functions, represented by the vector α, which provides a measure of the degree to which the governing equations are not satisfied. Each element of this array has a value of zero when the $\underline{n}$ equations are satisfied. Both $\underline{z}$ and $\underline{\alpha}$ are of length n.

Let k be the number of iterations, such that z_k and α_k are the iteration variable and residual function vectors on the k^{th} iteration. Let z_0 represent the set of starting estimates. An iteration step is made by calculating z_{k+1} from z_k. The Newton-Raphson method does this by computing a vector of correction terms, δ, by solving the matrix equation:

$$\underline{J}\,\underline{\delta} = -\underline{\alpha} \qquad \text{(D–70)}$$

Here $\underline{J}$ is the Jacobian matrix, defined as

$$\underline{J} = \left(\frac{\partial \alpha_i}{\partial z_j} \right) \tag{D–71}$$

where i and j are the matrix coordinates. The correction term is then used to generate a new iteration variable vector by adding it to the old one:

$$\underline{z}_{k+1} = \underline{z}_k + \underline{\delta}_k \tag{D–72}$$

If the iteration converges, all elements of both α and δ approach zero. It is useful to define another residual function vector β, which is identical to α, except that some elements may be normalized to provide a better measure of convergence. Such relative residuals for mass balances are defined by dividing the absolute residual by the corresponding total number of moles. The relative residual for the charge balance is defined as the absolute residual divided by the calculated total number of charge equivalents. It is then convenient to define β_{max} and δ_{max} as the largest absolute values of the elements of β and δ, respectively. Both β_{max} and δ_{max} may then be used in tests to determine if the iteration has converged satisfactorily.

Useful measures of how well convergence is proceeding may be constructed. The Newton-Raphson method is a so-called second-order method, meaning that convergence should be very rapid in a close neighborhood of the solution. This behavior suggests that in a such a region, $\delta_{max,k+1}$ should be much less than $\delta_{max,k}$. The function δ_{conv} (the variable **delfnc** in EQ3NR and EQ6) defined as:

$$\delta_{conv,k+1} = 1 - \left(\frac{\delta_{\max,k+1}}{\delta_{\max,k}} \right) \tag{D–73}$$

therefore tends to approach (from below) a value of unity when convergence is rapid. Convergence to a significantly lesser apparent limiting value, say 0.72 instead of 0.99, may imply an error in the Jacobian matrix. Similar behavior is expected from an analogous function defined in terms of the β_{max} residual (β_{conv}, the variable **betfnc** in EQ3NR and EQ6):

$$\beta_{conv,k+1} = 1 - \left(\frac{\beta_{\max,k+1}}{\beta_{\max,k}} \right) \tag{D–74}$$

First one makes a single Newton-Raphson step, and then recomputes activity coefficients and computes the number of moles of dependent species. The latter item is called between Newton-Raphson steps, in accordance with the single update method that was discussed earlier in this section.

The maximum number of iterations in a Newton-Raphson calculation is determined by the input file variable **itermx**. This has a default value of 40 in EQ6. Convergence is achieved when β_{max} is less than the tolerance parameter **tolbt**, δ_{max} is less than the tolerance parameter **toldl**, and max norms on the changes in the Σm function, the ionic strength, and the activity coefficients are all less than **tolbt**. The tolerance parameters **tolbt** and **toldl** both appear on the input file, and both have a default value of $1\text{x}10^{-6}$.

D.3.8. DERIVATION OF RESIDUAL FUNCTIONS AND THE JACOBIAN MATRIX

In this section, we shall derive the residual functions and the Jacobian matrix for the Newton-Raphson iteration procedures used by the EQ3NR code. Given a set of governing equations and an equal number of unknowns, there is no unique way to formulate residuals and Jacobians. The number of equations and unknowns may be reduced by means of substitutions. Furthermore, one may then construct the residual functions in any number of ways. Once the residual functions have been chosen, the form of the Jacobian is determined according to the partial derivatives of these functions.

We will now take each remaining governing equation, construct a corresponding pair of residual functions (α and β), and derive the corresponding row of elements in the Jacobian matrix by partial differentiation. The α residuals are the true Newton-Raphson residual functions and are the subject of partial differentiation to define the Jacobian matrix. The β residuals are better measures of satisfactory convergence. In the hybrid Newton-Raphson method currently used in EQ3/6, the activity coefficients of aqueous species are treated as known constants in a Newton-Raphson step. The partial differentiation of the α residuals therefore does not flow through these variables.

D.3.8.1. Mass Balance

As was pointed out in Section D.1.2.1, mass balances can be defined in two ways: in terms of basis species (excluding the fictive redox species O_2, which is the $s_B{}^{\text{th}}$ species), and in terms of chemical elements. The former treatment is more general, and permits the use of an auxiliary basis set. The latter, however, is the basis for the treatment in the present version of EQ6. The former treatment is equivalent to the latter if an auxiliary basis set is not used. Otherwise, they are not equivalent and the number of basis species minus one is greater than the number of chemical elements and there is hence a greater number of corresponding mass balance equations. As the former treatment is likely to be incorporated into future versions of EQ6, the equations for both treatments will be presented here.

Mass balance for the s'^{th} basis species is expressed by eq (D–2):

$$\sum_{s=1}^{s_T} u_{s's} n_s + \sum_{\phi=1}^{\phi_T} u_{s'\phi} n_\phi + \sum_{\psi=1}^{\psi_T} \sum_{\sigma=1}^{\sigma_{T,\psi}} u_{s'\sigma\psi} n_{\sigma\psi} = n_{T,s'} \qquad \text{(D–75)}$$

(see this appendix Section D.1). The corresponding residual functions are defined by:

$$\alpha_{s'} = -n_{T,s'} + \sum_{s=1}^{s_T} u_{s's} n_s + \sum_{\phi=1}^{\phi_T} u_{s'\phi} n_{\phi} + \sum_{\psi=1}^{\psi_T} \sum_{\sigma=1}^{\sigma_{T,\psi}} u_{s'\sigma\psi} n_{\sigma\psi} \tag{D–76}$$

$$\beta_{s'} = \frac{\alpha_{s'}}{n_{T,s'}} \tag{D–77}$$

Mass balance for the εth chemical element is expressed by eq (D–3):

$$\sum_{s=1}^{s_T} c_{\varepsilon s} n_s + \sum_{\phi=1}^{\phi_T} c_{\varepsilon\phi} n_{\phi} + \sum_{\psi=1}^{\psi_T} \sum_{\sigma=1}^{\sigma_{T,\psi}} c_{\varepsilon\sigma\psi} n_{\sigma\psi} = n_{T,\varepsilon} \tag{D–78}$$

(again, see Section D.1). The corresponding residual functions are defined by:

$$\alpha_{\varepsilon} = -n_{T,\varepsilon} + \sum_{s=1}^{s_T} c_{\varepsilon s} n_s + \sum_{\phi=1}^{\phi_T} c_{\varepsilon\phi} n_{\phi} + \sum_{\psi=1}^{\psi_T} \sum_{\sigma=1}^{\sigma_{T,\psi}} c_{\varepsilon\sigma\psi} n_{\sigma\psi} \tag{D–79}$$

$$\beta_{\varepsilon} = \frac{\alpha_{\varepsilon}}{n_{T,\varepsilon}} \tag{D–80}$$

The corresponding rows of the Jacobian matrix are obtained by partial differentiation of the α residuals with respect to the algebraic master variables. The numbers of moles variables appearing in the above equations are all directly related to the algebraic master variables, except those pertaining to the non-basis aqueous species. In deriving the Jacobian matrix, it is helpful to isolate the corresponding terms by rewriting eq (D–76) as:

$$\alpha_{s'} = -n_{T,s'} + u_{s's'} n_{s'} + \sum_{s''=s_Q+1}^{s_T} u_{s's''} n_{s''} + \sum_{\phi=1}^{\phi_T} u_{s'\phi} n_{\phi} \tag{D–81}$$

$$+ \sum_{\psi=1}^{\psi_T} \sum_{\sigma=1}^{\sigma_{T,\psi}} u_{s'\sigma\psi} n_{\sigma\psi}$$

Here s_Q is the number of basis species and s'' denotes a non-basis aqueous species Eq (D–79) can be similarly rewritten as:

$$\alpha_{\varepsilon} = - n_{T,\varepsilon} + \sum_{s'''=1}^{s_Q} c_{\varepsilon s'''} n_{s'''} + \sum_{s''=s_Q+1}^{s_T} c_{\varepsilon s''} n_{s''} + \sum_{\phi=1}^{\phi_T} c_{\varepsilon\phi} n_{\phi}$$

$$+ \sum_{\psi=1}^{\psi_T} \sum_{\sigma=1}^{\sigma_{T,\psi}} c_{\varepsilon\sigma\psi} n_{\sigma\psi} \tag{D–82}$$

It is then necessary to find the $n_{s''}$ as functions of the $n_{s'}$. The relevant mass action equation can be written in the following form:

$$\log K_r = b_{wr}(\log x_w + \log \lambda_w) + b_{s_B r} \log f_{O_2} + b_{s''r}(\log m_{s''} + \log \gamma_{s''})$$

$$+ \sum_{\substack{s'=1 \\ s' \neq w, s_B}}^{s_Q} b_{s'r}(\log m_{s'} + \log \gamma_{s'}) \tag{D–83}$$

where r is the reaction for the s''^{th} aqueous species, K_r is the thermodynamic equilibrium constant for the reaction b_{sr} is the reaction coefficient for the s^{th} species, λ_w is the activity coefficient of water, and γ_s ($s \neq w$) is the molal activity coefficient of the s^{th} species. As index labels, r and s'' are related in the present version of EQ3/6 by:

$$r = s'' - s_B \tag{D–84}$$

(recall that is the number of strict basis species). Eq (D–83) can be rearranged to give:

$$\log m_{s''} = \frac{\log K_r}{b_{s''r}} - \log \gamma_{s''} - \frac{b_{wr}}{b_{s''r}}(\log x_w + \log \lambda_w) - \frac{b_{s_B r}}{b_{s''r}} \log f_{O_2}$$

$$- \sum_{\substack{s'=1 \\ s' \neq w, s_B}}^{s_Q} \frac{b_{s'r}}{b_{s''r}}(\log m_{s'} + \log \gamma_{s'}) \tag{D–85}$$

Because EQ6 deals directly with the number of moles instead of the molality, it is useful for the purpose of deriving the Jacobian matrix to substitute eq (D–1) into eq (D–85) to obtain:

$$\log n_{s''} = \frac{\log K_r}{b_{s''r}} - \log \gamma_{s''} - \frac{b_{wr}}{b_{s''r}}(\log x_w + \log \lambda_w) - \frac{b_{s_B r}}{b_{s''r}} \log f_{O_2}$$

$$- b_{T,r} \log \Omega - \log n_w - \sum_{\substack{s'=1 \\ s' \neq w, s_B}}^{s_Q} \frac{b_{s'r}}{b_{s''r}}(\log n_{s'} + \log \gamma_{s'}) \qquad \text{(D–86)}$$

Here we have that:

$$b_{T,r} = b_{s''r} + \sum_{\substack{s'=1 \\ s' \neq w, s_B}}^{s_Q} b_{s'\phi} \qquad \text{(D–87)}$$

Note that x_w appearing in eq (D–86) depends on the $n_{s'}$: This can be first expressed by the following equation:

$$\log x_w = \log\left(\frac{\Omega}{\Omega + \sum_{s'=1}^{s_Q} m_{s'} + \sum_{s''=s_Q+1}^{s_T} m_{s''}} \right) \qquad \text{(D–88)}$$

We could proceed by substituting eq (D–1) into eq (D–88) and then carry out partial differentiation. Instead, we will take a different route and use results that were obtained for EQ3NR Appendix B.4. There is was necessary to find the following:

$$\mathrm{W}_{\mathrm{s'}} = \frac{\partial \log \mathrm{x_w}}{\partial \log \mathrm{m_{s'}}} \ , \qquad \mathrm{s'} \neq \mathrm{w},\ \mathrm{s_B} \qquad \text{(D–89)}$$

$$\mathrm{W}_{\mathrm{s_B}} = \frac{\partial \log \mathrm{x_w}}{\partial \log \mathrm{f_{O_2}}} \qquad \text{(D–90)}$$

It was shown that:

$$W_{s'} = \frac{-\frac{x_w}{\Omega}\left(m_{s'} - \sum_{s''=s_Q+1}^{s_T} \frac{b_{s'r}m_{s''}}{b_{s''r}}\right)}{\left(1 - \frac{x_w}{\Omega}\sum_{s''=s_Q+1}^{s_T} \frac{b_{wr}m_{s''}}{b_{s''r}}\right)}, \quad s' \neq w, s_B \tag{D–91}$$

$$W_{s_B} = \frac{-\frac{x_w}{\Omega}\left(-\sum_{s''=s_Q+1}^{s_T} \frac{b_{s'r}m_{s''}}{b_{s''r}}\right)}{\left(1 - \frac{x_w}{\Omega}\sum_{s''=s_Q+1}^{s_T} \frac{b_{wr}m_{s''}}{b_{s''r}}\right)} \tag{D–92}$$

In EQ3/6, the $\underline{W}$ vector corresponds to the **dlogxw** array.

For EQ6 the following related quantities are required:

$$\tilde{\mathrm{W}}_{\mathrm{s'}} = \frac{\partial \log \mathrm{x_w}}{\partial \log \mathrm{n_{s'}}}, \quad \mathrm{s'} \neq \mathrm{s_B} \tag{D–93}$$

$$\tilde{\mathrm{W}}_{\mathrm{s_B}} = \frac{\partial \log \mathrm{x_w}}{\partial \log \mathrm{f_{O_2}}} \tag{D–94}$$

Note that a $\tilde{W}$ parameter is required for water. Using the chain rule and eq (D–1), it can be shown that:

$$\frac{\partial \log \mathrm{x_w}}{\partial \log \mathrm{n_{s'}}} = \frac{\partial \log \mathrm{x_w}}{\partial \log \mathrm{m_{s'}}}, \quad \mathrm{s'} \neq \mathrm{w,\ s_B} \tag{D–95}$$

$$\frac{\partial \log \mathrm{x_w}}{\partial \log \mathrm{n_w}} = -\sum_{\substack{\mathrm{s'}=1 \\ \mathrm{s'} \neq \mathrm{w,\ s_B}}}^{\mathrm{s_Q}} \frac{\partial \log \mathrm{x_w}}{\partial \log \mathrm{n_{s'}}} \tag{D–96}$$

Therefore, the following results may be obtained:

$$\tilde{W}_w = - \sum_{\substack{s' = 1 \\ s' \neq w, s_B}}^{s_Q} W_{s'} \tag{D–97}$$

$$\tilde{W}_{s'} = W_{s'}, \; s' \neq w \tag{D–98}$$

The $\underline{\tilde{W}}$ vector is computed from $\underline{W}$ by EQ6. Because of the near-identity of these vectors (only the entry for water differs), the former is also represented by the **dlogxw** array.

Partial differentiation of the mass balance equation gives the elements of the corresponding row of the Jacobian matrix: Using mass balances for basis species, the Jacobian elements are given by:

$$J_{s'w} = 2.303 \; u_{s'w} n_w + \sum_{s'' = s_Q + 1}^{s_T} \tilde{H}_{s'r}(b_{T,r} - b_{wr}\tilde{W}_w) \tag{D–99}$$

$$J_{s's'''} = 2.303\left(u_{s's'''} m_{s'''} - \sum_{s'' = s_Q + 1}^{s_T} \tilde{H}_{s'r}(b_{s'''r} + b_{wr}\tilde{W}_{s'''}) \right), \quad s' \neq w, s_B \tag{D–100}$$

$$J_{s's_B} = 2.303 - \sum_{s'' = s_Q + 1}^{s_T} \tilde{H}_{s'r}(b_{s_B r} + b_{wr}\tilde{W}_{s_B}) \tag{D–101}$$

where:

$$\tilde{H}_{s'r} = \frac{n_{s''} u_{s's''}}{b_{s''r}}, \quad s' = 1, s_Q, \; s' \neq w \tag{D–102}$$

Note that $\tilde{H}_{s'r}$ is analogous to the $H_{s'r}$ used in EQ3NR. The difference is that the former is defined in terms of a number of moles variable instead of the corresponding molality. Note that $u_{s's''} = 1$ if $s' = s''$, otherwise $u_{s's''} = 0$.

Using mass balances for chemical elements and taking the ε^{th} chemical element to be associated with the s'^{th} basis species, the Jacobian elements are given by:

$$J_{s'w} = 2.303\ c_{\varepsilon w} n_w + \sum_{s'' = s_Q + 1}^{s_T} \tilde{H}_{s'r}(b_{T,r} - b_{wr}\tilde{W}_w) \qquad \text{(D–103)}$$

$$J_{s's_B} = 2.303\ - \sum_{s'' = s_Q + 1}^{s_T} \tilde{H}_{s'r}(b_{s_B r} + b_{wr}\tilde{W}_{s_B}) \qquad \text{(D–104)}$$

$$J_{s's''} = 2.303\left(c_{\varepsilon s'''}m_{s'''} - \sum_{s'' = s_Q + 1}^{s_T} \tilde{H}_{s'r}(b_{s'''r} + b_{wr}\tilde{W}_{s'''})\right), \qquad s' \neq w, s_B \qquad \text{(D–105)}$$

where:

$$\tilde{H}_{s'r} = \frac{n_{s''}c_{\varepsilon s''}}{b_{s''r}}, \quad s' = 1, s_Q, s' \neq w \qquad \text{(D–106)}$$

D.3.8.2. Electrical Balance

In the present version of EQ6, the fictive redox species O_2 (the $s_B{}^{th}$ species) is formally associated with the charge balance equation. We will use a subscript z to denote items having to do with this equation. The treatment is closely analogous to that for a mass balance equation. The electrical charge z_s takes the place of the stoichiometric equivalence factor $u_{s's}$ or the composition coefficient $c_{\varepsilon s}$. The fixed charge imbalance (Δ_z; see Section D.1.2.2) is included in the treatment. The governing equation can be written as:

$$\sum_{s' = 1}^{s_Q} z_{s'}n_{s'} + \sum_{s'' = s_Q + 1}^{s_T} z_{s''}n_{s''} = \Delta_z \qquad \text{(D–107)}$$

The corresponding residual functions are defined by:

$$\alpha_z = -\Delta_z + \sum_{s' = 1}^{s_Q} z_{s'}n_{s'} + \sum_{s'' = s_Q + 1}^{s_T} z_{s''}n_{s''} \qquad \text{(D–108)}$$

$$\beta_z = \frac{\alpha_z}{\sum_{s'=1}^{s_Q} z_{s'} n_{s'} + \sum_{s''=s_Q+1}^{s_T} z_{s''} n_{s''}}$$

The Jacobian elements are as follows:

$$J_{zw} = 2.303 \sum_{s''=s_Q+1}^{s_T} \tilde{H}_{zr}(b_{T,r} - b_{wr}\tilde{W}_w) \tag{D–110}$$

$$J_{zs_B} = 2.303 - \sum_{s''=s_Q+1}^{s_T} \tilde{H}_{zr}(b_{s_B r} + b_{wr}\tilde{W}_{s_B}) \tag{D–111}$$

$$J_{zs'''} = 2.303\left(z_{s'''}m_{s'''} - \sum_{s''=s_Q+1}^{s_T} \tilde{H}_{zr}(b_{s'''r} + b_{wr}\tilde{W}_{s'''}) \right), \qquad s' \neq w, s_B \tag{D–112}$$

where:

$$\tilde{H}_{zr} = \frac{n_{s''}z_{s''}}{b_{s''r}} \tag{D–113}$$

D.3.8.3. Mass Action For Pure Minerals

The governing mass action equation for pure minerals can be written as:

$$\log K_\phi = b_{w\phi}(\log x_w + \log \lambda_w) + b_{s_B\phi} \log f_{O_2} + \sum_{\substack{s'=1 \\ s'\neq w, s_B}}^{s_Q} b_{s'\phi}(\log m_s + \log \gamma_{s'}) \tag{D–114}$$

The corresponding residual functions are defined by:

$$\alpha_\phi = -\log K_\phi + b_{w\phi}(\log x_w + \log \lambda_w) + b_{s_B\phi} \log f_{O_2} + \sum_{\substack{s'=1 \\ s'\neq w, s_B}}^{s_Q} b_{s'\phi}(\log m_s + \log \gamma_{s'}) \tag{D–115}$$

$$\beta_\phi = \alpha_\phi \tag{D–116}$$

The residual function α_{ϕ} defined above is equivalent to the saturation index (SI). To facilitate the derivation of the corresponding elements of the Jacobian matrix, eq (D–83) can be written as:

$$\alpha_{\phi} = -\log K_{\phi} + b_{w\phi}(\log x_w + \log \lambda_w) + b_{s_B\phi} \log f_{O_2} + b_{T,\phi}(\log \Omega - \log n_w)$$

$$+ \sum_{\substack{s' = 1 \\ s' \neq w, s_B}}^{s_Q} b_{s'\sigma\psi}(\log n_{s'} + \log \gamma_{s'}) \tag{D–117}$$

where:

$$b_{T,\phi} = \sum_{\substack{s' = 1 \\ s' \neq w, s_B}}^{s_Q} b_{s'\phi} \tag{D–118}$$

Note that $b_{T,\phi}$ differs slightly in form from $b_{T.r}$: there is no term analogous to $b_{s''r}$. Similar forms can be written for components of solid solutions and for gas species ($b_{T,\sigma\psi}$ and $b_{T,g}$ being defined analogously to $b_{T,\phi}$, not $b_{T,r}$).

