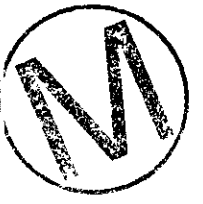
Title 40 CFR Part 191
Compliance Certification
Application
for the
Waste Isolation Pilot Plant

Appendix SECOFL2D

United States Department of Energy
Waste Isolation Pilot Plant

Carlsbad Area Office
Carlsbad, New Mexico

WIPP PA User's Manual for SECOFL2D



WIPP PA

User's Manual

for

SECOFL2D, Version 3.03

Document Version 1.01

WPO # 37271

May 7, 1996

Table of Contents

*The User's Manual in Appendix I has its own pagination.

1.0 INTRODUCTION

This document serves as a User's Manual for SECOFL2D, as used in the 1996 WIPP PA calculation. As such, it describes the code's purpose and function, the user's interaction with the code, and the models and methods employed by the code.

1.1 Software Identifier

Code Name: SECOFL2D (Sandia-ECOdynamics/FLow in 2 Dimensions)
WIPP Prefix: ST2D2
Version Number: 3.03, May 7, 1996.
Platform: FORTRAN 77 for OpenVMS AXP, ver 6.1, on DEC Alpha

1.2 Points of Contact

1.2.1 Code Sponsor

Rebecca L. Blaine
Ecodynamics Research Associates
P.O. Box 9229
Albuquerque, NM 87119
Voice: (505) 843-7445
Fax: (505) 843-9641

1.2.2 Code Consultant

Patrick J. Roache
Ecodynamics Research Associates
P.O. Box 9229
Albuquerque, NM 87119
Voice: (505) 843-7445
Fax: (505) 843-9641

2.0 FUNCTIONAL REQUIREMENTS

R.1 This code shall perform groundwater hydrology simulation by solving a partial differential equation for head. The code can model heterogeneous materials, steady state or transient solutions, and fixed gradient or flux boundary conditions on a quasi-horizontal grid.

R.2 This code shall solve for Darcy velocity using the solution for head.

R.3 Problems may be run on regional and local grids where the local grid is decoupled from the regional grid in space and time. (Decoupling in time not tested for WIPP QA purposes.)

R.4 The head boundary conditions for the local simulation are interpolated using bilinear interpolation from the regional solution.

3.0 REQUIRED USER TRAINING AND/OR BACKGROUND

Code user prerequisites are described in detail in Appendix I.

4.0 DESCRIPTION OF THE MODELS AND METHODS

Models and methods for SECOFL2D are described in detail in Appendix I.

NOTE

Sandia's Configuration Management System (CMS) is the operating system platform for the 1996 WIPP Performance Assessment (PA) calculation. All statements in Appendix I referencing CAMCON, the operating system upon which CMS is in part based, are applicable to the new CMS operating system.

NOTE

The reader should note that the following capabilities are not supported for the 1996 WIPP PA calculation:

- MAC = 0 option
- adaptive SOR relaxation factor
- unconfined flow option
- lake/river conductance and constant head region options
- quasi-climate factors
- well options
- solution=adaptive time-stepping option

The above options are explained in Appendix I, and Appendix I clearly states that they are not considered operative for the 1996 WIPP PA calculation.

5.0 CAPABILITIES AND LIMITATIONS OF THE SOFTWARE

Capabilities and limitations of the software are discussed in detail in Appendix I.

6.0 USER INTERACTIONS WITH THE SOFTWARE

User interactions with the software are discussed in detail in Appendix I.

7.0 DESCRIPTION OF INPUT FILES

Input files for SECOFL2D are discussed in detail in Appendix I.

8.0 ERROR MESSAGES

SECOFL2D outputs the following error messages, which cause SECOFL2D to abort:

- **"The build for SF2D had returned an error or warning . . ."** If SECOFL2D has a problem with the contents of the input control file supplied by PRESECOFL2D, it outputs this message.

- **"Inconsistency in file grid dimensions . . ."** If the grid dimensions in the input file are not equal to the code's compile dimensions, it outputs this message.

- **"Error in seco_2L; max # time steps > dimension, . . ."** If the number of time steps exceeds the dimensioned value, it outputs this message.

- **"Error in multigrid dimensioning . . ."** If the calculated dimensions for the multigrid solver are not correct for this problem, it outputs this message.

- **"No convergence in sor25g_dp . . ."** If SECOFL2D fails to converge in the maximum number of iterations, it outputs this message.

In addition, error messages resulting from invalid file names are output to the terminal and also cause execution to abort.

9.0 DESCRIPTION OF OUTPUT FILES

Output files for SECOFL2D are discussed in detail in Appendix I. A sample diagnostics/debug file is provided in Appendix II.

10.0 REFERENCES

References for SECOFL2D are listed in Appendix I.

11.0 APPENDICES

The following section provides the appendices for this document.

Appendix I: SECOFL2D User's Manual, by Patrick Roache, Rebecca Blaine, and Bruce Baker

SAND90-7096
Unlimited Release
Printed October 1995

Distribution
Category UC-721

SECOFL2D User's Manual

Version 3.03
WIPP Prefix SF2D2

Patrick J. Roache[1]
Rebecca L. Blaine[1]
Bruce L. Baker[2]

Sandia National Laboratories
Albuquerque, NM 87185

[1] Ecodynamics Research Associates, Albuquerque, NM.

[2] Technadyne, Albuquerque, NM.

ABSTRACT

This is a user's manual for the SECOFL2D (Sandia-ECOdynamics/FLow in 2 Dimensions) Code, Version 3.03, WIPP Prefix SF2D2. This code calculates single-phase Darcy flow for groundwater flow problems in two dimensions. The code is written in modern, structured FORTRAN for the VAX/Alpha class of computers. The formulation is based on a single partial differential equation for hydraulic head using fully implicit time differencing. Both confined and unconfined aquifer conditions are simulated. The flow is solved in both a regional and local grid, each of which is defined independently of the grid that defines the aquifer properties. A semi-coarsening multigrid solver is used to increase solution efficiency for large array dimensions. High-order accuracy particle tracking is available with the related SECOTR codes, three-dimensional flow calculations are possible with the related code SECOFL3D, and efficient second-order accurate radionuclide transport calculations are possible with the related SECOTP2 code (all separately documented). This manual describes how to use both a stand-alone version of SECOFL2D and the version within the Compliance Assessment Methodology CONtroller (CAMCON), the controller for compliance evaluations developed for the Waste Isolation Pilot Plant (WIPP).

PREFACE

This manual documents the SECOFL2D (Sandia-ECOdynamics/FLow in 2 Dimensions) code, Version 3.03, WIPP Prefix SF2D2, the groundwater flow program module for the Compliance Assessment Methodology CONtroller (CAMCON). The manual is written by Patrick J. Roache, Rebecca L. Blaine, and Bruce L. Baker, who are contractors to the Sandia National Laboratories (SNL) Performance Assessment Division. Although published as a contractor report, this report has successfully completed SNL peer review and SNL management review and is an official Sandia (SAND) report. Information on the code sponsor and code consultant follows.

Code Sponsor:

Rebecca L. Blaine
Ecodynamics Research Associates, Inc.
P. O. Box 9229
Albuquerque, New Mexico 87119-9229
Telephone (505) 843-7445

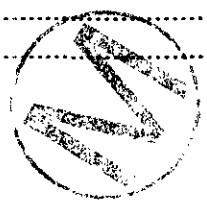
Code Consultant:

Dr. Patrick J. Roache
Ecodynamics Research Associates, Inc.
P. O. Box 9229
Albuquerque, New Mexico 87119-9229
Telephone (505) 843-7445

CONTENTS

Figures

CAMCON Keywords



LIST OF SYMBOLS

α	compressibility of the medium
β	compressibility of water
ϕ	porosity of the medium
μ	viscosity of water
ρ	mass density of water
ρ_f	mass density of freshwater
Δt	time step
Δt_{old}	previous time step
Δx, Δy	computational cell size in x, y directions
Δz	vertical cell thickness
adx, ady	arrays of Δx, Δy
amp	modeled amplitude of the variation
aq_bot	array of aquifer bottom elevation
aq_thick	array of aquifer thickness
A_T	aquifer thickness
C2, C4, C5, C6, C8, C10..	stencil elements
CONV	convergence tolerance
con_salt	saline concentration
CS5	the steady-state contribution to the central stencil element C5
CF	climate factor
C_{RIV}	surface water conductance term
cycle	modeled length of the climate cycle
E	elevation
E_{ET}	time discretization error estimator
e	density perturbation
elev	elevation calculated at the center of the aquifer
error_tol	acceptable level of error estimate E_{ET} before adaptation is activated
e_rho	density ratio perturbation
flux_u, *flux_v*	flux components arising from u and v
FNEW	array of new values of under-relaxed quantity
FOLD	array of old values of under-relaxed quantity
g	gravity vector
h	hydraulic head
$\hat{h}$	the new value of h predicted by an explicit algorithm
h_0	ghost cell value of h
h_1	first interior node value of h
$h_{boundary}$	Dirichlet boundary condition value of h
head	dependent variable of the simulation, h
$h_{i,j}$	head in the aquifer cell (i, j)
h^{n+1}	new head value predicted by the fully implicit algorithm

$\hat{h}^{n+1}$	new head values predicted by the explicit algorithm
h_{RANGE}	total range of h^{n+1}; maximum (over all i,j) - minimum
h_{RIV}	specified surface water (river/lake) head
i, j	indexes that refer to a location in space, denoted by (x_i, y_j)
K	hydraulic conductivity
m,n	node indexes
nadlmx	limit on the maximum number of intra-time step adaptations
O( )	order
p	pressure
Q	a general source/sink term including possible nonlinear terms
Q_{RIV}	source/sink term from surface water (river/lake)
R_{BOT}	elevation at the bottom of the riverbed
RELF	general under-relaxation factor
rf	SOR relaxation factor
RHS	Right-Hand-Side of an equation
S	storativity
S_S	specific storage
S_Y	specific yield or drainage porosity
SOR	Successive Over-Relaxation
T_F	transmissivity factor
t_{NORM}	normalized simulation time
u, v	Darcy velocity components (specific discharges) in x,y directions
uscf, vscf	arrays of Darcy velocities (specific discharges) in x,y directions
visc_ratio	viscosity ratio of saline water compared to fresh water
x_i, y_j	cartesian coordinates of the (i, j) computational cell
z_{top}, z_{bot}	the vertical coordinates of the top and bottom of the cell (z_{top} - z_{bot} = aquifer thickness)

1. INTRODUCTION

1.1 Purpose

SECOFL2D, Version 3.03, performs the hydrologic modeling subtask for the Waste Isolation Pilot Plant (WIPP) performance assessment. Before disposing of radioactive waste at the WIPP, the United States Department of Energy (DOE) must comply with the United States Environmental Protection Agency's Environmental Radiation Protection Standards for Management and Disposal of Spent Nuclear Fuel, High-Level and Transuranic Radioactive Wastes 40 CFR Part 191 (U. S. EPA, 1985). Key to assessment of the long-term performance of the WIPP is the demonstration of compliance with the Containment Requirements (§ 191.13) in Subpart B of the Standard.

This manual explains how to use Version 3.03 of the SECOFL2D (Sandia-ECOdynamics) two-dimensional flow code, which comprises one of the major programs controlled by the Compliance Assessment Methodology CONtroller (CAMCON) system (Rechard et al., 1989; Rechard, 1992). The results produced by the programs that CAMCON controls will ultimately determine whether WIPP complies with § 191.13.

The CAMCON executive program links distinct model components (including SECOFL2D) to insure repeatability and avoid misinterpretation, in addition to controlling Latin Hypercube sampling of parameters. Following the guidance of Subpart B of the Standard, code modules in the CAMCON system express the calculated releases of radionuclides as complementary cumulative distribution functions (CCDFs) that indicate the probability of exceeding various levels of cumulative radionuclide release. The CAMCON system is fully described in Rechard et al., 1989; and Rechard, 1992; the use of CCDFs for WIPP compliance assessment is explained in detail in Helton (1990), Howery et al. (1990), WIPP PA (1992), and Helton et al. (1995).

SECOFL2D is based on its predecessor, SECO2DH Version 2.1, and fulfills similar objectives as its three-dimensional counterpart SECOFL3D, presently under final development (Knupp and Askew, 1995); for the purposes of this document, these codes will be referred to as the SECO_FLOW codes, as they have been previously referred to generically in open-literature publications (Roache et al., 1990; Roache, 1991, 1992, 1993a,b, 1994b; Salari et al., 1995).

SECO_FLOW determines the regional and local flow field in the Culebra Dolomite Member, the lower of two water-bearing layers of dolomite within the Rustler Formation. The groundwater aquifer in the Culebra Dolomite has been identified (WIPP PA, 1992) as an important pathway to the accessible environment. Figure 1 shows how SECO_FLOW interacts with several of the major CAMCON modules. The Culebra flow field results from SECO_FLOW and the calculated nuclide transport from the waste panel into the borehole from the CAMCON module, PANEL, are input to another CAMCON module, SECOTR2D (or generically, SECO_TRANSPORT), which calculates nuclide transport in the Culebra to the accessible environment outside the controlled area (Salari et al., 1992; Salari and Blaine, 1995).

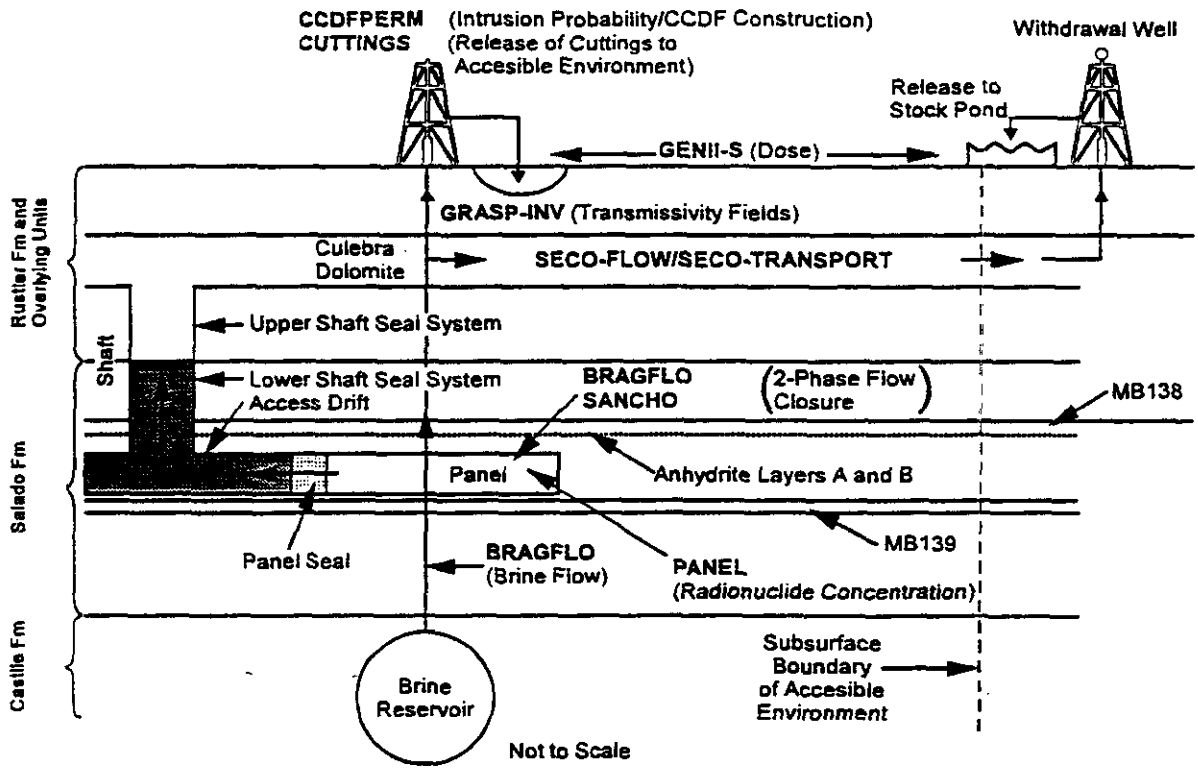


Figure 1. Major CAMCON Modules (after WIPP [1992])

1.2 Overview of SECO Codes

SECO_FLOW calculates single-phase Darcy flow for groundwater flow problems in two dimensions. The formulation is based on a single partial differential equation for hydraulic head using fully implicit time differencing. Both confined and unconfined aquifer conditions are simulated. The flow is solved in both a regional and a local grid, each of which may be defined independently of the grid that defines the aquifer properties. A semi-coarsening multigrid solver is used to increase solution efficiency for large array dimensions. High-order accuracy particle tracking is available for both grids with the SECO_TRACKER codes, in both two- and three-dimensional versions. Radionuclide transport calculations are performed using the 2-D and 3-D SECO_TRANSPORT codes. SECO_TRACKER and SECO_TRANSPORT are described in detail in separate documentation (Roache, 1995b; Salari and Blaine, 1995). These codes are all written in DEC VMS FORTRAN using modern, structured coding practice, extensive internal documentation, and clear mnemonics. The codes are designed specifically for execution on VAX/Alpha computers operating under the VMS operating system, but also have been adapted to a Cray YMP, 80486 and Pentium Personal Computers, and a massively parallel Paragon computer. Animation display of time-dependent 2D head contours and particle tracking is provided by the SECO_VIEW code (Salari, 1995) built for the Silicon Graphics IRIS 4D-25 workstation. The guiding philosophy for all the SECO codes is to make the problem definition convenient and to facilitate, as much as possible, the running of grid-convergence tests and local-area simulations within the larger regional-area simulation.

SECO_FLOW features generalized boundary conditions, lakes/rivers, and water wells, including quasi-climatic variations on each. These features make SECO_FLOW capable of solving more general groundwater flow problems. Groundwater management, aquifer pumping, and contaminant plume tracing are possible applications.

SECO_FLOW features two distinct interfaces. First, the user can interact directly with the SECOFL2D codes on a stand-alone basis. (The interactive capability of the predecessor code, SECO2DH Version 2.1, with input and controls entered via a terminal, has been dropped in favor of commented NAMELIST file input.) Second, for WIPP performance assessment, SECO_FLOW is a component within the CAMCON system, the controller for compliance evaluations developed for the WIPP. In the CAMCON version, SECO_FLOW is controlled by input files.

1.3 Manual Organization

This manual has seven additional chapters and three appendixes. Chapter 2 describes the general modeling capabilities of SECO_FLOW. Chapter 3 discusses the governing equations, discretizations and solvers. Chapter 4 introduces the configuration options. Chapter 5 describes the formulation for variable density flow of saline water in terms of freshwater head. Chapter 6 covers issues of compatibility, conservation and consistency between the SECO flow and transport codes. Chapter 7 provides all the necessary user information to run SECO_FLOW within CAMCON, and Chapter 8 guides the user through the solution of two simple example groundwater flow problems. Appendix A is a reprint of Roache et al. (1990), which discusses verification of spatial accuracy and benchmarking a variety of groundwater hydrology computational fluid dynamic codes, including SECO_FLOW. Appendix B presents verification of temporal accuracy. Appendix C describes the program structure of the stand-alone version of SECO_FLOW, not used in the WIPP PA CCDF calculations but which has been used in scoping calculations and which has the advantage of being independent of the executive controller system CAMCON.

1.4 Remarks on WIPP QA Requirements

All code features described in this manual have been exercised and verified, but not all have been rigorously tested and thoroughly documented to the high level required for WIPP Performance Assessment Software Quality Assurance (QA). Not all of these features will be used for WIPP PA, so they need not be subjected to the time- and resource-consuming WIPP QA process. All the code features to be used for future WIPP PA calculations have been thoroughly tested and verified, and have attained the status of QA Qualified Code. For WIPP QA purposes, the QA unqualified features are considered inoperative in Version 3.03. Inadvertent use by WIPP PA analysts (users) in the CAMCON system is prevented by checks in the Preprocessor. (See Section 4.3 below for special consideration for quasi-climate factors.)

In this manual, the sections describing such QA unqualifed features will include the following notice.

NOTICE

"For WIPP QA purposes, [this feature] is considered inoperative in Version 3.03."

1.5 Code User Prerequisites

In order for the theoretical sections of this manual to be useful, the user will need the following.

- A first-year graduate level understanding of partial differential equations.
- Senior level undergraduate course or equivalent in linear algebra.
- Graduate or senior level undergraduate course or equivalent in numerical methods.
- First level undergraduate course or equivalent in groundwater flow and transport.

To apply SECO_FLOW effectively, the user should be aware of the code's capabilities and limitations. It is recommended that the user run the sample problems provided in Chapter 8 to gain understanding in using the code.

2. MODELING CAPABILITIES

2.1 Problem Decoupling and Facilitation of Grid Convergence Tests in Multiple Domains

Grid convergence tests, or other reliable truncation error estimators, are important to any computational fluid dynamics (CFD) project (e.g., see Roache, 1972, 1990, 1993c, 1994a,b, 1995; Roache et al., 1990). Unfortunately, in groundwater flow modeling (as in other geophysical studies) such elementary but time-consuming exercises are exceptional (Roache, 1994b; Westerink and Roache, 1995). Unavoidable uncertainty in the governing physical parameters is often used as the rationale for neglecting grid convergence tests. The WIPP PA group tries to improve accuracy whenever possible.

The guiding philosophy for the SECO codes is to make the problem definition convenient and to facilitate the running of grid convergence tests and local-area simulations within the larger regional-area simulation. Accordingly, considerable effort has been expended in the SECO codes to facilitate the performance of grid convergence tests by the users. Although these features involve few algorithmic innovations, they represent a considerable improvement to change and depart from traditional groundwater code design.

One aspect of this philosophy is the decoupling of the problem definition from the computational grid. The aquifer properties are defined on a discrete data base (called the aquifer-defining-grid) that can be independent of the computational grids. A subroutine ("prop_range", included in the preprocessor in the CAMCON version) then calculates the actual ranges of property values in the aquifer and compares these to acceptable range values set in (user-accessible) data statements, and outputs the actual ranges and a warning when the acceptable range has been exceeded. The properties on the computational grids (local and regional) are then obtained automatically by interpolation using the conservation-preserving interpolation method of Dukowitz (1985), extended by P. Knupp of Ecodynamics (Knupp, 1992) to handle unaligned boundaries. Likewise, boundary conditions are not specified in the computational grids but in the continuum, and the location and properties of production wells, recharge wells, and surface water features (lakes/rivers) are defined only once in map coordinates, i.e., the user need not locate a well in a cell.