Each part of the equation for α_{ϕ} is now explicit in terms of the algebraic master variables, with one exception, the term in log x_w. The partial differentiation flowing through this term has been discussed previously. The corresponding Jacobian elements are as follows:

$$J_{\phi w} = -b_{T,\phi} + b_{w\phi}\widetilde{W}_w \tag{D–119}$$

$$J_{\phi s_B} = b_{s_B\phi} + b_{w\phi}\widetilde{W}_{s_B} \tag{D–120}$$

$$J_{\phi s'} = b_{s'\phi} + b_{w\phi}\widetilde{W}_{s'}, \;\; s \neq \dot{w}, s_B \tag{D–121}$$

D.3.8.4. Mass Action For End Member Components Of Solid Solutions

The governing mass action equation for the σ^{th} end member component of the ψ^{th} solid solution is:

$$\log K_{\sigma\psi} = b_{\sigma\psi\sigma\psi}(\log x_{\sigma\psi} + \log\lambda_{\sigma\psi}) + b_{w\sigma\psi}(\log x_w + \log\lambda_w)$$

$$+ b_{s_B\sigma\psi}\log f_{O_2} + \sum_{\substack{s' = 1 \\ s' \neq w,\, s_B}}^{s_Q} b_{s'\sigma\psi}(\log m_{s'} + \log\gamma_{s'}) \qquad \text{(D–122)}$$

The corresponding residual functions are defined by:

$$\alpha_{\sigma\psi} = -\log K_{\sigma\psi} + b_{\sigma\psi\sigma\psi}(\log x_{\sigma\psi} + \log\lambda_{\sigma\psi}) + b_{w\sigma\psi}(\log x_w + \log\lambda_w)$$

$$+ b_{s_B\sigma\psi}\log f_{O_2} + \sum_{\substack{s' = 1 \\ s' \neq w,\, s_B}}^{s_Q} b_{s'\sigma\psi}(\log m_{s'} + \log\gamma_{s'}) \qquad \text{(D–123)}$$

$$\beta_{\sigma\psi} = \alpha_{\sigma\psi} \qquad \text{(D–124)}$$

Here $x_{\sigma\psi}$ is the mole fraction of the end member component and $\lambda\sigma\psi$ is the corresponding activity coefficient. The residual function $\alpha\sigma\psi$ defined above is again equivalent to the saturation index (SI). For the purpose of deriving the corresponding elements of the Jacobian matrix, it is more convenient to write the equation for this residual function in the form:

$$\alpha_{\sigma\psi} = -\log K_{\sigma\psi} + b_{\sigma\psi\sigma\psi}(\log x_{\sigma\psi} + \log\lambda_{\sigma\psi}) + b_{w\sigma\psi}(\log x_w + \log\lambda_w)$$

$$+ b_{s_B\sigma\psi}\log f_{O_2} + b_{T,\,\sigma\psi}(\log\Omega - \log n_w) + \sum_{\substack{s' = 1 \\ s' \neq w,\, s_B}}^{s_Q} b_{s'\sigma\psi}(\log n_{s'} + \log\gamma_{s'}) \qquad \text{(D–125)}$$

where:

$$b_{T,\sigma\psi} = \sum_{\substack{s' = 1 \\ s' \neq w,\, s_B}}^{s_Q} b_{s'\sigma\psi} \tag{D–126}$$

Each part of the equation for $\alpha_{\sigma\psi}$ is now explicit in terms of the master iteration variables, with two exception, the terms in *log* x_w and *log* $x_{\sigma\psi}$. We have dealt previously with the flow of partial differentiation through the former, and will now deal with it through the latter.

The mole fraction of the end member component can be written as:

$$x_{\sigma\psi} = \frac{n_{\sigma\psi}}{\sum_{i=1}^{\sigma_{T,\psi}} n_{i\psi}} \tag{D–127}$$

where $\sigma_{T,\psi}$ is the number of end member components in the ψ^{th} solid solution. It follows that:

$$\frac{\partial \log x_{\sigma\psi}}{\partial \log n_{\sigma\psi}} = 1 - x_{\sigma\psi} \tag{D–128}$$

$$\frac{\partial \log x_{\sigma\psi}}{\partial \log n_{i\psi}} = -x_{i\psi} \tag{D–129}$$

The Jacobian elements are as follows:

$$J_{\sigma\psi w} = -b_{T,\sigma\psi} + b_{w\sigma\psi} W_w \tag{D–130}$$

$$J_{\sigma\psi s_B} = b_{s_B\sigma\psi} + b_{1\sigma\psi} W_{s_B} \tag{D–131}$$

$$J_{\sigma\psi s'} = b_{s'\sigma\psi} + b_{1\sigma\psi} W_{s'} \ , \qquad s \neq w,\ s_B \tag{D–132}$$

$$J_{\sigma\psi\sigma\psi} = b_{\sigma\psi\sigma\psi}\left((\Lambda_{\sigma\sigma\psi} + 1)(1 - x_{\sigma\psi}) - \sum_{\substack{i = 1 \\ i \neq \sigma}}^{\sigma_{T,\psi}} \Lambda_{\sigma i\psi} x_{i\psi} \right) \tag{D–134}$$

$$J_{\sigma\psi j\psi} = b_{\sigma\psi\sigma\psi}\left(- x_{j\psi} + \Lambda_{\sigma j\psi} 1 - x_{j\psi} - \sum_{\substack{i = 1 \\ i \neq j}}^{\sigma_{T,\psi}} \Lambda_{\sigma i\psi} x_{i\psi} \right)$$

where:

$$\Lambda_{\sigma i\psi} = \frac{\partial \log x_{\sigma\psi}}{\partial \log n_{i\psi}} \qquad \text{(D–135)}$$

These $\Lambda_{\sigma i\psi}$ parameters are specific to the solid solution thermodynamic models being employed. For examples, see Wolery (1979) where the notation uses "S" in place of Λ. The presence of such factors in the Jacobian matrix places the correction of activity coefficients of solid solution components in Newton-Raphson iteration; i.e., these activity coefficients are not corrected in the same manner as those of aqueous species. This is likely to be changed in a future version of EQ6.

D.3.9.FIND PHASES TO PRECIPITATE TO SATISFY PARTIAL EQUILIBRIUM

Each hybrid Newton-Raphson calculation requires that a phase assemblage be specified in advance. EQ6 finds the correct phase assemblage by making a sequence of calculations in which the phase assemblage is changed from calculation to calculation using algorithms previously described by Wolery (1979).

If such a Newton-Raphson calculation converges, EQ6 searches for cases of supersaturation, the extent of which is measured by the precipitation affinity ($A_{-,j}$, where j denotes a phase such as the ϕ^{th} pure mineral or the ψ^{th} solid solution. Phases which are suppressed or whose affinity functions are less than a specified tolerance (**tolsst**) are ignored. The code picks one supersaturated phase, adds it to the phase assemblage, and tries again. In principle, more than one such phase could be added simultaneously, but it is not generally profitable to do so. The number of supersaturations may far exceed the number of new phases that actually need to be precipitated. This is because precipitation of a phase reduces not only its own affinity, but also the affinities of other phases that are made of the same components.

EQ6 chooses the phase to be added as the one with the greatest scaled precipitation affinity. The scaled affinity is calculated by dividing the standard affinity by a scaling factor:

$$A_{-,j,\text{scaled}} = \frac{A_{-,j}}{b_{j,\text{scaled}}} \qquad \text{(D–136)}$$

The scaling factor is arbitrarily defined as the sum of the absolute values of the reaction coefficients:

$$b_{j,\,scale} = \sum_{s'=1}^{s_Q} b_{s'j} \tag{D–137}$$

Other definitions (i.e., the sum of the molecular weights of the species appearing in the reaction, the number of atoms appearing in the reaction) are possible and might work as well. The justification for scaling is that it improves the probability, as shown by experience, of choosing the right phases. About 80% or more of the choices are correct when there is a large number of supersaturations, say twenty or more, and the figure gets better when there are only a few. Scaling helps because it tends to remove a bias in the unscaled affinity that favors phases with large molecular formulas, such as clay minerals.

EQ6 also must occasionally delete phases from the phase assemblage. This is needed in part because the algorithm for choosing phases to precipitate is not 100% accurate. This capability is also needed in reaction path calculations, where a phase in the equilibrium system may become exhausted.

There are two paths in EQ6 leading to deletion from the phase assemblage. One occurs when the matrix is singular at the start of the iteration, indicating that the phase assemblage being tried probably violates the so-called “apparent” or “mineralogic” phase rule (Wolery, 1979). In this case, EQ6 looks for linear dependence in the rows of the Jacobian that describe heterogeneous equilibria (e.g., mineral solubilities). If such dependence is found, the code calculates conditional affinities for the phases involved, scales them, and deletes the phase with the most negative scaled conditional affinity.

A “conditional” affinity is the affinity of a phase which is uniquely determined by mutual solubility for a subset of other phases. Violation of the mineralogic phase rule means that such subsets exist. Mathematically, conditional affinities can always be defined for linearly dependent sets. For example, if a solution is in equilibrium with cristobalite (a form of SiO_2), this is sufficient to fix the affinity of quartz, its polymorph. See Wolery (1979) for further discussion of this topic.

In the second path, the iteration proceeds for one or more iterations and subsequently diverges. Divergence generally is due to one of the following conditions: (a) a mineral should be deleted, (b) the starting value for the oxygen fugacity is too far off the mark, or (c) the system is so ill-poised that the oxygen fugacity can not be defined with acceptable precision. Experience with the code shows that the use of the convergence enhancement features described earlier usually limits the possibilities to one of these.

EQ6 analyzes the situation to determine what action to take in response to divergence. It uses four independent algorithms to attempt to pick candidate phases for deletion. An object function (O_{nj}) is calculated for the n^{th} algorithm and the j^{th} phase in the assemblage. Each algorithm produces a candidate phase for removal from the phase assemblage. This is the phase with the most negative value of the object function for that algorithm. This value (denoted simply as O_n) must be negative, otherwise no candidate is produced. The overall choice for deletion, if any, is the candidate among the possible four with the most negative object function. EQ6 deletes this phase, unless it determines that the problem lies instead with the redox variable and takes other corrective action (see below).

One of the four deletion algorithms referred to above is much more important than the other three. Recall that one of the under-relaxation controls (**screw5**) limits the magnitude of the Newton-Raphson correction terms. This causes the calculation to diverge more slowly than it would otherwise. If a mineral does not belong in the phase assemblage, the strongest indication of this, after a mineralogic phase rule violation, is that its logarithmic mass variable plunges downward. The corresponding correction term commonly approaches a value of –**screw5** (the **screw5** default is 4.0, so this value is usually –4.0) before the matrix becomes numerically singular and the iteration crashes. The object function for the j^{th} phase for the first algorithm is:

$$O_{1,j} = \log n_{j,k} - \log n_{j,0} \tag{D–138}$$

where $\log n_{j,k}$ is the value of $\log n_j$ at the last successful (k^{th}) iteration and $\log n_{j,0}$ is the value prior to the start of iteration.

The other three algorithms were programmed into EQ6 prior to the full development of the convergence enhancement techniques that are now in the code. No study has been made of their significance to the operation of the code in its present state. Casual observation suggests that their current role may be largely vestigial. See Wolery (1979) for a description of these algorithms.

If the starting value of the oxygen fugacity variable is more than about five log units away from the correct value, experience has shown that convergence is not highly probable. This divergence tends to show up most strongly in the correction term for the log oxygen fugacity variable. Commonly, this correction term is at or near either ±**screw5** the last couple of iterations. This condition indicates that a better value for the starting value of this variable is required. The code goes into the redox scan mode in an attempt to find a good starting value if the magnitude of the last correction term for the log oxygen fugacity variable is **screw5**. This condition overrides any choice for phase deletion.

If the correction terms for the log oxygen fugacity variable oscillate in sign with magnitudes of the order of **screw5**, the system is probably ill-poised. If the residual functions have small magnitudes at this point, ill-poising is almost a certainty. The problem then is that the oxygen fugacity is so sensitive to the masses of the components in the system that the addition or subtraction of even one molecule of O_2 per kilogram of solvent can change the oxygen fugacity by orders of magnitude (or the corresponding Eh by hundreds of millivolts). An extremely ill-poised system causes trouble in the calculation because the machine floating point precision, even at 64 bits, is insufficient to handle the situation.

There are two methods of trying to deal with this condition. One would be to accept the results of the iteration process if the residual functions are all close to zero, even if the correction terms are not. The code presently does not do this. The second way, which the code does employ, is to assume that the ill-poising is limited to a very narrow range of reaction progress (encountered at a so-called redox jump), and make several attempts to step over it, using the redox scan feature to pick up the calculation on the other side of the jump.

D.3.10. THE REDOX SCAN FEATURE

The redox scan feature is used to try to generate a starting value for the log oxygen fugacity variable that will lead to convergence. As noted above, the starting value normally must be within about five log units to obtain convergence. A redox scan is a sequence of Newton-Raphson calculations with increasing or decreasing starting values for this variable. It terminates when either convergence is achieved or the whole range of the stability of water at the specified temperature and pressure has been covered.

The upper range of the stability field of water is taken to correspond to an oxygen fugacity of 1 bar. Thus:

$$\left(\log f_{O_2}\right)\left\{\text{upper limit}\right\} = 0 \tag{D–139}$$

The lower range is taken to correspond to a hydrogen fugacity of one bar and is determined by reference to the reaction:

$$2H_{2(g)} + O_{2(g)} = 2H_2O_{(l)} \tag{D–140}$$

The corresponding mass action equation can be written as:

$$\log f_{O_2} = -\log K_{H_{2(g)}} - 2\log f_{H_2} + 2\log a_w \tag{D–141}$$

The lower limit is calculated by assuming that $\log a_w = 0$. Thus,

$$\left(\log f_{O_2}\right)\left\{\text{lower limit}\right\} = -\log K_{H_{2(g)}} \tag{D–142}$$

At 25°C and 1.013 bar pressure, it has a value of about –83.1. It decreases in magnitude as the temperature is raised.

The above limits are what is in the present version of EQ6. They are calculated assuming a pressure limit of 1 bar and unit fugacity coefficients. For the purposes of establishing scan limits, assuming unit fugacity coefficients is not likely to be of much concern. However, the pressure limit should match the actual pressure (*P*), which may be considerably more than 1 bar. Thus, a better set of scanning limits would be:

$$\left(\log fO_2\right)\left\{\text{upper limit}\right\} = P \tag{D–143}$$

$$\left(\log fO_2\right)\{\text{upper limit}\} = -\log K_{H_{2(g)}} - 2\log P \qquad \text{(D–144)}$$

For P = 1000 bars, the upper limit of the scanning range would be increased by 3 log units, and the lower limit would be decreased by 6 log units. This is probably marginal in terms of affecting the outcome of the scanning process.

D.4. REFERENCES

Aagaard, P. and Helgeson, H.C. 1982. "Thermodynamic and kinetic constraints on reaction rates among minerals and aqueous solutions. I. Theoretical considerations." *American Journal of Science*, *282*, 237–285. New Haven, CT: Scholarly Publications. TIC: 225516.

Bourcier, W.L. 1985. *Improvements in the Solid Solution Modeling Capabilities of the EQ3/6 Geochemical Code*. UCID-20587. Livermore, CA: Lawrence Livermore National Laboratory. TIC: 209067.

Bourcier, W.L. 1986. "Improvements in the Solid Solution Modeling Capabilities of EQ3/6." *Proceedings of the Workshop on Geochemial Modeling CONF-8609134,* Editors Jackson, K.J. and Bourcier, W.L. Livermore, CA: Lawrence Livermore National Laboratory, 41–48. TIC: 202125.

Carroll-Webb, S.A. and Walther, J.V. 1988. "A Surface Complex Reaction Model for the pH-Dependence of Corundum and Kaolinite Dissolution Rates." *Geochimica et Cosmochimica Acta, 52,* 2609–2623. New York, NY: Pergamon Press. TIC: 225937.

Chou, L. and Wollast, R. 1984. "Study of the Weathering of Albite at Room Temperature and Pressure with a Fluidized Bed Reactor." *Geochimica et Cosmochimica Acta, 48,* 2205–2217. New York, NY: Pergamon Press. TIC: 238818.

Davies, C. W., 1962. *Ion Association.* Washington, D.C.: Butterworths. TIC: 244840.

Delany, J.M., Puigdomenech, I. and Wolery, T.J. 1986. *Precipitation Kinetics Option for the EQ6 Geochemical Reaction Path Code.* UCRL-53642. Livermore, California: Lawrence Livermore National Laboratory. TIC: 203008.

Delany, J.M. and Wolery, T.J. 1984. *Fixed Fugacity Option for the EQ6 Geochemical Reaction Path Code.* UCRL-53598. Livermore, California: Lawrence Livermore National Laboratory. TIC: 208799.

Harvie, C.E., Møller, N. and Weare J.H. 1984. "The Prediction of Mineral solubilities in Natural Waters: The Na-K-Mg-Ca-H-Cl-SO_4-OH-HCO_3-CO_3.$CO_2$$H_2O$ System to High Ionic Strengths at 25ºC." *Geochimica et Cosmochimica Acta*, *48*, 723–751. New York, NY: Pergamon Press. TIC: 239849.

Helgeson, H. C. 1968. "Evaluation of irreversible reactions in geochemical processes involving minerals and aqueous solutions– I. Thermodynamic relations." *Geochimica et Cosmochimica Acta, 32*, 853–877. New York, New York: Pergamon Press. TIC: 220011.

Helgeson, H. C., 1969. "Thermodynamics of hydrothermal systems at elevated temperatures and pressures." *American Journal of Science, 267*, 729–804. New Haven, CT: Scholarly Publications. TIC: 241817.

Helgeson, H.C., Brown, T.H., Nigrini, A. and Jones, T.A. 1970. "Calculation of Mass Transfer in Geochemical Processes Involving Aqueous Solutions." *Geochimica et Cosmochimica Acta, 34*, 569–592. New York, NY: Pergamon Press. TIC: 225570.

Helgeson, H.C. and Murphy, W.M. 1983. "Calculation of Mass Transfer Among Minerals and Aqueous Solutions as a Function of Time and Surface Area in Geochemical Processes. I. Computational Approach." *Mathematical Geology, 15,* 109–130. TIC 220014.

Helgeson, H.C., Murphy, W.M. and Aagaard, P. 1984. "Thermodynamic and Kinetic Constraints on Reaction Rates Among Minerals and Aqeous Solutions. II. Rate Constants, Effective Surface Area, and the Hydrolysis of Feldspar." *Geochimica et Cosmochimica Acta, 48*, 2405–2432. New York, NY: Pergamon Press. TIC: 220177.

Holdren, G.R., Jr. and Berner, R.A. 1979. "Mechanism of Feldspar Weathering. 1. Experimental Studies." *Geochimica et Cosmochimica Acta, 43*, 1161–1171. New York, NY: Pergamon Press. TIC: 239690.

Knauss, K.G. and Wolery, T.J. 1986. "Dependence of Albite Dissolution Kinetics on pH and Time at 25ºC and 70ºC." *Geochimica et Cosmochimica Acta, 50*, 2481–2497. New York, NY: Pergamon Press. TIC: 221756.

Knauss, K.G. and Wolery, T.J. 1988. "The Dissolution Kinetics of Quartz as a Function of pH and Time at 70ºC." *Geochimica et Cosmochimica Acta, 50*, 2481–2497. New York, NY: Pergamon Press. TIC: 223212.

Lasaga, A.C. 1981a. "Rate Laws of Chemical Reactions." Chapter 1, *Kinetics of Geochemical Processes.* Lasaga, A.C. and Kirkpatrick, R.J., eds. Reviews in Mineralogy Volume 8, 1–68. Washington, D.C.: Mineralogical Society of America. TIC: 240354.

Lasaga, A.C. 1981b. "Transition state theory." Chapter 4, *Kinetics of Geochemical Processes.* Lasaga, A.C. and Kirkpatrick, R.J., eds. Reviews in Mineralogy Volume 8, 135–169. Washington, D.C.: Mineralogical Society of America. TIC: 240353.

Murphy, W.M. and Helgeson, H.C. 1987. "Thermodynamic and Kinetic Constraints on Reaction Rates Among Minerals and Aqueous Solutions. III. Activated Complexes and the pH-Dependence of the Rates of Feldspar, Pyroxene, Wollastonite, and Olivine Hydrolysis." *Geochimica et Cosmochimica Acta, 51*, 3137–3153. New York, NY: Pergamon Press. TIC: 239172.

Murphy, W.M. and Helgeson, H.C. 1989. "Thermodynamic and Kinetic Constraints on Reaction Rates Among Minerals and Aqueous Solutions. IV. Retrieval of Rate Constants and Activation Parameters for the Hydrolysis of Pyroxene, Wollastonite, Olivine, Andalusite, Quartz, and Nepheline." *American Journal of Science, 289,* 17–101. New Haven, CT: Scholarly Publications. TIC: 240953.

Parkhurst, D.L., Thorstenson, D.C. and Plummer, L.N. 1980. "PHREEQE - A Computer Program for Geochemical Calculations." USGS WRI 80-96. Reston, VA: U.S. Geological Survey. TIC: 223154.

Petrovich, R. 1981a. "Kinetics of Dissolution of Mechanically Comminuted Rock-Forming Oxides and Silicates. I. Deformation and Dissolution of Quartz Under Laboratory Conditions." *Geochimica et Cosmochimica Acta, 45*, 1665–1674. New York, NY: Pergamon Press. TIC: 253651.

Petrovich, R. 1981b. "Kinetics of Dissolution of Mechanically Comminuted Rock-Forming Oxides and Silicates. II. Deformation and Dissolution of Oxides and Silicates in the Laboratory and at the Earth's Surface." *Geochimica et Cosmochimica Acta, 45*, 1675–1686. New York, NY: Pergamon Press. TIC: 253652.

Pitzer, K.S. 1973. "Thermodynamics of Electrolytes. I. Theoretical Basis and General Equations." *Journal of Physical Chemistry, 77,* 268–277. Washington, D.C.: American Chemical Society. TIC: 239503.

Pitzer, K. S. 1975. "Thermodynamics of Electrolytes. V. Effects of higher-order electrostatic terms." *Journal of Solution Chemistry, 4*, 249–265. New York, NY: Plenum Publishing. TIC: 248997.

Pitzer, K.S. 1979. "Theory: Ion Interaction Approach." Pytkowicz, R.M., editor. *Activity Coefficients in Electrolyte Solutions.* Boca Raton, FL: CRC Press. TIC: 239527.

Pitzer, K. S. 1987. "Thermodynamic model for aqueous solutions of liquid-like density." Editors: Carmichael, I. S. E. and Eugster, H. P. *Thermodynamic Modeling of Geological Materials: Minerals, Fluids and Melts, Reviews in Mineralogy, 17*, 97–142. Washington, D. C.: Mineralogical Society of America. TIC: 253653.

Rimstidt, J.D. and Barnes, H.L. 1980. "The Kinetics of Silica-Water Reactions." *Geochimica et Cosmochimica Acta, 44*, 1683–1699. New York, NY: Pergamon Press. TIC: 219975.

Talman, S.J. and Nesbitt, H.W. 1988. "Dissolution of Populations of Ultrafine Grains with Applications to Feldspars." *Geochimica et Cosmochimica Acta, 52*, 1467–1471. New York, NY: Pergamon Press. TIC: 253655.

Van Zeggeren, F. and Storey, S.H. 1970. *The Computation of Chemical Equilibrium Analysis.* London, England: Cambridge University Press. TIC: 252891.

Wolery, T.J. 1979. *Calculation of Chemical Equilibrium Between Aqueous Solution and Minerals: The EQ3/6 Software Package.* UCRL-52658. Livermore, California: Lawrence Livermore Laboratory. ACC: HQS.19880517.2586.

APPENDIX E

ODE INTEGRATION METHODS

INTENTIONALLY LEFT BLANK

CONTENTS

FIGURES

E.1. INTRODUCTION

The issue of ODE (Ordinary Differential Equation) integration is closely tied to the reaction-path tracing methodology used in EQ6. In the context of geochemical modeling, the ordinary differential equations in question are (or at least include) the rate equations for the advancement of chemical reactions. A reaction-path problem usually requires solving a combination of such ODEs and various Algebraic Equations (AEs) describing partial equilibrium. Early geochemical reaction path modeling (Helgeson 1968; Helgeson et al. 1970) did not incorporate true rate equations for chemical reactions, only specified relative rates of irreversible reactions (relative to a reaction progress variable ξ). Problems of this type, which are still utilized today, do not have a definite time frame, and only require that AEs be solved at successive points of reaction progress. Extension to deal with explicit kinetics (introducing a definite time frame and equations that are inherently ODEs) came later (Aagaard and Helgeson 1982; Helgeson et al. 1984).

ODE solution techniques have long influenced geochemical reaction-path modeling. The original numerical approach of Helgeson (1968) was in fact to differentiate the AEs with respect to the progress variable, thus transforming them into ODEs. Helgeson's approach was to develop matrix equations to obtain derivatives of a set of master variables with respect to reaction progress. One matrix equation yielded first-order derivatives. A second (Helgeson et al. 1970) provided the corresponding second-order derivatives. Moving from one point of reaction progress to another was accomplished by using these derivatives to evaluate a set of Taylor's series. As an ODE solver, this technique was not very efficient. In theory, it could have been improved by solving additional matrix equations to obtain higher-order derivatives. However, the process for deriving the appropriate matrix equations (the matrix is the same for any order; the right-hand-side vector changes) quickly becomes quite daunting in view of rapidly increasing complexity. A more logical approach, consistent with standard ODE integration methodology (Gear 1971a, 1971b), would have been to utilize information at preceding points to built backward finite-difference approximations to increase the order of the method.

The lack of an ability to solve AEs directly often meant that this code could not initialize a system so as to be able to apply the ODE integrator. A requirement was that the starting fluid could not be supersaturated with respect to any mineral of interest. The code could not precipitate such a mineral so as to satisfy such a condition. Setting up a problem that would run was therefore somewhat difficult. Another problem was "drift error," the development of cumulative integration errors in the governing algebraic equations. This manifested as errors in mass and charge balance. No rigorous method was available to correct these. When EQ6 was first developed, a decision was made to avoid these problems by treating the AEs as AEs. Since true rate expressions were not initially provided for, there were no ODEs to deal with. Later, however, consideration of such expressions was added (as an optional capability).

It was then necessary for EQ6 to deal with ODEs in addition to AEs. Basically, the AE solver was nested inside the ODE solver. As will be discussed below, the ODE solver technology used in EQ6 through version 7.2c was fairly primitive. After solving the AEs at a new point of reaction progress (or time), the rate equations were evaluated and compared with predicted values. If an accuracy criterion was not satisfied, the only option was to cut the step size. This could lead to situations in which the step size was reduced to a minimum value and held there repeatedly until the code stopped. The ODE solver was of the so-called predictor-corrector variety (see for example the method of Gear, 1971a, 1971b), but without the corrector part. Various types of ODE correctors exist. In discussing ODE solvers, the usual question about correctors is whether or not they will solve so-called "stiff" systems. Corrector types will be discussed later in this report. The main thing to note now is that EQ6 through version 7.2c lacked any corrector.

E.2. BACKWARD FINITE-DIFFERENCE METHODOLOGY AND USAGE

Even in the absence of true rate equations, the reaction path methodology used in EQ6 borrows heavily from ODE finite difference techniques to provide a local continuous representation of the path itself. In the context of a variable step size method such as that used in EQ6, this local representation is closely tied to the determination of the size of the next step. In the context of limiting error in stepping forward from some base point, this generally focuses on limiting the magnitude of the first neglected term in the representation, something that is normally carried as a "hidden" term (e.g., Gear 1971a, 1971b), though the last term included can also be used for this purpose.

Local representations in EQ6 are mostly (entirely, in versions through 7.2c) made by backward finite differences, as is illustrated in Figure E–1. Here a single function y(x) is depicted; in general, there are many possible such y and two typical x; reaction progress (ξ) and time (t). Discrete data at n + 1 successive points, including the most recent point (point "0") are used to build an nth order representation that can be extrapolated forward to a new point (point "–1"). In EQ6, the order automatically varies from 0 (approximate y_{-1} by y_0) to a maximum value that is ordinarily six. The methodology is one that accommodates variable step size and variable order. The value of the step size ($\Delta x = x_{-1} - x_0$) and the actual order used at each step is determined by choosing the order that has the largest corresponding step size, using the "hidden" term methodology for each order up to the lesser of the maximum order available or the maximum order allowed.

The maximum order available depends on the number of preceding points that can be used. This is zero, for example, when starting a reaction path calculation. It must also be zero at a point for which a represented quantity may be expected to have a discontinuity in its derivatives, for example at phase boundaries (where a new secondary phase appears, or an existing one disappears). When all calculations have been completed for the new point, the point numbering is shifted so that the new point becomes point 0. Then a new local representation is calculated with the new set of points (the new point 0 and the old set of points, usually less the point farthest back).

These backward difference representations have been used in EQ6 for three distinct purposes. One is to represent each of the members of the z vector of master variables used in solving the algebraic equations describing chemical equilibrium. Here the objective has been to provide accurate starting values for the Newton-Raphson based algebraic equation solver used in the code. The associated effect on step size control allows the step size to become large when the elements of this vector are changing slowly, but forces it to become smaller when one or more are changing rapidly.

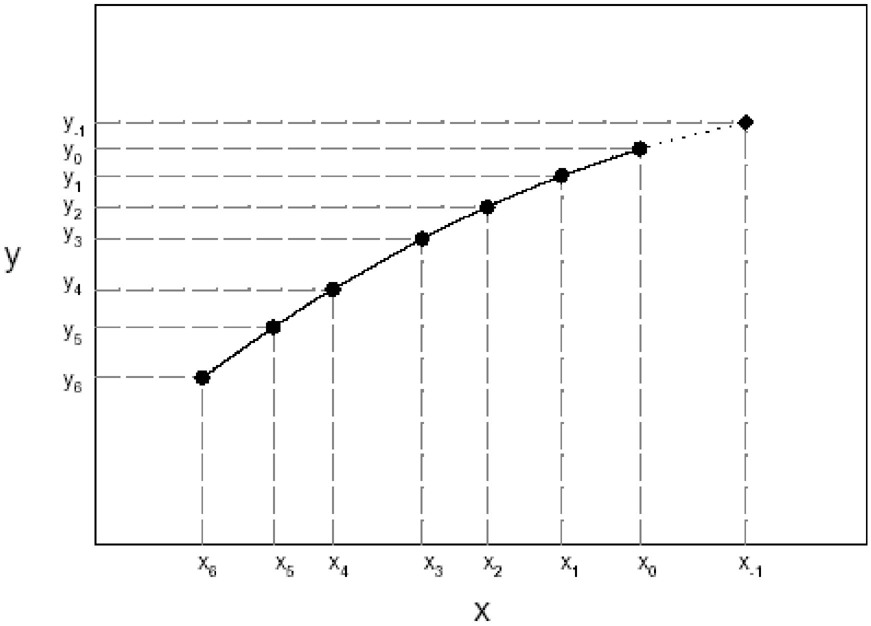

Figure E–1. A generalized scheme for backward finite differences. Here point "0" is the most recent point used in constructing a finite difference representation. Point "1" is the immediately preceding point, point "2" the point preceding that, and so forth. Point "–1" is a new point to which an extrapolation is made.

A second use of such backward finite difference representations has been to aid in the location of "events" of interest, for example a phase boundary at which a secondary mineral starts to precipitate or the complementary phase boundary at which such a phase disappears. In the case of the first example, the saturation index (SI) of a mineral would be negative (undersaturation) at point 0, but positive (supersaturation) and exceeding a tolerance at the initial choice of the new point. The new point is then adjusted to a lesser value (e.g., the step size is reduced) so that the calculated SI is close to zero (actually, between some minimum tolerance greater than zero and the maximum tolerance involved in initiating the step size reduction. The step size adjustment is made by a search that utilizes the local backward differences representation. In older versions of EQ6 (up through 7.2c), the representation of a quantity such as the SI of a mineral was obtained by expanding the representations of the elements of the z vector. Thus, there was no direct finite difference representation of such a quantity. In version 8.0 of EQ6, such quantities are represented by direct finite difference representations. This increases demands on memory, which is now cheap and abundant, and avoids the computing time overhead of expanding representations of the z vector.

The third use of backward differences representations in EQ6 has been to assist in the integration of rate equations. Thus, we are now talking about the use of such representations in ODE solvers. It will be helpful to note that EQ6 has always in operated in reaction progress-time mode, in which an overall reaction progress variable ξ is advanced, and the corresponding model time t is then calculated (a concept first described by Helgeson et al. 1984). Here representations of relative rates $d\xi_i/d\xi$ and the so-called inverse rate $dt/d\xi$ are integrated over the step size in overall reaction progress ($\Delta\xi$) to obtain, respectively, extents of reaction $\Delta\xi_i$ and the time step Δt. Here overall reaction progress is conceptually defined by $\xi = \Sigma_i\ \xi_i$, where the ξ_i are individual reaction progress variables. Differentiation with respect to t gives as $d\xi/dt = \Sigma_i\, d\xi_i/dt$. The $d\xi_i/dt$ are actual reaction rates defined by kinetic rate laws. The relative rates are obtained via $d\xi_i/d\xi = (d\xi_i/dt)/(d\xi/dt)$ and the inverse rate via $dt/d\xi = 1/(d\xi/dt)$. In time mode, time itself would be directly advanced. There, representations of the actual rates $d\xi_i/dt$ would be integrated over a step size in time (Δt) to obtain extents of reaction ($\Delta\xi_i$), and then the overall reaction progress step could be obtained (if desired) via $\Delta\xi = \Sigma_i\, \Delta\xi_i$. The treatment of backward finite-differences as part of an ODE solver will be reviewed here in order to provide an integrated discussion of the overall methods used in ODE integration. The notation used here is slightly different.

A backward finite-difference representation such as the one depicted in Figure E–1 can be represented mathematically in various equivalent forms. One (e.g., Carnahan et al. 1969, p. 9–26) is the so-called "finite difference function" or "Newton's divided-difference interpolating polynomial." The basic form truncated at nth order (corresponding to a fitting of n points) is:

$$y(x) = y_0 + \sum_{i=1}^{n} f_0^{(i)} \prod_{j=0}^{i-1} (x - x_j) \tag{E–1}$$

where the $f_0^{(i)}$ are so-called "finite differences" of order 1 through n. These are defined by:

$$f_0^{(1)} = \frac{(y_0 - y_1)}{(x_0 - x_1)} \tag{E–2}$$

$$f_0^{(i+1)} = \frac{(f_0^{(i)} - f_1^{(i)})}{(x_0 - x_{i+1})} \tag{E–3}$$

Here the superscript in parentheses denotes the order of a difference function and the subscript denotes the point to which the difference function corresponds. Note that eqs (2) and (3) are special cases, respectively, of:

$$f_j^{(1)} = \frac{(y_j - y_{j+1})}{(x_j - x_{j+1})} \tag{E–4}$$

$$f_j^{(i+1)} = \frac{(f_j^{(i)} - f_{j+1}^{(i)})}{(x_j - x_{i+j+1})} \tag{E–5}$$

Note the element of recursion: finite differences of order greater than one at the most recent point (x_0) can be calculated from the finite differences at the immediately preceding point (x_1). Equations (E–4) and (E–5) need never be programmed. For example, starting with two points 0 and 1, $f_0^{(1)}$ is calculated using eq (E–2). Advancing one point, these two points become, respectively, points 1 and 2, point 0 now being the new point. Then the finite difference that was $f_0^{(1)}$ becomes $f_1^{(1)}$. The new $f_0^{(1)}$ is calculated using eq (E–2), and $f_0^{(2)}$ is calculated using eq (E–3) and $f_0^{(1)}$ and $f_1^{(1)}$. Again advancing one point, $f_0^{(1)}$ and $f_0^{(2)}$ respectively become

$f_1^{(1)}$ and $f_1^{(2)}$. Application of eq (E–2) yields $f_0^{(1)}$, and two applications of eq (E–3) gives $f_0^{(2)}$ and $f_0^{(3)}$. In this way, the order (or the potential order) builds up to whatever maximum value is permitted.

The above finite difference representation is one possible basis for deriving quadrature formulas for the numerical integration of ODEs (e.g., Carnahan et al. 1969). An equivalent representation, the Lagrangian, is more popular for deriving quadrature formulas and is more likely to be encountered by the reader in surveying such formulas in the literature. Instead of recording the information at previous points in the form of finite differences, it does so in the form of the values of *y* at those points. The approach used in EQ6 is based on storing finite differences, because it is easy to translate these functions into the derivatives appearing in an equivalent truncated Taylor's series. For further information on the Lagrangian representation, see for example Carnahan, Luther, and Wilkes (1969, p. 27–34).

The finite difference representation of eq (E–1) is not used directly in EQ6. Rather, the code makes extensive use of the equivalent truncated Taylor's series:

$$y(x) = y_0 + \sum_{i=1}^{n} \frac{1}{i!} d_0^{(i)} (x - x_0)^i \tag{E–6}$$

where:

$$d_0^{(i)} = \left(\frac{d^i y}{dx^i}\right)_0 \tag{E–7}$$

The Taylor's series form is favored for several reasons. It is very familiar to anyone who has studied calculus. It is easily differentiated and integrated. Also, the derivatives that appear in it for point 0, or that can be calculated for any x by differentiating it, particularly the first-order derivatives, are sometimes of intrinsic interest. For example, if y describes the number of moles of a mineral in partial equilibrium with aqueous solution, and x is time, then dy/dx if positive is the rate of growth, else -dy/dx is the rate of dissolution.