The aquifer-defining-grid is not used for computation. For example, if the aquifer properties varied linearly with direction, a grid with three points in the x and y directions would be sufficient to define the aquifer properties, whereas the CFD solution for many wells might require hundreds of points in each direction, and the user might want to run a sequence of CFD grid solutions (e.g., 20×20, 40×40, 80×80, 160×160) to verify the numerical accuracy. This sequence of solutions should not require the user to define aquifer properties point by point in each computational grid.

A similar aspect of this philosophy is the decoupling of the problems in the regional computational grid from the local computational grid, both in space and time. The 2-D SECO_FLOW code uses two domains, a local grid embedded in a larger regional grid. This allows both higher resolution in the local grid directly over the waste repository, and closer alignment of grid boundaries with natural features particular to the site. A number of parameters, including the boundaries of the computational regions, the spatial increments (cell sizes), the simulation times, and the time steps, are all decoupled in both space and time. The only requirement is that the local grid problem domain of definition must lie within the regional grid

problem domain of definition. For example, the regional grid problem could be defined over a 34- by 40-km area over 1,000 years, while the local grid problem could be limited to a 15- by 20-km area rotated 35 degrees within the regional grid and cover only 250 years beginning 100 years after the regional grid problem begins. Likewise, definition of boundary conditions (types and values) and wells (locations and pumping schedules) are decoupled from the computational grid and are defined in the continuum.

2.2 Transmissivity Factors

The earlier SECO2DH code required that the user determine confined (artesian) or unconfined (water table) conditions. In SECOFL2D, transmissivity factors are automatically calculated, which permits the determination of the confined or unconfined condition at each time step. Conditions may change as the simulation progresses. (See Sections 3.4 and 3.5 below for details.) Significantly, if a computational cell de-waters, so that the transmissivity factor becomes zero, SECO_FLOW allows the cell to recharge later and the transmissivity factor automatically to reactivate. This is in contrast to many groundwater hydrology codes (e.g., the original MODFLOW code, McDonald and Harbaugh, 1988) and avoids a potentially serious simulation error. The fixes employed in SWIFT II (Reeves et al., 1986) and the later version of MODFLOW (McDonald et al., 1991) are non-ordered approximations. The indication is that their treatments are also not correct even before the limit of de-watered conditions is reached.

NOTICE

For WIPP QA purposes, the water table condition is considered inoperative in Version 3.03.

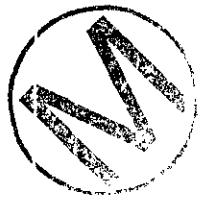
2.3 Hydraulic Head Calculation

Earlier versions of the SECO_FLOW codes used simple point- or line-iterative methods to solve the implicit equation for hydraulic head. Such methods are inefficient for large arrays, and computational time increases rapidly as grid resolution increases. (A doubling of array sizes in each direction, i.e., a doubling of resolution, increases the number of unknowns by a factor of 4. For an optimal solution method, the computational time would also increase by a factor of 4, but for the point and line solvers, the computational time would increase by roughly an order of magnitude.) SECO_FLOW utilizes a specially developed semi-coarsening multigrid algorithm that exhibits optimal performance. This algorithm is described in Section 3.2.

2.4 Additional Capabilities

The SECOFL2D code solves a two-dimensional version of essentially the same fundamental equation for hydraulic head as the United States Geological Survey (USGS) code MODFLOW (McDonald and Harbaugh, 1988). However, the SECOFL2D code has the following additional capabilities.

- Regional and local grid solutions
- General boundary conditions
- Option for interactive problem definition and output
- Options for initial condition specification
- Options for either cell-centered or node-centered grids



- Automated specification of grid spacing, including uniform spacing or power-law stretching for increased resolution near physical features
- Automated specification of time steps, including uniform spacing or power-law stretching for increased time resolution near events
- Temporal error estimation and option for automated (solution adaptive) determination of time steps
- Option for variable density flow calculation
- Parameterized quasi-climatic variations of boundary conditions
- Correct treatment of de-watered cells

The regional and local grid capabilities include the following.

- Independent specification of aquifer properties in an aquifer-defining grid (independent of the computational grids) and checking for acceptable parameter ranges
- Convenient specification of regional and local grid translation and rotation without the need for redefining aquifer properties
- A single specification of well properties and locations applicable to both the regional and local grids
- Independent specification of time stepping for regional and local grids
- Automated, conservative interpolation of time-dependent or steady boundary conditions from the regional grid solution to the local grid boundaries

2.5 Block-Centered or Cell-Centered Discretization

SECO_FLOW has been written with an option flag called MAC (after the original Marker And Cell method, Harlow and Welch, 1965; see also Roache, 1972, pp. 196 ff) to select either the most common block-centered discretization (MAC=1), with the cell edge coincident with the aquifer edge, or node-centered discretization (MAC=0), with the cell center (or node) on the aquifer edge. (Details are presented in Section 3.2 below.)

Unless required by a specific study, which is not anticipated, the default cell configuration will be MAC=1. This configuration clearly more accurately locates the aquifer edge for both Dirichlet (fixed head) and Neumann (fixed gradient) boundary conditions. Nevertheless, there is already considerable investment in coding both options (MAC=1 and MAC=0), and the capability of operating under the MAC=0 option could be useful for comparison to other codes such as HST3D (Kipp, 1986) that do use this configuration. It is clearly not advisable to remove the MAC=0 option.

NOTICE

For WIPP QA purposes, the MAC = 0 option is considered inoperative in Version 3.03.

3. GOVERNING EQUATIONS, DISCRETIZATION, AND SOLVERS

3.1 Governing Equations

The 2-D partial differential equation solved for hydraulic head, h, is the following:

$$S\,\partial h / \partial t = \nabla \cdot (A_T K \nabla h) - Q \tag{1}$$

where S is the storativity, A_T is the aquifer thickness, K is the (tensor) hydraulic conductivity, and Q is a general source/sink term including possible nonlinear terms, as in Section 3.6 below. (A modification of Eqn. (1) is required for unconfined conditions in which the head is below the top level of the cell; see Section 3.4 below.) We write the terms A_T and K separately for clarity since they are separate input data in the code; more often, Eqn. (1) is written with this product consolidated into the transmissivity $T = A_T K$.

This formulation is based on Darcy's law for porous media flow velocity, approximately constant density, the conservation of mass, and the small angle approximation that the aquifer flow is essentially horizontal (or parallel to the confining beds). (Historically, the small angle approximation is associated with the Dupuit hypothesis only in the context of unconfined flows.) For derivations, see Bear (1972), De Marsily (1987, Section 5.3), Bear and Verruijt (1987, Section 4.2), Anderson and Woessner (1992, Chapter 2), or any elementary text on groundwater flows. The principal axes of K must be aligned along the coordinate directions x and y in Version 3.03. (Although not installed in the QA version of the code, SECO_FLOW modules for non-aligned general tensor properties have been developed and thoroughly verified; see Roache et al., 1990.) The metric system is used, so all heads and distances are expressed in meters and time is expressed in seconds. Chapters 7 and 8 describe the units for all input data. (Actually, the computer arithmetic does not recognize units, and any consistent set of units could be used except for the data set value of water compressibility.) Time may optionally be input to the code in years; all times are output in both seconds and years.

S is the product of the specific storage S_S and the aquifer thickness A_T,

$$S = S_S A_T \tag{2}$$

The "specific storage" S_S (as in MODFLOW, McDonald and Harbaugh, 1988) or "specific storage" or "medium's storativity", as in Bear (1972), is the volume of water stored, per unit volume of aquifer, per unit head (metric units in meters). S_S is calculated externally to SECO_FLOW proper (in the aquifer-defining part of the analysis) from the basic properties of rock compressibility, water compressibility, water density, and porosity:

$$S_s = \rho g (\alpha + \beta \phi) \tag{3}$$

where α is the medium's compressibility, ρ and β are the density and compressibility of water (set in a data statement) and ϕ is the porosity.

This 2-D Eqn. (1) cannot be obtained from the full 3-D equation in cartesian coordinates (x,y,z) by simply eliminating variations in the third coordinate z, because Eqn. (1) allows the

aquifer thickness A_T to vary in space, i.e., $A_T = A_T(x,y)$. (In fluid dynamics texts, this approach is sometimes referred to as a "quasi-two-dimensional formulation" because it allows for metric or area variations in the third direction although it is still a two-coordinate problem.)

3.2 Discretization and Solvers

The above equation (or the steady-state version with $\partial h/\partial t = 0$) is discretized using standard second-order differences in space and first-order backward (fully implicit) differences in time (McDonald and Harbaugh, 1988; Roache, 1972).

For the node-centered option (MAC = 0), all variables *u, v, h, T* are *co-located*, meaning that the indexes *i, j* refer to the same location in space, denoted by (x_i, y_j). (The code is written for tensor product grids, which allow for non-uniform cell spacing but are restricted to orthogonal grids; hence, variation of Δx with j or Δy with i is not allowed.) The MAC = 0 cell structure is shown in Figure 2. Using any of several standard techniques of discretization (e.g., see Roache, 1972) the discrete analog of the x-derivative term of the Right-Hand-Side of Eqn. (1) may be derived as

$$RHS_x(i,j) = \frac{1}{\Delta x_i}\left[\frac{T_{i+1/2}(h_{i+1}-h_i)}{(\Delta x_{i+1}+\Delta x_i)/2} - \frac{T_{i-1/2}(h_i-h_{i-1})}{(\Delta x_i+\Delta x_{i-1})/2}\right] \tag{4}$$

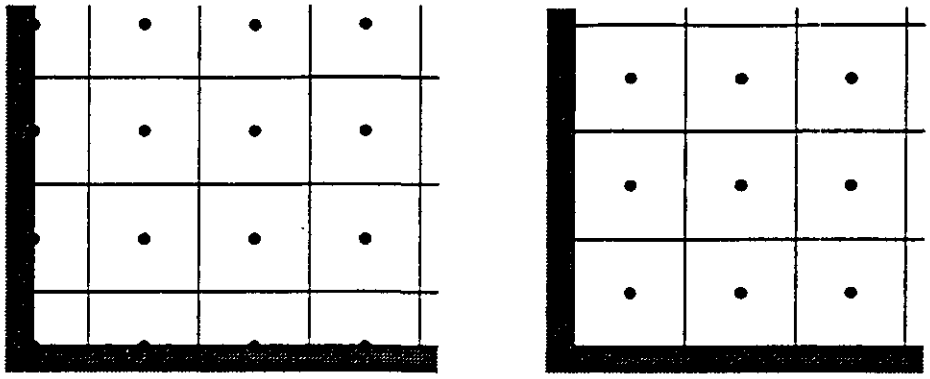
Similarly for RHS_y. Then

$$RHS(i,j) = RHS_x(i,j) + RHS_y(i,j) - Q(i,j) \tag{5}$$

The notation $T_{+1/2}$ does *not* imply a simple averaging of T_i and T_{i+1}, nor does it imply a location half-way between i and $i+1$ in physical coordinates. It does imply the evaluation (to be discussed below) of T at the cell interface between cells i and $i+1$ (which interface is "half way" in logical space, i.e., in the indexes i and $i+1$.) This discretization of *RHS* is a standard second-order accurate block-centered finite difference or finite volume discretization for the strictly *planar* 2-D continuum equation.

Other straight-forward algebraic relations (e.g., inclusion of the cell volume term $A_T\Delta x\Delta y$ on the left side time derivative, grouping of product terms, introduction of a scaling factor to minimize computer round-off error) are used in the assembly of the matrix to be solved. (As in finite element practice, the Fortran arrays defining this matrix are "actualized", rather than being reconstructed during iterative sweeps.) These are algebraically grouped into "stencil elements" in the symbolic form

$$\begin{matrix} & & C2 & & \\ [C4 & C5 & C6] & h = & C10 \\ & & C8 & & \end{matrix} \tag{6}$$



(a) MAC = 0 (b) MAC = 1

Figure 2. MAC = 0 and 1 cell structure.

or in expanded form with Fortran array notation,

$$\begin{aligned} &\text{C2(i,j)} * \text{h(i,j+1)} \\ +&\text{C4(i,j)} * \text{h(i-1,j)} + \text{C5(i,j)} * \text{h(i,j)} + \text{C6(i,j)} * \text{h(i+1,j)} \\ &+ \text{C8(i,j)} * \text{h(i,j-1)} \\ &= \text{C10(i,j)} \end{aligned} \tag{7}$$

In the assembly, use is made of the symmetries arising from the conservation nature of the fluxes (e.g., see Roache, 1972). For example, the x-direction flux out of the cell (i) across the face at (i+1/2) is identical to the flow into the cell (i+1) across the same face, giving

$$\text{C6(i,j)} = \text{C4(i+1,j)} \tag{8}$$

Similarly in y,

$$\text{C8(i,j)} = \text{C2(i,j-1)} \tag{9}$$

By the same reasoning, the steady-state contribution to the central stencil element C5, denoted here by CS5, is evaluated as

$$\text{CS5(i,j)} = -\{\text{C2(i,j)} + \text{C4(i,j)} + \text{C6(i,j)} + \text{C8(i,j)}\} \tag{10}$$

Using these symmetries in the assembly of the matrix minimizes coding, saves computer time in the set up (which would be significant in a later extension to time-varying properties due, e.g., to subsidence), and minimizes round-off error.

A key aspect of the discretization is the evaluation of the hydraulic conductivity K at the cell interfaces, e.g., K(i-1/2, j). In principle, it could be evaluated from the original data at the location of the interface, x(i-1/2, j), found by interpolation. This approach is commonly used in continuum mechanics problems wherein the underlying physical assumption is usually that of

smooth variation of physical properties. However, in groundwater flow modeling, the usual underlying physical assumption allows for discontinuous variation of K, and K data are specified on blocks corresponding to the cells (i, j). Some interpolation to find $K(i\text{-}1/2, j)$ is required. Although any second-order interpolation would suffice, it is customary in groundwater flow modeling to use the harmonic average, defined as the product divided by the arithmetic average.

$$K_{i-1/2,j} = \frac{2K_i \cdot K_{i-1}}{K_i + K_{i-1}} \tag{11}$$

The primary motivation for use of this form (e.g., McDonald and Harbaugh, 1988, de Marsily, 1986) is that it produces the exact answer for steady Darcy flow in one dimension for piece-wise constant properties. However, for unsteady flow or multidimensional flow, this desirable property is not obtained. The method is still second-order accurate, but has no significant advantage in accuracy over linear interpolation (e.g., see Das, et al., 1994) which results in the simple arithmetic average

$$K_{i-1/2,j} = \frac{K_i + K_{i-1}}{2} \tag{12}$$

for the case of constant Δx. User options for both evaluations are provided in SECO_FLOW.

The significant distinction between the MAC = 0 and MAC = 1 options occurs in the boundary definitions. The location of the edge of the computational boundary is a conceptual modeling assumption, not determined by the code but by the analyst (user). For MAC = 0, the nodes are located on the boundary, as shown in Fig. 2a. For MAC = 1, the cell edges are located on the boundary, as shown in Fig. 2b. The latter (which is also called a block-centered discretization, or a finite volume method) is more common in groundwater flow modeling, but not universal. The MAC = 0 option treats Dirichlet boundary conditions (specified values of head) in a straightforward way, e.g.,

$$h_1 = h_{boundary} \tag{13}$$

By contrast, the MAC = 1 option requires interpolation. With the left boundary located between the first interior node h_1 and the ghost cell h_0 (e.g., see Roache, 1972) the evaluation gives

$$\frac{h_1 + h_0}{2} = h_{boundary} \tag{14}$$

for constant Δx. (Regardless of the cell spacing in the interior, the ghost cells are always defined as mirror images of the first interior cell, so that the cell spacing is constant at the boundaries.) This is the equation for the ghost cell value h_0, which is solved as part of the linear algebra of the time-stepping procedure. Thus, there is no discretization error involved in the application of Dirichlet boundary conditions for MAC = 0, but there is a second-order error for MAC = 1. The complimentary advantage occurs for Neumann boundary conditions (specified values of head gradient normal to the boundary, proportional to the flux.) In this case, MAC = 0 gives

$$\frac{h_2 - h_1}{\Delta x} = \partial h / \partial x_{boundary} \tag{15}$$

which is only first order accurate (e.g., see Roache, 1972). MAC = 1 gives

$$\frac{h_1 - h_0}{\Delta x} = \partial h / \partial x_{boundary} \tag{16}$$

which is second-order accurate. Thus, MAC = 0 has some advantage for Dirichlet conditions, but MAC = 1 is advantageous for Neumann conditions or combinations.

NOTICE

For WIPP QA purposes, the MAC = 0 option is considered inoperative in Version 3.03.

The time differencing used is the fully implicit method, which is only first-order accurate but very robust and free from overshoot. The time level is denoted by n, and the calculation advances from level n to $n+1$; where time $t^{n+1} = t^n + \Delta t$. The *RHS* of Eqn. (1) is evaluated from Eqn. (5) with all values of head at the advance time level. Then the fully implicit time-stepping equation is

$$S(i,j) \cdot (h_{i,j}^{n+1} - h_{i,j}^{n}) / \Delta t = RHS^{n+1} \tag{17}$$

Since *RHS* involves values of h not only at the location (i, j) but also at the neighboring locations $(i\pm1,j)$ and $(i,j\pm1)$ the equation is implicit, i.e., it involves the matrix solution of head h at all values of i and j simultaneously.

The fully implicit time differencing produces unconditional stability for this linear equation but requires the solution of an elliptic (Helmholtz) equation at each time step. In MODFLOW (McDonald and Harbaugh, 1988), the linear, elliptic equation is solved by Strongly Implicit Procedure or planar successive over-relaxation (SOR) iterative methods. Other common solution methods include point SOR, two-line SOR, and direct solution by (banded) Gaussian elimination. The direct solver is not considered to be practical for a realistic grids (sufficiently fine resolution), being excessively sensitive to computer round-off error and very slow. In SECOFL2D, the solver options are point or (single) line SOR (e.g., see Roache, 1972), and the semi-coarsening multigrid solver MGSS2, which was developed at Ecodynamics (Schaffer, 1995; see also Dendy et al., 1989,1992).

The semi-coarsening multigrid solver (MGSS2) is the default option. For very coarse resolution (e.g., a 6×6 grid that might be used for development of code enhancements), the point SOR solver is fastest. However, MGSS2 results in significantly increased efficiency for problems with fine resolution and strongly varying conductance (due to either hydraulic conductivity variations or stretched grids). Further, MGSS2 removes from the user the burden of searching for an optimum relaxation factor.

It is not necessary to "actualize" the Darcy velocity in order to solve for the heads, as shown above. The flux component arrays at the cell trailing faces are calculated during the iterative solution for head as

$$flux\_u(i,j) = c4(i,j) \cdot \left(h(i-1,j) - h(i,j)\right) \tag{18}$$

$$flux\_v(i,j) = c8(i,j) \cdot \left(h(i,j-1) - h(i,j)\right) \tag{19}$$

where $c4$ and $c8$ are the stencil coefficients corresponding to the trailing cell faces. The use of these stencil elements in the calculation of fluxes, rather than reconstruction of the algebra of the discretization, assures compatibility with the head solution regardless of harmonic vs. linear (or other) transmissivity evaluation, unconfined flow transmissivity factors, etc. Then the Darcy velocities (or specific discharges) are calculated from the fluxes as

$$uscf(i,j) = flux\_u(i,j) / \left(ady(j) \cdot aq\_thick(i,j)\right) \tag{20}$$

$$vscf(i,j) = flux\_v(i,j) / \left(adx(i) \cdot aq\_thick(i,j)\right) \tag{21}$$

where *ady(j)*, *adx(i)* and *aq_thick(i,j)* are the Fortran variables for Δy , Δx, and aquifer thickness A_T. Likewise, these fluxes are used to calculate flow out of the region along each boundary, with second-order quadrature consistent with the head equation.

The 2-D discrete equation (7) using the harmonic average option is equivalent to a single layer of the 3-D equation solved in MODFLOW (McDonald and Harbaugh, 1988) or SWIFT II (Reeves et al., 1986) with the aquifer thickness A_T replaced by the vertical cell thickness $\Delta z(i,j,k)$. The small angle approximation is implicit in this formulation when x and y are taken as horizontal coordinates and Δz varies with x and/or y (as is common), since area and volume effects are accounted for but not the curvature terms of a full coordinate transformation. This is clear from the fact that the formulation in a vertical plane would result in only a 5-point stencil (or 7-point in full 3-D) whereas a full coordinate transformation must necessarily include cross-derivative terms and would result in a 9-point stencil in a 2-D vertical plane (19-point in full non-orthogonal 3-D grids). The small angle approximation is an intrinsic part of a (quasi) 2-D formulation, allowing the inclusion of effects due to variable aquifer thickness (and elevation) in a two-coordinate description rather than a full 3-D description. The approximation is "ordered" not in mesh increments Δx and Δy but in the geometric properties (thickness and dip) of the aquifer itself. Thus, it is properly a *conceptual modeling assumption* rather than a numerical approximation *per se*. It is worthwhile to recognize that, in a full 3-D code, this approximation does not vanish in the limit of $\Delta z \rightarrow 0$. Thus, typical uses of MODFLOW and SWIFT II also are limited by the small-angle approximation. The full 3-D SECO_FLOW code (Knupp and Askew, 1995) includes all the terms of the full transformation. Note, however, that the geometric properties for the WIPP studies (Howery et al., 1990; WIPP PA, 1992; Corbet, 1994; Helton et al., 1995) are well within this approximation.

3.3 Quality Assurance Considerations for the Multigrid Solver Module

MGSS2 is a very complex module, perhaps more so than the rest of SECOFL2D combined. (It is not nearly as long as the rest of SECOFL2D, but it involves much more complex algorithms and is much less transparent.) As with any elaborate code, the verification process is complicated

by interaction of options; the required number of verification cases generally increases geometrically with the number of options. Even though MGSS2 has been exercised on a variety of problems, the nature of the multigrid, semi-coarsening algorithm makes it possible for non-convergence to appear in particular grid configurations. For one case exercised so far, which involved a rather extreme variation of properties, an earlier version of MGSS2 failed to converge when the number of grid points in the *j* direction was even.