The task here is to convert is the array of backward finite differences:

$$f_0 = \begin{bmatrix} f_0^{(1)} \\ f_0^{(2)} \\ f_0^{(3)} \\ \vdots \\ f_0^{(n)} \end{bmatrix} \tag{E–8}$$

into an equivalent array of derivatives:

$$d_0 = \begin{bmatrix} d_0^{(1)} \\ d_0^{(2)} \\ d_0^{(3)} \\ \vdots \\ d_0^{(n)} \end{bmatrix} \tag{E–9}$$

This means to accomplish this conversion can be derived by comparing eq (E–6) [truncated Taylor's series] with eq (E–1) [finite difference function]. The latter can be written as:

$$\begin{aligned} y(x) = y_0 &+ (x - x_0) f_0^{(1)} + (x - x_0)(x - x_1) f_0^{(2)} + (x - x_0)(x - x_1)(x - x_2) f_0^{(3)} \\ &+ (x - x_0)(x - x_1)(x - x_2)(x - x_3) f_0^{(4)} + \ldots \end{aligned} \tag{E–10}$$

This can be expanded to:

$$\begin{aligned} y(x) = y_0 &+ (x - x_0) f_0^{(1)} + (x - x_0)\left[(x - x_0) + \Delta x_{01}\right] f_0^{(2)} \\ &+ (x - x_0)\left[(x - x_0) + \Delta x_{01}\right]\left[(x - x_0) + \Delta x_{02}\right] f_0^{(3)} \\ &+ (x - x_0)\left[(x - x_0) + \Delta x_{01}\right]\left[(x - x_0) + \Delta x_{02}\right]\left[(x - x_0) + \Delta x_{03}\right] f_0^{(4)} + \cdots \end{aligned} \tag{E–11}$$

where:

$$\Delta x_{0i} = x_0 - x_i \tag{E–12}$$

Collecting terms in like powers of $x - x_0$ gives:

$$\begin{aligned} y(x) \;=\; & y_0 + (x - x_0)\left[f_0^{(1)} + \Delta x_{01} f_0^{(2)} + \Delta x_{01}\Delta x_{02} f_0^{(3)} + \Delta x_{01}\Delta x_{02}\Delta x_{03} f_0^{(4)} + \ldots\right] \\ & + (x - x_0)^2\left[f_0^{(2)} + (\Delta x_{01} + \Delta x_{02}) f_0^{(3)} + (\Delta x_{01}\Delta x_{02} + \Delta x_{01}\Delta x_{03} + \Delta x_{02}\Delta x_{03}) f_0^{(4)} + \ldots\right] \\ & + (x - x_0)^3\left[f_0^{(3)} + (\Delta x_{01} + \Delta x_{02} + \Delta x_{03}) f_0^{(4)} + \ldots\right] \\ & + (x - x_0)^4\left[f_0^{(4)} + \ldots\right] \\ & + \ldots \end{aligned} \tag{E–13}$$

The Taylor's series can be written in the analogous form:

$$\begin{aligned} y(x) \;=\; & y_0 + (x - x_0) d_0^{(1)} + (x - x_0)^2 \frac{d_0^{(2)}}{2!} \\ & + (x - x_0)^3 \frac{d_0^{(3)}}{3!} + (x - x_0)^4 \frac{d_0^{(4)}}{4!} + \ldots \end{aligned} \tag{E–14}$$

Comparison of like terms gives:

$$\begin{aligned}
d_0^{(1)} &= f_0^{(1)} + \Delta x_{01} f_0^{(2)} + \Delta x_{01}\Delta x_{02} f_0^{(3)} + \Delta x_{01}\Delta x_{02}\Delta x_{03} f_0^{(4)} + \ldots \\
\frac{d_0^{(2)}}{2!} &= f_0^{(2)} + (\Delta x_{01} + \Delta x_{02}) f_0^{(3)} + (\Delta x_{01}\Delta x_{02} + \Delta x_{01}\Delta x_{03} + \Delta x_{02}\Delta x_{03}) f_0^{(4)} + \ldots \\
\frac{d_0^{(3)}}{3!} &= f_0^{(3)} + (\Delta x_{01} + \Delta x_{02} + \Delta x_{03}) f_0^{(3)} + \ldots \\
\frac{d_0^{(4)}}{4!} &= f_0^{(4)} + \ldots \\
&\ldots
\end{aligned} \qquad \text{(E–15)}$$

The required transformation can be written as:

$$d_0 = FBf_0 \qquad \text{(E–16)}$$

where F is the “diagonal factorial matrix” defined by:

$$F = \begin{bmatrix} 1 & 0 & 0 & 0 & \ldots \\ 0 & 2! & 0 & 0 & \ldots \\ 0 & 0 & 3! & 0 & \ldots \\ 0 & 0 & 0 & 4! & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots \end{bmatrix}$$

and the matrix B is given by:

$$B = \begin{array}{ccccc} 1 & \Delta x_{01} & \Delta x_{01}\Delta x_{02} & \Delta x_{01}\Delta x_{02}\Delta x_{03} & \ldots \\ 0 & 1 & (\Delta x_{01} + \Delta x_{02}) & (\Delta x_{01}\Delta x_{02} + \Delta x_{01}\Delta x_{03} + \Delta x_{02}\Delta x_{03}) & \ldots \\ 0 & 0 & 1 & (\Delta x_{01} + \Delta x_{02} + \Delta x_{03}) & \ldots \\ 0 & 0 & 0 & 1 & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots \end{array} \qquad \text{(E–18)}$$

B is upper triangular. The diagonal consists of ones. The following recursions permit calculation of the off-diagonal elements in the upper triangle. For the first row, we have that:

$$B_{1,i} = B_{1,j-1}\Delta x_{0(j-1)} \qquad \text{(E–19)}$$

where j goes from 2 to n. The remaining elements may be calculated from:

$$B_{i,j} = B_{i-1,j-1} + B_{i,j-1}\Delta x_{0(j-1)} \qquad \text{(E–20)}$$

where i goes from 2 to n and j goes from i + 1 to n. These recursions permit B to be calculated for any desired order.

Eq (E–16) is appropriate for mathematical description, but a bit elaborate for computational work. The diagonal factorial matrix F need not appear in any programming. The matrix that is the product FB can be obtained from B by multiplying each row by the factorial of the row number. Alternatively, this matrix can be obtained by loading the diagonal with the factorials of the row numbers and then applying the following recursions:

$$(\mathrm{FB})_{1,\mathrm{i}} = (\mathrm{FB})_{1,\mathrm{j}-1}\Delta \mathrm{x}_{0(\mathrm{j}-1)} \tag{E–21}$$

where j goes from 2 to n and:

$$(\mathrm{FB})_{\mathrm{i},\mathrm{j}} = \mathrm{i}(\mathrm{FB})_{\mathrm{i}-1,\mathrm{j}-1} + (\mathrm{FB})_{\mathrm{i},\mathrm{j}-1}\Delta \mathrm{x}_{0(\mathrm{j}-1)} \tag{E–22}$$

where i goes from 2 to n and j goes from i + 1 to n. Another possibility is to use B without F to obtain the vector d_0 * whose elements are the $\frac{d_0^{(i)}}{i!}$: The relation for that is:

$$d_0^* = Bf_0 \tag{E–23}$$

The error in a truncated Taylor's series is usually estimated from the magnitude of the first neglected term in the series. Such an error expression is used to derive an algorithm for bounding the step size ($\Delta x = x - x_0$) so as to keep the estimated error within some predetermined limit. In many finite difference algorithms, therefore, an estimate is constructed of this extra term. In EQ6, however, the last term in the truncated series is used instead, which has the effect of using a more conservative bound. Therefore, letting m be the order of the term used to estimate the error (m = n in EQ6, m = n + 1 in the more traditional treatment):

$$Error[y] \cong \frac{1}{m!}\left(\frac{d^m y}{dx^m}\right)_0 (\Delta x)^m \tag{E–24}$$

where *Error* [*y*] is the error in *y*. Requiring Δx to satisfy the condition:

$$|Error[y]| \le Tolerance[y] \tag{E–25}$$

where *Tolerance* [*y*] is the required error tolerance on *y*, leads to the following limit on the step size:

$$\Delta x \leq \left(\frac{m!\,Tolerance\left[y\right]}{\left(\dfrac{d^m y}{dx^m}\right)_0} \right)^{\frac{1}{m}} \tag{E–26}$$

The above equation for determining the step size is used to test each of the possible orders. The order chosen is that which gives the largest step size satisfying the specified error tolerance.

The step size and order control discussed above is applied to basically two groups of things in EQ6. One main application is to the representations of the elements of the z vector of master algebraic variables. This is intended to guarantee that starting estimates for these at the new point are not too far off the mark, and has the benefit of assuring that step sizes will become small when the system is rapidly changing. Otherwise, step sizes may become large. The actual error in the predicted values at the new point may or may not be constrained by this mechanism to the level indicated by the tolerance value used. In EQ6, this actual error is controlled by the Newton-Raphson based AE solver, which acts as a corrector to the backward difference based predictor functions for the elements of the z vector.

The other main application of this step size and order control method is to the representations of the rate functions discussed previously. The actual error in the predicted values of the rate functions at the new point, and the actual error in the integrations of the predictor representations from the old point to the new point, may or may not be constrained to the level indicated by the value of the tolerance parameter used. The AE solver cannot correct these errors. The AE solver does correct chemical parameters that appear in the governing rate laws (assuming that the ODE solver has performed accurately). The rate values that are recalculated from those rate laws after AE correction can be compared with the corresponding predicted values obtained from the backward difference based representations (the predictor functions). The differences are a measure of the actual error in using the predictor functions to move from the old point to the new one.

To assure accurate ODE integration, such differences (usually in the form of a max norm) are required to satisfy an accuracy test. If this test is not satisfied, the usual response is to apply a corrector function. This is an attempt to iteratively improve the finite difference representation by using information at the new point. Usually the information at the point farthest back is dropped to keep the order constant. The corrector function is a mixed backward/forward finite difference function. This is the “corrector” of “predictor-corrector” ODE solvers (e.g., Gear 1971a, 1971b). Various approaches to corrector functions are possible, as will be discussed below. In EQ6 through version 7.2c, a corrector function was not implemented. Therefore, the only way to reduce ODE integration error was to drop the step size. This has been found in various cases to be less than efficient, and in others to lead to run failure as the step size would become stuck at the minimum value. The code is programmed to stop if the step size remains at the minimum value many times in a row.

It should be noted here that similar failure could occur even with the use of a corrector function. Because a corrector is associated with an interactive process, it is subject to all of the usual behaviors of such a process. In solving AEs, for example, simple back-substitution methods (which are usually so-called "first order" methods) may fail to converge (requiring step size reduction to avoid loss of accuracy), while more sophisticated methods like Newton-Raphson (which are "second order") may succeed. This carries over into methods for ODE correctors. It is more than an analogy, as ODE corrector schemes are essentially the usual methods for solving algebraic equations, only now applied to corrector representations. If a back-substitution corrector fails to converge and leads to a persistent state of minimal step size (usually leading to termination of the run without running to completion), the system of ODE's is often said to be "stiff" at that point. Stiffness is often said to be associated with (if not defined by) a situation in which one of the derivatives (in the present case, reaction rates) defined by an ODE is changing extremely rapidly compared with another. This may suggest that more than one ODE (here, more than one rate law) must be in the set being integrated in order to encounter that problem. In fact, a single ODE will do. Consider that a back-substitution method for solving a single algebraic equation may fail to converge. Furthermore, employing a second-order method (here a so-called "stiff system" corrector) does not guarantee success, much as failure to converge can also occur when applying a second-order method to solve an algebraic equation or set of such equations.

E.3. CORRECTOR FINITE-DIFFERENCE METHODOLOGY AND USAGE

The basic approach to a finite difference based corrector representation is illustrated in Figure E–2. The situation is much like that in Figure E–1. However, the new point (point –1) replaces the farthest back point (here point 6). The farthest back point could be retained with an increase in order, but this is uncommon in ODE integration methods. An important point is that the y-coordinate of the new point (y_{-1}) is not the value that was obtained by extrapolating the predictor function (that is depicted by the farthest right triangle in Figure E–2). In a back-substitution method, this is the value obtained by evaluating the associated ODE (here the value obtained by evaluating an equation for a reaction rate). Otherwise, it is the previous value (starting with the predictor value) corrected by some algorithm, such as the Newton-Raphson method. The level of ODE correction is selected in the EQ6 input file by **iopt(14)** (see Section 3.4.8)

In the context of the finite difference methodology previously applied in EQ6, the easiest way to handle things is to calculate a vector of finite differences of varying order at the new point, using the step-forward rotation discussed previously. The development in the previous section could then be applied to obtain a corresponding vector of derivatives at that point. However, these derivatives would not directly support a Taylor's series expansion about point 0. An expansion about point –1 could be integrated between point –1 and point 0, but would require the use of modified formulas for the integration. It is simpler just to derive a modified linear relation that gives the derivatives at point 0 from the finite differences at point –1.

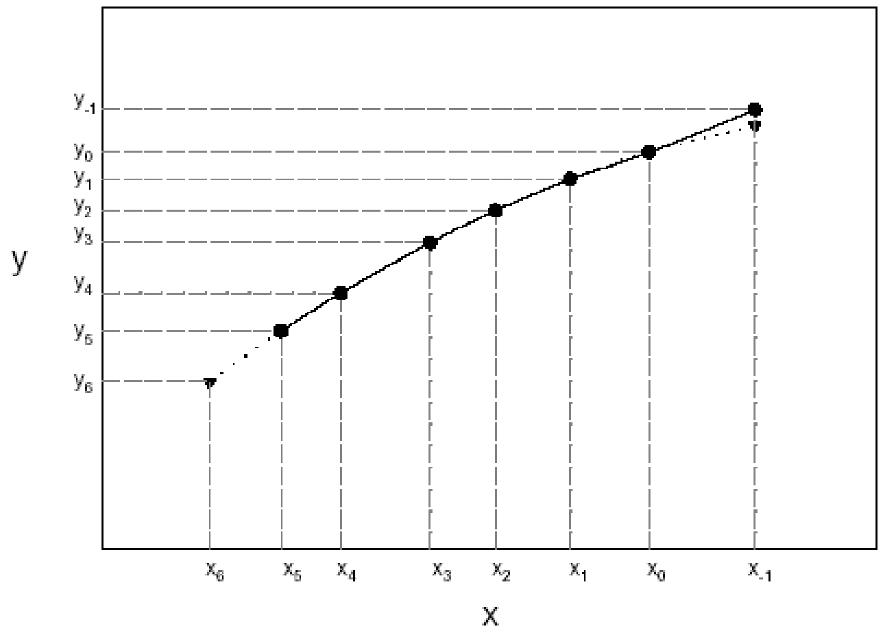

Figure E–2. A generalized scheme for a finite difference based corrector representation. Here point 0 is the point being stepped from, point 1 is the immediately preceding point, point 2 the point preceding that, and so forth. Point –1 is the new point (the point being stepped to). Point 6 (triangle, far left) represents the farthest back point used in the corresponding predictor representation (dotted line, mostly covered by the solid line representing the corrector). The farthest back point is dropped here so as to keep the order constant (point –1 replaces it). Note that y_{-1} is now slightly different than in the case for the predictor (Figure E–1). The new value is obtained by evaluating a rate at the new point, and is influenced by the action of the AE solver at this point. The point corresponding to the predicted value is show as the triangle farthest to the right.

The necessary finite differences are obtained from the following equations, which are just special cases of eqs (E–4) and (E–5):

$$f_{-1}^{(i+1)} = \frac{(f_{-1}^{(i)} - f_0^{(i)})}{(x_{-1} - x_i)} \tag{E–27}$$

$$f_{-1}^{(1)} = \frac{(y_{-1} - y_0)}{(x_{-1} - x_0)} \tag{E–28}$$

The analogue of eq (E–10) [finite difference function] is:

$$\begin{aligned} y(x) = {} & y_{-1} + (x - x_{-1})f_{-1}^{(1)} + (x - x_{-1})(x - x_0)f_{-1}^{(2)} + (x - x_{-1})(x - x_0)(x - x_1)f_{-1}^{(3)} \\ & + (x - x_{-1})(x - x_0)(x - x_1)(x - x_2)f_{-1}^{(4)} + \ldots \end{aligned} \tag{E–29}$$

This expansion about point –1 needs to be converted into one about point 0. This requires that the leading term y_{-1} be replaced by one of y_0, and that the $(x - x_{-1})$ factor in the following terms be replaced by one of $(x - x_0)$. We will work on the latter first. Note that:

$$(x - x_{-1}) = (x - x_0) + \Delta x_{0,-1} \tag{E–30}$$

where $\Delta x_{0,-1} = x_0 - x_{-1}$. This last relation is a special case of eq (E–12): $\Delta x_{0i} = x_0 - x_2$. Eq (E–29) can then be modified by substituting the right hand side of eq (E–30) for each factor of $(x - x_{-1})$. This, along with other applications of eq (E–12) yields:

$$\begin{aligned} y(x) = {} & y_{-1} + [(x - x_0) + \Delta x_{0,-1}]f_{-1}^{(1)} + [(x - x_0) + \Delta x_{0,-1}](x - x_0)f_{-1}^{(2)} \\ & + [(x - x_0) + \Delta x_{0,-1}](x - x_0)[(x - x_0) + \Delta x_{01}]f_{-1}^{(3)} \\ & + [(x - x_0) + \Delta x_{0,-1}](x - x_0)[(x - x_0) + \Delta x_{01}][(x - x_0) + \Delta x_{02}]f_{-1}^{(4)} + \ldots \end{aligned} \tag{E–31}$$

Multiplying out the factors on the right hand side and arranging to obtain terms in simple powers of $x - x_0$ gives:

$$\begin{aligned} y(x) = {} & y_{-1} + (x - x_0)f_{-1}^{(1)} + \Delta x_{0,-1}f_{-1}^{(1)} \\ & + (x - x_0)\Delta x_{0,-1}f_{-1}^{(2)} + (x - x_0)^2 f_{-1}^{(2)} \\ & + (x - x_0)\Delta x_{0,-1}\Delta x_{01}f_{-1}^{(3)} + (x - x_0)^2(\Delta x_{0,-1} + \Delta x_{01})f_{-1}^{(3)} + (x - x_0)^3 f_{-1}^{(3)} \\ & + (x - x_0)\Delta x_{0,-1}\Delta x_{01}\Delta x_{02}f_{-1}^{(4)} + (x - x_0)^2\Delta x_{0,-1}\Delta x_{01}f_{-1}^{(4)} \\ & + (x - x_0)^3(\Delta x_{0,-1} + \Delta x_{01} + \Delta x_{02})f_{-1}^{(4)} + (x - x_0)^4 f_{-1}^{(4)} + \ldots \end{aligned} \tag{E–32}$$

It remains to change the leading term. By definition, we have that:

$$y_{-1} = y_0 - \Delta x_{0,-1}f_{-1}^{(1)} \tag{E–33}$$

Substituting the right hand side of this equation for the leading term on the right hand side of eq (E–32) and gathering terms in common powers of $x - x_0$ leads to:

$$\begin{aligned} y(x) = {} & y_0 + (x - x_0)[f_{-1}^{(1)} + \Delta x_{0,-1}f_{-1}^{(2)} + \Delta x_{0,-1}\Delta x_{01}f_{-1}^{(3)} + \Delta x_{0,-1}\Delta x_{01}\Delta x_{02}f_{-1}^{(4)} + \ldots] \\ & + (x - x_0)^2[f_{-1}^{(2)} + (\Delta x_{0,-1} + \Delta x_{01})f_{-1}^{(3)} + (\Delta x_{0,-1}\Delta x_{01} + \Delta x_{0,-1}\Delta x_{02} + \Delta x_{01}\Delta x_{02})f_{-1}^{(4)} + \ldots] \\ & + (x - x_0)^3[f_{-1}^{(3)} + (\Delta x_{0,-1} + \Delta x_{01} + \Delta x_{02})f_{-1}^{(4)} + \ldots] \\ & + (x - x_0)^4[f_{-1}^{(4)} + \ldots] \\ & + \ldots \end{aligned} \tag{E–34}$$

The desired Taylor's series is again eq (E–14):

$$\begin{aligned} y(x) &= y_0 + (x - x_0) d_0^{(1)} + (x - x_0)^2 \frac{d_0^{(2)}}{2!} \\ &+ (x - x_0)^3 \frac{d_0^{(3)}}{3!} + (x - x_0)^4 \frac{d_0^{(4)}}{4!} + \ldots \end{aligned} \tag{E–35}$$

Comparison of like terms gives:

$$\begin{aligned} d_0^{(1)} &= f_{-1}^{(1)} + \Delta x_{0,-1} f_{-1}^{(2)} + \Delta x_{0,-1} \Delta x_{01} f_{-1}^{(3)} + \Delta x_{0,-1} \Delta x_{01} \Delta x_{02} f_{-1}^{(4)} + \ldots \\ \frac{d_0^{(2)}}{2!} &= f_{-1}^{(2)} + (\Delta x_{0,-1} + \Delta x_{01}) f_{-1}^{(3)} + (\Delta x_{0,-1} \Delta x_{01} + \Delta x_{0,-1} \Delta x_{02} + \Delta x_{01} \Delta x_{02}) f_{-1}^{(4)} + \ldots \\ \frac{d_0^{(3)}}{3!} &= f_{-1}^{(3)} + (\Delta x_{0,-1} + \Delta x_{01} + \Delta x_{02}) f_{-1}^{(3)} + \ldots \\ \frac{d_0^{(4)}}{4!} &= f_{-1}^{(4)} + \ldots \\ &\ldots \end{aligned} \tag{E–36}$$

Letting d_0 again be the array of derivatives of various order (at point 0) and f_{-1} the corresponding array of finite differences at point –1, the transformation can be written as:

$$d_0 = F C f_{-1} \tag{E–37}$$

where F is the "diagonal factorial matrix" defined previously and the matrix C is given by:

$$C = \begin{array}{ccccc} 1 & \Delta x_{0,-1} & \Delta x_{0,-1} \Delta x_{01} & \Delta x_{0,-1} \Delta x_{01} \Delta x_{02} & \ldots \\ 0 & 1 & (\Delta x_{0,-1} + \Delta x_{01}) & (\Delta x_{0,-1} \Delta x_{01} + \Delta x_{0,-1} \Delta x_{02} + \Delta x_{01} \Delta x_{02}) & \ldots \\ 0 & 0 & 1 & (\Delta x_{0,-1} + \Delta x_{01} + \Delta x_{02}) & \ldots \\ 0 & 0 & 0 & 1 & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots \end{array} \tag{E–38}$$

which is again an upper triangular matrix. The similarity to the matrix B is high. In fact, C can be obtained from the formula for B by substituting "0, –1" for "01", followed by "01" for "02", "02" for "03", and so forth. The following recursions permit calculation of the off-diagonal elements in the upper triangle. They are almost identical to those used for the predictor function.