For this or any other case in which the multigrid solver might fail to converge, there is no danger of an incorrect answer being produced because of the following strategy employed for SECOFL2D and all codes developed or modified at Ecodynamics. The efficient but complex solver (including multigrid MGSS2 in any later versions and any other possible solver like preconditioned conjugate gradient methods, etc.) is used only to provide an initial estimate of the solution to the point SOR solver. That is, no matter what efficient solver is used, it is followed by a call to the point SOR solver. If the multigrid solver is working correctly, the SOR solver will simply verify the solution (and incidentally perform an additional smoothing iteration) and then exit. If the multigrid solver is not working correctly, the solution can still be obtained by the point SOR solver.

The cost of this procedure is minimal. If there are no difficulties the cost is one additional "work unit," whereas the typical multigrid run requires 25 to 29 work units. (One work unit is the cost equivalent of one relaxation sweep on the finest grid level.) The cost, then, is 3 to 4% in solver time and less for the overall program run time.

The advantages of this procedure are significant.

(1) No separate verification or documentation is required for the multigrid solver MGSS2 (or for any other optional solver). Our QA philosophy does not insist on code robustness or efficiency but only requires that the codes do not produce unreliable output (Rechard et al., 1991). If the multigrid solver were to fail, it would only increase run time or, at the very worst, prevent a run from being completed because of stop checks in the SOR solver (e.g., maximum number of SOR iterations, etc.).

(2) The internal iterative convergence test that is built into MGSS2 and is natural to the semi-coarsening multigrid algorithm is based on the L2 norm of the residuals (Schaffer, 1995). The more conservative and more easily interpretable criterion used in the SECOFL2D SOR solver is based on the L-infinity (maximum) norm. These two criteria are not related directly. (For easy problems, there is a reasonably sharp bound on the difference between the two norms, but it becomes less sharp for large property variations, e.g., the 5 order-of-magnitude variation in conductivity considered for WIPP PA calculations; see WIPP PA, 1992; Helton et al., 1995.) Also, the point SOR solver in SECOFL2D incorporates more than usual sophistication in the scaling of the convergence test. The post-processing of the multigrid solver output by the SOR routine obviates the need for any further consideration of convergence criteria and makes the results obtained using different solvers easily comparable.

The point SOR solver option also has an option for adaptively adjusting the SOR relaxation factor *rf*. The algorithm is based on that in the BOAST code (Fanchi et al., 1982) which itself is based on that in Young (1971), modified herein by normalizing the successive ratio test, by requiring a minimum of 4 steps before adjusting *rf*, by smoothing the value of the successive *rf* ratio to damp oscillations before estimating the new spectral radius, and by limiting *rf* by user-accessible min and max values. The algorithm is not described in detail herein because (1) the semicoarsening multigrid algorithm has proven so robust and is so much more efficient that SOR

is only used for verification (see above), (2) the algorithm is not very effective in cases of moderate to high resolution, which is when it would be most useful, and (3) in any case, it is not used in the WIPP PA calculations.

NOTICE

For WIPP QA purposes, the adaptive SOR relaxation factor is considered inoperative in Version 3.03.

3.4 Governing Equations and Discretization for Unconfined Flow

A modification to the partial differential equation solved for hydraulic head, h, is required for unconfined conditions in which the head is below the top level of the cell. Following the common formulation (e.g., de Marsily, 1986, Section 5.1; Anderson and Woessner, 1992. Chapter 2), Eqn. (1) is modified by (a) replacing the S of confined flow with the corresponding term for unconfined flow, the specific yield (or drainage porosity) S_Y, and (b) by the inclusion of the transmissivity factor T_F.

$$S_Y \, \partial h / \partial t = \nabla \cdot (T_F \, A_T \, K \nabla h) \tag{22}$$

Numerically, the change from S to S_Y is inconsequential. Only if highly compressible porous media were considered, so that S_Y in Eqn. 22 were replaced by $[S_Y + (h - z_{bot}) S_S]$, would the numerical procedure be affected by the additional nonlinearity. For most applications, including WIPP PA, the S_S term is negligible compared to S_Y. In all cases, only the transient response would be affected, not the steady-state solutions.

T_F for a computational cell is simply the fraction of the cell vertical thickness (aquifer thickness) occupied by water. $T_F = 1$ for confined flow. For unconfined flow, it represents the reduction in flow area across a cell interface caused by the fact that the head is below the top of the cell (aquifer). The inter-cell volume fluxes driven by head (pressure) gradients as obtained from Darcy's law must be reduced by T_F appropriately averaged at cell faces. For example, if the head location is at half the height of the cell, then $T_F = 1/2$ for the cell. T_F for the cell is determined by the following equation in which z_{top} and z_{bot} are the vertical coordinates of the top and bottom of the cell ($z_{top} - z_{bot}$ = aquifer thickness).

$$T_F = \frac{(h - z_{bot})}{(z_{top} - z_{bot})} \quad for \;\; z_{bot} < h < z_{top} \tag{23}$$

Although ostensibly a simple modification of the confined flow equation, in fact the use of the transmissivity factor involves subtle approximations and serious ramifications.

- This formulation is based on the Dupuit hypothesis (e.g., see De Marsily, 1987 or any other standard groundwater flow text) that the aquifer flow is essentially horizontal, i.e., the small angle approximation. Although a legitimate assumption for many aquifer flows including the Culebra in the WIPP PA, it is not strictly compatible with a rigorous formulation in which the free surface is tracked exactly with non-orthogonal coordinates and in which derivatives with respect to the vertical coordinate (i.e., vertical flows) are included. The approximation would fail if the free surface varied rapidly in x or y, even if the aquifer itself were horizontal and constant thickness.

Wherever the vertical flow component is not negligible, e.g., at locations near seepage faces, the full 3-D equations would be required as in SECO_FLOW_3D (Knupp and Askew, 1995). (Note that MODFLOW [McDonald and Harbaugh, 1988; McDonald et al., 1991), HST3D (Kipp, 1986) and SWIFT II [Reeves et al., 1986] do not treat the free surface equations rigorously even in their 3-D codes.)

- T_F depends on h, therefore the modified equation is nonlinear. This nonlinearity requires (outer) iteration to solve if T_F is evaluated (as are the other terms in the RHS) at the advanced time level $n+1$, even if a direct solver was used for the unmodified equation. The nonlinearity also raises questions of nonuniqueness.

- Incorrect evaluation of T_F leads to serious non-physical behavior in the case of cell de-watering (see Section 3.5 below).

3.5 Cell De-Watering

Cell de-watering occurs if $h \le z_{bot}$ develops, due to pumping, etc. In SECOFL2D, if a cell de-waters so that the transmissivity factor becomes zero, the cell can later recharge and the transmissivity factor will reactivate.

This ability of the SECO_FLOW code corrects an error found in (at least) two commonly used groundwater flow codes (the MODFLOW code, McDonald and Harbaugh, 1988, and the SWIFT II code, Reeves et al., 1986). The inter-cell volume fluxes driven by head (pressure) gradients as obtained from Darcy's law must be reduced by T_F appropriately averaged at cell faces. Inter-cell conductance terms (the product of conductivity K and the cell face area) are typically obtained by harmonic averaging. The error comes from applying harmonic averaging not only to conductances but also to the transmissivity factors. This is not simply a matter of choosing between two different but consistent discretizations; the harmonic averaging of transmissivity factors is wrong.

This is easily shown by considering a de-watered cell, for which $T_F = 0$. Then application of harmonic averaging gives $T_F = 0$ at all the faces of that cell, regardless of T_F values in neighboring cells. Now all conductances into the cell are zero, and the head (pressure) can never change. Even though neighboring cells might develop arbitrarily large heads, no flow can occur into the cell, which is now *permanently* de-watered. The rational and simple solution is to allow the harmonic averaging option only for the other terms in the conductances, but to use only linear averaging for the transmissivity factors.

This simple approach avoids the necessity of the two non-ordered approximations of the later MODFLOW version (requiring a 100 page documentation in McDonald et al., 1991) or the artifice of limiting T_F to a minimum of 0.1 as used in SWIFT II (Reeves et al., 1986). The SECO_FLOW method can be shown to enhance accuracy for small but non-zero values of transmissivity factor.

The non-linearity of the unconfined flow equations requires (outer) iteration to solve. Also, the code structure is designed with the option available to re-evaluate all the stencil arrays of Eqn. (7) at each time step, with the view toward future code maintenance including possible inclusion of subsidence causing time-dependent properties of permeability, porosity, and/or aquifer thickness. The outer iteration is performed with the common form of under-relaxation (by the

factor RELF < 1) between the previous (n+1) head values FOLD and the new values (before under-relaxation) FNEW, as in the following pseudo-code segment.

$$\text{FNEW(I,J)} \leftarrow \text{RELF} * \text{FNEW(I,J)} + (1.0 - \text{RELF}) * \text{FOLD(I,J)} \tag{24}$$

The (outer, nonlinear) convergence test is performed *before* the under-relaxation step as

$$\text{IF (|FOLD - FNEW| < CONV) THEN convergence accepted} \tag{25}$$

Thus, no false indication of convergence can be caused by small RELF (Roache, 1972; Ferziger, 1993). The convergence tolerance CONV is user-accessible.

NOTICE

For WIPP QA purposes, the unconfined flow option is considered inoperative in Version 3.03.

3.6 Lake/River Conductances and Constant Head Regions

The contribution of surface water features (lakes/rivers) to the groundwater flow is implemented following MODFLOW (McDonald and Harbaugh, 1988, Chapter 6) and SWIFT II (Reeves et al., 1986). When the aquifer head is above the (effective) river bottom, then the river drains the aquifer at a rate proportional to the difference in heads; the proportionality constant is defined as the river conductance, denoted here as C_{RIV}. This adds to the continuity equation a source/sink term Q_{RIV} evaluated as

$$Q_{RIV} = -C_{RIV} \cdot (h_{i,j} - h_{RIV}) \tag{26}$$

where h_{RIV} is the specified river head and $h_{i,j}$ is the head in the aquifer cell. (Note h_{RIV} may be made time-dependent through the use of the climate factor routines.) However, when the aquifer head is below the (effective) river bottom, the flow is from the riverbed into the aquifer, but it is not realistic to set the flow proportional to the difference in heads as above; rather, the flow rate is limited by the elevation at the bottom of the riverbed, denoted here by R_{BOT}.

$$Q_{RIV} = -C_{RIV} \cdot (h_{RIV} - R_{BOT}) \tag{27}$$

The algebraic formulation is trivial and coding implementation is straightforward; but the conceptualization requires insight. The burden is on the modeler to determine the effective riverbed elevation R_{BOT} and especially the conductance term C_{RIV}. The discussion in the MODFLOW manual (McDonald and Harbaugh, 1988, Chapter 6) is recommended. (For the variable-density or saline water option, the elevation test for aquifer head relative to R_{BOT} is based on environmental head, whereas the flow rates are driven by freshwater head differences, leading to some inconsistency. See Section 5 below.)

The SECOFL2D code also contains an input option for specifying constant head. In an earlier (developmental) version, this was implemented as in MODLFOW by an integer flag to designate a constant head cell. This implementation was replaced because an integer flag does not interpolate correctly, e.g., between a regional and local grid. The present implementation is simpler and interpolates correctly. The implementation uses the same algebra and code segments as the

river/lake conductance option. The user-specified constant head value is set to the Fortran variable for R_{BOT} and the code assigns a very large number (1.E+10) to C_{RIV}. This has the effect of driving $h_{i,j}$ towards R_{BOT}. In practice, $h_{i,j}$ is constant to at least the significant figures given in the output files.

NOTICE

For WIPP QA purposes, the lake/river conductance and constant head region options are considered inoperative in Version 3.03.

4. CONFIGURATION OPTIONS

SECOFL2D permits configuration of initial conditions, boundary conditions, climate factors, surface water features (lakes/rivers), and wells. A description of the SECOFL2D configuration options follows.

4.1 Initial Conditions

For regional grid simulations, initial conditions on hydraulic head may be specified: (1) by using the values set in the aquifer-defining grid; or (2) by solving the steady-state problem with the specified boundary conditions and all wells turned off, a typical option for groundwater modeling in the absence of extensive field data.

For the local grid simulations, initial conditions are obtained from the (interior) regional grid solution by linear interpolation in time, by reading the output files from the regional grid solution. The algorithm uses simple linear interpolation in time, which is second-order accurate (higher than the interior time differencing scheme). The procedure used allows small time extrapolation (beyond the last time level calculated in the regional grid) in order to avoid possible premature end-of-file encounter due to round-off error discrepancy between the local-grid last simulation time and the regional-grid last output time.

4.2 Boundary Conditions

Unlike most groundwater hydrology codes, SECOFL2D allows a fairly general specification of boundary conditions. For regional grid simulations, the SECOFL2D boundary conditions can be one of the following types: Dirichlet (fixed head), Neumann (fixed gradient, user-specified either as a head gradient or as a flux), Robin (mixed, or linear combination of Dirichlet and Neumann), or adaptive, which sets specified flux at inflow boundaries and specified head at outflow boundaries (a nonlinear condition). For all of these, the values may be either user-specified as input values, or may be taken from initial condition values.

These types of boundaries may be set independently along each of the four rectangular boundaries of the grid or along an arbitrary number of user-specified sections on each boundary. Following the basic philosophy of the SECO codes, the specification of these boundary sections is done in the continuum rather than being tied into the discretization. In particular, sections of specified-gradient boundaries can be used to simulate recharge boundaries; these values can be modified by a quasi-climatic variation in time.

Constant-head regions may also be set on interior regions, as can time-independent wells and lake/river levels, which differ from simple constant-head regions in that they affect the cell block heads via a riverbed conductance term. (The formulation follows MODFLOW; see McDonald and Harbaugh, 1988.) The specification of these interior boundaries is not automated at present: the user must specify each interior boundary on a cell-by-cell basis in the aquifer-defining grid, as is the case with other aquifer properties. However, once established, these values can be used without further user specification in any regional or local grid. In this sense, the interior boundaries are still defined independently of the discretization of the computational grids.

For the local grid solution, the code sets possibly time-dependent boundary conditions by reading the regional-grid solution output for head in time and interpolating in space and time. As is the case for the local grid initial conditions (see above), the algorithm uses simple linear interpolation in time and allows small time extrapolation in order to avoid possible premature end-

of-file encounter due to round-off error discrepancies. The spatial interpolation is also linear and second-order accurate. The property data (e.g., transmissivity) is obtained by conservative interpolation using the Dukowitz-Knupp algorithm (Dukowitz, 1985; Knupp, 1990). Thus, the local grid boundary conditions are usually Dirichlet (specified dependent variable) values of head (time-dependent values for the time-dependent regional grid problem). The result is that Darcy velocities (specific discharges) are preserved. That is, the Darcy velocity normal to the local grid boundary is algebraically the same as would be obtained by trigonometric projection of the Darcy velocity components in the regional grid in the direction of the local grid boundary normals.

This is a nice property, but we stress that it is not essential to the modeling. The use of the regional-local grids is a modeling concept to improve the far-field boundary conditions for the local grid simulation. Like other modeling concepts, it is not an "ordered approximation" (Roache, 1972, 1993a,b,c,d, 1994a,b; 1995a; Westerink and Roache, 1995), i.e., the error does not vanish as the grid increments approach zero. For example, one could also consider any approximate solution obtained on a wider area than the local grid, e.g., an analytical solution obtained using constant properties. The only "assumption" involved is that a coarse grid solution over a much larger area provides better (less arbitrarily constrained) boundary conditions for the local grid than the usual hydrogeologic practice of setting no-flow boundary conditions (at other than a true groundwater divide). The method used here has obvious advantages.

Although not an ordered approximation, the method has been tested and exercised in many calculations. A revealing test involves over-plotting head contours in the same region of space obtained by both the regional and local grid solutions. When the grid increments in both regional and local grids are comparable, the contour plots virtually overlay, as expected. When the local grid resolution is much higher, as intended in the concept and as used in the WIPP PA calculations (WIPP PA, 1992), there is some difference (as hoped, otherwise the whole procedure would be useless since virtually the same results could have been obtained with a single coarse grid calculation in the regional grid) but not any gross discrepancy. Most importantly, the head contour lines in the local grid solution vary smoothly and align with the regional grid solution at the boundary of the local grid, indicating no distortion from the interpolation procedure (again, as expected from the theory). It is also worth noting that transport calculations in the 1992 and later WIPP PA calculations do not extend across the local grid boundaries, i.e., the regulatory boundaries are inside the local grid. Thus, there is no question about the boundary conditions for the transport equations. (Note, however, that we have exercised the option of continuing particle tracking via SECO_TRACKER through the boundary, and again, the trajectory plots transitioning through the local grid boundary are well behaved.)

4.3 Quasi-Climate Factors

SECOFL2D includes an option for inclusion of quasi-climate factors on the boundary conditions. By "quasi-climate" we mean time-dependent factors which may be used with a climate model to investigate sensitivity of the solutions to an expected range of climatic variations. The qualification occurs because no attempt is made to model the detailed mechanics of climatic variation, e.g., increased rainfall recharge through distributed source terms (varying in space as well as time) and interacting with the aquifer by leakance terms through poorly characterized overburden. This level of modeling has been achieved (Corbet, 1994) with full three-dimensional basin-scale modeling using the SECOFL3D code (Knupp and Askew, 1995). For the two-dimensional simulations, the much simpler quasi-climatic modeling was used to

estimate a general level of sensitivity of the simulations to climate variations through time-varying boundary conditions.

Climate-factor subroutines are used to parameterize the quasi-climatic variability of the simulation. These subroutines apply four separate factors that are functions of time to (1) each lake/river head, (2) flux boundaries (Neumann conditions or specified head gradients), (3) head boundaries, both interior and at the edge of the region, and both Dirichlet (specified head) and Robin (mixed Dichlet-Neumann) boundary conditions, and (4) to internal recharge wells.

Consistent with the basic philosophy of the codes, these climate factors are defined in terms of the continuum time and are not tied into any time discretization or time simulation. (That is, the factors are defined in terms of absolute time regardless of the time period of the simulation. However, "hooks" are provided in the subroutine argument lists to allow a user to tie in the climate factors with the simulation time intervals if desired.) The generic climate-factor subroutines, which are easily modified, provide a simple, sinusoidal variation with user-specified period, amplitude, and phase shift (allowing the recharge boundary flux factor to lag the lake/river head factor by a seasonal shift, for example). The generic default form used for all four climate factors is

$$CF = 1 + amp \cdot \sin(2\pi \cdot cycle \cdot t_{NORM}) \tag{28}$$

where CF is the climate factor, amp is the modeled amplitude of the variation, cycle is the modeled length of the climate cycle, and t_{NORM} is the simulation time normalized by a user-defined value including the phase shift. The time interval of the simulation is *not* used as the period for the climatic variation, in order to avoid a user trap; the lake/river subroutine can be called for both the regional simulation and for the local simulation, and these may have different simulation time intervals, yet the climate factor must be the same at the same simulation times. Also, to avoid inconsistencies with steady state solutions, it is advisable (but not strictly necessary) to set the climate lake/river factor = 1 at time = 0 and for steady state solutions.

If the climate lake/river factor is small enough to give

$$lake\ /\ river\ head \times CF \langle\ head \tag{29}$$

where *head* is the dependent variable of the simulation, then flow from the aquifer to the lake/river bed will result. If this simulation is judged unrealistic, then the user must take appropriate action to limit the climate factor.

These quasi-climate features were used in previous WIPP PA calculations (WIPP PA, 1992; Helton et al., 1995) but are not expected to be used in the final Compliance Certification Application (CCA) for WIPP. Because of their previous use, the default condition in the CAMCON version sets these features on. If, as anticipated, they will not be used in the final WIPP CCA, the analyst will turn them off. (See second example problem in Section 8.2.)

NOTICE

For WIPP QA purposes, the options for quasi-climate factors are considered inoperative in Version 3.03.

4.4 Wells

SECO_FLOW provides two distinct approaches to well modeling: a parameterized model and an arbitrary direct time history model.

4.4.1 Parameterized Well Models

Parameterized well models use a generally small number of parameters to calculate a continuous function of time for the well rate. The default nominal well specifications are set in DATA statements in modular code segments. In the stand-alone version, the user can change the well specifications through the input file, by user-interactive input, by using DEPOSIT commands (see Chapter 7 of this manual), or by writing customized source code modules. In the CAMCON version, the user can set these parameters only by way of the preprocessor input file, for reasons of QA Analysis control and documentation.

The number of wells, the well schedules, the completion depths, and the well locations are each separately defined in separate modules. Thus, once a well is defined in terms of its nominal schedule (pumping rate versus time), it can be independently located in the region. Following the basic philosophy of the SECO codes, the specification of these well locations is done in the continuum rather than being tied into the discretization. The codes then search for the cell location corresponding to the continuum location. Since no local well analytic solutions are presently used, a well is only located to within the cell discretization; effectively, this places the well location at the center of each cell. Thus, the simulation only locates the well to first order spatial accuracy. Rigorous second-order spatial accuracy can still be maintained by the user during a grid convergence test by redefining the wells for each grid refinement, distributing the well terms over the same physical space and location as the coarse grid simulation. For example, in a grid doubling a single well term can be divided into four wells, each with 1/4 the strength, in each of the four cells covering the same continuum space as the single cell in the coarse grid (Salari et al., 1995). However, the process is not automated in the SECO_FLOW codes. Also, note that the presently documented version of SECO_FLOW does not include the more realistic nonlinear well modeling in which the pumping rate depends on bottom-hole pressure (e.g., see the SWIFT II code of Reeves et al., 1986).

In earlier versions of the SECO codes, the user confirmed that the aquifer type definition (confined = artesian, unconfined = water table) associated with the well was consistent with its location within the aquifer. In the present codes, this task is automated. Note, however, that the well specification defines only a *nominal* pumping rate for conditions in which the head stays above the well completion depth. If the head drops below the well completion depth, the *actual* rate of pumping from the aquifer becomes zero, and the water budget for the aquifer cannot be calculated by simply integrating the nominal well specification of pumping rate. This feature (that the actual well pumping rate is turned off automatically if the cell block head drops below the completion depth) and the transmissivity factor calculation for unconfined flow are the only aspects of the flow solution that depend directly on the head level rather than simply on gradients. If the cell block head later returns to a value higher than the completion depth, the well will reactivate at its nominal rate.

The default generic nominal well schedules are of the ramp type. The flow rate varies linearly from zero at a start time to a specified level, which then holds constant until a specified time and then shuts down linearly. See Figure 3.

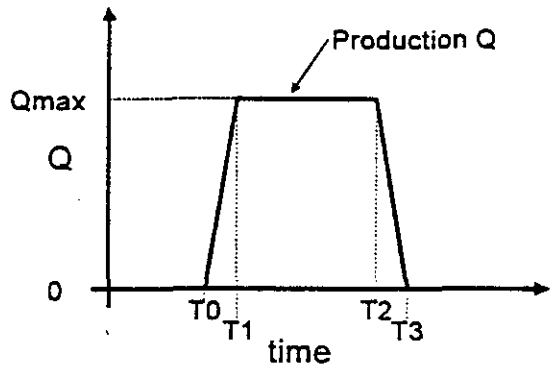


Figure 3. Generic well schedule. This well model is ramped on from T0 to T1 seconds to a value of Qmax, and ramped off from T2 to T3. For steady-state simulation, Qmax is used.

For transient simulations, the code calculates the source term *not* by sampling instantaneous well rates Q as defined above, but by the time integral of Q from time = 0 to the simulation time. This greatly improves accuracy for poorly time-resolved simulations. In the extreme case, a well could be ramped on and off within a single Δt; this would give Q = 0 and no effect if the instantaneous value of Q were used, but gives a reasonable effect when the integrated value non-zero Q is used. For steady-state calculations, the value of Q_{MAX} is used.

By suitable selection of the ramp times T0 - T3, the user can set constant well rates, linearly increasing well rates, etc. The default subroutines are readily modified to change the functional form of the well schedule. Alternately, the analyst can use the default ramp schedules to define different wells at the same location, and their effects will be additive; an arbitrary well schedule can thus be approximated by composite ramp schedules.

4.4.2 Direct Time-History Well Models

The CAMCON version of SECO_FLOW also permits an arbitrary direct time history well model to be defined by an input history file of a single integrated well rate vs. time. The code interpolates linearly as needed during the simulation. This generality may be needed for the modeling of the source term from the intrusion boreholes in the WIPP PA.

NOTICE

For WIPP QA purposes, the well options are considered inoperative in Version 3.03.

4.5 Time-Step Specification, Error Estimator, and Solution Adaptivity

In typical use in WIPP PA calculations (WIPP PA, 1992; Helton et al., 1995) the time-steps for SECO_FLOW are defined in elementary fashion as fixed Δt. The code has five options for defining the time sequence: (1) automatically set simulation times by constant time steps Δt, defined by the total simulation time and the user-specified number of time steps n_{timei}, (2) automatically set simulation times for a specified n_{timei} by power-law packing of Δt around some user-specified event time (e.g., a borehole intrusion time), (3) user-specified sequence of Δt, (4)

user-specified sequence of simulation times t, and (5) solution-adaptive time stepping in which the total number of time steps resulting is not predetermined.

The option for solution adaptive time-stepping is based on a temporal error estimator (Roache, 1993a,b). This inexpensive time error estimator uses the difference between backward and forward time integration, implemented as an extrapolation. The method is very cheap to implement because it does not require another implicit matrix solution, nor even another explicit stencil evaluation. It includes the effects of time-dependent boundary conditions and wells.

In the current time step, Eqn. (17) is advancing the solution for h from time level n to time level $(n+1)$ with increment Δt using fully implicit (backward) time differencing, so that the *RHS* is being evaluated at $(n+1)$. In the previous time step, the solution was advanced from $(n-1)$ relative to the current indexing to n with increment Δt_{OLD} and the *RHS* evaluated at n. We could explicitly evaluate RHS^n and make a separate, parallel estimate of the values h^{n+1} with an explicit step, as in

$$S(i,j)\cdot(\hat{h}_{i,j}^{n+1} - h_{i,j}^{n}) / \Delta t = RHS^n \tag{30}$$

where $\hat{h}$ signifies the new value predicted by the explicit algorithm. (Since the equation involves values of $\hat{h}$ only at the location (i, j) but not at the neighboring locations $(i\pm1,j)$ and $(i,j\pm1)$ the equation is explicit, i.e., it does not involve a matrix solution of head h at all values of i and j simultaneously.) The difference between the new value of h^{n+1} predicted by the implicit algorithm and $\hat{h}^{n+1}$ predicted by the explicit algorithm, both of which predictions are $O(\Delta t)$ accurate, is itself an error estimator of accuracy $O(\Delta t)$ for the time discretization error for that time step. Explicit evaluation of RHS^n would not be expensive compared to the computer time necessary for the implicit matrix solution, but it does involve coding storage penalties and complexities (storage of old values of boundary conditions, well terms, etc.) An economical and elegant approach is to recognize that the *RHS* from the previous implicit step is identical to the *RHS* for the present explicit step. Thus, the *RHS* for the present explicit step can be evaluated from the knowledge of the previous change in h, requiring only the temporary storage of previous solution arrays. The explicit solution for head at $(n+1)$ is then obtained by simple linear extrapolation of previous solutions; for constant Δt,

$$\hat{h}_{i,j}^{n+1} = 2h_{i,j}^{n} - h_{i,j}^{n-1} \tag{31}$$

For the more general case of variable Δt,

$$\hat{h}_{i,j}^{n+1} = h_{i,j}^{n} + \frac{\Delta t}{\Delta t_{OLD}}(h_{i,j}^{n} - h_{i,j}^{n-1}) \tag{32}$$

Finally, the error estimator E_{ET} is calculated for the time discretization error for heads for the single time step.

$$E_{ET} = 100\cdot \max_{I,J}\{abs(\hat{h}_{i,j}^{n+1} - h_{i,j}^{n+1})\} / h_{RANGE} \tag{33}$$

E_{ET} is thus calculated as the % maximum deviation (or L^{∞} norm) of the absolute value of the difference between the new head values h^{n+1}, predicted by the fully implicit algorithm, and $\hat{h}^{n+1}$ predicted by the explicit algorithm, normalized by h_{RANGE}, which is the total range of h^{n+1}. Ghost point evaluations with Dirichlet boundary conditions tend to exaggerate the error (though slight and smooth) so error estimates and h_{RANGE} are calculated only over interior points. E_{ET} is included in output files so that the user can inspect it for time accuracy.

Note that the explicit calculation is used only as an error estimator within a time step of an implicit method, not as the solution algorithm, i.e., its effects do not accumulate. Thus, stability limitations and/or conservation issues of explicit time stepping as a solution algorithm are irrelevant.

Although the extrapolation procedure is equivalent to explicit time-stepping, the extrapolation cannot be started until there are two time levels. Also, in the event that the initial conditions are set arbitrarily by the analyst (without setting initial conditions as a steady-state solution), it is likely that the initial conditions are incompatible with the boundary conditions applied at the first time step. This means that the change in boundary values during the first time step is fixed, i.e., does not depend on the time resolution. Consequently, the first time step would not provide a meaningful estimate of $\partial h/\partial t$ and the error estimator would be invalid. Also, for a local grid solution, the initial time may or may not be interpolated from such an inconsistent initial condition, depending on the time schedule for the local solution, and that information for the regional grid solution is not available. Therefore, in the local grid solution procedure, the conservative assumption is made, and the error estimator is not activated until the fourth time step.

E_{ET} is also used internally as the basis for the option (not used in WIPP PA) for a solution-adaptive time-stepping algorithm. The code can adjust Δt so that the *error_estimate* = E_{ET} is acceptable. The term *error_tol* is the acceptable level of the error-estimate before adaptation is activated. The algorithm is best explained by the following pseudo-code segment.

```
      IF (error_estimate .gt. error_tol) THEN
c     IF ... GOTO is a WHILE loop.
c       Recalculate the last time step with multiple steps.
        nad2 = 1 + (1 + nad2) * (error_estimate / error_tol) * error_sf
        nad2 = min (nad2, nad1mx)
        Δt = Δt / nad2
        GOTO 950
      ENDIF
   ...
   ...
950          CONTINUE
```

The term *nad1mx* is a user-accessible limit on the maximum number of intra-time step adaptations. The algorithm could be set up to give new *error_estimate* = *error_tol*. To compensate for the $O(\Delta t)$ accuracy of the error estimator, there are two safety factors built into the coding. The first is the use of integer values of *nad2*. The second is the parameter *error_sf*, set in a user-accesible DATA statement to 1.3. If *error_sf* = 1, then the number of adaptive time steps would give the new *error_estimate* = *error_tol* if *nad2* were real.

NOTICE

For WIPP QA purposes, the solution-adaptive time-stepping option is considered inoperative in Version 3.03.

4.6 Code Verifications

The numerical accuracy of the SECO_FLOW code has been verified on model problems. The flow codes experimentally exhibit the expected $O(\Delta x^2, \Delta t)$ accuracy. None of the time-dependent aspects are claimed to be more than first order accurate. Coding errors which might destroy first-order accuracy would be expected to produce gross errors in the results which would show up in qualitative behavior errors in the solution, e.g., a sign error. The user-defined time schedules for well activity are not used in the WIPP PA calculations, and therefore their verifications need not be documented in this manual. The only complexity in time-stepping is involved in the coding for the solution-adaptive time-stepping algorithm. Since all such algorithms are heuristic, their detailed performance is not critical; regardless of whether the adaptive algorithm works in detail as intended, the solution is still $O(\Delta x^2, \Delta t)$ accurate and the adequacy of the solution accuracy must be determined by grid (space and time) convergence tests.

The verification of the order of spatial accuracy is not trivial. Details are given in Appendix A, which is a reprint of Roache et al. (1990). As seen therein, other codes which are nominally second-order accurate fail rigorous tests for some conditions. The verification of the order of temporal accuracy is straightforward; results are presented in Appendix B.

5. BRINE DARCY FLOW IN TERMS OF FRESHWATER HEAD

The use of hydraulic head, rather than pressure, as the dependent variable in constant density Darcy flow codes has advantages, not the least of which is that head can be measured directly in the field. Also, head being the sum of pressure and elevation, it is the gradient of head, not pressure, which causes flow.

However, for a variable density code, e.g., including brine (more properly, variable unsaturated saline concentration), pressure is used virtually universally. The well-known paper by Lusczynski (1961) indicates that, even for variable density flows, the horizontal flow is driven only by gradients in freshwater head. Although true, we find this misleading (like some other theorems). It is true for 3-D flows but only for strictly horizontal coordinates in quasi-2-D codes, whereas real aquifers have some dip.

Nevertheless, we find some advantage in formulating the variable density (and variable viscosity) flow equations using freshwater head as the dependent variable (Roache, 1993a,b). This results in the addition of a buoyancy term involving saline solution to the usual Darcy equation. With z vertical (collinear with the gravity vector g), define freshwater head as

$$H_f = p/\left(\rho_f |g|\right) + E \tag{34}$$

where p is pressure, ρ_f is freshwater density, and elevation E is measured along x. Define the usual hydraulic conductivity

$$K = -k|g|\rho_f/\mu \tag{35}$$

and the density perturbation

$$e = \left(\rho - \rho_f\right)/\rho_f \tag{36}$$

Then the Darcy equation with buoyancy (saline water) using freshwater head is

$$V = -K\left|\nabla H_f + e\nabla E\right| \tag{37}$$

$$\nabla E = \left(0,0,\partial E/\partial z\right) \tag{38}$$

Saline concentration also enters into the evaluation of K through viscosity μ. The saline water continuity equation is just

$$S_s \frac{\partial H_f}{\partial t} = -\nabla \cdot V \tag{39}$$

but the specific storativity S_S depends on true (environmental) head for an unconfined aquifer. We assume S_S is not affected by saline concentration.

The elevation presently is calculated at the center of the aquifer, as

$$\text{elev} = \text{aq\_bot} + \text{aq\_thick} / 2 \tag{40}$$

but could readily be modified for unconfined flow when *head* < *aq_top* as

$$\text{elev} = \text{aq\_bot} + (\text{head} - \text{aq\_bot}) / 2 \tag{41}$$

The linear relation for density ratio perturbation is curve fit from the SWIFT-II code documentation (Reeves et al., 1986) as

$$\text{e\_rho} == (\text{rho} - \text{rho\_fresh}) / (\text{rho\_fresh}) = 0.7 * \text{con\_salt}. \tag{42}$$

The viscosity ratio (compared to fresh water) is again curve fit from the SWIFT-II code documentation (Reeves et al., 1986) as

$$\text{visc\_ratio} = 1.0 + 0.6731 * \text{con\_salt} \tag{43}$$

The combined equations (37) and (39) are convenient in the interior, but some of the appeal may be lost when boundary conditions are considered. There still is an inherent appeal to formulating the equations in such a way as to isolate the effect of a physically small perturbation. However, we stress that this is not a "small perturbation" equation, in spite of the fact that it looks reminiscent of the Boussinesq approximation. There is no approximation involved. Also, to allay the concerns of readers who know of misinterpretations caused by the Lusczynski (1961) paper, we emphasize that there is nothing mathematically significant about using freshwater head as the dependent variable, aside from convenience of interpretation. Freshwater head in this formulation is just a normalization of pressure using the density of fresh water. The density of mercury could just as well have been used, and the equations would still be valid. The formulation and coding have been verified, demonstrating $0(\Delta x^2)$ convergence towards an analytical solution, provided that some care is exercised in specifying ghost-cell values of elevation.

The convergence rate of this buoyancy formulation has been verified by Knupp and Dombroski (1993). A somewhat surprising sensitivity to a boundary formulation (not a boundary condition) was noted. When the ghost cell values for aquifer properties, including the elevation of the aquifer bottom, were evaluated by zeroth-order extrapolation, unequivocal *first*-order convergence was obtained. Only when linear extrapolation was used did the performance match the expected second-order convergence.

Consistent physics requires that the saline (or other) density perturbation be moved with the flow. A later version of SECOFL2D is planned using SECO_TRANSPORT modules (Salari et al., 1992; Salari and Blaine, 1995) for the advection, diffusion, and dispersion of the saline water.

6. COMPATIBILITY, CONSERVATION, AND CONSISTENCY BETWEEN FLOW AND TRANSPORT CODES

Compatibility between flow and transport codes is necessary in order to achieve conservation. For example, if the flow code uses different metrics for the variable aquifer thickness than does the transport code, then a flow solution for steady-state Darcy velocities (specific discharges) at cell faces will not be a steady-state solution when evaluated in the transport code, leading to artificial (but ordered) source/sink terms. ("Ordered" means that the artificial terms vanish as Δx, $\Delta y \rightarrow 0$.) The obvious method of achieving compatibility is to use consistent discretization in the flow and transport codes. Indeed, it might seem that this is the only way to achieve compatibility. However, there is another approach that is equally effective and has advantages (Roache, 1993a,b).

For a constant density code, instead of passing the specific discharges (in units of velocity [L/T], more fundamentally of volumetric discharge per unit cell area, $[L^3/(T \times L^2)] = [L/T]$) from the flow to the transport code, one passes the total volumetric discharge $Q[L^3/T]$. It is this total discharge Q that satisfies the steady volume conservation equation at each node,

$$\sum_m Q_{nm} = 0 \tag{44}$$

where the index m for each node n ranges over the connecting neighboring nodes. The transport code then reconstructs the Darcy velocities as needed. Generally, the Darcy velocities in the flow code can differ from those reconstructed in the transport code (converging in the limit of vanishing grid size, of course) yet the conservation relation is algebraically conserved in each. The concept extends readily to variable density and unsteady flows. In this manner, flow and transport codes can be developed independently.

Specifically, a flow code based on a small-angle formulation in the metrics can be used for input to a transport code such as SECO_TRANSPORT that uses the full metric evaluation. Also, a finite difference or finite volume (or pseudo-spectral!) flow code could be coupled to a finite element transport code, etc., with no generation of spurious source terms.

7. USER INTERACTIONS, INPUT FILES AND OUTPUT FILES

In order to run SECOFL2D, a preprocessor, PRESECOFL2D must first be run to setup all of the input files that are needed by SECOFL2D. This section contains the specific information required to run PRESECOFL2D including the input commands. It also contains specific information required to run SECOFL2D.

7.1 User Interactions with PRESECOFL2D

PRESECOFL2D can be exercised interactively (Section 7.1.1) or from a command line (Section 7.1.2). Regardless of the approach, the user must specify up to 10 files. A description of the 10 files follows:

Files 1-5 are files that already exist. Files 6-10 are files that are created by PRESECOFL2D. Files 6-9 are files that are used to run SECOFL2D. As the input files 1-5 already exist, they already have assigned names. The output files 6-10 have names assigned by the user.

file 1 is the input aquifer defining CAMDAT database corresponding to the sampled vector. It contains the aquifer grid and material properties.

File 2 is the input regional CAMDAT database corresponding to the sampled vector. If the region is defined on a different grid than the aquifer, this database would contain only the grid information. The material properties would be interpolated from the values on the aquifer database.

FOR ALL RUNS FOR THE 1996 WIPP PA CALCULATION, THE AQUIFER DATABASE (FILE 1) AND REGIONAL DATABASE (FILE 2) ARE THE SAME. THE SAME CAMDAT DATABASE NAME IS ENTERED FOR BOTH FILES.

File 3 is the input local CAMDAT database corresponding to the sampled vector. This database contains only the local grid as the material property data is interpolated from the regional data.

File 4 is a CAMDAT database that has time dependent well information on it. This database is necessary only if a time dependent well is being modeled.

THE SOURCE DATABASE FILE (FILE 4) IS NOT USED FOR THE 1996 WIPP PA CALCULATION. THEREFORE THE USER ENTERS "CANCEL" WHEN SPECIFYING THE SOURCE DATABASE FILE.

File 5 is the ASCII input file that controls PRESECOFL2D. The commands entered in this file direct how SECOFL2D is to be run. Section 7.2.4 contains a detailed explanation of all the commands that go into this input file.

File 6 is an ASCII input file corresponding to the sampled vector created to run SECOFL2D.

Files 7 and 8 are the input property files for the regional and local grids corresponding to the sampled vector. These are binary files that are needed to run SECOFL2D.

File 9 is an ASCII file containing the time dependent well information if a time dependent well is being modeled.

THE TIME DEPENDENT WELL DATA INPUT FILE (FILE 9) IS NOT USED FOR THE 1996 WIPP PA CALCULATION. THEREFORE THE USER ENTERS "CANCEL" WHEN SPECIFYING THE WELL DATA FILE.

File 10 is the diagnostic/debug file containing information about the PRESECOFL2D run. If errors occur during execution of PRESECOFL2D, this file can contain useful information about the errors. It is an optional file, but it is necessary that it be specified if the user wants to take advantage of the error reporting.

Some of these files are designated as optional. There are many configurations that can be used to run the PRESECOFL2D code depending on how the user intends to run the SECOFL2D code.

7.1.1 Exercising PRESECOFL2D Interactively

To execute PRESECOFL2D interactively, type PRESECOFL2D followed by a carriage return at the main menu of *CAMCON* or at the OpenVMS "$" prompt.

A banner scrolls down the screen and then the following information describing the file definitions is printed on the screen.

```
**************************************************************************
PRESECOFL2D expects the following files:
   1.  Input CAMDAT aquifer definition database (optional)
   2.  Input CAMDAT regional grid database (optional)
   3.  Input CAMDAT local grid database (optional)
   4.  Input CAMDAT source database (optional)
   5.  PRESECOFL2D input filename
   6.  SECOFL2D input (transfer) filename
   7.  SECOFL2D regional property data input filename (optional)
   8.  SECOFL2D local property data input filename (optional)
   9.  SECOFL2D time dependent well data input filename (optional)
   10. PRESECOFL2D diagnostic/debug filename (optional)
**************************************************************************
```

The following prompts for filenames appear after the above list of files is printed on the screen. The file names in the angle brackets are the default names that will be chosen if only a carriage return is entered. Type 'cancel' for no file to be chosen when a file name is optional. In this example, the CAMDAT aquifer database and the CAMDAT regional database are the same. It is not unusual for the aquifer and regional definitions to be the same.

The symbols `PRESECOFL2D$INPUT_DIRECTORY` and `PRESECOFL2D$OUTPUT_DIRECTORY` shown below may be assigned with the OpenVMS DEFINE command to be a specific directory. If they are not defined, the user must type [] to specify files in the current directory or [dir_spec], where [dir_spec] is the specification of a particular directory.

```
Enter the name of the Input CAMDAT aquifer definition database (CANCEL for no
file)
   <PRESECOFL2D$INPUT_DIRECTORY:SECO2CAMDAT_AQ.CDB> []region_test.cdb
```

```
Enter the name of the Input CAMDAT regional grid database (CANCEL for no file)
   <PRESECOFL2D$INPUT_DIRECTORY:SECO2CAMDAT_REG.CDB> []region_test.cdb

Enter the name of the Input CAMDAT local grid database (CANCEL for no file)
   <PRESECOFL2D$INPUT_DIRECTORY:SECO2CAMDAT_LOC.CDB> []local_test.cdb

Enter the name of the Input CAMDAT source database (CANCEL for no file)
   <PRESECOFL2D$INPUT_DIRECTORY:CAMDAT_SRC.CDB> cancel

Enter the name of the PRESECOFL2D input file
   <PRESECOFL2D$INPUT_DIRECTORY:PRESECOFL2D.INP> []presecofl2d_test.inp

Enter the name of the SECOFL2D input (transfer) file
   <PRESECOFL2D$OUTPUT_DIRECTORY:SECOFL2D.INP> []secofl2d_test.inp

Enter the name of the SECOFL2D regional property data input file (CANCEL for
no file)
   <PRESECOFL2D$OUTPUT_DIRECTORY:REGDAT.INP> []regdat_test.inp

Enter the name of the SECOFL2D local property data input file (CANCEL for no
file)
   <PRESECOFL2D$OUTPUT_DIRECTORY:LOCDAT.INP> []locdat_test.inp

Enter the name of the SECOFL2D time dependent well data input file (CANCEL for
no file)
   <PRESECOFL2D$OUTPUT_DIRECTORY:WELDAT.INP> cancel

Enter the name of the PRESECOFL2D diagnostics/debug file (CANCEL for no file)
   <PRESECOFL2D$OUTPUT_DIRECTORY:PRESECOFL2D.DBG> []presecofl2d_test.dbg
```

If the program completes without errors, the message

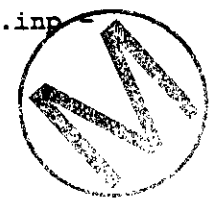
```
PRESECOFL2D Normal Completion
```

appears on the screen. If FORTRAN STOP appears on the screen, an error has occurred. The PRESECOFL2D diagnostics/debug file should be consulted to find a description of the error condition. This can be done using the editor.