The main difference is that it is first necessary to load $\Delta x_{0,-1}$ into C_{12}. Then for the rest of the first row, we have that:

$$C_{1,i} = C_{1,j-1}\Delta x_{0(j-1)} \tag{E–39}$$

(j goes from 3 to n). The remaining elements may be calculated from:

$$C_{i,j} = C_{i-1,j-1} + C_{i,j-1}\Delta x_{0(j-1)} \tag{E–40}$$

where i goes from 2 to n and j goes from i + 1 to n. These recursions permit C to be calculated for any desired order.

The matrix that is the product FC can be obtained from C by multiplying each row by the factorial of the row number. Alternatively, this matrix can be obtained by loading the diagonal with the factorials of the row numbers, loading $\Delta\chi_{0,-1}$ into C_{12}, and then applying the following recursions:

$$(FC)_{1,i} = (FC)_{1,j-1}\Delta x_{0(j-1)} \tag{E–41}$$

where j goes from 2 to n and:

$$(FC)_{i,j} = i(FC)_{i-1,j-1} + (FC)_{i,j-1}\Delta x_{0(j-1)} \tag{E–42}$$

where i goes from 2 to n and j goes from i + 1 to n. Another possibility is to use C without F to obtain $d_0{}^*$ via:

$$d_0^* = Cf_{-1} \tag{E–43}$$

Now that we have the corrector finite differences worked out, they can be applied to make an ODE corrector. It is only necessary to come up with a way to make new estimates of y_{-1}. The simplest method is back-substitution. But many other methods are possible. In fact, the finite difference representation has served to reduce an ODE problem to an AE problem. Any method for solving AEs can now be employed in the context of an ODE solver. Methods more sophisticated than back-substitution will be discussed in the following section.

In EQ6, the ODE corrector mechanism is only applied to the true ODEs, which is to say the rate functions. The local finite difference representations of other things could also be updated in like manner, for example for the elements of the z vector. However, there would probably be no real advantage to doing so. There might appear to be some advantage in such updating, for example in regard to the efficacy of searching for events. However, local finite difference representations are only used to provide approximations as part of a larger search mechanism, which does not assume that such representations (updated or not) are necessarily as accurate as the associated tolerances would imply. One could also consider the possibility of updating the step size and/or the order when the finite difference representations are updated by applying a corrector cycle. Again, however, there is little likelihood of any real advantage to doing so.

With back-substitution, the method developed here is equivalent to the predictor-corrector method of C.W. Gear (1971a, 1971b). The principle difference is that the finite difference representations here are cast in the form of Taylor's series. In addition to the advantages previously noted (easy differentiability and integrability, access to derivatives of intrinsic interest), this approach also has the advantage of avoiding having to deal with a large set of associated constants that fall out of using a Lagrangian finite difference representation. Problems sometimes occur because of difficulties associated with the precision of these constants (e.g., these constants are given for some floating point precision, and one then tries to use them for calculations involving a higher precision). The approach presented here requires instead only dealing with factorials, which are a lot simpler to deal with and scale to whatever floating point precision one might happen to be using.

E.4. IMPROVING ON A SIMPLE BACK-SUBSTITUTION ODE SOLVER

E.4.1. THE BASIC FRAMEWORK

In the simple back-substitution case, each y_{-1} is recalculated according to the relevant rate equation, the corresponding finite difference representation is updated, and the updated representation is integrated between point 0 and point –1. Let $r_{j,-1,i}$ is the value of the j^{th} rate function at point –1 that is consistent with the finite difference representation for the i^{th} corrector iteration. Consider i to be zero when the predictor function is employed. Because that function is not based on a value at the point in question, this is simply the predicted or extrapolated value. When a corrector function is employed, $r_{j,-1,i}$ is the value that was used in constructing that representation (e.g., this is an example of a y_{-1}). This value will necessarily be reproduced by evaluating the corrector at x_{-1}. Let $r^{(c)}_{j,-1,i}$ be the value that is calculated from the corresponding rate equation (the ODE). This value is obtained after the AE solver has been applied, at the end of the i^{th} corrector iteration.

A measure of the error in the ODE integration is the residual function defined by:

$$\alpha_{j,i} = r^{(c)}_{j,-1,i} - r_{j,-1,i} \qquad \text{(E–44)}$$

Here α for any i is a vector, each element corresponding to one of the rate functions appearing in the set of ODEs. α is also used in EQ3/6 to describe a similar residual function vector used in the AE solver. In fact, it generically denotes the residual function vector used in Newton-Raphson and similar AE solvers. The meaning of α in the present document should be obvious by context. However, if necessary to avoid confusion, $\alpha[r]$ may be used to denote the present residual function, and $\alpha[z]$ to denote that associated with the EQ6 AE solver.

In a simple back-substitution scheme, one simply applies:

$$r_{j,-1,i+1} = r^{(c)}_{j,-1,i} \tag{E–45}$$

A more general way to write the correction is:

$$r_{j,-1,i+1} = r_{j,-1,i} + \kappa\delta_{j,i} \tag{E–46}$$

where $\delta_{j,i}$ is a correction term (as generated by some method) and κ is a relaxation factor. For the case of back-substitution, we have that:

$$\delta_{j,-1} = r^{(c)}_{j,-1,i} - r_{j,-1,i} \tag{E–47}$$

Values of κ less than unity are often employed to damp oscillations. For simple back substitution, κ has a value of unity. Here δ is a correction term vector. By context, this should be recognized as δ[r], the vector applying to rates. It is the analogue of a similar vector used in the AE solver (δ[z]).

The AE solver in EQ6 (see Appendix D.3) is based on the Newton-Raphson method. It generates a correction vector from the classical Newton-Raphson relation $J\delta = -\alpha$. The Jacobian matrix J is obtained by partial differentiation of the elements of α with respect to the vector of unknowns. More specifically, the AE solver employs this relation in the form $J[z]\delta[z] = -\alpha[z]$, where J[z] is the Jacobian matrix for correcting the z vector (algebraic master variables). This matrix is obtained by partial differentiation of the α[z] vector with respect to each of the elements of the z vector. Correction in vector notation is given by $z_{i+1} = z_i + \kappa\delta[z]_i$ where i is iteration number (here for the AE solver).

The same general methodology can be applied here to correcting a vector of rates r, whose elements are the $r_{j,-1}$. The Newton-Raphson relation then takes the form:

$$J[r]]\delta[r] = -\alpha[r] \tag{E–48}$$

This matrix is obtained analogously, by partial differentiation of the α[r] vector with respect to each of the elements of the r vector. Correction in vector notation is given by:

$$r_{i+1} = r_i + \kappa\delta[r]_i \tag{E–49}$$

where i is iteration number (here for the ODE solver). A more powerful ODE solver can be created by using this approach. We have already shown how to obtain the vector α[r] and how to use the correction vector δ[r]. The only difficulty is to obtain the Jacobian matrix J[r]. Because we are solving a system of AEs inside the integration of a system of ODEs, the flow of partial differentiation needed to obtain J[r] flows through the AE solver. As we will see, the limiting J[z] (the matrix J[z] when the AE solver has converged) appears in the construction of J[r].

In the language of ODE solvers, correctors based on back-substitution are usually referred to as “simple” or “non-stiff” correctors, whereas more elegant correctors usually fall under the heading of “stiff system” correctors. “Stiffness” basically refers to a situation in which a simple corrector fails to converge. This generally results in cutting the step size in order to preserve accuracy, to the point that further progress in integrating the ODEs is effectively nil. Although there is often reference to “stiff systems of ODEs,” stiffness is a condition that is encountered somewhere along the path of integration. A simple corrector may work fine up to some point, say a model time of 50 years. If stiffness then ensues, greater model times are in a practical sense inaccessible. Either the run terminates before reaching the desired limit (say 1000 years), or the run grinds on for long periods of real time with little advancement of model time.

The definition of stiffness given above is a practical one. Stiffness is sometimes defined as a situation in which two rates have very different magnitudes, and one of those rates starts to change rapidly. This is basically a condition, not necessarily the only one, which can result in a practical case of stiffness. If this definition were rigorously correct, stiffness could not occur when integrating a single ODE. Note that we have mapped finite-difference-based ODE solver techniques to AE solver techniques (e.g., back substitution, Newton-Raphson method). All the usual problems associated with AE solver techniques therefore apply. There are many examples of back-substitution AE solvers failing to converge when applied to a system consisting of a single AE. The same problem can happen when a finite-difference-based ODE solver is applied to a system consisting of a single ODE.

Most “stiff system” ODE solvers are based in some fashion on the Newton-Raphson method. One of the main issues when so doing is whether to calculate a true Jacobian or to rely instead on some approximation of the Jacobian. A true Jacobian may be difficult to derive. A true Jacobian reflects the form of the underlying equations and may involve rather complex formulas. If the form of these equations changes, a new formula for the Jacobian must be derived and programmed. A true Jacobian may also be time-consuming to compute. An approximation generally involves numerical differentiation of the α vector. A major advantage is that it is then unnecessary to provide coding for the Jacobian that is specific to a given system of ODEs. It is only necessary to code a generic finite difference approximation. An approximate Jacobian may also be time consuming to compute, perhaps more so than the corresponding true Jacobian. Much depends on the actual structure of the true Jacobian.

Here we consider a true Jacobian approach. Eq (E–44) can be written in generalized vector form as:

$$\alpha = r^{(c)} - r \tag{E–50}$$

(here it is understood that α is $\alpha[r]$). Letting j denote one rate function and j' another, partial differentiation first yields:

$$J_{jj} = \frac{\partial \alpha_j}{\partial r_j} = \frac{\partial r_j^{(c)}}{\partial r_j} - 1 \tag{E–51}$$

$$J_{jj'} = \frac{\partial \alpha_j}{\partial r_{j'}} = \frac{\partial r_j^{(c)}}{\partial r_{j'}} \tag{E–52}$$

(here it is understood that J is J[r]). It remains to obtain the partial derivatives on the right hand sides of the above equations.

EQ6 does not use true reaction rates directly (see Appendix D.1.3.2). The code uses an overall reaction progress variable ξ as the primary variable (x) instead of time t. A relative rate conceptually defined by $v_{rel,j} = \partial \xi_j / \partial \xi$ corresponds to each true rate, which is conceptually defined as $v_j = \partial \xi_j / \partial t$. Here ξ_j is a reaction progress variable associated with the *j*th rate law. The actual expression used to calculate v_j is some rate equation written in terms of state variables, such as a surface area and one or more products of thermodynamic activities of selected chemical species. Overall reaction progress is related to the progress variables of the individual kinetically-controlled reactions by $\xi = \Sigma \mid \xi_j \mid$. Partial differentiation with respect to time gives an "overall" rate $\partial \xi / \partial t = \Sigma_j \mid \partial \xi_j \mid \partial t \mid$. Evaluating the true rates allows calculation of this overall rate, hence evaluation of the relative rates $[\partial \xi_j / \partial \xi = (\partial \xi_j / \partial t) / (\partial \xi / \partial t)]$. Integration of the j^{th} relative rate with respect to ξ gives the progress increment $\Delta\xi_j$ for that reaction (as would integration of the j^{th} true rate with respect to t). From such progress increments, one may calculate the total associated mass transfer (new mass balance totals). In addition, the treatment used in EQ6 requires the use of an inverse rate conceptually defined as $v_{1/t} = \partial t / \partial \xi$. This is evaluated as the inverse of the overall rate $[\partial t / \partial \xi = 1 / (\partial \xi / \partial t)]$. The inverse rate is used to calculate the time step Δt corresponding to the reaction progress step $\Delta\xi$. We will continue to use *v* here instead of *r* to symbolize "rate", consistent with the use of symbols in the prior EQ6 User's Manual (Wolery and Daveler 1992). The symbol *r* itself will be retained to refer to "r vector" associated with the ODE integrator.

If one were to use time as the primary variable, then it would be natural to use the true reaction rates v_j as the rate functions that comprise the r vector. One would then have that:

$$r = \begin{matrix} v_1 \\ v_2 \\ \cdots \\ v_{j_\kappa} \end{matrix} \tag{E–53}$$

where j_K is the number of rate laws and it is understood that the v_j here are the *presumed* values of the actual rates at point –1 (not necessarily values obtained by evaluating the corresponding rate laws). Using overall reaction progress instead as the primary variable, as is currently done in EQ6, the reaction rates in the r vector are replaced by the corresponding relative rates. The inverse rate must also appear, so the length of the r vector is increased by one. That vector can be written as:

$$r = \begin{matrix} v_{rel,1} \\ v_{rel,2} \\ \dots \\ v_{rel,j_K} \\ v_{1/t} \end{matrix} \tag{E–54}$$

where again it is understood that the inverse rate and the relative rates appearing in the vector on the right hand side are the *presumed* values at point –1.

Having defined the r vector in sufficient detail, it is now helpful to change the notation a bit so that "*v*" henceforth implies a rate of some kind calculated directly or indirectly from the actual rate laws. Thus:

$$v = \begin{matrix} v_1 \\ v_2 \\ \dots \\ v_{j_K} \end{matrix} \tag{E–55}$$

is the vector of *calculated* actual rates, and

$$v_x = \begin{matrix} v_{rel,1} \\ v_{rel,2} \\ \dots \\ v_{rel,j_K} \\ v_{1/t} \end{matrix} \tag{E–56}$$

is the vector whose elements include the *calculated* relative rates and the *calculated* inverse rate. Thus for a framework direct in time, eq (E–50) takes the form:

$$\alpha = v - r \tag{E–57}$$

For a framework direct in overall reaction progress, eq (E–50) takes the form:

$$\alpha = v_x - r \tag{E–58}$$

E.4.2. DERIVING THE TRUE JACOBIAN J[R], PART I: FRAMEWORK AND PIECES

We will now derive the Jacobian for the ODE integrator (J[r]) for the latter case (that for the former case will fall out of this). Basically, this involves a large-scale application of the chain rule of partial differentiation. The path of partial differentiation goes roughly as follows (it will be assumed here that temperature and pressure are known constants). The calculated relative rates and calculated inverse rate depend on the calculated actual rates. The actual rates depend on various state variables, such as surface areas and products of thermodynamic activities. The surface areas may be constants or depend on the number of moles of associated solid species. The thermodynamic activity products depend on the number of moles of associated species, generally aqueous species. These number of moles variables depend on a reduced set of number of moles variables for the so-called basis species. Those variables in turn depend on the limiting z vector (i.e., the z vector in the limit of convergence of the AE solver). The limiting z vector depends on the total number of moles quantities that constrain the algebraic mass balance relations. These total number of moles quantities depend on (a) the corresponding quantities at point 0 and (b) correction terms that are linearly related to the integration of the finite difference-based representations of the rate functions from point 0 to point –1. Those representations depend on the r vector (it is again emphasized that this contains presumed values of the contained rate quantities).

To assist in presenting the overall picture, we will utilize a special notation in which y and x are vectors and "[$\partial y / \partial x$]" is the matrix of partial derivatives $\partial y_i / \partial x_j$: As a first step, we have that:

$$J[r] = \frac{\partial \alpha}{\partial r} = \frac{\partial v_x}{\partial r} - I \tag{E–59}$$

where I is the identity matrix containing 1's on the diagonal and 0's elsewhere. We have then that:

$$\frac{\partial v_x}{\partial r} = \frac{\partial v_x}{\partial v} \frac{\partial v}{\partial r} \tag{E–60}$$

Note that the matrix on the left hand side is $j_K + 1$ by $j_K + 1$; on the right hand side, the first matrix $j_K + 1$ by j_K, while the second is j_K by $j_K + 1$. We will show that the first matrix on the right hand side is simple. It is the second one that is more complicated. This matrix would also be required if we were using time as the primary variable. In that case, the α vector is defined by eq (E–57) and eq (E–59) would be replaced by:

$$J[r] = \frac{\partial \alpha}{\partial r} = \frac{\partial v}{\partial r} - I \tag{E–61}$$

We will first address with $[\partial v_x / \partial v]$, then deal with $[\partial v / \partial r]$. Once v is calculated, the total rate can be obtained as:

$$v_{tot} = \sum_{j=1}^{j_K} v_j \tag{E–62}$$

From that, one can calculate the relative rates from:

$$v_{rel,j} = \frac{v_j}{v_{tot}} \tag{E–63}$$

and the inverse rate from:

$$v_{1/t} = \frac{1}{v_{tot}} \tag{E–64}$$

Partial differentiation of eqs (E–63) and (E–64) yields the following elements of $[\partial v_x / \partial v]$:

$$\frac{\partial v_{rel,j}}{\partial v_j} = v_{1/t}(1 - v_{rel,j}) \tag{E–65}$$

$$\frac{\partial v_{rel,j}}{\partial v_{j' \neq j}} = -v_{1/t} v_{rel,j} \tag{E–66}$$

$$\frac{\partial v_{1/t}}{\partial v_j} = -v_{1/t}^2 \tag{E–67}$$

This is a very simple matrix, containing only $2j_K + 1$ distinct elements out of $j_K(j_K + 1)$ total elements. It is not efficient to calculate and store the matrix as a programming object and then use matrix-matrix multiplication to obtain the product $[\partial v_x / \partial v][\partial v / \partial r]$. It is more efficient to code the equivalent special loops.

We move on now to $[\partial v / \partial r]$. The kinetic rate laws currently programmed into EQ6 are discussed in Appendix D.1.3.3. These all basically apply to mineral dissolution and precipitation kinetics. With adaptation, these forms could be applied to other types of reactions not currently treated by EQ6. The rate law most commonly used is a "one-term" formulation for net dissolution based on transition-state theory (TST). It can be written most simply for the present purposes as:

$$v_j = s_j k_{+,j} q_{+,j} \; 1 - \left(\frac{Q_{+,j}}{K_{+,j}}\right)^{m_{+,j}} \tag{E–68}$$

This will be used to springboard the derivation of J[r]. We will assume for the moment that all rate laws in the ODE set follow this form. We will subsequently address other rate laws, including the "multi-term" TST form, other "multi-term forms, the corresponding rate laws describing net precipitation, and ODE sets of mixed rate law forms. Here the "+" subscripts denotes the reaction proceeding in the forward direction (dissolution); s_j is the reactive surface area, $k_{+,j}$ the rate constant, $q_{+,j}$ the kinetic activity product, $Q_{+,j}$ the thermodynamic activity product, $K_{+,j}$ the corresponding equilibrium constant, and $m_{+,j}$ a parameter that modifies the response to changes in *Q/K* and has both theoretical and semi-empirical roots. At equilibrium, the *Q/K* ratio is unity and the net rate is zero. The kinetic activity product is typically either unity or the activity of the hydrogen ion raised to some power. The *m* exponent is most often taken as unity, or some value within an order of magnitude of unity. The thermodynamic activity product depends on how the dissolution reaction is written and is typically more complex.