7.1.2 Exercising PRESECOFL2D from a Command Line or a Command File

To exercise PRESECOFL2D from a command line, type PRESECOFL2D at the OpenVMS "$" prompt, but do not strike the carriage return key. Instead, follow the PRESECOFL2D command with the necessary filenames in the sequence indicated at the beginning of Section 7.1. (Up to 10 filenames are required [see above]; filenames not being used require the qualifier "cancel.") Use the hyphen ("-") at the end of lines on the computer screen as a continuation symbol. The Alpha computer will read it to mean "continued without break on the next line." Thus, although the command procedure is typed on several lines, because of the hyphens, the computer reads it as being typed entirely on one line.
A sample command line procedure that executes PRESECOFL2D could be:

```
$ PRESECOFL2D region_test.cdb region_test.cdb local_test.cdb cancel -
_$presecofl2d_test.inp secofl2d_test.inp regdat_test.inp locdat_test.inp
```



```
_$cancel presecofl2d_test.dbg
```

The comand line may also be placed into a command file and the command file may be executed at the command line or in batch mode. In the command file, the continuation symbol may be used, but the "_$" would not appear at the beginning of the line.

7.2 Description Of Input Files For PRESECOFL2D

7.2.1 Regional/Aquifer CAMDAT Input Files

The data required to exist on the regional (or aquifer) input CAMDAT database are listed below.

Data Name	CAMDAT Default symbol	Data type
Solid properties		
Hydraulic conductivity in x direction	Hycnd_X	Attribute
Hydraulic conductivity in y direction	Hycnd_Y	Attribute
Bulk material compressibility	CmprsBlk	Attribute
Porosity	Porosity	Attribute
Depth	Depth	Attribute
Thickness	Thick	Attribute

7.2.2 Local CAMDAT Input File

This file contains only the local grid. SECOFL2D interpolates the material property data from the regional data contained in the regional/aquifer CAMDAT input files.

7.2.3 CAMDAT Source Input File

This file contains time-dependent well information.

THE CAMDAT SOURCE INPUT FILE IS NOT USED FOR THE 1996 WIPP PA CALCULATION.

7.2.4 Input Control File

Of the five input files to PRESECOFL2D described in Section 6.0, only one of them is created directly by the user. This file is the PRESECOFL2D ASCII input control file. This file provides input not only to PRESECOFL2D, but is also the primary user interface to the entire SECOFL2D triad of codes.

This section contains a description of the valid input that can be entered in the PRESECOFL2D ASCII input file. The format of the input file follows the standard format established for CAMCON input files as stated in the User's Reference Manual for CAMCON:

Compliance Assessment Methodology Controller, Version 3.0, section 2.9.4 (Rechard, 1992). The input file contains keywords, denoted by a leading asterisk (*), and parameters associated with the keywords that can function as secondary keywords or have input values associated with them. A sample input control file is provided in Appendix A.

THE INPUT FILE DESCRIPTION THAT FOLLOWS IS FOR THE COMPLETE PRESECOFL2D COMMAND SET, OF WHICH THE COMMAND SET FOR THE 1996 WIPP PA CALCULATIONS IS A SUBSET.

Twelve keywords are used in the PRESECOFL2D input file:

1. *RUN_TYPE
2. *CONSTants
3. *ATTRibutes
4. *AQUIFER_source_terms
5. *INITial_conditions
6. *SOLVing_parameters
7. *REG_TIMEstepping
8. *LOCAL_TIMEstepping
9. *BOUNDary_conditions
10. *WELL
11. *REG_CLIMate_factors
12. *END

These keywords and any parameter keywords may be abbreviated. The required letters of the abbreviations appear in the following description as capital letters, with the unnecessary letters in lower case. When entering any keywords in the input file, upper case, lower case, or any combination may be used.

***RUN_TYPE**

The *RUN_TYPE keyword is always required as it directs what sections of the SECOFL2D code are to be executed. Each of the parameters is optional, but certain parameters sometimes require that other parameters are included. The exceptions are noted below. If a parameter does not appear, the section of code that it corresponds to will not be executed.

SET_AQFR, *variable_name*$_1$=*value*$_1$, *variable_name*$_2$=*value*$_2$, ...

- *variable_name*$_n$ is either

  TYPE <no default> - determines how the aquifer properties will be set; choices are:

  DEFinition - Uses the attributes in the CAMDAT to define the aquifer properties needed by SECOFL2D that are not in the CAMDAT database

  READ - Uses a database that has all of the necessary

aquifer properties already defined

AQFR_ID <no default> - arbitrary user assigned aquifer identification number; it has no meaning in the CAMCON system, must be greater than 0, and can be a real number

NOTE: SET_AQFR, TYPE=DEF or SET_AQFR, TYPE=READ must appear when the SET_GREG, TYPE=DEF keyword is present to assure the aquifer data is present.

SET_GREG, *variable_name*$_1$=*value*$_1$, *variable_name*$_2$=*value*$_2$, ...

- *variable_name*$_n$ is either

TYPE <no default> - determines how the regional properties will be set; choices are:
DEFinition - Uses the attributes in the CAMDAT database to define the regional properties needed by SECOFL2D that are not in the CAMDAT database
READ - Uses a database that has all of the necessary regional properties already defined

GREG_ID <no default> - arbitrary user assigned regional identification number; it has no meaning in the CAMCON system, must be greater than 0, and can be a real number

X_REL <0.0 meters> - relative offset of the regional grid in reference to the aquifer grid in the x direction; must be within the bounds of the aquifer grid or 0

Y_REL <0.0 meters> - relative offset of the regional grid in reference to the aquifer grid in the y direction; must be within the bounds of the aquifer grid or 0

THETA <0.0 degrees> - rotation of regional grid; positive theta is clockwise rotation; range is 0 to 360°

NOTE: SET_AQFR, TYPE=DEF or SET_AQFR, TYPE=READ must appear when the SET_GREG, TYPE=DEF parameter is present to assure the aquifer data is present. SET_GREG, TYPE=DEF or SET_GREG, TYPE=READ must appear if the SET_GLOC, TYPE=DEF keyword, the FLOW, REGION keyword or the FLOW, BOTH keyword appears to assure that the regional data is present.

SET_GLOC, *variable_name*$_1$=*value*$_1$, *variable_name*$_2$=*value*$_2$, ...

- *variable_name*$_n$ is either

TYPE <no default> - determines how the local properties will be set; choices are:
DEFinition - Uses the attributes in the CAMDAT to define the local properties needed by SECOFL2D that are not in the CAMDAT database
READ - Uses a database that has all of the necessary regional properties already defined

GLOC_ID <no default> - arbitrary user assigned local identification number; it has no meaning in the CAMCON system, must be greater than 0, it can be a real number

X_REL <0.0 meters> - relative offset of the local grid in reference to the regional grid in the x direction; must be within the bounds of the regional grid or 0

Y_REL <0.0 meters> - relative offset of the local grid in reference to the regional grid in the y direction; must be within the bounds of the regional grid or 0

THETA <0.0 degrees> - rotation of local grid; positive theta is clockwise rotation; range is 0 to 360°

NOTE: SET_GREG, TYPE=DEF or SET_GREG, TYPE=READ must appear when the SET_GLOC, TYPE=DEF keyword appears to assure that the regional data is present. SET_GLOC, TYPE=DEF or SET_GLOC, TYPE=READ must appear if the FLOW, LOCAL parameter or the FLOW, BOTH parameter appears to assure local data is present

FLOW, *variable_name*$_1$=*value*$_1$, *variable_name*$_2$=*value*$_2$, ...

- *variable_name*$_n$ is either

REGIONal <no default> - Solve the flow equations on the regional grid by one of the methods listed; choices are:
STEADY - steady state simulation
TRANSient - transient simulation
Note: only one of the keywords, REGION, LOCAL, or BOTH may appear

REG_FLOW_ID <no default> - arbitrary user assigned run identification number; it has no meaning in the CAMCON system, must be greater than 0, and can be a real number

LOCAL <no default> - solve the flow equations on the local grid by one of

methods listed; choices are:
STEADY - steady state simulation
TRANSient - transient simulation
Note: this choice must match the choice for the regional simulation
Note: only one of the keywords, REGION, LOCAL, or BOTH may appear

LOC_FLOW_ID <no default> - arbitrary user assigned run identification number; it has no meaning in the CAMCON system, must be greater than 0, and can be a real number

BOTH <no default> - solve the flow equations on both the regional and local grids by one of the methods listed; choices are:
STEADY - steady state simulation
TRANSient - transient simulation
Note: only one of the keywords, REGION, LOCAL, or BOTH may appear

LOC_BOUNDary <NO> - specifies if a local boundary conditions file should be written and/or read; choices are:
YES - if both regional and local simulations, or only a local simulation is asked for
NO - if only a regional simulation is asked for
The user must set this correctly or SECOFL2D will crash.

WELLS <OFF> - determines if wells or a time dependent source are to be activated; choices are:
OFF - all of the wells or a time dependent source set up using the *WELL keyword are to be turned off in the regional simulation
ON - all of the wells or a time dependent source set up using the *WELL keyword are to be turned on in the regional simulation

SALT_BUOYancy <OFF> - determines if the salt buoyancy factor should be on or off; choices are:
OFF - salt buoyancy factor is off
ON - salt buoyancy factor is on

SALT_VISCosity <OFF> - Determines if the salt viscosity factor should be on or off; choices are:
OFF - salt viscosity factor is off
ON - salt viscosity factor is on

***CONSTants**

The *CONSTants keyword is necessary only if the default values of any of the parameters listed under this keyword are to be changed. The default values are in < >.

SCALE_factor, *scale_factor* <1.0E-09>

scale_factor - used to prevent underflow in the calculation; set to the appropriate value, it can also be used to change units

COMPRESsibility, *compressibility* <4.4E-10 Pa^{-1} (water)>

compressibility - the fluid compressibility; must be greater than 0

TIME_units, *time_units* <YEARS>

time_units - set to either SEConds or YEARS for the regional and/or local flow simulation and well schedules.

SCREEN_IO, *variable_name* <NONE>

- *variable_name* is either

NONE - no run information will be printed to the debug file

BRIEF - a brief version of run information will be printed to the debug file; basic run information, boundary run setup, convergence information, boundary flux information, and reports on some arrays are written to the output file.

FULL - the full version of run information will be printed to the debug file; contains all the information provided in brief plus a report of arrays read from property file or written to the output file.

BRIEF+ARRAYS - the brief version of run information and array information will be printed to the debug file; the array information includes the time array, three internal boundary condition arrays, and (for very small problems) the head array

FULL+ARRAYS - the full version of run information and array information will be printed to the debug file; the array information includes the time array, three internal boundary condition arrays, and (for very small problems) the head array

***ATTRibutes**

The *ATTRibutes keyword is necessary only if any of the default CAMDAT attribute names are to be changed. (See Section 7.1 for default CAMDAT attribute names.)

HYCND_X	<HYCND_X>	- CAMDAT attribute name for the hydraulic conductivity in the x direction; must be a valid CAMDAT attribute name
HYCND_Y	<HYCND_Y>	- CAMDAT attribute name for the hydraulic conductivity in the y direction; must be a valid CAMDAT attribute name
BULK_COMPRES	<CMPRSBLK>	- CAMDAT attribute name for rock compressibility; must be a valid CAMDAT attribute name
POROSITY	<POROSITY>	- CAMDAT attribute name for the porosity; must be a valid CAMDAT attribute name
THICK	<THICK>	- CAMDAT attribute name for the thickness; must be a valid CAMDAT attribute name
DEPTH	<DEPTH>	- CAMDAT attribute name for the depth; must be a valid CAMDAT attribute name

***AQUIFER_source_terms**

The *AQUIFER_source_terms keyword allows the user to set up time independent source terms (wells, constant head cells and rivers with finite riverbed conductance) as part of the aquifer definition. All of the *AQUIFER_source_terms parameters are optional.

PRECIPitation, *variable_name*$_1$=*value*$_1$, *variable_name*$_2$=*value*$_2$, ...

- *variable_name*$_n$ is either

IRANGE	<no default>	- the range of cell positions of the recharge in the I direction in the aquifer grid; I must correspond to interior cells; entered as num1, num2
JRANGE	<no default>	- the range of cell position of the recharge in the J direction in the aquifer grid; J must correspond to interior cells, entered as num1, num2

RATE <no default> - recharge rate for precipitation (cm**3/sec); must be greater than 0

RIVER, *variable_name$_1$=value$_1$, variable_name$_2$=value$_2$, ...*

- *variable_name$_n$* is either

IRANGE <no default> - cell range of the river in the I direction in the aquifer grid; cells in IRANGE must be interior cells; entered as num1, num2

JRANGE <no default> - cell range of the river in the J direction in the aquifer grid; cells in JRANGE must be interior cells; entered as num1, num2

CONDUCTance <no default> - conductance of the riverbed (m**2/sec); must be greater than or equal to 0

HEAD <no default> - river head (meters); must be greater than or equal to 0, and greater than or equal to BOTTOM

BOTTOM <no default> - river bottom (meters); must be greater than or equal to 0

HEAD, *variable_name$_1$=value$_1$, variable_name$_2$=value$_2$, ...*

- *variable_name$_n$* is either

IRANGE <no default> - cell range of the area of constant head in the I direction in the aquifer grid; cells in IRANGE must be interior cells; entered as num1, num2

JRANGE <no default> - cell range of the area of constant head in the J direction in the aquifer grid; cells in JRANGE must be interior cells; entered as num1, num2

HEAD <no default> - constant head value in region defined by IRANGE and JRANGE; must be greater than or equal to 0

CELL_type <ACTIVE> - type of cell; choices are:
ACTIVE - sets head as specified in METHOD (see below) and cell remains active, overwriting the value

set by the initial conditions; head does not stay constant
INACTIVE - constant head and no flow into or out of range of cells specified

METHOD <no default> - method of accumulating head in a cell; choices are:
ABSolute - overwrite any value of initial condition for head which may be present for HEAD in this range
INCrement - add HEAD value to initial condition for head which may be present for HEAD in this range

***INITial_conditions**

Initial conditions are required if and only if *RUN_TYPE, FLOW,REGION=TRANS or *RUN_TYPE, FLOW,BOTH=TRANS are selected. The initial conditions are set from variables on the CAMDAT database, from the calculated steady state solution (RESET), or from a constant value (CONSTANT).

CAMDAT, *variable_name*$_1$=*value*$_1$, *variable_name*$_2$=*value*$_2$, ...

- *variable_name*$_n$ is either

HEAD <HEAD> - the name of the CAMDAT attribute or element from which the initial conditions are read; must be a valid CAMDAT attribute or element name

TYPE <ATTR> - the type of CAMDAT variable that contains the head initial condition; choices are:
ATTR - for an attribute variable
ELEM - for an element variable

RESET

appears if the interior head initial conditions are to be reset by steady state solution with time dependent wells turned off

CONSTant, *variable_name, value*

variable_name is

VALUE <0.0> - set the initial conditions to this constant value; must be greater than or equal to 0

SALT_CONCentration, *name* <BRINEEL>

name - the CAMDAT element variable name that contains the initial condition for the brine; must be a valid CAMDAT element name

***SOLVing_parameters**

The *SOLVing_parameters keyword is required only if the *RUN_TYPE, FLOW parameters is selected and the user does not wish to use the default values listed below for a regional or local run. The regional parameters are set with the REG keyword. The local parameters are set the LOC keyword, but only if both regional and local simulation are being performed.

REGional, *variable_name*$_1$=*value*$_1$, *variable_name*$_2$=*value*$_2$, ...

- *variable_name*$_n$ is either

SOLVER <MULTIgrid> - method of solving matrix equations; choices are:
MULTIgrid - use the multigrid solver
LINE_sor - use the line SOR 2D solver
POINT_sor - use the point SOR 2D solver

CONDUCTance <HARMONIC> - method of calculating the mean hydraulic conductance; choices are:
HARMONIC - use the harmonic mean
ARITHMETIC - use the arithmetic mean

RELAX_factor <1.0> - relaxation factor for SOR solver; range is 0.1 to 1.5

CONVergence <1.0E-06> - relative convergence criteria for SOR solver; range is 1.0E-10 to 1.0

MAX_ITER_factor <10> - factor used to determine the maximum number of iterations for SOR solver; maxiter=MAX_ITER_factor*max(nrows,ncols) must be greater than or equal to 1

MIN_ITER <1> - minimum number of iterations for SOR solver; must be greater than or equal to 1

AQUIFER <CONfined> - type of aquifer; choices are CONfined or UNCONfined; this choice is constant for each grid block throughout the simulation

LOCAL, *variable_name$_1$=value$_1$, variable_name$_2$=value$_2$, ...*

- *variable_name$_n$* is either

SOLVER <MULTIgrid> - method of solving matrix equations; choices are:
MULTIgrid - use the multigrid solver
LINE_sor - use the line SOR 2D solver
POINT_sor - use the point SOR 2D solver

CONDUCTance <HARMONIC> - method of calculating the mean hydraulic conductance; choices are:
HARMONIC - use the harmonic mean
ARITHMETIC - use the arithmetic mean

RELAX_factor <1.0> - relaxation factor for SOR solver; range is 0.1 to 1.5

CONVergence <1.0E-06> - relative convergence criteria for SOR solver; range is 1.0E-10 to 1.0

MAX_ITER_factor <10> - factor used to determine the maximum number of iterations for SOR solver; maxiter=MAX_ITER_factor*max(nrows,ncols) must be greater than or equal to 1

MIN_ITER <1> - minimum number of iterations for SOR solver; must be greater than or equal to 1

AQUIFER <CONfined> - type of aquifer; choices are CONfined or UNCONfined; this choice is constant for each grid block throughout the simulation

***REG_TIMEstepping**

The *REG_TIMEstepping keyword is required only for a transient simulation using the regional grid, i.e. if *RUN_TYPE, FLOW, REGION=TRANS or *RUN_TYPE, FLOW, BOTH=TRANS has been selected. For steady state calculations, these commands are ignored.

SEQuence, *variable_name$_1$=value$_1$, variable_name$_2$=value$_2$, ...*

- *variable_name$_n$* is either

NUM_steps	<no default>	- number of time steps; must be greater than or equal to 1
START	<no default>	- time for simulation to begin; must be greater than or equal to 0
FINISH	<no default>	- time for simulation to terminate; must be greater than 0
ADAPTive	<OFF>	- turn the adaptive time step option ON or OFF

AUTO, *variable_name*$_1$=*value*$_1$, *variable_name*$_2$=*value*$_2$, ...

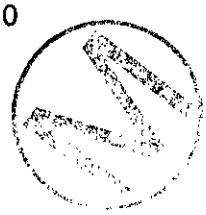
- *variable_name*$_n$ is either

OPTION	<no default>	- option for setting up automatic time stepping; choices are: DEL_TIME - uses the start and finish times and number of time steps to calculate a constant delta time EVENT_PACKing - packs time steps around an event time; requires entering the event time and a power ratio; NUM_STEPS must be greater than 4
EVENT_TIME	<no default>	- event time needed when using the event packing option; must be contained in the specified time interval
POWER_ratio	<no default>	- power ratio needed when using the event packing option; range is 0.0 to 5.0

MANual, *variable_name*, *value*$_1$, *value*$_2$, ... *value*$_n$

- *variable_name* is either

DEL_TIME	<no default>	- a list of the delta times; must be greater than 0
TIME	<no default>	- a list of the time steps; must be greater than 0 (the initial time may equal 0)



***LOCAL_TIMEstepping**

The *LOCAL_TIMEstepping keyword is required only for a transient simulation using the local grid, i.e. if *RUN_TYPE, FLOW, LOCAL=TRANS or *RUN_TYPE, FLOW, BOTH=TRANS has been selected. For steady state calculations, these commands are ignored.

SEQuence, *variable_name*$_1$=*value*$_1$, *variable_name*$_2$=*value*$_2$, ...

- *variable_name*$_n$ is either

NUM_steps	<no default>	- number of time steps, must be greater than or equal to 1
START	<no default>	- time for simulation to begin; must be greater than or equal to 0
FINISH	<no default>	- time for simulation to terminate; must be greater than 0
ADAPTive	<OFF>	- turn the adaptive time step option ON or OFF

AUTO, *variable_name*$_1$=*value*$_1$, *variable_name*$_2$=*value*$_2$, ...

- *variable_name*$_n$ is either

OPTION	<no default>	- option for setting up automatic time stepping; choices are: DEL_TIME - uses the start and finish times and number of time steps to calculate a constant delta time EVENT_PACKing - packs time steps around an event time; requires entering the event time and a power ratio; NUM_STEPS must be greater than 4
EVENT_TIME	<no default>	- event time needed when using the event packing option; must be contained in the specified time interval
POWER_ratio	<no default>	- power ratio needed when using the event packing option; range is 0.0 to 5.0

MANual, *variable_name, value$_1$, value$_2$, ... value$_n$*

- *variable_name* is either

DEL_TIME	<no default>	- a list of the delta times; must be greater than 0
TIME	<no default>	- a list of the time steps; must be greater than 0 (the initial time may equal 0)

***BOUNDary_conditions**

Boundary conditions for each of the four boundaries can be entered in sections. Each section for a regional simulation requires the REGional parameter with the necessary information including the position in meters where the section ends and each section for a local simulation requires the LOCAL parameter with the appropriate information. The LOCAL parameter is used only to set what type of boundary condition will be used for each defined section. The values of the local boundary conditions are interpolated from the regional solution.

REGional, *variable_name$_1$=value$_1$, variable_name$_2$=value$_2$, ...*

- *variable_name$_n$* is either

BOUNDARY	<no default>	- boundary specification; choices are: LOWER, UPPER, LEFT, RIGHT
TYPE	<GRAD>	- type of boundary condition; choices are: HEAD - specified head (Dirichlet) FLUX - specified flux GRAD - specified gradient (Neumann)
END	<no default>	- position in meters where this boundary condition section ends; must be within the regional domain
VALUES	<0.0,0.0>	- values of boundary condition to overwrite the initial condition; the first value applies to the beginning of the section (lower or left), the second value applies to the end of the section (upper or right), linear interpolation is used in between; the two values may be equal; must be greater than or equal to 0
CLIM_FRAC	<1.0>	- fraction of climate condition to use for a HEAD boundary; = 1.0 for full climate factor; = 0.0 to

turn off climate; range is 0.0 to 1.0; the value for HEAD is set with the *REG_CLIM keyword

USER_FUNC <NO> - specifies if a user defined function has been inserted into the SECOFL2D code and is to be used for this boundary section; choices are YES or NO. To use this option, see code sponsor for instructions.

LOCAL, *variable_name*$_1$=*value*$_1$, *variable_name*$_2$=*value*$_2$, ...