In broad outline, the $[\partial v/\partial r]$ matrix is obtained as follows:

$$\frac{\partial v}{\partial r} = \left\{ \left(\frac{\partial v}{\partial q} \frac{\partial q}{\partial a} + \frac{\partial v}{\partial Q} \frac{\partial Q}{\partial a} \right) \frac{\partial a}{\partial z} \frac{\partial z}{\partial n_T} \frac{\partial n_T}{\partial \Delta\xi} + \frac{\partial v}{\partial s} \frac{\partial s}{\partial n} \frac{\partial n}{\partial \Delta\xi} \right\} \frac{\partial \Delta\xi}{\partial r} \qquad \text{(E–69)}$$

This is just a fancy application of the chain rule of differentiation. Every item in brackets is a matrix, and each variable appearing within brackets is a vector. Here q is the vector of kinetic activity products, Q is the vector of thermodynamic activity products, and s is the vector of reactive surface areas. These all have the same length as v and r. Here a is the vector of thermodynamic activities. Its length is subject to choice. It could be the vector of thermodynamic activities of all chemical species in the model, but we will presume that it is the smaller vector comprised of just the activities of the relevant basis species. It should be noted that by its definition, Q can contain activities of only basis species (species used as general building blocks in writing reactions). On the other hand, q could potentially be written to contain activities of non-basis species. However, those activities would then depend on the activities of basis species. Thus q can be effectively rewritten only in terms of basis species. So we can assume that a includes only basis species with no loss of generality. The a vector may include fugacities as well as thermodynamic activities. In the following derivations, it may be assumed for example that "a_{O_2}" is really f_{O_2} if the gas O_2 (real or fictive) is employed as a basis species. The a vector depends on the z vector, the vector of master variables used in the AE solver.

The z vector is obtained from the vector n_T of mass balance totals by the application of the AE solver. The length of z is the number of master algebraic variables; that of n_T is the number of components for which mass balances are defined, which is at least equal to the the number of master algebraic variables. Deriving the matrix $[\partial z/\partial n_T]$ in the absence of explicit equations to partially differentiate is at first a bit of a puzzle. We will later show that this matrix is the inverse of $J[z]$ (if z and n_T have the same length) or a part of that inverse (if they differ in length). The n_T vector is obtained from $\Delta\xi$, which is here the vector whose elements are the $\Delta\xi_j$. Note that we also use "$\Delta\xi$ " for the step size in overall reaction progress. This dual-usage must be resolved by analyzing the context.

The surface area vector *s* depends on surface area models. The surface areas could all be constants, in which case the $[\partial s/\partial n]$ matrix becomes zero and the term in which it appears vanishes. Otherwise, each surface area depends in some fashion on the number of moles of the associated solid. These numbers of moles variables comprise the *n* vector. The *n* vector in turn depends on the $\Delta\xi$ vector.

Finally, the dependence of the $\Delta\xi$ vector on the *r* vector goes back to the corresponding finite-difference-based corrector functions. This dependence follows from the calculation of each element of this vector by integration of the corrector function from point 0 to point –1.

We begin with $[\partial v/\partial q]$. Partial differentiation yields:

$$\frac{\partial v_j}{\partial q_{+,j}} = \frac{v_j}{q_{+,j}} \tag{E–70}$$

A rate depends on its own kinetic activity product, but not the kinetic activity product of another reaction. Therefore $\partial v_j / \partial q_{+,j^i} = 0$ for $j \neq j^i$ and $[\partial v/\partial q]$ is a diagonal matrix (a matrix whose off-diagonal elements are all zero). Programming-wise, only the diagonal of such a matrix need be calculated and stored. However, this is such a simple diagonal that even creating it as a programming object (a 1-D array) is unnecessary.

The matrix $[\partial q/\partial a]$ is more interesting. The kinetic activity product can be written in general form as:

$$\mathrm{q}_{+,\,\mathrm{j}} = \prod_{\mathrm{n}} \mathrm{a}_{\mathrm{n}}^{-\mathrm{N}_{+,\,\mathrm{nj}}} \tag{E–71}$$

where a_n is the thermodynamic activity of the n^{th} species and $-N_{+,nj}$ is an associated exponent. This can be factored as follows:

$$\mathrm{q}_{+,\,\mathrm{j}} = \left(\prod_{\mathrm{s}'} \mathrm{a}_{\mathrm{s}'}^{-\mathrm{N}_{+,\,\mathrm{s}'\mathrm{j}}}\right)\left(\prod_{\mathrm{s}''} \mathrm{a}_{\mathrm{s}''}^{-\mathrm{N}_{+,\,\mathrm{s}''\mathrm{j}}}\right) \tag{E–72}$$

where s' implies a basis species and s'' a non-basis species. The activity of a non-basis species is related to the activities of the basis species by the associated mass action equation:

$$\mathrm{K}_{\mathrm{s}''} = \mathrm{a}_{\mathrm{s}''}^{\mathrm{b}_{\mathrm{s}''\mathrm{s}''}} \prod_{\mathrm{s}'} \mathrm{a}_{\mathrm{s}'}^{\mathrm{b}_{\mathrm{s}'\mathrm{s}'}} \tag{E–73}$$

where $K_{s''}$ is the associated equilibrium constant, $b_{s''s''}$ is the reaction coefficient for species s'' in its own reaction (which almost always has a value of –1), and $b_{s's''}$ is the reaction coefficient for the basis species s' in the same reaction. These reaction coefficients are defined to be positive for products and negative for reactants. Partial differentiation of eq (E–72) yields:

$$\frac{\partial \mathrm{q}_{+,\,\mathrm{j}}}{\partial \mathrm{a}_{\mathrm{s}'}} = \mathrm{q}_{+,\,\mathrm{j}}\left(\frac{-\mathrm{N}_{+,\,\mathrm{s}'\mathrm{j}}}{\mathrm{a}_{\mathrm{s}'}} + \sum_{\mathrm{s}''}\frac{-\mathrm{N}_{+,\,\mathrm{s}''\mathrm{j}}}{\mathrm{a}_{\mathrm{s}''}}\frac{\partial \mathrm{a}_{\mathrm{s}''}}{\partial \mathrm{a}_{\mathrm{s}'}}\right) \tag{E–74}$$

Rearrangement of eq (E–73) gives:

$$a_{s''} = K_{s''}^{\frac{1}{b_{s''s''}}} \prod_{s'} a_{s'}^{\frac{b_{s's''}}{b_{s''s''}}} \tag{E–75}$$

Partial differentiation of this yields:

$$\frac{\partial a_{s''}}{\partial a_{s'}} = -\frac{b_{s's''}}{b_{s''s''}}\frac{a_{s''}}{a_{s'}} \tag{E–76}$$

Substitution into eq (E–75) gives the desired result:

$$\frac{\partial q_{+,j}}{\partial a_{s'}} = \frac{q_{+,j}}{a_{s'}}\left(-N_{+,s'j} + \sum_{s''} N_{+,s''j}\frac{b_{s's''}}{b_{s''s''}}\right) \tag{E–77}$$

This result fills up the $[\partial q / \partial a]$ matrix. In general, each element is unique. In practice, zero elements would likely be common owing to the simple forms commonly taken for kinetic activity products (e.g., $q_{+,j} = 1$ or $q_{+,j} = a_{H^+}^{-N_{+,H^+,j}}$). In fact, the whole matrix can be zero, in which case the dependency of $[\partial v / \partial r]$ on the q vector vanishes.

We now consider $\partial v / \partial Q$. Partial differentiation yields:

$$\frac{\partial v_j}{\partial Q_{+,j}} = \frac{-s_j k_{+,j} q_{+,j} m_{+,j}\left(\frac{Q_{+,j}}{K_{+,j}}\right)^{m_{+,j}}}{Q_{+,j}} \tag{E–78}$$

A rate depends on its own thermodynamic activity product, but not the thermodynamic activity product of another reaction. Therefore $\partial v / \partial Q$ is another diagonal matrix like its analog $\partial v / \partial q$.

A small complication is that the one-term TST rate law actually used in EQ6 is not eq (E–68) but the equivalent affinity-based form:

$$v_j = s_j k_{+,j} q_{+,j} \left(1 - \mathrm{e}^{\frac{-A}{\sigma_{+,j} RT}} \right) \tag{E–79}$$

Here $A_{+,j}$ is the thermodynamic affinity:

$$A_{+,j} = -RT \ln \left(\frac{Q_{+,j}}{K_{+,j}} \right) \tag{E–80}$$

and $\sigma_{+,j} = 1 / m_{+,j}$. In terms of $A_{+,j}$ and $Q_{+,j}$, eq (E–78) can be written as:

$$\frac{\partial v_j}{\partial Q_{+,j}} = \frac{- s_j k_{+,j} q_{+,j} \left(\mathrm{e}^{\frac{-A_{+,j}}{\sigma_{+,j} RT}} \right)}{\sigma_{+,j} Q_{+,j}} \tag{E–81}$$

It is also possible to write the right hand side in terms of $A_{+,j}$ and $K_{+,j}$; however, it is more useful to retain the form given above, as the $Q_{+,j}$ in the numerator will be neatly cancelled out later in taking the product $[\partial v / \partial Q] / [\partial Q / \partial a]$.

The matrix $[\partial Q / \partial a]$ is somewhat less interesting than its analog $[\partial q / \partial a]$. That is because thermodynamic activity products, at least as they are dealt with in EQ3/6, can include the thermodynamic activities of only basis species. A thermodynamic activity product appearing in a rate law can be written in general form as:

$$Q_{+,j} = \prod_{s'} a_{s'}^{b_{s'j}} \tag{E–82}$$

Partial differentiation yields:

$$\frac{\partial \mathrm{Q}_{+,\,\mathrm{j}}}{\partial \mathrm{a}_{\mathrm{s}'}} = \frac{\mathrm{Q}_{+,\,\mathrm{j}} \mathrm{b}_{\mathrm{s}'\mathrm{j}}}{\mathrm{a}_{\mathrm{s}'}} \tag{E–83}$$

Though relatively simple, $[\partial Q / \partial a]$ is a full matrix like its analog $[\partial q / \partial a]$.

The matrix $[\partial Q / \partial z]$ is one of the more interesting cases. The *a* vector contains the activities of only basis species, which in EQ3/6 must all be of type aqueous. In general, this includes the activity of solvent water (a_w) and (as noted earlier) the fugacity of oxygen (f_{O_2}). The *z* vector in EQ6 has the following form (from Appendix D.3.2, equation D–61):

$$z = \begin{array}{c} \log n_w \\ \cdot \\ \cdot \\ \log n_{s'}\ ,\ s' = 1, s_B - 1\ ,\ s \neq w \\ \cdot \\ \cdot \\ \log f_{O_2} \\ \cdot \\ \cdot \\ \log n_{s'}\ ,\ s' = s_B + 1, s_Q \\ \cdot \\ \cdot \\ \log n_{\phi}\ ,\ \phi = 1, \phi_T \\ \cdot \\ \cdot \\ \log n_{\sigma\psi}\ ,\ \sigma = 1, \sigma_{T,\psi}\ ,\ \psi = 1, \psi_T \end{array} \qquad \text{(E–84)}$$

It basically consists of log number of moles (log *n*) variables. The first group of these is for the active basis species, which number s_Q. This starts with $\log n_w$. A fictive O_2 species is included in the general case, and is described in Section 2.1.1 and denoted as the s_Bth active basis species. The *z* vector may include other elements, *log n* variables for pure solids (ϕ_T of these) and end-members of solid solutions ($\sigma_{T,\varphi}$ end-members of the Ψth solid solution, ψ_T solid solutions). The *a* vector has no dependence on these, at least not in the way that the *z* vector is defined in EQ6. In the following derivations, it is necessary to consider various cases in which solvent water (*w*) and fictive O_2 (O_2 or s_B) are distinguished from other basis species.

The activity of the kth basis species (excluding solvent water and fictive O_2) is related to the molality by:

$$a_k = \gamma_k m_k \qquad \text{(E–85)}$$

where γ_k is the molal activity coefficient. The molality m_k is defined as:

$$m_k = \frac{\Omega n_k}{n_w} \qquad \text{(E–86)}$$

where Ω is a constant equal to 1000 divided by the molecular weight of water ($\Omega \approx 5.51$), n_k is the number of moles of species *k*, and n_w is the number of moles of solvent water. Eqs (E–85) and (E–86) apply to any aqueous solute species, of course. Here they are only required for basis species. Treating the activity coefficient as a known constant (this assumption will be discussed later), partial differentiation (through a number of steps not explicitly shown here) yields:

$$\frac{\partial a_k}{\partial z_k} = 2.303 a_k \tag{E–87}$$

$$\frac{\partial a_k}{\partial z_w} = 2.303 a_k \tag{E–88}$$

Here "2.303" really stands for the natural logarithm of ten, which in calculations needs to be represented to the full precision of the type of floating point used in the computations (usually 64 bits). Note that a_k has no dependency on any of the other elements of the *z* vector.

In the case of the fictive O_2 species, "a_{O_2}" is f_{O_2}. By definition we have that $z_{O_2} = \log f_{O_2}$. Rearrangement followed by partial differentiation gives the result:

$$\frac{\partial f_{O_2}}{\partial z_{O_2}} = 2.303 f_{O_2} \tag{E–89}$$

Note that f_{O_2} has no dependency on any of the other elements of the *z* vector.

In the case of solvent water, the picture becomes more complicated.

$$a_w = \lambda_w x_w \tag{E–90}$$

where λ_w is the mole fraction activity coefficient of water and x_w the mole fraction. We again treat the activity coefficient as a known constant in the partial differentiation. The details in the partial differentiation will be skipped over here, as the equivalent necessary result is derived in detail elsewhere (in Appendix D.3.8.1). The pertinent result can be written as:

$$\frac{\partial a_w}{\partial z_{s'}} = 2.303 \widetilde{W}_{s'} a_w \tag{E–91}$$

where s' denotes any basis species. Again letting k denote any basis species other than water or fictive O_2, the $W_{s'}$ factors are:

$$\tilde{W}_k = \frac{-\frac{x_w}{\Omega}\left(m_k - \sum_{s''} \frac{b_{s's''} m_{s''}}{b_{s''s''}}\right)}{\left(1 - \frac{x_w}{\Omega} \sum_{s''} \frac{b_{ws''} m_{s''}}{b_{s''s''}}\right)} \tag{E–92}$$

$$\tilde{W}_{O_2} = \frac{-\frac{x_w}{\Omega}\left(- \sum_{s''} \frac{b_{s's''} m_{s''}}{b_{s''s''}}\right)}{\left(1 - \frac{x_w}{\Omega} \sum_{s''} \frac{b_{wr} m_{s''}}{b_{s''s''}}\right)} \tag{E–93}$$

$$\tilde{W}_w = -\tilde{W}_{O_2} - \sum_k \tilde{W}_k \tag{E–94}$$

Logically, $\tilde{W}_{s'\neq O_2} = \partial \log x_w / \partial \log n_{s'}$ and $\tilde{W}_{O_2} = \partial \log x_w / \partial \log f_{O_2}$. These factors are computed during the calculation of the $J[z]$ matrix and need only to be saved when the AE solver completes its task for subsequent calculation of $J[r]$.

We stated earlier that the matrix $[\partial z / \partial n_T]$ was either the inverse of $J[z]$ ($J^{-1}[z]$)or a part of that inverse. Thus, $J[z]$ or $J^{-1}[z]$) also needs to be saved and used in the calculation of $J[r]$. In general, whether the full matrix is required depends on whether the column size of the $[\partial a / \partial z]$ matches the length of the full z vector. It is always safe to use the full size case. However, it was pointed out earlier that the a vector as EQ6 is presently constructed consists only of the thermodynamic activities of aqueous species. These have no dependency on the pure minerals and solid solutions parts of the z vector. This would result in a corresponding set of columns consisting of nothing but zeros on the right hand size of $[\partial a / \partial z]$. Thus the corresponding set of rows in the bottom part of $J^{-1}[z]$ could contribute nothing to the product matrix $[\partial a / \partial z] J^{-1}[z]$.

The matrix $[\partial n_T / \partial \Delta\xi]$ is relatively simple to deal with. For a detailed discussion of the dependence of the n_T vector on the $\Delta\xi$ vector, see Appendix D.1.3.1. The basic result dependency is expressed by:

$$n_{T,s'} = n_{T,s',0} + \sum_{j=1}^{j_T} b_{s'j} \Delta\xi_j \tag{E–95}$$

where $n_{T,s'}$ is the total number of moles of the s'-th basis species, $n_{T,s',0}$ is the corresponding value at point 0, and the $b_{s'j}$ are reaction coefficients for the kinetically controlled reactions. Partial differentiation yields:

$$\frac{\partial n_{T,s'}}{\partial \Delta\xi_j} = b_{s'j} \tag{E–96}$$

In older versions of EQ3/6 (up to and including the version 7 series) the mass balance totals in the n_T vector are for chemical elements, not basis species. In that case, eq (E–95) is replaced by:

$$n_{T,\varepsilon} = n_{T,\varepsilon,0} + \sum_{s'=1}^{s_Q} c_{\varepsilon s'} \sum_{j=1}^{j_T} b_{s'j}\Delta\xi_j \tag{E–97}$$

where ε denotes a chemical element and $c_{\varepsilon s'}$ is a compositional coefficient, the number of moles of element ε in one mole of basis species s'. Partial differentiation then yields:

$$\frac{\partial n_{T,\varepsilon}}{\partial \Delta\xi_j} = \sum_{s'=1}^{s_Q} c_{\varepsilon s'} b_{s'j} \tag{E–98}$$

The matrix $[\partial v / \partial s]$ is another simple diagonal matrix. Partial differentiation of eq (E–68) yields:

$$\frac{\partial v_j}{\partial s_j} = \frac{v_j}{s_j} \tag{E–99}$$

More interesting is the companion matrix $[\partial s / \partial n]$. This must be a diagonal matrix at most, because s_j can depend only on n_j. One possibility is that s_j is treated as a constant. Then one has that:

$$\frac{\partial s_j}{\partial n_j} = 0 \tag{E–100}$$

Two other models are of interest. The first has the specific surface area (s_j, cm^2/g) held constant. Then one has that:

$$s_j = \hat{s}_j M_j n_j \tag{E–101}$$

where M_j is the molecular weight. Partial differentiation then yields:

$$\frac{\partial s_j}{\partial n_j} = \hat{s}_j M_j \tag{E–102}$$

The second is for particle growth with a fixed number of particles. For this, one has that:

$$s_j = s_{j,0}\left(\frac{h_j}{n_{j,0}}\right)^{2/3} \tag{E–103}$$

where $s_{j,0}$ is the surface area and $n_{j,0}$ the number of moles at point 0. For this, partial differentiation yields:

$$\frac{\partial s_j}{\partial n_j} = \frac{2}{3}\frac{s_j}{n_j} \tag{E–104}$$

The matrix $[\partial n / \partial \Delta\xi]$ is relatively straightforward. The key relation is:

$$n_j = n_{j,0} + \hat{b}_j \Delta\xi_j \tag{E–105}$$

where $n_{j,o}$ is the number of moles at point 0 and $\hat{b}_j$ is the coefficient for the j^{th} reactant in its own reaction. Usually this has a value of –1 (it is negative because it is consumed in that reaction). Partial differentiation yields:

$$\frac{\partial n_j}{\partial \Delta\xi_j} = \hat{b}_j \tag{E–106}$$

Thus $[\partial n / \partial \Delta\xi]$ is yet another diagonal matrix.