- *variable_name*$_n$ is either

BOUNDARY <no default> - boundary specification; choices are: LOWER, UPPER, LEFT, RIGHT

TYPE <GRAD> - type of boundary condition; choices are:
HEAD - specified head (Dirichlet)
FLUX - specified flux
GRAD - specified gradient (Neumann)

END <no default> - position in meters where this boundary condition section ends; must be in the local domain

***WELL**

The *WELL keyword allows the user to define the time dependent wells or a time dependent source. One *WELL data input set is needed for each well or time dependent source. If the user-specified completion depth is not within the aquifer thickness at the well location, the well will be inactive. The completion depth is measured from the same datum as head.

LOCation, *variable_name*$_1$=*value*$_1$, *variable_name*$_2$=*value*$_2$, ...

- *variable_name*$_n$ is either

X <no default> - x coordinate (in meters) of well or time dependent source; must be in the regional domain

Y <no default> - y coordinate (in meters) of well or time dependent source; must be in the regional domain

COMPLETION <no default> - completion depth (in meters) of well; must be within the aquifer or the well will be inactive

TIME_SRC, *status* <NO>

status - specifies if there is a time dependent source or not; the choices are YES or NO.

NAME_FLOW, *name* <FLOW>

name - name of the history variable on the source database containing the values for the time dependent source; must be a valid CAMDAT name

FLOW_rate, *flow_rate* <no default>

flow_rate - the base volumetric flow rate for well (m**3/sec); flow is positive for a production well and negative for an injection well

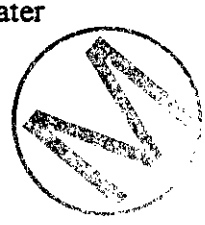
SCHEDULE, $variable\_name_1$=$value_1$, $variable\_name_2$=$value_2$, ...

- $variable\_name_n$ is either

FLOW_method <no default> - method for applying the flow_rate; choices are:
CONSTant - use a constant flow rate from the beginning to end of the simulation
RAMP - ramp the flow rate on from time START_RAMP_ON until END_RAMP_ON, use FLOW_rate from time END_RAMP_ON until START_RAMP_OFF, ramp the flow rate off from time START_RAMP_OFF until END_RAMP_OFF

START_RAMP_ON <no default> - start time the well will be ramped on; must be within the specified time interval

END_RAMP_ON <no default> - end time the well will be ramped on; must be within the specified time interval and greater than the preceding value



START_RAMP_OFF <no default> - start time the well will be ramped off; must be within the specified time interval and greater than the two preceding values

END_RAMP_OFF <no default> - end time the well will be ramped off; must be within the specified time interval and greater than the three preceding values

***REG_CLIMate_factors**

The climate factors are used to multiply previously set values of lakes/rivers, recharge wells, boundary heads and boundary fluxes. All of the *REG_CLIMate_factors parameters are all optional and would be included if the user wants to change the default values listed below.

LAKE, *variable_name*$_1$=*value*$_1$, *variable_name*$_2$=*value*$_2$, ...

- *variable_name*$_n$ is either

AMPlitude <0.2> - lake climate factor amplitude; must be greater than or equal to 0

CYCLES <1.0> - lake climate factor cycles; must be greater than or equal to 0

START_time <0.0> - time to turn on the climate factor; must be greater than or equal to 0

FINISH_time <100 years> - time to turn off the climate factor; must be greater than 0

RATIO <2.0> - ratio of river to precipitation; must be greater than or equal to 0

RECHARGE_bn, *variable_name*$_1$=*value*$_1$, *variable_name*$_2$=*value*$_2$, ...

- *variable_name*$_n$ is either

AMPlitude <0.25> - recharge climate factor amplitude; can be the name of a CAMDAT property variable or a value; must be greater than or equal to 0 or a valid CAMDAT property name

CYCLES	<1.0>	- recharge climate factor cycles; can be the name of a CAMDAT property variable or a value; must be greater than or equal to 0 or a valid CAMDAT property name
START_time	<3 months>	- time to turn on the climate factor; must be greater than or equal to 0
FINISH_time	<100 years>	- time to turn off the climate factor; must be greater than or equal to 0

HEAD_bn, *variable_name*$_1$*=value*$_1$*, variable_name*$_2$*=value*$_2$*, ...*

- *variable_name*$_n$ is either

AMPlitude	<0.25>	- head climate factor amplitude; can be the name of a CAMDAT property variable or a value; must be greater than or equal to 0 or a valid CAMDAT property name
CYCLES	<1.0>	- head climate factor cycles; can be the name of a CAMDAT property variable or a value; must be greater than or equal to 0 or a valid CAMDAT property name
HEAD_FAC	<0.0>	- factor used in head climate calculation; can be the name of a CAMDAT property variable or a value; must be greater than or equal to 0 or a valid CAMDAT property name
START_time	<3 months>	- time to turn on the climate factor; must be greater than or equal to 0
FINISH_time	<100 years>	- time to turn off the climate factor; must be greater than 0

***END**

The *END keyword signals the end of the input file. There are no other keywords or parameters associated with it.

SUMMARY OF COMMANDS

The following summarizes all PRESECOFL2D input keyword and parameters:

***RUN_TYPE**

1. SET_AQFR, TYPE=, AQFR_ID=

2. SET_GREG, TYPE=, GREG_ID=, X_REL=, Y_REL=, THETA=

3. SET_GLOC, TYPE=, GLOC_ID=, X_REL=, Y_REL=, THETA=

4. FLOW, REGIONAl=, REG_FLOW_ID=, LOCAL=, LOC_BOUNDary=,
   LOC_FLOW_ID=, BOTH=, WELLS=, SALT_BUOYancy=,
   SALT_VISCosity=

***CONSTants**

1. SCALE_factor=

2. COMPRESsibility=

3. TIME_units=

4. SCREEN_IO=

***ATTRibutes**

1. HYCND_X=

2. HYCND_Y=

3. BULK_COMPRES=

4. POROSITY=

5. THICK=

6. DEPTH=

***AQUIFER_source_terms**

1. PRECIPitation, IRANGE=, JRANGE=, RATE=

2. RIVER, IRANGE=, JRANGE=, CONDUCTance=, HEAD=, BOTTOM=

3. HEAD, IRANGE=, JRANGE=, HEAD=, CELL_type=, METHOD=

***INITial_conditions**

1. CAMDAT, HEAD=, TYPE=
2. RESET=
3. CONSTant, VALUE=
4. SALT_CONCentration=

***SOLVing_parameters**

1. REGional, SOLVER=, CONDUCTance=, RELAX_factor=,
   CONVergence=, MAX_ITER_factor=, MIN_ITER=,
   AQUIFER=
2. LOCAL, SOLVER=, CONDUCTance=, RELAX_factor=, CONVergence=,
   MAX_ITER_factor=, MIN_ITER=, AQUIFER=

***REG_TIMEstepping**

1. SEQuence, NUM_steps=, START=, FINISH=, ADAPTive=

2. AUTO, OPTION=, EVENT_TIME=, POWER_ratio=

3. MANual, DEL_TIME=, TIME=

***LOCAL_TIMEstepping**

1. SEQuence, NUM_steps=, START=, FINISH=, ADAPTive=

2. AUTO, OPTION=, EVENT_TIME=, POWER_ratio=

3. MANual, DEL_TIME=, TIME=

***BOUNDary_conditions**

1. REGional, BOUNDARY=, TYPE=, END=, VALUES=, CLIM_FRAC=,
   USER_FUNCtion=

2. LOCAL, BOUNDARY=, TYPE=, END=

***WELL**

1. LOCation, X=, Y=, COMPLETION=

2. TIME_SRC=

3. NAM_FLOW=

4. FLOW_rate=

5. SCHEDULE, FLOW_method=, START_RAMP_ON=, END_RAMP_ON=, START_RAMP_OFF=, END_RAMP_OFF=

***REG_CLIMate_factors**

1. LAKE, AMPlitude=, CYCLES=, START_time=, FINISH_time=, RATIO=

2. RECHARGE_bn, AMPlitude=, CYCLES=, START_time=, FINISH_time=,

3. HEAD_bn, AMPlitude=, CYCLES=, START_time=, FINISH_time=, HEAD_FAC=

***END**

7.3 Interactions with SECOFL2D

To execute SECOFL2D for each of the sampled vectors, type SECOFL2D at the Alpha system "$" prompt. SECOFL2D will request the names of seven files. Alternatively, the user may append the names of the seven files (in the order listed below) to the SECOFL2D command line. The seven files that the user must specify are listed below:

1. The input control file. This INP file specifies the processing options. The user has no direct control over the contents of this file other than through the preprocessor, PRESECOFL2D.

2. The regional property binary data rile corresponding to the sampled vector. This file, generated by PRESECOFL2D, contains the regional input data. These data include material properties from MATSET and the sampled transmissivity field data from GRASP.

3. The local property binary data file. This file, also generated by PRESECOFL2D, contains the grid data from the CDB file generated by GENMESH. However, the CAMDAT database file from GENMESH is not populated with property data; all local grid property data is interpolated from the regional grid by PRESECOFL2D.

4. The regional binary output data file corresponding to the sampled vector. This file contains the regional results that are output to POSTSECOFL2D for conversion into standard CAMDAT database format. (The regional flow CAMDAT database file is not used for transport calculations because previous simulations have shown radionuclide transport stays well within the local domain.)

5. The local binary output data file corresponding to the sampled vector. This file contains the local results that are output to POSTSECOFL2D for conversion into standard CAMDAT database format.

6. The binary boundary condition file for the local flow field. This file contains the local boundary conditions interpolated from the regional solution for head by SECOFL2D.

7. The diagnostics/debug file. This file contains run-time information on the execution of SECOFL2D for the sampled vector.

7.4 Description Of Input Files

Because SECOFL2D is but one module of the SECOFL triade of codes (PRESECOFL2D, SECOFL2D, and POSTSECOFL2D), no input files are supplied by the user directly to SECOFL2D. The input files are:

- The input control file to SECOFL2D (File 1 described in Section 7.3) is generated by PRESECOFL2D based in part on its own input control file. For all practical purposes, the input control file to SECOFL2D: is transparent to the user.

- The regional property data file and the local property data file are binary files generated without user intervention by PRESECOFL2D. Because they are binary, these files are also transparent to the user.

7.5 Description Of Output Files

Because SECOFL2D is but one module of the SECOFL triade of codes (PRESECOFL2D, SECOFL2D, and POSTSECOFL2D), only one output file can be read by the user, the diagnostics/debug file (File 7 described in Section 7.3), which can be found in Appendix II. The regional output data file (File 4 in Section 7.3), the local output data file (File 5 in Section 7.3), and the boundary condition file for the local flow field (File 6 in Section 7.3) are binary files that are transparent to the user.

8. EXAMPLE PROBLEMS

The use of SECOFL2D to solve groundwater flow problems similar to those used for the WIPP PA is demonstrated with the following examples.

8.1 Example 1

This steady-state problem is run in the 1992 PA regional domain covering a 25 x 30 km area. The grid consists of 50 x 57 cells and has variable spacing to reflect the spatial distribution of transmissivity data collected from wells in the area of the WIPP site (WIPP PA, 1992). Grid-block dimensions range from 50 m near the center of the site to approximately 2800 m at the boundary. The material properties are all constants except for the hydraulic conductivity which was obtained from one of the transmissivity fields generated for the 1992 WIPP PA exercise. The boundary conditions used are no-flow and fixed head. The no-flow boundaries are set up to correspond to natural geological boundaries or regions of low permeability in the material. The fixed head boundary conditions are based on the steady-state solution for the selected transmissivity field. The problem was run steady state with no climate conditions. The following is the ASCII input file used to run the pre-processor. A CAMDAT database, containing the grid and material property information, is also used as input into the pre-processor. Figure 4 is a contour plot of the resulting hydraulic head.

```
!~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
! Sample problem for SECOFL2D, steady state, no climate
!~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
!
*RUN_TYPE
  SET_AQFR, TYPE=DEF, AQFR_ID=1.001
  SET_GREG, TYPE=DEF, GREG_ID=1.002
  FLOW, REGION=STEADY
*CONSTANT
  SCREEN_IO=BRIEF
*INITIAL_CONDITIONS
  CAMDAT, HEAD=HEADAT, TYPE=ATTR
*ATTR
HYCND_Y=HYCND_X
POROSITY=POROSITY
DEPTH=THICK
*BOUNDARY
  REG, BOUNDARY=LOWER, TYPE=HEAD, END=25000.
  REG, BOUNDARY=UPPER, TYPE=HEAD, END=10000.
  REG, BOUNDARY=UPPER, TYPE=HEAD, END=17300.
  REG, BOUNDARY=UPPER, TYPE=GRAD, END=25000.
  REG, BOUNDARY=RIGHT, TYPE=HEAD, END=27240.
  REG, BOUNDARY=RIGHT, TYPE=GRAD, END=30000.
  REG, BOUNDARY=LEFT,  TYPE=HEAD, END=4000.
```

```
   REG, BOUNDARY=LEFT,  TYPE=GRAD, END=18595.
   REG, BOUNDARY=LEFT,  TYPE=HEAD, END=30000.
*REG_CLIMATE
! TURNS CLIMATE FACTORS OFF
   LAKE, AMP=0.0, CYCLES=1.0, START=0.0, FINISH=10000., RATIO=2.0
   RECHARGE, AMP=0.0, CYCLES=1.0, START=0.0, FINISH=10000.
   HEAD, AMP=0.0, CYCLES=1.0, HEAD_FAC=0.0, START=0.0, FINISH=10000.
*END
```

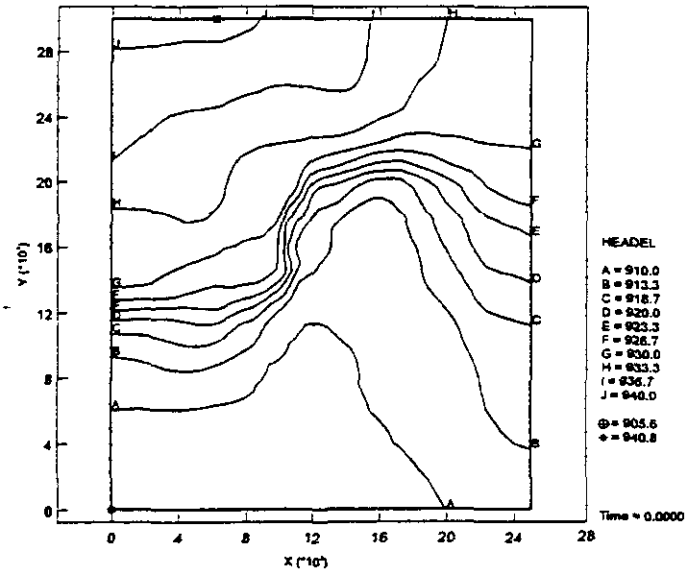


Figure 4. Steady-state solution of hydraulic head.

8.2 Example 2

This transient problem is run on the 1992 WIPP PA regional domain used in example 1 and on the local domain used in the 1992 WIPP PA exercise. The local domain covers a 5.75 x 6.625 km area and is rotated 38° clockwise and translated 12826 m in the x direction and 10665.8 m in the y direction with respect to the origin (lower left corner) of the regional domain. The grid consists of 46 x 53 uniform cells, 125 x 125 m. The boundary and initial conditions for the local domain are determined by the solution of flow in the regional domain. The regional simulation applies a climate condition, modeled as a time-dependent head boundary condition, to the upper left corner of the regional domain. The simulation is run for 10,000 years. The following is the ASCII input file used to run the pre-processor. Two CAMDAT databases, one regional and one local, containing the grid and material property information, are also used as input into the pre-processor. Figures 5 and 6 are contour plots of the resulting hydraulic head for the regional and local domains respectively at time = 10,000 years.

```
!-------------------------------------------------------------------------------
! Sample problem for SECOFL2D, transient with climate and local simulation
```

```
!-------------------------------------------------------------------------------
!
*RUN_TYPE
   SET_AQFR, TYPE=DEF, AQFR_ID=1.001
   SET_GREG, TYPE=def, GREG_ID=1.002
   SET_GLOC, TYPE=DEF, GLOC_ID=1.025, X_REL=12826.1, Y_REL=10665.8,
             THETA=38.0
   FLOW, BOTH=TRANSIENT, LOC_BOUND=YES
*CONSTANT
   SCREEN_IO=BRIEF
*INITIAL_CONDITIONS
   CAMDAT, HEAD=HEADAT, TYPE=ATTR
   RESET
*ATTR
 HYCND_Y=HYCND_X
 POROSITY=POROSITY
 DEPTH=THICK
*REG_TIME
   SEQ, NUM_STEP=10, START=0.0, FINISH=10000.
   AUTO, OPTION=DEL_TIME
*LOCAL_TIME
   SEQ, NUM_STEP=10, START=0.0, FINISH=10000.
   AUTO, OPTION=DEL_TIME
*BOUNDARY
   REG, BOUNDARY=LOWER, TYPE=HEAD, END=25000., CLIM_FRAC=0.
   REG, BOUNDARY=UPPER, TYPE=HEAD, END=10000., CLIM_FRAC=1.
   REG, BOUNDARY=UPPER, TYPE=HEAD, END=17300., CLIM_FRAC=0.
   REG, BOUNDARY=UPPER, TYPE=GRAD, END=25000., CLIM_FRAC=0.
   REG, BOUNDARY=RIGHT, TYPE=HEAD, END=27240., CLIM_FRAC=0.
   REG, BOUNDARY=RIGHT, TYPE=GRAD, END=30000., CLIM_FRAC=0.
   REG, BOUNDARY=LEFT,  TYPE=HEAD, END=4000.,  CLIM_FRAC=0.
   REG, BOUNDARY=LEFT,  TYPE=GRAD, END=18595., CLIM_FRAC=0.
   REG, BOUNDARY=LEFT,  TYPE=HEAD, END=30000., CLIM_FRAC=1.
   LOCAL, BOUNDARY=LOWER, TYPE=HEAD, END=5750.
   LOCAL, BOUNDARY=UPPER, TYPE=HEAD, END=5750.
   LOCAL, BOUNDARY=RIGHT, TYPE=GRAD, END=6625.
   LOCAL, BOUNDARY=LEFT,  TYPE=HEAD, END=6625.
*REG_CLIMATE
   LAKE, AMP=0.0, CYCLES=1.0, START=0.0, FINISH=10000., RATIO=2.0
   RECHARGE, AMP=0, CYCLES=1., START=0.0, FINISH=10000.
   HEAD, AMP=AMPLITUD, CYCLES=CYCLES, HEAD_FAC=CLIMTIDX, &
      START=0.0, FINISH=40000.
*END
```

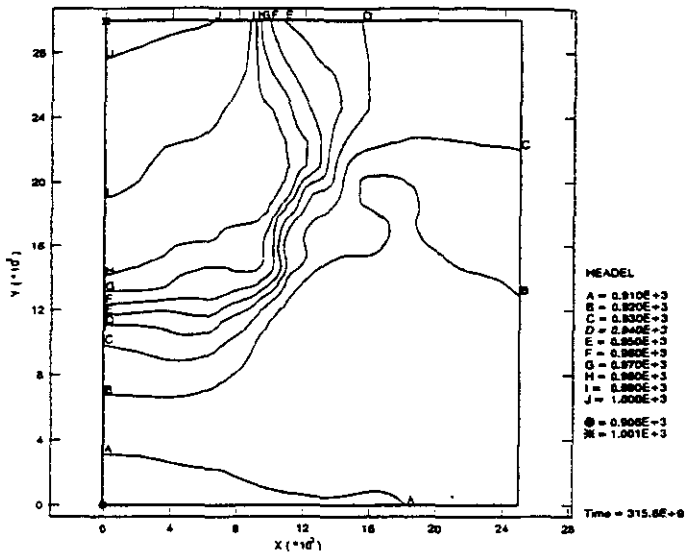


Figure 5. Transient solution of hydraulic head in the regional domain time = 10,000 years.

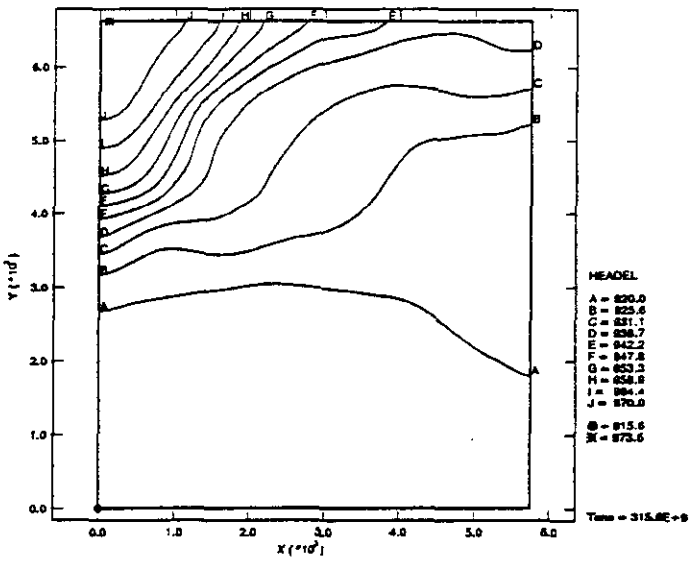


Figure 6. Transient solution of hydraulic head in the local domain time = 10,000 years.

REFERENCES

Anderson, M. P. and Woessner, W. W. (1992), *Applied Groundwater Modeling: Simulation of Flow and Advective Transport*, Academic Press, San Diego.

Bear, J. (1972), *Dynamics of Fluids in Porous Media*, American Elsevier, New York.

Bear, J. and Verruijt, A. (1987), *Modeling Groundwater Flow and Pollution*, Reidel Publishing Co., Dordrecht, Netherlands.

Corbet, T. F. (1994), "Evolution of Patterns of Regional Groundwater Flow in Southeastern New Mexico; Response to Post-Pleistocene Changes in Climate", 1994 Annual Meeting of the Geological Society of America, Oct 24-27 1994, Seattle, WA.

Das, B., Steinberg, S., Weber, S., and Schaffer, S. (1994), "Finite Difference Methods for Modeling Porous Media Flows", *Transport in Porous Media*, Vol. 17, pp. 171-200.

de Marsily, G. (1986), *Quantitative Hydrogeology: Groundwater Hydrology for Engineers*, Academic Press, San Diego.

Dendy, J. E. Jr., McCormick, S. F., Ruge, J. W., Russell, T. F., and Schaffer, S., (1989), "Multigrid Methods for Three-Dimensional Petroleum Reservoir Simulation", Proc. 10th Symposium on Reservoir Simulation, Society of Petroleum Engineers, Houston, Texas, 6-8 Feb. 1989, pp. 19-25.

Dendy, J. E. Jr., Ida, M. P., and Rutledge, J. M. (1992), "A Semicoarsening Multigrid Algorithm for SIMD Machines", *SIAM Journal on Scientific and Statistical Computing*, Vol. 24, pp. 1460-1469.

Dukowitz J. D. (1985), "Conservative Rezoning Algorithm for Generalized Two-Dimensional Meshes", *Journal of Computational Physics*, Vol. 59, 1985, pp. 193-199.

Fanchi, J. R., et al. (1982), "BOAST; a Three-Dimensional, Three-Phase Black Oil Applied Simulation Tool (Version 1.1). Vol. I: Technical Description and Fortran Code", Keplinger and Associates, Inc., Tulsa, OK. NTIS DE83 000529, Sept. 1982, pp. 43-44.

Ferziger, J. H. (1993), "Estimation and Reduction of Numerical Error", ASME FED-Vol. 158, *Quantification of Uncertainty in Computational Fluid Dynamics*, ASME Fluids Engineering Division Summer Meeting, Washington, DC, 20-24 June 1993. Celik., I, Chen, C. J., Roache, P. J., and Scheurer, G., Eds., pp. 18.

Harlow, F. and Welch, J. E. (1965), "Numerical Calculation of Time-Dependent Viscous Incompressible Flow of Fluid with Free Surface", *Physics of Fluids*, Vol. 8, No. 12, pp. 2182-2189.

Helton, J. C. (1990), "Sensitivity Analysis Techniques and Results for Performance Assessment of the Waste Isolation Pilot Plant", SAND90-7103, Albuquerque, NM, Sandia National Laboratories.

Helton, J. C., Anderson, D. R., Baker, B. L., Bean, J. E., Berglund, J. W., Beyeler, W., Garner, J. W., Iuzzolino, H. J., Marietta, M. G., Rechard, R. P., Roache, P. J., Rudeen, D. K., Schreiber, J. D., Swift, P. N., Tierney, M. S., and Vaughn, P. (1995), "Effect of Alternative Conceptual Models in a Preliminary Performance Assessment for the Waste Isolation Pilot Plant", *Nuclear Engineering and Design*, Vol. 154, pp. 251-344.

Howery, S. G., M. G. Marietta, R. P. Rechard, P. N. Swift, D. R. Anderson, B. Baker, J. Bean, W. Beyeler, K. F. Brinster, R. V. Guzowski, J. Helton, R. D. McCurley, D. K. Rudeen, and J. Schreiber, (1990), "Preliminary Comparison with 40 CFR Part 191, Subpart B for the Waste Isolation Pilot Plant". SAND90-2347, Albuquerque, NM, December 1990, Sandia National Laboratories.

Kipp, K. L., Jr.(1986). "HST3D: A Computer Code for Simulation of Heat and Solute Transport in Three-Dimensional Ground-Water Flow Systems", Water Resources Investigations Report 86-4095 .U.S. Geological Survey, 1986.

Knupp, P. M. (1990) , "Dukowicz/Ramshaw Mesh Interpolation", Ecodynamics Memo 16 October 1990. "Dukowicz/Ramshaw Mesh Interpolation - Part II", Ecodynamics Memo 18 October 1990. Ecodynamics Research Associates, Inc., P. O. Box 9229, Albuquerque, NM 87119.

Knupp, P. M. and Askew, S. (1995), "User's Guide for SECOLFL3D, Version 1.8x", Ecodynamics Research Associates, Inc., P. O. Box 9229, Albuquerque, NM 87119.

Knupp, P. M. and Dombroski, E. S. (1993), "Grid Convergence Study for SECO2D_FLOW Salt Benchmark Problem", 7 April 1993, Ecodynamics Research Associates, Inc., P. O. Box 9229, Albuquerque, NM 87119.

Lusczynski, N. J., (1961), "Head and Flow of Ground Water of Variable Density", *Journal of Geophysical Research*, Vol. 66, No. 12, December 1961, pp. 4247-4256.

McDonald, M. G., and Harbaugh, A. W. (1988), "A Modular Three-Dimensional Finite-Difference Ground-Water Flow Model", Techniques of Water-Resources Investigations of the United States Geological Survey, Book 6, Modeling Techniques, Scientific Software Group, Washington, D.C.

McDonald, M. G., Harbaugh, A. W., Orr, B. R., and Ackerman, D. J. (1991), "A Method of Converting No-Flow Cells to Variable-Head Cells for the U. S. Geological Survey Modular Finite-Difference Ground-Water Flow Model", United States Geological Survey, Open-File Report 91-536, U. S. Geological Survey, MS 411 Federal Center, Box 25425, Denver, CO 80225.

Rechard, R. P., Iuzzolino, H. J., Rath, J. S., McCurley, R. D., Rudeen, D. K., and Gilkey, A. P., (1989), "User's Manual for CAMCON: Compliance Assessment Methodology Controller", SAND88-1496, Albuquerque, NM, Sandia National Laboratories.

Rechard, R. P., Roache, P. J., Blaine, R. L., Gilkey, A. P., and Rudeen, D. K., (1991), "Quality Assurance Procedures for Computer Software Supporting Performance Assessments of the Waste Isolation Pilot Plant", SAND90-1240, Albuquerque, NM, Sandia National Laboratories.

Rechard, R. P., Ed. (1992), "User's Reference Manual for CAMCON: Compliance Assessment Methodology Controller, Version 3.0", SAND90-1983, Albuquerque, NM, Sandia National Laboratories.

Reeves, M., Ward, D. S., Johns, N. D., and Cranwell, R. M., (1986), "Theory and Implementation for SWIFT II The Sandia Waste-Isolation Flow and Transport Model for Fractured Media", SAND83-1159, Sandia National Laboratories, Albuquerque, New Mexico, August 1986.

Roache, P. J. (1972), *Computational Fluid Dynamics*, Albuquerque, NM: Hermosa Publishers. Revised Edition, 1976.

Roache, P. J. (1990), "Need for Control of Numerical Accuracy", *Journal of Spacecraft and Rockets*, Vol. 27, No. 2, 1990, pp. 98-102.

Roache, P. J., Knupp, P. M., Steinberg, S. and Blaine, R. L. (1990), "Experience with Benchmark Test Cases for Groundwater Flow", ASME FED-Vol. 93, *Benchmark Test Cases for Computational Fluid Dynamics*, I. Celik and C. J. Freitas, eds., Book No. H00598-1990, pp. 49-56.

Roache, P. J. (1991), "Computational Fluid Dynamics Algorithms and Codes Developed for WIPP Site Simulations", Proceedings of the Asian Pacific Conference on Computational Mechanics, 11-13 December 1991, University of Hong Kong, Y. K. Cheung, J. H. W. Lee, and A. Y. T. Leung, eds., A. A. Balkema, Rotterdam, Vol. 2, pp. 1325-1335.

Roache, P. J. (1992), "Computational Fluid Dynamics Algorithms Developed for WIPP Site Simulations", Proceedings IX International Conference on Computational Methods in Water Resources, Denver, Colorado, 9-12 June 1992, T. Russell, et al., eds., pp. 375-382.

Roache, P. J. (1993a), "The SECO Suite of Codes for Site Performance Assessment", Proceedings 1993 Intl. High-Level Radioactive Waste Management Conf., April 26-30, 1993, Las Vegas, NV.

Roache, P. J. (1993b), "The SECO Code Algorithms for Groundwater Flow and Transport", *Finite Elements in Fluids: New Trends and Applications*, Part II, Proc. VIII International Conference on Finite Elements in Fluids, Univ. Politecnica de Cataluna, Barcelona, Spain, 20-23

Sept. 1993, K. Morgan, E. Onate, J. Periaux, J. Peraire, and O. C. Zienkiewicz, Eds., pp. 939-948.

Roache, P. J. (1993c), "A Method for Uniform Reporting of Grid Refinement Studies", ASME FED-Vol. 158, *Quantification of Uncertainty in Computational Fluid Dynamics*, ASME Fluids Engineering Division Summer Meeting, Washington, DC, 20-24 June 1993. Celik., I, Chen, C. J., Roache, P. J., and Scheurer, G. Eds., pp. 109-120.

Roache, P. J. (1993d), "A Method for Uniform Reporting of Grid Refinement Studies", Proc. 11th AIAA Computational Fluid Dynamics Conference, July 6-9, 1993, Orlando, Fl., Part 2, pp. 1-57-1058.

Roache, P. J. (1994a), "Perspective: A Method for Uniform Reporting of Grid Refinement Studies", *ASME Journal of Fluids Engineering*, Vol. 116, No. 3, Sept.1994, pp. 405-413.

Roache, P. J. (1994b), "Modeling Uncertainty and Numerical Uncertainty in Atmospheric Models, Ocean Circulation, and Groundwater Flow and Transport", Proc. Basel World CFD User Days 1994, Second World Conference in Applied Computational Fluid Dynamics, Basel, Switzerland, 1-5 May 1994, *Workshop # 5, "How is CFD Helping in Environmental Risk Assessments?"*, pp. 20.7 -20.10.

Roache, P. J. (1995a), "Verification of Codes and Calculations", AIAA 95-2224, 26th AIAA Fluid Dynamics Conference, June 19-23, 1995, San Diego, CA.

Roache, P. J. (1995b), "SECOTR2D and SECOTR3D User's Manual", Ecodynamics Research Associates, Inc., P. O. Box 9229, Albuquerque, NM 87119.

Salari, K., Knupp, P., Roache, P., and Steinberg, S. (1992), "TVD Applied to Radionuclide Transport in Fractured Porous Media", Proceedings IX International Conference on Computational Methods in Water Resources, Denver, Colorado, 9-12 June 1992, T. Russell, et al., eds., pp. 141-148.

Salari, K., Blaine, R. L., Economy, K., and Roache, P. J. (1995), "Grid Resolution Studies of Radionuclide Transport in Fractured Porous Media", ASME FED-Vol. 213, *Quantification of Uncertainty in Computational Fluid Dynamics - 1995*, ASME/JSME Fluids Engineering and Laser Anemometry Conference, August 13-18, 1995, Hilton Head, SC. R. W. Johnson and E. D. Hughes, Eds., pp. 25-30.

Salari, K., and Blaine, R. L. (1995), "SECOTP2D User's Manual, Version 1.21ZO", SAND95-xx (to appear), Albuquerque, NM, Sandia National Laboratories.

Salari, K. (1995), "SECO_VIEW User's Manual", Ecodynamics Research Associates, Inc., P. O. Box 9229, Albuquerque, NM 87119. To appear.

Schaffer, S. (1995), "A Semicoarsening Multigrid Method for Elliptic Partial Differential Equations with Highly Discontinuous and Anisotropic Coefficients," *SIAM Journal on Scientific Computing*, accepted for publication.

U. S. Environmental Protection Agency (1985), "Environmental Standards for the Management and Disposal of Spent Nuclear Fuel, High-Level and Transuranic Radioactive Waste; Final Rule, 40 CFR Part 191", Federal Register 50:38066-38089.

WIPP PA (1992), Performance Assessment Division, "Preliminary Comparison with 40 CFR Part 191, Subpart B for the Waste Isolation Pilot Plant, Volume 1: Methodology and Results." SAND92-0700/1, Sandia National Laboratories, Albuquerque, NM, December 1992.

Young, D. M. (1971), *Iterative Solution of Large Linear Systems*, Academic Press, NY, Ch. 3,

Westerink, J. J. and Roache, P. J. (1995), "Issues in Convergence Studies in Geophysical Flow Computations", ASME FED-Vol. 213, *Quantification of Uncertainty in Computational Fluid Dynamics - 1995*, ASME/JSME Fluids Engineering and Laser Anemometry Conference, August 13-18, 1995, Hilton Head, SC. R. W. Johnson and E. D. Hughes, Eds., pp. 1-6.

APPENDIX A. VERIFICATION OF SPATIAL NUMERICAL ACCURACY

The following is a reprint of Roache et al. (1990). The format, font and some symbols have been changed to enhance readability and typographical errors have been corrected, but the substantive information has all been previously published in Roache et al. (1990).

EXPERIENCE WITH BENCHMARK TEST CASES FOR GROUNDWATER FLOW

P. J. Roache, P. M. Knupp, S. Steinberg, and R. L. Blaine
Ecodynamics
Albuquerque, New Mexico

ABSTRACT

Experience with formulating and applying several benchmark test cases in groundwater hydrology CFD problems is described. Three problem categories are considered: (1) full 2-D groundwater hydrology problem codes tested on a steady problem with scalar conductivities in stretched cartesian coordinates; (2) Fortran subroutines produced by computer Symbolic Manipulation for the stencil array evaluation of tensor conductivity in general non-orthogonal 2-D and 3-D coordinates; (3) particle tracking in 2-D and 3-D. Freedom from coding errors, consistency of the discretization, and order of convergence are verified. Examples are given of successful code verification, successful error detection, and unsuccessful "false negative" tests. The codes tested involve a commercial code (SWIFT II), two codes produced by USGS personnel (MODFLOW and HST3D), and our own codes (SECO, for Sandia-ECOdynamics).

INTRODUCTION

The basic equations for ground water hydrology with constant composition are simple enough; nonetheless, applications are still error-prone. Even if the code is correct, there is a substantial chance for user error to occur. This is most evident when using a code developed by someone other than the user, especially when the code has more physics than required for the user's problem, and when the codes have an ancient history (e.g. "card image" orientation in input files, spaghetti coding, uneven documentation, etc.). But errors also occur even when the code developer is applying his or her own code.

We recognize five distinct regimes where errors can be made: (1) in code generation (either by hand or using computer Symbolic Manipulation; see below); (2) in code instructions (e.g. in user manual or comment cards); (3) in problem set-up; (4) in defining and coding a benchmark case; (5) in the interpretation of results. The first two are errors of the code builder. The last three are errors of the code user, although ambiguous or scant code documentation can put some of the responsibility back onto the code builder. "Verification" of a code removes (1) and, if done thoroughly, (2), but (3-5) still contain the potential for errors in any new application. We reluctantly conclude that there will be a continuing need for users to construct and exercise benchmark test cases *even when using verified and validated codes.*

In this paper, we describe some our of experiences with formulating and applying several benchmark test cases in groundwater hydrology CFD problems. The problems are all single phase, Darcy law flows. Three problem categories are considered: (1) full 2-D groundwater hydrology flow codes tested on a steady problem with scalar conductivities in stretched cartesian coordinates; (2) Fortran subroutines produced by computer Symbolic Manipulation for the stencil array evaluation of tensor conductivity in general non-orthogonal 2-D and 3-D coordinates; (3) particle tracking in 2-D and 3-D. Freedom from coding errors, consistency of the discretization, and order of convergence are verified. Examples are given of successful code verification, successful error detection, and unsuccessful "false negative" tests.

2-D FLOW WITH SCALAR CONDUCTIVITY IN STRETCHED CARTESIAN COORDINATES

Four codes were considered in this part of the study: SWIFT II (Reeves, et al., 1986), HST3D (Kipp, 1987), MODFLOW (McDonald and Harbaugh, 1988), and SECO (Roache, et al., 1990). SWIFT II is a code developed by Intera Technologies for Sandia Laboratories. HST3D and MODFLOW were developed by the USGS (United States Geological Survey). SECO (for Sandia-**ECO**dynamics) is a suite of codes under development at Ecodynamics for Sandia. SWIFT II and HST3D use primitive variables and allow for varying fluid compositions (e.g. brine, heat) whereas MODFLOW and SECO (presently) allow for only a pure component. SWIFT II and MODFLOW use a Marker-And-Cell (MAC, or Arakawa "C") staggered grid, HST3D uses a non-staggered grid (i.e., collocated variables) and SECO has options for either. All four use fully implicit (backward Euler) time differencing, and all four appear from the problem formulations and code descriptions to be uniformly 2nd order accurate in space and 1st order accurate in time.

The simplest formulation of the governing equations for this problem gives a single parabolic equation in terms of hydraulic (piezometric) head h.

$$S_s \, \partial h/\partial t = \nabla \cdot (k \nabla h) \tag{1}$$

The codes considered all use stretched cartesian (i.e. planar orthogonal) coordinates, so the conductivity k may be tensor, provided that the principal axes of k are aligned with x and y; this is effectively a scalar conductivity assumption, since it is equivalent to a simple rescaling of x or y. All four use or have options to use harmonic averaging for the conductance (a combination of the physical variable conductivity and discretization terms); for a uniform grid, harmonic averaging is equivalent to defining k at the cell face $(i + 1/2)$ as

$$k(i+1/2) = 2k(i)\,k(i+1) / \left[k(i) + k(i+1)\right] \tag{2}$$

as opposed to the more naïve linear average

$$k(i+1/2) = \left[k(i) + k(i+1)\right] / 2 \tag{3}$$

SWIFT II and HST3D treat much more complicated problems, so that many of their calculations are null for the simple problem described here. However, it is noteworthy that MODFLOW and SECO are also fairly complex codes, owing not to the complexity of (1) itself,

but to (2) and many *modeling* issues such as definition of aquifer properties, locally confined (artesian) or unconfined aquifer conditions, simulation of rivers, recharge, well schedules, inactive regions, etc. All of these introduce significant nonlinearities (step function dependencies on the dependent variable h) and coding complexities. Also, SECO has both regional and local area grid solution capability (i.e., an elementary domain decomposition approach) and can treat discontinuous boundary definition of the general Robin type, whereas HST3D allows Dirichlet or non-homogeneous Neumann, and the other codes all use only homogeneous Neumann boundary conditions.

A benchmark test case for steady flow is formulated as follows. With constant aquifer properties on a square domain of (0,a) × (0,a) and no wells, the boundary conditions are as follows. ("h" = head for MODFLOW and SECO, or pressure for SWIFT II and HST3D.) At $y = a$, the Dirichlet condition applies as

$$h(x,a) = c\left(\cos(\pi \cdot x / a) + 1\right) \tag{4}$$

$$c = \cosh(\pi) = 11.59... \tag{5}$$

and on the other three sides of the domain, the homogeneous Neumann condition applies as

$$\partial h / \partial n = 0. \tag{6}$$

The analytic solution is

$$h(x,y) = \cos(\pi \cdot x / a)\cosh(\pi \cdot y / a) + c \tag{7}$$

The domain chosen was 40 km × 40 km, but these values, like the aquifer properties, scale out of the results for the test problem. The analytic solution was used to set initial conditions at all points.

SECO uses this discretized analytic solution to set discrete boundary conditions, which means boundary values and values at the first interior cells. Thus, at boundaries with a homogeneous Neumann condition, the two-point difference equation for $\partial h / \partial n$ is not set to zero, but to the discrete values from the exact solution, which only approach zero as the grid is refined.

Two problem sets are presented. Table 1 presents results for a uniform staggered (MAC) grid, and Table 2 for a stretched MAC grid. The Tables show maximum error ERR_MAX, and COEFF_MAX = ERR_MAX · (IL-1+mac)^2, where mac = 1 for the MAC grid, or mac = 0 for the collocated grid option. For a uniformly 2nd order accurate solution, the value of COEFF_MAX should become roughly a constant (or decrease) as the grid is refined.

TABLE 1. CONVERGENCE OF SECO: UNIFORM GRID.

IL×JL	ERR_MAX	COEFF_MAX
5×5	1.129E-03	2.82E-02
10×10	4.142E-04	4.14E-02
20×20	1.390E-04	4.14E-02
40×40	2.274E-05	3.64E-02
80×80	4.902E-06	3.14E-02

The results display the expected second-order accuracy. The maximum error occurs well away from boundaries, e.g., in the 40×40 grid, at (I, J) = (7, 34).

Table 2 presents the same type results, but for a stretched grid. The grid stretching used in x concentrated (arbitrarily) the stretching at $x = 20$ km in the 40 km × 40 km grid, with a power-law stretching with $p = 0.5$. This gives a ratio of maximum to minimum x-spacing of 1.72 in the 5×5-cell grid, and 6.33 in the 80×80-cell grid. The grid stretching used in y concentrated the stretching at $y = 23$ km with $p = 0.4$. This gives a ratio of maximum to minimum y-spacing of 2.97 in the 5×5-cell grid, and 5.92 in the 80×80-cell grid.

TABLE 2. CONVERGENCE OF SECO: STRETCHED GRID.

IL×JL	ERR_MAX	COEFF_MAX
5×5	5.906E-04	1.48E-02
10×10	5.354E-04	5.35E-02
20×20	4.242E-04	1.70E-01
40×40	1.704E-04	2.73E-01
80×80	4.101E-05	2.62E-01

Again, the results display the expected 2nd order accuracy, and the maximum error occurs well away from boundaries, e.g., in the 40×40 grid, at (I, J) = (35, 34). Asymptotic behavior is reached more slowly, and the maximum truncation error is larger, for the stretched grid than for the uniform grid, as is to be expected for a smoothly varying solution. This is the cost for the increased resolution near the grid concentration point (20 km, 23 km).

A similar second test problem, previously used on SWIFT II, was also exercised on SECO. At $y = a$, the Dirichlet condition applies as

$$h(x,a) = c \cdot \cos(0.5 \cdot \pi \cdot x / a) \tag{8}$$

$$c = \cosh(\pi / 2) = 2.509... \tag{9}$$

At $x = a$, the Dirichlet condition applies as

$$h(a,y) = 0 \tag{10}$$

and on the other two sides of the domain, the homogeneous Neumann condition applies as

$$\partial h / \partial n = 0. \tag{11}$$

The analytic solution is

$$h(x,y) = \cos(\pi \cdot x / 2a)\cosh(\pi \cdot y / 2a). \tag{12}$$

This problem is considered for SECO only because it caused difficulties with SWIFT II. Although the solution behavior is not significantly more difficult than the previous problem, it does contain Dirichlet boundary conditions on two adjacent boundaries. This causes a large local error in SWIFT II near that corner. No such difficulty arises with SECO, as expected. In the 80×80 uniform grid, ERR_MAX = 6.265E-06 and COEFF_MAX = 4.01E-02, which are comparable to the errors for the first problem. Also, the maximum error occurs at (I, J) = (22, 41) indicating no difficulty near the corner, in contrast to the SWIFT II results, which were clearly indicative of an error. The SWIFT II user manual indicates correctly that Dirichlet conditions at a corner cell, where different Dirichlet values apply at each side, will be over-written so that only the second entered value is active. Thus, the two adjacent faces of a corner cell cannot have different Dirichlet values, i.e., due to coding peculiarities, SWIFT II needs to have $h(a - \Delta x / 2, a) = h(a, a - \Delta y / 2)$.