We now address the matrix $[\partial \Delta\xi / \partial r]$. Integration of the Taylor's series-formatted corrector function for the j^{th} relative rate gives:

$$\Delta\xi_j = r_{j,0}\Delta x + \sum_{n=1}^{\hat{n}} \frac{d_0^{(n)}[r_j]}{(n+1)!}\Delta x^{n+1} \tag{E–107}$$

Here the "$\lfloor r_j \rfloor$" is a modifier to indicate what is represented by the indicated derivatives $d_0^{(n)}$, r_j is the j^{th} element of the r vector (defined of course at point –1), while $r_{j,0}$ is the corresponding value at point 0, and $\hat{n}$ is the order of the series. We use x here as the stepping variable, so as to be general; x could be overall reaction progress (ξ) or time (t). The dependence on r_j goes entirely through the derivatives, which depend on the corresponding finite differences, which finally depend on r_j. Applying the chain rule, we have that:

$$\frac{\partial \Delta\xi_j}{\partial r_j} = \sum_{n=1}^{\hat{n}} \frac{\partial \Delta\xi_j}{\partial d_0^{(n)}[r_j]} \sum_{i=1}^{\hat{n}} \frac{\partial d_0^{(n)}[r_j]}{\partial f_{-1}^{(i)}[r_j]} \frac{\partial f_{-1}^{(i)}[r_j]}{\partial r_j} \tag{E–108}$$

Partial differentiation of eq (E–107) gives:

$$\frac{\partial \Delta\xi_j}{\partial d_0^{(n)}[r_j]} = \frac{\Delta x^{n+1}}{(n+1)!} \tag{E–109}$$

Note that this result is independent of j and it depends only on the recent stepping history. It is convenient to put the partial derivatives in eq (E–109) into the following special vector:

$$g = \begin{matrix} \dfrac{\Delta x^2}{2!} \\ \dfrac{\Delta x^3}{3!} \\ \dfrac{\Delta x^4}{4!} \\ \dots \end{matrix} \tag{E–110}$$

The dependency of the derivatives on the corresponding finite differences was given by eq (E–37), which here takes the following form:

$$d_0[r_j] = FCf_{-1}[r_j] \tag{E–111}$$

Partial differentiation yields:

$$\left[\frac{\partial d_0[r_j]}{\partial f_{-1}[r_j]}\right] = FC \tag{E–112}$$

This result also has no dependency on j, depending only on the recent stepping history.

The dependency of the f_{-1} vector applied to the j^{th} reaction rate is expressed by the following forms of eqs (E–27) and (E–28):

$$f_{-1}^{(1)}[r_j] = \frac{\left(r_j - r_{j,0}\right)}{\left(x_{-1} - x_0\right)} \tag{E–113}$$

$$f_{-1}^{(1+1)}[r_j] = \frac{\left(f_{-1}^{(i)}\left\lfloor r_j \right\rfloor - f_0^{(i)}\left\lfloor r_j \right\rfloor\right)}{\left(x_{-1} - x_i\right)} \tag{E–114}$$

Here the "$\left\lfloor r_j \right\rfloor$" is a modifier to indicate what is represented by the finite differences indicated; $\left\lfloor r_j \right\rfloor$ is the j^{th} element of the r vector defined for point –1, while $\left\lfloor r_{j,0} \right\rfloor$ is the corresponding value at point 0. Partial differentiation gives:

$$\frac{\partial f_{-1}^{(i)}\left\lfloor r_j \right\rfloor}{\partial r_j} = \prod_{m=0}^{i-1} \frac{1}{\left(x_{-1} - x_m\right)} \tag{E–115}$$

Note that this result too is independent of j and it depends only on the recent stepping history. This result can be written in vector notation as:

$$\frac{\partial f_{-1}[r_j]}{\partial r_j} = h \tag{E–116}$$

where in the notation for stepping history used earlier:

$$\mathrm{h} \;=\; \begin{matrix} \dfrac{1}{\Delta \mathrm{x}_{-1,\,0}} \\ \dfrac{1}{\Delta \mathrm{x}_{-1,\,0}\Delta \mathrm{x}_{-1,\,1}} \\ \dfrac{1}{\Delta \mathrm{x}_{-1,\,0}\Delta \mathrm{x}_{-1,\,1}\Delta \mathrm{x}_{-1,\,2}} \\ \cdots \end{matrix} \tag{E–117}$$

Eq (E–108) can now be written in vector-matrix notation as:

$$\frac{\partial \Delta \xi_j}{\partial r_j} = g \bullet FCh = w \tag{E–118}$$

where w is a scalar independent of j. Thus, one can write that:

$$\left[\frac{\partial \Delta \xi}{\partial r}\right] = wI \tag{E–119}$$

where I is again the identity matrix.

This concludes the base-case run-through for the computation of the Jacobian matrix $J[r]$ required for the stiff system ODE solver. This base case was for each rate equation being of the one-term TST form applied to the forward direction (dissolution of a mineral in EQ6):

$$v_j = s_j k_{+,j} q_{+,j} \left[1 - \left(\frac{Q_{+,j}}{K_{+,j}}\right)^{m_{+,j}}\right] \tag{E–120}$$

In theory, the dependence on $Q_{+,j} / K_{+,j}$ allows this form to apply also to reaction in the backward direction. If $Q_{+,j} / K_{+,j}$ is greater than one, then the reaction rate v_j is negative, implying that the reaction is moving backwards (precipitation of a mineral in EQ6). EQ6 allows the user to use a forward direction rate law in this manner. Whether that is a good idea in practice is another thing altogether. Typically, a rate law of the form of eq (E–120) is obtained by empirically fitting it to some dissolution rate measurements. This activity does not prove that the reaction mechanism implied by the form of the equation is the actual reaction mechanism. There is thus no guarantee that the resulting equation will accurately predict precipitation rates.

Backward direction rates are more often specified directly. In effect, one specifies one rate law for dissolution and an independent one for precipitation. The one-term TST form is given for the backward direction is given by:

$$-v_j = s_j k_{-,j} q_{-,j} \left[1 - \left(\frac{Q_{-,j}}{K_{-,j}}\right)^{m_{-,j}}\right] \tag{E–121}$$

where $-v_j$ is the rate in the backward direction and the "–" subscript denotes the backward direction. The corresponding affinity-based form is:

$$-v_j = s_j k_{-,j} q_{-,j} \left(1 - \mathrm{e}^{\frac{-A_{-,j}}{\sigma_{-,j} RT}}\right) \tag{E–122}$$

Simple thermodynamic considerations require that $Q_{-,j} = 1 / Q_{+,j}$, $K_{-,j=1K_{+,j}}$, and $A_{-,j} = -A_{+,j}$. Kinetic theory furthermore requires that $q_{-,j} / q_{+,j} = Q_{+,j}^{m_{+,j}}$, $k_{+j} / k_{-,j} = K_{+,j}^{m_{+,j}}$, and $m_{-,j} = m_{+,j}$ (and hence $\sigma_{-,j} = \sigma_{+,j}$). If these latter relations all hold, then eq (E–121) for example is equivalent to eq (E–120). In practice, this is not always assumed to be the case, and a backward direction rate law may be specified in which $q_{-,j}$, $k_{-,j}$, and $m_{-,j}$ (or $\sigma_{-,j}$) are defined independently of these theoretical relations.

If the j^{th} rate law is written in terms of the backward form, how is the computation of $J[r]$ affected? For the most part, the answer is that the corresponding "+" quantities are replaced by their "–" counterparts. The v vector and related vectors may include mixed elements of the "+" and "–" type. For example, consider $[\partial v / \partial q]$. For cases in which v_j is specified by the backward form, partial differentiation yields:

$$\frac{\partial v_j}{\partial q_{-,j}} = \frac{v_j}{q_{-,j}} \tag{E–123}$$

This is completely analogous to eq (E–70), the corresponding "+" case. Here the "–" subscript just replaces the "+" one. In general, that is the rule that applies to all the other partial derivatives involving variables found in the rate laws. There is one exception to that involving the matrix $[\partial v / \partial Q]$. For the former, one obtains for the case of the j^{th} rate law having the backward form:

$$\frac{\partial v_j}{\partial Q_{-,j}} = \frac{s_j k_{-,j} q_{-,j} m_{-,j} \left(\frac{Q_{-,j}}{K_{-,j}} \right)^{m_{-,j}}}{Q_{-,j}} \tag{E–124}$$

This differs from its analog, eq (E–78), in that in addition to "+" labels being replaced by "–", there is a sign change. In terms of affinity, one obtains that:

$$\frac{\partial v_j}{\partial Q_{-,j}} = \frac{s_j k_{-,j} q_{-,j} \left(e^{\frac{-A_{-,j}}{\sigma_{-,j} RT}} \right)}{\sigma_{-,j} Q_{-,j}} \tag{E–125}$$

This has the same form as its analog, eq (E–81), but again with a sign change.

Another sign change (canceling out that found in the $\partial v_j / \partial Q_{-,j}$) comes about in the partial differentiation of $Q_{-,j}$ with respect to thermodynamic activities. Recall that:

$$Q_{+,j} = \prod_{s'} a_{s'}^{b_{s'j}} \tag{E–126}$$

Hence one has that:

$$Q_{-,j} = \prod_{s'} a_{s'}^{-b_{s'j}} \tag{E–127}$$

Partial differentiation yields:

$$\frac{\partial Q_{-,j}}{\partial a_{s'}} = \frac{Q_{-,j} b_{s'j}}{a_{s'}} \tag{E–128}$$

which has the form of its analog, eq (E–83), but includes also a sign change.

To simplify matters somewhat, one might wish to write the backward rate law in terms of just the forward direction thermodynamic quantities. Thus eq (E–121) could be replaced by:

$$-v_j = s_j k_{-,j} q_{-,j} \left[1 - \left(\frac{K_{+,j}}{Q_{+,j}} \right)^{m_{-,j}} \right] \tag{E–129}$$

Partial differentiation with respect to $Q_{+,j}$ then yields:

$$\frac{\partial v_j}{\partial Q_{+,j}} = \frac{-s_j k_{-,j} q_{-,j} m_{-,j} \left(\frac{K_{+,j}}{Q_{+,j}} \right)^{m_{-,j}}}{Q_{+,j}} \tag{E–130}$$

For the affinity-based form, one obtains that:

$$\frac{\partial v_j}{\partial Q_{+,j}} = \frac{-s_j k_{-,j} q_{-,j} \left(\mathrm{e}^{\frac{A_{+,j}}{\sigma_{-,j} RT}} \right)}{\sigma_{-,j} Q_{+,j}} \tag{E–131}$$

These forms may be useful if it is desired not to carry program variables for the "–" thermodynamic quantities. Also, the Q vector can be defined then entirely in terms of $Q_{+,j}$.

Multi-term rate laws can be dealt with as follows. The multi-term TST form can be written as:

$$\mathrm{v_j} = \mathrm{s_j} \sum_{\mathrm{i}=1}^{\mathrm{i_{T,j}}} \mathrm{k_{+,ij} q_{+,ij}}\ 1 - \left(\frac{\mathrm{Q_{+,j}}}{\mathrm{K_{+,j}}} \right)^{\mathrm{m_{+,ij}}} \tag{E–132}$$

where $i_{T,j}$ is the number of terms in the j^{th} rate law. Note that an *i* subscript has been added to the rate constant, the kinetic activity product, and the *m* parameter, as these are now term-specific. Distributing the surface area over the summation allows one to write this in the general form:

$$\mathrm{v_j} = \sum_{\mathrm{i = 1}}^{\mathrm{i_{T,\,j}}} \mathrm{v_{ij}} \tag{E–133}$$

where v_{ij} is the contribution to v_j from the i^{th} term. Recall that in broad outline, the $\left[\partial v/\partial r\right]$ matrix was obtained for a set of one-term rate laws in the following manner:

$$\frac{\partial \mathrm{v}}{\partial \mathrm{r}} = \left\{\left(\frac{\partial \mathrm{v}}{\partial \mathrm{q}}\frac{\partial \mathrm{q}}{\partial \mathrm{a}} + \frac{\partial \mathrm{v}}{\partial \mathrm{Q}}\frac{\partial \mathrm{Q}}{\partial \mathrm{a}}\right)\frac{\partial \mathrm{a}}{\partial \mathrm{z}}\frac{\partial \mathrm{z}}{\partial \mathrm{n_T}}\frac{\partial \mathrm{n_T}}{\partial \Delta\xi} + \frac{\partial \mathrm{v}}{\partial \mathrm{s}}\frac{\partial \mathrm{s}}{\partial \mathrm{n}}\frac{\partial \mathrm{n}}{\partial \Delta\xi}\right\}\frac{\partial \Delta\xi}{\partial \mathrm{r}} \tag{E–134}$$

For the multi-term case, one can replace this by:

$$\frac{\partial \mathrm{v}}{\partial \mathrm{r}} = \left(\left\{\sum_{\mathrm{i}}\left(\frac{\partial \mathrm{v_i}}{\partial \mathrm{q_i}}\frac{\partial \mathrm{q_i}}{\partial \mathrm{a}} + \frac{\partial \mathrm{v_i}}{\partial \mathrm{Q}}\frac{\partial \mathrm{Q}}{\partial \mathrm{a}}\right)\right\}\frac{\partial \mathrm{a}}{\partial \mathrm{z}}\frac{\partial \mathrm{z}}{\partial \mathrm{n_T}}\frac{\partial \mathrm{n_T}}{\partial \Delta\xi} + \frac{\partial \mathrm{v}}{\partial \mathrm{s}}\frac{\partial \mathrm{s}}{\partial \mathrm{n}}\frac{\partial \mathrm{n}}{\partial \Delta\xi}\right)\frac{\partial \Delta\xi}{\partial \mathrm{r}} \tag{E–135}$$

where v_i is the vector of i^{th} terms for the various reaction rates and q_i is the corresponding vector of kinetic activity products. Basically, all the previous one-term results can be applied by adding an "*i*" subscript in all the appropriate places. For example, eq (E–70) becomes:

$$\frac{\partial v_{ij}}{\partial q_{+,ij}} = \frac{v_{ij}}{q_{+,ij}} \tag{E–136}$$

Eq (E–77) takes the form:

$$\frac{\partial q_{+,ij}}{\partial a_{s'}} = \frac{q_{+,ij}}{a_{s'}}\left(-N_{+,s'ij} + \sum_{s''} N_{+,s''ij}\frac{b_{s's''}}{b_{s''s''}}\right) \tag{E–137}$$

Eq (E–78) becomes:

$$\frac{\partial v_{ij}}{\partial Q_{+,j}} = \frac{-s_j k_{+,ij} q_{+,ij} m_{+,ij}\left(\dfrac{Q_{+,j}}{K_{+,j}}\right)^{m_{+,ij}}}{Q_{+,j}} \tag{E–138}$$

Other kinds of rate laws can be dealt with by making the proper modifications. The activity-term rate law of Plummer et al. (1978) can be dealt with in the above format simply by recognizing the absence of any explicit dependency on Q_{+j}. The dependency on the a vector then flows solely through the q_{+ij}. A specified constant rate can be dealt with similarly by using just one term, equal to $s_j k_{+,j}$, with the surface area held constant. The rate can be held constant on a per unit area basis by doing the same thing, but choosing a different model for the surface area.

One can see that a true Jacobian for a stiff system ODE solver (where $J[r]$) can be rather imposing. In the case dealt with here, it is certainly not trivial, but it is not as formidable as one might have thought. It consists of a lot of pieces, but most of these are relatively simple and tractable. The most difficult parts are or are obtained from things calculated by the AE solver. The matrix $J^{-1}[z]$ is obtained by inverting $J[z]$, the AE solver's Jacobian matrix. The $\tilde{W}$ vector is left over from the computation of $J[z]$. Once the formula for $J[r]$ has been derived and programmed, the key to its successful usage is that not too much CPU time be consumed in repeatedly evaluating it during a model simulation.

Inverting $J[z]$ is fairly straightforward. By definition, one has that:

$$J[z]J^{-1}[z] = I \tag{E–139}$$

The i^{th} column of $J^{-1}[z]$ can be obtained by solving the matrix equation:

$$J[z]x_i = u_i \tag{E–140}$$

where x_i is that column and u_i is a unit vector matching the i^{th} column of I. This repeated solution of a matrix equation for a series of right-hand-side vectors (here the u_i) is especially facilitated by LU decomposition of the matrix. LU decomposition is already the standard method for solving matrix equations in EQ3/6.

E.4.3. DERIVING THE TRUE JACOBIAN J[R], PART II: DISTILLING IT DOWN

In theory, one could treat all the elements on the right hand side of eq (E–134) as programming objects, calculate them according to the formulas derived in the previous section, and plug in the results to obtain the matrix $\partial v / \partial r$. In practice, that would not be very efficient in terms of storage, computation speed, or economy of programming. The goal of this section is to distill things down to something that is both more comprehensible and more efficient. Most of this effort will focus on simplifying the computations for obtaining $\partial v / \partial r$, as that is where most of the complexity lies.

Eq (E–134) was given as:

$$\frac{\partial v}{\partial r} = \left(\left\{ \sum_i \left(\frac{\partial v_i}{\partial q_i} \frac{\partial q_i}{\partial a} + \frac{\partial v_i}{\partial Q} \frac{\partial Q}{\partial a} \right) \right\} \frac{\partial a}{\partial z} \frac{\partial z}{\partial n_T} \frac{\partial n_T}{\partial \Delta\xi} + \frac{\partial v}{\partial s} \frac{\partial s}{\partial n} \frac{\partial n}{\partial \Delta\xi} \right) \frac{\partial \Delta\xi}{\partial r} \tag{E–141}$$

We previously noted that $\left[\partial z / \partial n_T\right]$ is closely related to $J^{-1}[z]$, the inverse of the limiting Jacobian matrix used in the AE solver. It would be this inverse if the full z vector were involved and the n_T vector were extended to include zeros corresponding to elements of z not associated with basis species. However, as noted earlier, the $\left[\partial z / \partial z\right]$ only requires the first part of z that does deal with such species. As the EQ6 AE solver is presented constructed, there is no part of the a vector on the other part of the z vector. Thus here we need deal with the z vector in abbreviated form and n_T only in the true (not extended) form. In this case, we have that:

$$\left[\frac{\partial z}{\partial n_T}\right] = \Phi \tag{E–142}$$

where Φ is the upper left quadrant of $J^{-1}[z]$, a square whose dimension matches the number of aqueous basis species (or the number of mass balances).

Recall from eq (E–119) that:

$$\left[\frac{\partial \Delta\xi}{\partial r}\right] = wI \tag{E–143}$$

where w is a scalar. If A is a square matrix with the same dimension as I, then:

$$AwI = wA \tag{E–144}$$

Using these relationships, we can now write eq (E–141) somewhat more simply as:

$$\frac{\partial \mathrm{v}}{\partial \mathrm{r}} = \mathrm{w}\left(\left\{\sum_{\mathrm{i}}\left(\frac{\partial \mathrm{v_i}}{\partial \mathrm{q_i}}\frac{\partial \mathrm{q_i}}{\partial \mathrm{a}} + \frac{\partial \mathrm{v_i}}{\partial \mathrm{Q}}\frac{\partial \mathrm{Q}}{\partial \mathrm{a}}\right)\right\}\frac{\partial \mathrm{a}}{\partial \mathrm{z}}\,\Phi\,\frac{\partial \mathrm{n_T}}{\partial \Delta\xi} + \frac{\partial \mathrm{v}}{\partial \mathrm{s}}\frac{\partial \mathrm{s}}{\partial \mathrm{n}}\frac{\partial \mathrm{n}}{\partial \Delta\xi}\right) \tag{E–145}$$

Clearly, much of the remaining complexity lies in the summation in braces. If we were to evaluate this summation as written, we would have to run i up to the largest term number ($i_{T,j}$) for any rate law in the set. In the case of rate laws with lower term number, the non-existent higher terms would yield a potentially large collection of zeros. Furthermore, in evaluating the summation literally, a large number of programming objects would have to be created and dealt with. A lot of simplification is possible because $\left[\partial v \partial q_i\right]$ and $\left[\partial v \partial Q\right]$ are diagonal matrices. Furthermore, the matrix $\partial a / \partial z$ following the summation also has a special structure (only the first row, the first column, and the diagonal are non-zero). We will now show that it is possible to combine the summation and the matrix that follows it into a more convenient and efficient matrix:

(E–146)

$$\Xi = \left\{ \sum_{i} \left(\frac{\partial v_i}{\partial q_i} \frac{\partial q_i}{\partial a} + \frac{\partial v_i}{\partial Q} \frac{\partial Q}{\partial a} \right) \right\} \frac{\partial a}{\partial z}$$

Here logically $\Xi = [\partial v / \partial z]$. Consider the first term in this summation ($[\partial v_i / \partial q_i][\partial q_i / \partial a]$). Letting the directionality ("+" or "-") be implied to simplify the notation (directionality being determined by the rate law form, forward or backward, specified for each rate), we have from eq (E–70) (applied term-wise) that:

$$\frac{\partial v_{ij}}{\partial q_{ij}} = \frac{v_{ij}}{q_{ij}} \qquad \text{(E–147)}$$

and similarly from eq (E–77) that:

$$\frac{\partial q_{ij}}{\partial a_{s'}} = \frac{q_{ij}}{a_{s'}} \left(-N_{s'ij} + \sum_{s''} N_{s''ij} \frac{b_{s's''}}{b_{s''s''}} \right) \qquad \text{(E–148)}$$

To simplify the notation a bit, we define:

$$\tilde{N}_{s'ij} = N_{s'ij} - \sum_{s''} N_{s''ij} \frac{b_{s's''}}{b_{s''s''}} \qquad \text{(E–149)}$$

Then eq (E–148) can be written as:

$$\frac{\partial q_{ij}}{\partial a_{s'}} = \frac{-q_{ij} \tilde{N}_{s'ij}}{a_{s'}} \qquad \text{(E–150)}$$

Thus, one can write that:

$$\frac{\partial v_i}{\partial q_i} \frac{\partial q_i}{\partial a} = \begin{array}{cccccccccc}
\frac{v_{i1}}{q_{i1}} & 0 & 0 & 0 & \cdots & \frac{-q_{i1}\tilde{N}_{1i1}}{a_1} & \frac{-q_{i1}\tilde{N}_{1i2}}{a_2} & \frac{-q_{i1}\tilde{N}_{1i3}}{a_3} & \frac{-q_{i1}\tilde{N}_{1i4}}{a_4} & \cdots \\
0 & \frac{v_{i2}}{q_{i2}} & 0 & 0 & \cdots & \frac{-q_{i2}\tilde{N}_{2i1}}{a_1} & \frac{-q_{i2}\tilde{N}_{2i2}}{a_2} & \frac{-q_{i2}\tilde{N}_{2i3}}{a_3} & \frac{-q_{i2}\tilde{N}_{2i4}}{a_4} & \cdots \\
0 & 0 & \frac{v_{i3}}{q_{i3}} & 0 & \cdots & \frac{-q_{i3}\tilde{N}_{3i1}}{a_1} & \frac{-q_{i3}\tilde{N}_{3i2}}{a_2} & \frac{-q_{i3}\tilde{N}_{3i3}}{a_3} & \frac{-q_{i3}\tilde{N}_{3i4}}{a_4} & \cdots \\
0 & 0 & 0 & \frac{v_{i4}}{q_{i4}} & \cdots & \frac{-q_{i4}\tilde{N}_{4i1}}{a_1} & \frac{-q_{i4}\tilde{N}_{4i2}}{a_2} & \frac{-q_{i4}\tilde{N}_{4i3}}{a_3} & \frac{-q_{i4}\tilde{N}_{4i4}}{a_4} & \cdots \\
\cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots
\end{array} \qquad \text{(E–151)}$$

Carrying through, one obtains that:

$$\frac{\partial v_i}{\partial q_i}\frac{\partial q_i}{\partial a} = -\begin{matrix} \frac{v_{i1}\tilde{N}_{1i1}}{a_1} & \frac{v_{i1}\tilde{N}_{1i2}}{a_2} & \frac{v_{i1}\tilde{N}_{1i3}}{a_3} & \frac{v_{i1}\tilde{N}_{1i4}}{a_4} & \cdots \\ \frac{v_{i2}\tilde{N}_{2i1}}{a_1} & \frac{v_{i2}\tilde{N}_{2i2}}{a_2} & \frac{v_{i2}\tilde{N}_{2i3}}{a_3} & \frac{v_{i2}\tilde{N}_{2i4}}{a_4} & \cdots \\ \frac{v_{i3}\tilde{N}_{3i1}}{a_1} & \frac{v_{i3}\tilde{N}_{3i2}}{a_2} & \frac{v_{i3}\tilde{N}_{3i3}}{a_3} & \frac{v_{i3}\tilde{N}_{3i4}}{a_4} & \cdots \\ \frac{v_{i4}\tilde{N}_{4i1}}{a_1} & \frac{v_{i4}\tilde{N}_{4i2}}{a_2} & \frac{-v_{i4}\tilde{N}_{4i3}}{a_3} & \frac{-v_{i4}\tilde{N}_{4i4}}{a_4} & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \end{matrix} \qquad \text{(E–152)}$$

Summation over terms then yields:

$$\sum_i \frac{\partial v_i}{\partial q_i}\frac{\partial q_i}{\partial a} = -\begin{matrix} \frac{\sum_{i=1}^{i_{T,1}} v_{i1}\tilde{N}_{1i1}}{a_1} & \frac{\sum_{i=1}^{i_{T,1}} v_{i1}\tilde{N}_{1i2}}{a_2} & \frac{\sum_{i=1}^{i_{T,1}} v_{i1}\tilde{N}_{1i3}}{a_3} & \cdots & \cdots \\ \frac{\sum_{i=1}^{i_{T,2}} v_{i2}\tilde{N}_{1i2}}{a_1} & \frac{\sum_{i=1}^{i_{T,2}} v_{i2}\tilde{N}_{2i2}}{a_2} & \frac{\sum_{i=1}^{i_{T,2}} v_{i2}\tilde{N}_{2i3}}{a_3} & \cdots & \cdots \\ \frac{\sum_{i=1}^{i_{T,3}} v_{i3}\tilde{N}_{3i1}}{a_1} & \frac{\sum_{i=1}^{i_{T,3}} v_{i3}\tilde{N}_{3i2}}{a_2} & \frac{\sum_{i=1}^{i_{T,3}} v_{i3}\tilde{N}_{3i3}}{a_3} & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \end{matrix} \qquad \text{(E–153)}$$

where $i_{T,j}$ is the number of terms in the j^{th} rate law. Note that the summation is now much more logically organized; it is no longer necessary to sum over unused (zero) terms.

The product $\left[\partial v_i / \partial Q\right]\left[\partial Q / \partial a\right]$ that composes the second term in the summation in eq (E–146) can be treated in similar fashion. Analogous to eq (E–152), one obtains that:

$$\frac{\partial v_i}{\partial Q}\frac{\partial Q}{\partial a} = \begin{matrix} \frac{\theta_{i1} b_{11}}{a_1} & \frac{\theta_{i1} b_{21}}{a_2} & \frac{\theta_{i1} b_{31}}{a_3} & \frac{\theta_{i1} b_{41}}{a_4} & \cdots \\ \frac{\theta_{i2} b_{12}}{a_1} & \frac{\theta_{i2} b_{22}}{a_2} & \frac{\theta_{i2} b_{32}}{a_3} & \frac{\theta_{i2} b_{42}}{a_4} & \cdots \\ \frac{\theta_{i3} b_{13}}{a_1} & \frac{\theta_{i3} b_{23}}{a_2} & \frac{\theta_{i3} b_{33}}{a_3} & \frac{\theta_{i3} b_{43}}{a_4} & \cdots \\ \frac{\theta_{i4} b_{14}}{a_1} & \frac{\theta_{i4} b_{24}}{a_2} & \frac{\theta_{i4} b_{34}}{a_3} & \frac{\theta_{i3} b_{44}}{a_4} & \cdots \\ \cdots & \cdots & \cdots & \cdots & \cdots \end{matrix} \tag{E–154}$$

where:

$$\theta_{ij} = -s_j k_{+,\,ij} q_{+,\,ij} m_{+,\,ij} \left(\frac{Q_{+,\,j}}{K_{+,\,j}}\right)^{m_{+,\,ij}} \tag{E–155}$$

if the j^{th} rate law is written in the forward direction and:

$$\theta_{ij} = -s_j k_{-,\,ij} q_{-,\,ij} m_{-,\,ij} \left(\frac{K_{+,\,j}}{Q_{+,\,j}}\right)^{m_{-,\,ij}} \tag{E–156}$$

if it is written in the backward direction. Note that we are presuming to deal with rate laws in either direction in terms of $Q_{+,j}$ and $K_{+,j}$ as was suggested earlier. Thus, the Q vector consists uniformly of $Q_{+,j}$. This simplifies treatment of the matrix $\left[\partial Q / \partial a\right]$, as there is then no need there to accommodate a possible mixture of forward and backward cases.

If the affinity-based rate law treatment is employed (as it is in EQ6), then eq (E–155) takes the form:

$$\theta_{ij} = \frac{-s_j k_{+,ij} q_{+,ij} \left(e^{\frac{-A_{+,j}}{\sigma_{+,ij} RT}} \right)}{\sigma_{+,ij}} \tag{E–157}$$

while eq (E–156) takes the form:

$$\theta_{ij} = \frac{-s_j k_{-,ij} q_{-,ij} \left(e^{\frac{-A_{+,j}}{\sigma_{-,ij} RT}} \right)}{\sigma_{-,ij}} \qquad \text{(E–158)}$$

Here we consistently treat the rate law in terms of $A_{+,j}$, regardless of its directionality. This is consistent with the above treatment in using $Q_{+,j}$ and $K_{+,j}$ in the same way.

Applying term-wise summation to eq (E–154) yields:

$$\sum_{i} \frac{\partial v_i}{\partial Q} \frac{\partial Q}{\partial a} = \begin{matrix} \frac{b_{11} \sum_{i=1}^{i_{T,1}} \theta_{i1}}{a_1} & \frac{b_{21} \sum_{i=1}^{i_{T,1}} \theta_{i1}}{a_2} & \frac{b_{31} \sum_{i=1}^{i_{T,1}} \theta_{i1}}{a_3} & \dots\ \dots \\ \frac{b_{12} \sum_{i=1}^{i_{T,2}} \theta_{i2}}{a_1} & \frac{b_{22} \sum_{i=1}^{i_{T,2}} \theta_{i2}}{a_2} & \frac{b_{32} \sum_{i=1}^{i_{T,2}} \theta_{i2}}{a_3} & \dots\ \dots \\ \frac{b_{13} \sum_{i=1}^{i_{T,3}} \theta_{i3}}{a_1} & \frac{b_{23} \sum_{i=1}^{i_{T,3}} \theta_{i3}}{a_2} & \frac{b_{33} \sum_{i=1}^{i_{T,3}} \theta_{i3}}{a_3} & \dots\ \dots \\ \dots & \dots & \dots & \dots\ \dots \\ \dots & \dots & \dots & \dots\ \dots \end{matrix} \qquad \text{(E–159)}$$

Combining eqs (E–153) and (E–159) yields:

$$\sum_{i} \left(\frac{\partial v_i}{\partial q_i} \frac{\partial q_i}{\partial a} + \frac{\partial v_i}{\partial Q} \frac{\partial Q}{\partial a} \right) = \begin{matrix} \frac{M_{11}}{a_1} & \frac{M_{12}}{a_2} & \frac{M_{13}}{a_3} & \frac{M_{14}}{a_4} & \dots \\ \frac{M_{21}}{a_1} & \frac{M_{22}}{a_2} & \frac{M_{23}}{a_3} & \frac{M_{24}}{a_4} & \dots \\ \frac{M_{31}}{a_1} & \frac{M_{32}}{a_2} & \frac{M_{33}}{a_3} & \frac{M_{34}}{a_4} & \dots \\ \frac{M_{41}}{a_1} & \frac{M_{42}}{a_2} & \frac{M_{43}}{a_3} & \frac{M_{44}}{a_4} & \dots \\ \dots & \dots & \dots & \dots & \dots \end{matrix} \qquad \text{(E–159)}$$

where:

$$M_{jk} = -\left(\sum_{i=1}^{i_{T,1}} v_{ij}\tilde{N}_{kij}\right) + b_{kj}\left(\sum_{i=1}^{i_{T,1}} \theta_{ij}\right) \tag{E–161}$$

Combining the two summations, this can be written as:

$$M_{jk} = \sum_{i=1}^{i_{T,1}} (-v_{ij}\tilde{N}_{kij} + b_{kj}\theta_{ij}) \tag{E–162}$$

Now the matrix $[\partial a/\partial z]$ is can be written as (Note: the pieces were derived in the previous section):

$$\frac{\partial a}{\partial z} = 2.303 \begin{array}{ccccc} a_1\tilde{W}_1 & a_1\tilde{W}_2 & a_1\tilde{W}_3 & a_1\tilde{W}_4 & \ldots \\ -a_2 & a_2 & 0 & 0 & \ldots \\ -a_3 & 0 & a_3 & 0 & \ldots \\ -a_4 & 0 & 0 & a_4 & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots \end{array} \tag{E–163}$$

Substituting this and eq (E–160) into eq (E–146) and carrying out the matrix multiplication gives:

$$\Xi = 2.303 \begin{array}{ccccc} \left(M_{11}\tilde{W}_1 - \sum_{s' \neq 1} M_{1s'}\right) & (M_{11}\tilde{W}_2 + M_{12}) & (M_{11}\tilde{W}_3 + M_{13}) & \ldots & \ldots \\ \left(M_{21}\tilde{W}_1 - \sum_{s' \neq 1} M_{2s'}\right) & (M_{21}\tilde{W}_2 + M_{22}) & (M_{21}\tilde{W}_3 + M_{23}) & \ldots & \ldots \\ \left(M_{31}\tilde{W}_1 - \sum_{s' \neq 1} M_{3s'}\right) & (M_{31}\tilde{W}_2 + M_{32}) & (M_{31}\tilde{W}_3 + M_{33}) & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots & \ldots \end{array} \tag{E–164}$$

Eq (E–145) can now be written as:

$$\frac{\partial v}{\partial r} = w\left(\Xi\Phi \frac{\partial n_T}{\partial \Delta\xi} + \frac{\partial v}{\partial s}\frac{\partial s}{\partial n}\frac{\partial n}{\partial \Delta\xi}\right) \tag{E–165}$$

We will define another matrx $\Theta = [\partial n_T / \partial \Delta \xi]$. If mass balances are handled in terms of basis species, then from eq (E–96) we have that:

$$\Theta_{s'j} = b_{s'j} \tag{E–166}$$

(Θ is an array of reaction coefficients). If mass balances are handled in terms of chemical elements, then from eq (E–98) we have instead that:

$$\Theta_{\varepsilon j} = \sum_{s'=1}^{s_Q} c_{\varepsilon s'} b_{s'j} \tag{E–167}$$

where $c_{\varepsilon s'}$ is a composition coefficient.

We can now write eq (E–165) as:

$$\frac{\partial v}{\partial r} = w\left(\Xi\Phi\Theta + \frac{\partial v}{\partial s}\frac{\partial s}{\partial n}\frac{\partial n}{\partial \Delta\xi}\right) \tag{E–168}$$

Clearly Θ (and the matrix *M* used to compute it as well as the $\hat{\Theta}$ vector used to calculate *M*) are worthy of being treated as programming objects (here meaning set up as allocated arrays). Φ essentially is obtained as a programming object, but there is no need to create it as a separate object from $J^{-1}[z]$. Clearly Θ is not sufficiently complex to warrant treatment as a programming object. However, as an intermediate step, it is probably logical to treat the product $\Phi\Theta$ as such an object. The product $\Xi\Phi\Theta$ need not be set up as an independent programming object. It can be stored in the object allocated for the matrix $[\partial v / \partial r]$. The product of three matrices to the right in eq (E–168) can then be added to this, and the result then multiplied by the scalar *w* to obtain $[\partial v / \partial r]$.

We now consider the product $[\partial v / \partial s][\partial s / \partial n][\partial n / \partial \Delta\xi]$. All three matrixes involved here are diagonal. Therefore, the product must also be a diagonal matrix, each diagonal element of which will be the product of the corresponding diagonal elements of the matrices being multiplied. We will call this product matrix ζ. Using eqs (E–99) and (E–106), one can obtain that the general form is given by:

$$\zeta = \begin{matrix} \frac{v_1\hat{b}_1}{s_1}\frac{\partial s_1}{\partial n_1} & 0 & 0 & \dots \\ 0 & \frac{v_2\hat{b}_2}{s_2}\frac{\partial s_2}{\partial n_2} & 0 & \dots \\ 0 & 0 & \frac{v_3\hat{b}_3}{s_3}\frac{\partial s_3}{\partial n_3} & \dots \\ \dots & \dots & \dots & \dots \end{matrix} \qquad \text{(E–169)}$$

What $\partial s_j / \partial n_j$ is in each instance depends on the surface area law assigned to the j^{th} reactant mineral. For the case of a constant surface area:

$$s_j = const \qquad \text{(E–170)}$$

one has that:

$$\frac{\partial s_j}{\partial n_j} = 0 \qquad \text{(E–171)}$$

and hence:

$$\zeta_{jj} = 0 \qquad \text{(E–172)}$$

For the case of constant specific surface area, the surface area is given by:

$$s_j = \hat{s}_j M_j n_j \qquad \text{(E–173)}$$

where M_j is the molecular weight. Partial differentiation then yields:

$$\frac{\partial s_j}{\partial n_j} = \hat{s}_j M_j \qquad \text{(E–174)}$$

and hence:

$$\zeta_{jj} = \frac{v_j \hat{b}_j \hat{s}_j M_j}{s_j} \qquad \text{(E–175)}$$

For the case of growth of a fixed number of particles, one has that:

$$s_j = s_{j,0}\left(\frac{n_j}{n_{j,0}}\right)^{2/3} \tag{E–176}$$

where $s_{j,0}$ is the surface area and $n_{j,0}$ the number of moles at point 0. For this, partial differentiation yields:

$$\frac{\partial s_j}{\partial n_j} = \frac{2}{3}\frac{s_j}{n_j} \tag{E–177}$$

and hence:

$$\zeta_{jj} = \frac{2}{3}\frac{v_j \hat{b}_j}{n_j} \tag{E–178}$$

Now we can write eq (E–168) as:

$$\left[\frac{\partial v}{\partial r}\right] = w(\Xi\Phi\Theta + \zeta) \tag{E–179}$$

This is far less imposing that than eq (E–135), which is what we started with for the case of multi-term rate laws.

Because of the relative simplicity of ζ, it is not worth the bother of treating it as a programming object. Note that ζ has the same dimensions as the product $\Xi\Phi\Theta$.

E.5. REFERENCES

Aagaard, P., and Helgeson, H. C. 1982. “Thermodynamic and kinetic constraints on reaction rates among minerals and aqueous solutions. 1. Theoretical considerations.” *American Journal of Science, 282*, 237–285. New Haven, Connecticut: Kline Geology Laboratory, Yale University. TIC: 225516.

Carnahan, B., Luther, H. A., and Wilkes, J. O. 1969. *Applied Numerical Methods.* New York, NY: John Wiley & Sons. TIC: 245078.

Gear, C. W. 1971a. “The automatic integration of ordinary differential equations.” *Communications of the ACM, 14*, 176–179.

Gear, C. W. 1971b. “Algorithm 407-DIFSUB for solution of ordinary differential equations.” *Communications of the ACM, 14*, 185–190.

Helgeson, H.C. 1968. "Evaluation of Irreversible Reactions in Geochemical Processes Involving Minerals and Aqueous Solutions - I. Thermodynamic Relations." *Geochimica et Cosmochimica Acta, 32,* 853–877. New York, New York: Pergamon Press. TIC: 220011.

Helgeson, H.C.; Brown, T.H.; Nigrini, A.; and Jones, T.A. 1970. "Calculation of Mass Transfer in Geochemical Processes Involving Aqueous Solutions." *Geochimica et Cosmochimica Acta, 34,* 569–592. New York, New York: Pergamon Press. TIC: 225570.

Helgeson, H.C.; Murphy, W.M.; and Aagaard, P. 1984. "Thermodynamic and Kinetic Constraints on Reaction Rates Among Minerals and Aqueous Solutions, II. Rate Constants, Effective Surface Area, and the Hydrolysis of Feldspar." *Geochimica et Cosmochimica Acta, 48,* 2405–2432. New York, New York: Pergamon Press. TIC: 220177.

Plummer, L.N., Wigley, T.M.L. and Parkhurst, D.L. 1978. “The Kinetics of Calcite Dissolution in CO_2-water systems at 5°C to 60°C and 0.0 to 1.0 atm CO_2.” *American Journal of Science, 278,* 179–216. New Haven, Connecticut: Yale University, Kline Geology Laboratory. TIC: 240650.

Wolery, T.J. and Daveler, S.A. 1992. *EQ6, A Computer Program for Reaction Path Modeling of Aqueous Geochemical Systems: Theoretical Manual, User's Guide, and Related Documentation (Version 7.0).* UCRL-MA-110662 PT IV. Livermore, California: Lawrence Livermore National Laboratory. ACC: MOL.19980701.0459.

INTENTIONALLY LEFT BLANK