Even with the first problem (involving 3 Neumann and 1 Dirichlet boundaries) neither SWIFT II nor MODFLOW displayed 2nd order convergence rates in the resolution ranges tested. Both of these use MAC grids. HST3D, which uses a collocated variable grid, did display 2nd order convergence rates (as did SECO above, also using MAC grids). However, for a related problem with specified non-zero flux boundary conditions, HST3D proved to have an error by a factor of 2 in the magnitude of the computed flux boundary value, only for a steady state calculation. (The pressure solution itself was correct, but the post-processing evaluation of boundary flux was in error.) It turned out that this coding error could easily be averted by changing the time differencing scheme selected for the steady state calculation.

TENSOR CONDUCTIVITY IN 2-D and 3-D NON-ORTHOGONAL COORDINATES

Versions of the SECO codes are presently under development which use 2-D and 3-D non-orthogonal coordinates and allow for tensor conductivity. The Fortran source codes for the stencil loading are produced using computer Symbolic Manipulation, rather than hand coding. (For example, see Roache and Steinberg, 1984, Steinberg and Roache, 1985, 1986a, 1986b). These code subroutines are verified by solving several test problems, the most difficult of which involves an ellipse shaped region on the unit square with a discontinuous jump in tensor conductivities across the boundary.

A driver code was written to test the code and theory of automatic generation of Symmetric Finite Difference Stencils in Generalized 3-D Coordinates (Steinberg and Roache, 1990). The goal of the exercise was to verify 2nd order accuracy of the finite difference approximations to the

continuum problem, the latter consisting of any second-order symmetric elliptic operator with tensor coefficients applied to a continuous function on an arbitrary connected domain in the plane.

A discontinuous coefficient tensor model is included to simulate the case of abrupt changes in the permeability of adjacent geological formations.

The basic capabilities of the driver code were, therefore, to generate (i) a set of transformations from logical space to physical space in the plane, (ii) a symmetric elliptic operator, and (iii) a set of boundary conditions based on the choice of solution function. With these inputs, the stencil-loader code was called to generate the symmetric stencils. A point SOR solver then used the stencils to obtain the discrete solution. Finally, the discrete solution was compared to the known exact solution to obtain the global truncation error.

The "irregular" domain results in a general non-orthogonal grid, the most general problem for which the stencil-loader was designed. A number of other domains (e.g., rectangles, parallelograms, trapezoids) were also used during the debugging phase of this exercise, but not in the final verification. Also, separate 2-D tests were run, but only the 3-D results are shown herein.

The Transformation

Two transformations were used.

A. Unit Cube:

$$\begin{aligned} x(\chi,\xi,\zeta) &= \chi \\ y(\chi,\xi,\zeta) &= \xi \\ z(\chi,\xi,\zeta) &= \zeta \end{aligned} \tag{13}$$

B. Irregular Domain:

$$\begin{aligned} x(\chi,\xi,\zeta) &= \chi + \sin\left[(\pi/4)\cdot\xi\right]/2 \\ y(\chi,\xi,\zeta) &= \xi + \sin\left[(\pi/8)\cdot(\chi+\zeta)\right]/2 \\ z(\chi,\xi,\zeta) &= \zeta + \sin\left[(\pi/8)\cdot(\chi+\xi+\zeta)\right]/2 \end{aligned} \tag{14}$$

Using the bound $|\cos(x)| \leq 1$, it is possible to show that the Jacobian of this transformation is strictly positive everywhere. The domain looks basically like a distorted cube.

The Operator

Three possible choices of coefficients define the Operator.

A. Laplace Operator.

$$
\begin{aligned}
&k11(x,y,z) = 1.\\
&k12(x,y,z) = k21(x,y,z) = 0.\\
&k13(x,y,z) = k31(x,y,z) = 0.\\
&k22(x,y,z) = 1.\\
&k23(x,y,z) = k32(x,y,z) = 0.\\
&k33(x,y,z) = 1.
\end{aligned}
\tag{15}
$$

B. Arbitrary Elliptic Operator with Continuous Coefficients.

$$
\begin{aligned}
&e1(x,y,z) = 1 + 2x^2 + y^2 + z^2,\\
&e2(x,y,z) = 1 + x^2 + 2y^2 + z^2,\\
&e3(x,y,z) = 1 + x^2 + y^2 + 2z^2
\end{aligned}
\tag{16}
$$

$$
\mathbf{D} = \begin{bmatrix} e1 & 0 & 0 \\ 0 & e2 & 0 \\ 0 & 0 & e3 \end{bmatrix} \tag{17}
$$

$$
\mathbf{P} = \begin{bmatrix} \cos(u)\cdot\cos(v) & \sin(u) & \cos(u)\cdot\sin(v) \\ -\sin(u)\cdot\cos(v) & \cos(u) & -\sin(u)\cdot\sin(v) \\ -\sin(v) & 0 & \cos(v) \end{bmatrix} \tag{18}
$$

with $0 < u, v < 2\cdot\pi$, and

$$
\mathbf{K} = \begin{bmatrix} k11 & k12 & k13 \\ k21 & k22 & k23 \\ k31 & k32 & k33 \end{bmatrix} \tag{19}
$$

Then **D** has positive eigenvalues. If we let

$$
\mathbf{K} = \mathbf{P}^{-1}\mathbf{D}\mathbf{P}. \tag{20}
$$

then **K** has the same eigenvalues. Further, since **P** is an orthogonal matrix, **K** is symmetric. The elements of **K** are as follows.

$$
\begin{aligned}
k11(x,y,z) = {} & e1\cdot\cos(u)\cdot\cos(u)\cdot\cos(v)\cdot\cos(v) + \\
& e2\cdot\sin(u)\cdot\sin(u)\cdot\cos(v)\cdot\cos(v) + \\
& e3\cdot\sin(v)\cdot\sin(v)
\end{aligned}
\tag{21}
$$

$$k12(x,y,z) = k21(x,y,z) = (e1 - e2)\cdot\cos(u)\cdot\sin(u)\cdot\cos(v) \tag{22}$$

$$k13(x,y,z) = k32(x,y,z) = \cos(v)\cdot\sin(v)\cdot\left[e1\cdot\cos(u)\cdot\cos(u) + e2\cdot\sin(u)\cdot\sin(u) - e3\right] \tag{23}$$

$$k22(x,y,z) = e1\cdot\sin(u)\cdot\sin(u) + e2\cdot\cos(u)\cdot\cos(u) \tag{24}$$

$$k23(x,y,z) = k32(x,y,z) = (e1 - e2)\cdot\cos(u)\cdot\sin(u)\cdot\sin(v) \tag{25}$$

$$k33(x,y,z) = e1\cdot\cos(u)\cdot\cos(u)\cdot\sin(v)\cdot\sin(v) + e2\cdot\sin(u)\cdot\sin(u)\cdot\sin(v)\cdot\sin(v) + e3\cdot\cos(v)\cdot\cos(v) \tag{26}$$

For the numerical calculations, we have taken

$$u = \frac{5\cdot\pi}{12} \quad \text{and} \quad v = \frac{2\cdot\pi}{9} \tag{27}$$

C. Elliptic Operator with Discontinuous Coefficients.

To model the geologic situation, normal fluxes across a geological boundary are assumed continuous:

$$\left[\mathbf{K}\,\partial f / \partial n\right]_I = \left[\mathbf{K}\,\partial f / \partial n\right]_{II} \tag{28}$$

$$\mathbf{K}\,\partial f / \partial n = \begin{bmatrix} n1 & n2 & n3 \end{bmatrix} \begin{bmatrix} k11 & k12 & k13 \\ k21 & k22 & k23 \\ k31 & k32 & k33 \end{bmatrix} \begin{bmatrix} fx \\ fy \\ fz \end{bmatrix} = n\mathbf{K}\nabla f \tag{29}$$

The unit cube domain is divided into two regions *W:I* and *W:II* where

$$W{:}I = \left\{(x,y,z) \text{ in } U \mid r < 2\right\} \tag{30}$$

$$W{:}II = \left\{(x,y,z) \text{ in } U \mid r > 2\right\} \tag{31}$$

$$r = \sqrt{(1+x)^2 + 12y^2 + 12z^2}. \tag{32}$$

On region *I*, let

$$\mathbf{D} = \begin{bmatrix} 1.5 & 0 & 0 \\ 0 & 1.0 & 0 \\ 0 & 0 & 0.75 \end{bmatrix} \tag{33}$$

with $u = \frac{5 \cdot \pi}{12}$ and $v = \frac{2 \cdot \pi}{9}$. Then

$$\begin{aligned} k11 &= 0.916361,\ k12 = 0.095755, \\ k13 &= 0.139593,\ k22 = 1.466506, \\ k23 &= 0.080348,\ k33 = 0.867133. \end{aligned} \tag{34}$$

On region *II*, let

$$\mathbf{K}_{II} = \rho \cdot \mathbf{K}_{I} \tag{35}$$

with $\rho > 1$. For our tests, $\rho = 2$. If we now take as the solution $f(x, y) = F(r)$:

$$F_I(r) = \rho \cdot \ln(r) \tag{36}$$

$$F_{II}(r) = \rho \cdot \ln(2) + 0.5 \cdot \ln(r - 1), \tag{37}$$

then the flux continuity condition reduces to

$$[\mathbf{K} \quad \mathbf{F}_r]_I = [\mathbf{K} \quad \mathbf{F}_r]_{II}. \tag{38}$$

This is automatically satisfied by the choice of $F(r)$. Furthermore, $F(r)$ is continuous and differentiable (even at $r = 2$), as is needed if F is to represent pressure or head.

The Boundary Conditions

The general Robin boundary condition is

$$\alpha \cdot \partial f / \partial n + \beta \cdot f = \gamma \tag{39}$$

for each (continuum) point on the boundary of domain. For each of the four boundaries, the user selects from the following choices.

A. All Dirichlet: $\alpha = 0, \beta = 1$
B. All Neumann: $\alpha = 1, \beta = 0$
C. All Robin: $\alpha = 1, \beta = 1$

D. Random: $0 < \alpha, \beta < 1$

(D uses α and β varying randomly along a boundary.)

The γ is then computed, based on the user's choice of solution function, f.

A. $f = 1$ on all of domain ($g = 0$ for Laplace Operator) (40)

B. $$f = \sin(\pi \cdot x) \cdot \sin(\pi \cdot y) \sin(\pi \cdot z) \quad (41)$$

C. $f = F(r)$ for discontinuous tensor. (42)

The source term g is also computed from f, using $g = Lf$. g and f are computed at cell centers, while γ is computed on boundary cell faces. For the discontinuous tensor case, $F(r)$ given in (36-37) must be selected.

Interface Conductivity Evaluation

Finally, the user-can define PDE coefficients at the cell interfaces either by linear or harmonic averages.

Results

Several bugs in the 2-D codes were uncovered during the course of exercising these benchmarks.

(i) The coefficient averaging procedure was not defined on the physical boundary of the problem, giving zero coefficient values there. This was easily remedied by changing the way in which the metric coefficients were computed in the stencil-loader.

(ii) An error in the point-SOR routine at the corner stencils prevented 2nd order convergence there.

(iii) Two of the boundary loops used to compute the right-hand-side arrays in the stencil-loader were indexed over "i" when they should have been indexed over "j", causing incorrect zeros in the right hand side array.

(iv) The corner stencil formula, e.g., at the lower left corner (i, j), was modified from

$$f(i,j) = \left[f(i,j+1) + f(i,j-1) \right] / 2 \quad (43)$$

to

$$f(i,j) = f(i+1,j) + f(i,j-1) - f(i+1,j+1) \quad (44)$$

to preserve 2nd order convergence there. (This also required adding a few lines of code setting the right-hand-side arrays to zero during the computation of the corner stencils).

With the 2-D code fixes built into the n-dimensional Symbolic Manipulation code, the 3-D tests uncovered only one additional bug.

(v) The right-hand-side arrays on the edges of the cube were not properly initialized to zero (being outside the needed loops).

When these items were corrected, the 2-D and 3-D stencil-loader routines were verified to produce 2nd order accurate stencils for problems on general domains using non-orthogonal grids, provided smooth PDE coefficients were used. Solution function (41) was used in all the runs shown herein.

The full set of 3-D problems is shown in Table 3. Linear averaging of the conductivities was used except in Run 10.

TABLE 3. CONDUCTIVITY PROBLEM CONFIGURATIONS.

RUN	Domain	Operator	B.C.
1	Unit Cube	Laplace	Dirichlet
2	Irregular	Laplace	Dirichlet
3	Unit Cube	Continuous	Dirichlet
4	Irregular	Continuous	Dirichlet
5	Unit Cube	Laplace	Robin
6	Irregular	Laplace	Robin
7	Unit Cube	Continuous	Robin
8	Irregular	Continuous	Robin
9	Unit Cube	Discontinuous	Dirichlet
10	Unit Cube	Discontinuous, harmonic ave.	Dirichlet

TABLE 4. RUN # 1.

N	xerr	at	xerr · N^2
5×5×5	0.03356	(3, 3, 3)	0.839
10×10×10	0.00796	(6, 6, 6)	0.796
20×20×20	0.00202	(11, 11, 11)	0.808
40×40×40	0.00051	(21, 21, 21)	0.816
5×5×40	0.02224	(3, 3, 21)	
5×40×5	0.02226	(3, 21, 3)	
40×5×5	0.02226	(21, 3, 3)	

Because of space limitations for this paper, numerical results are given here only for Runs 1, 8, 9 and 10.

The symmetry of the solution and boundary conditions made this Run # 1 ideal to check the case $\Delta x \neq \Delta y \neq \Delta z$. As seen from the results in Table 4 above, the truncation error is the same no matter which direction is refined, and the maximum error occurs in the 21st cell in the direction having 40 cells. Therefore, we can safely conclude there are no coding errors related to $\Delta x \neq \Delta y \neq \Delta z$.

We note that the 40×40×40 grids were run only after the RAM on the MicroVAX II was increased from 5MB to 16 MB.. The 5 MB configuration produced severe page faulting on the

system, with the result that little CPU time was being accumulated and all the time was being spent on the page-file handling. There are 31 single-precision arrays in the code, each 40×40×40, so approximately 7.9 MB of memory are needed to avoid swapping.

In another experiment, Run # 1 was repeated with the PDE coefficients set to zero in the ring of cells outside the physical boundary to obtain some idea of the effect of extrapolating these coefficients. The results showed that the accuracy was degraded thoroughly 1st order.

Run # 8 in Table 5 shows the 2nd order convergence of the continuous operator with Robin boundary conditions on an irregular domain, exercising all the metric terms in the 3-D boundary fitted coordinate transformation.

TABLE 5. RUN # 8.

N	xerr	$xerr^2$
5	0.06970	1.743
10	0.02119	2.119
20	0.00559	2.236
40		

Runs 9 and 10 in Tables 6 and 7 show the convergence of the discontinuous coefficient problem on the unit cube, for linear and harmonic averaging of conductivities.

TABLE 6. RUN # 9: LINEAR AVERAGING FOR K.

N	xerr	xerr · N
5	0.05395	0.27
10	0.02808	0.28
20	0.01442	0.29
40		

TABLE 7. RUN # 10: HARMONIC AVERAGING FOR K.

N	xerr	xerr · N
5	0.05906	0.29
10	0.02079	0.21
20	0.01301	0.26
40	0.00688	0.28

The theoretical basis for harmonic averaging is well established (e.g. see McDonald and Harbaugh, 1988) in 1-D. We found 2nd order convergence in 1-D provided that the discontinuity stayed on a cell interface as the grid was refined, and that harmonic averaging was used; linear averaging produced only 1st order convergence. However, our nonorthogonal grid multidimensional results in Tables 6 and 7 above are not impressive. For a discontinuous k with

ratio $\rho = 2$ (which is mild by geologic standards), 3-D harmonic averaging is slightly worse than linear averaging at N = 5, and slightly better at higher resolution. Both certainly give a consistent discretization, but both are only 1st order accurate. The order of convergence was not restored to 2nd order by aligning the grid with the discontinuity.

PARTICLE TRACKER CODES

The particle tracker codes developed by Ecodynamics are incorporated into the SECO suite of codes, but are also used with the output of other groundwater hydrology codes. In fact, they are not limited at all to Darcy flow, but work with any steady or time-dependent velocity fields. 2-D and 3-D cartesian versions are available, and like the SECO codes allow user selection of collocate variable grids (with mac = 0) or a staggered MAC or Arakawa "C" grid (with mac = 1).

The SECO tracker codes use linear (2nd order) interpolation of the velocity fields, and 5th order Runge-Kutta-Fehlberg integration in time using a modified version of RKF45 (Shampine et al., 1976).

Spatial Convergence of Tracker Codes

The first benchmark test case for the Tracker codes is designed to verify spatial convergence. The benchmark solution is a 1-D trajectory. The 1-D velocity field varies in space and time as

$$|V| = 0.5 \cdot \cos(time) / \cos(x). \tag{45}$$

The 1-D solution is

$$x\_part = \mathrm{asin}[0.5 \sin(time)] \tag{46}$$

which solution has sufficient structure to exercise all terms in the Taylor series expansion of the discretization error. The solution is rotated with the direction cosine $dcos1 \leq 1$, and $dcos2 = sqrt\,[1 - dcos1^2]$. Using $dcos1 = 0.5$ gives $\alpha 1 = 60°$, so the trajectory is not through the grid corners.

TABLE 8. 2-D TRACKER RESULTS FOR TEST #1, UNIFORM GRID, MAC = 0.

il×jl	relerr_p (%)	relerr_coef	error ratio
5×5	2.88 E+00 %	4.61 E-01	
9×9	5.74 E-01 %	3.68 E-01	5.0
17×17	6.97 E-02 %	1.78 E-01	8.2
33×33	5.06 E-02 %	5.18 E-01	1.4

The term relerr_p is the relative error in final position, and relerr_coef is calculated as relerr_p·$(il - 1)^2$. For a method which is 2nd order accurate in space, this coefficient should become asymptotically constant as *il* is increased. The time discretization error is $O(t^5)$ and is negligible compared to the spatial discretization error for this problem.

As seen in Table 8 above, this term is indeed roughly constant. (The slight decrease in relerr_coef as the grid is refined would theoretically indicate faster than 2nd order convergence, but this distinction is not significant, and is evidence of less than asymptotic behavior, round-off error, and time discretization error.)

The successive ratios of relative error in particle position are displayed in the last column of Table 8. The ratio of successive errors as the spatial step is halved, from (il-1) = N = 4, 8, 16, is better than the theoretical value for the 2nd order method, $2^2 = 4$. This test of 2nd order spatial discretization error is not as sensitive to computer round-off error as is the 5th order time discretization test. The better-than-theoretical performance will not generally hold for arbitrary velocity fields. The failure at N = 32 is indicative of round-off error.

The above test results are obtained for collocated grids (mac = 0); similar results are obtained for a staggered (Marker-and-Cell or Arakawa-C grid; mac = 1) in Table 9, and for a stretched grid in Table 10.

TABLE 9. 2-D TRACKER RESULTS FOR TEST #1, UNIFORM GRID, MAC = 0.

il×jl	relerr_p (%)	relerr_coef
5×5	1.6238 E+00 %	2.5981 E-01
9×9	4.0202 E-01 %	2.5729 E-01
17×17	5.3066 E-02 %	1.3585 E-01
33×33	5.1116 E-02 %	5.2343 E-01

TABLE 10. 2-D TRACKER RESULTS FOR TEST #1, STRETCHED GRID, MAC = 0.

il×jl	relerr_p (%)	relerr_coef
5×5	4.4230 E+00 %	7.0768 E-01
9×9	7.6609 E-01 %	4.9030 E-01
17×17	8.4578 E-02 %	2.1652 E-01
33×33	4.7120 E-02 %	4.8250 E-01

Temporal Convergence of Tracker Codes

The second benchmark test case for the particle tracker validates the temporal accuracy. A solid-body rotation velocity field with rotational speed = $2 \cdot \pi$ radians/unit time should return the particle to its original position at final time = 1.

For an initial position at *x_part* =(0.75, 0.0) the following errors in the final x-position at N time steps are obtained.

TABLE 11. 2-D TRACKER RESULTS FOR TEST #2, UNIFORM GRID, MAC = 0.

N	error in x_part	error ratio
10	4.9E-02	
20	1.7E-03	28.8
40	5.5E-05	30.9
80	4.0E-05	1.4

The ratio of successive errors as the time step is halved, from (il-1) = N = 10, 20, 40, very closely fits the theoretical value for the 5-th order method, $2^5 = 32$. The failure from N = 40 to 80 is indicative of accumulated round-off error using the single precision version of RKF45 on a Vax with about 7-8 significant figures of accuracy for floating point calculations. (The single precision version was developed for later conversion to the Cray-XMP, for which any use of double precision variables prevents vectorizing.) Single precision is expected to be adequate for our anticipated applications, even using a Vax or equivalent computer.

Although this benchmark test case #2 does verify the 5th order time discretization error, there is no spatial interpolation error, since the velocity field for solid body rotation is linear in space.

This is an important distinction. In actual application, a serious coding error was not detected by this seemingly convincing benchmark test case, i.e., we obtained a "false negative" error test. The code logic which located the cell indexes containing the particle position was in error. However, since the solid body rotation gives $u = - cy$ and $v = + cx$, i.e. a linear variation in velocity components, the linear interpolation to find $u\ (x, y, t)$ and $v\ (x, y, t)$ is algebraically exact, no matter what cell is used as the basis for the interpolation. (The wrong cell merely results in linear *extrapolation*, which is still algebraically exact.) Only when the previous benchmark test case #1, involving the rotated 1-D trajectory, was exercised was the coding error discovered.

Similar results are obtained from the 3-D tracker codes. The benchmark test case is the 3-D analog of the first 2-D problem above. The same 1-D solution (46) is rotated with direction cosines $dcos1$, $2 \le 1$, and $dcos3 = sqrt\ [1 - dcos1^2 - dcos2^2]$. Using $dcos1 = 0.5$ and $dcos2 = 0.6$ gives $\alpha 1 = 60°$ and $\alpha 2 = 53.13°$, so the trajectory is not through the grid corners. Results are shown in Tables 12 and 13.

TABLE 12. 3-D TRACKER RESULTS FOR TEST #1, UNIFORM GRID, MAC = 0.

il = jl = kl	relerr_p (%)	relerr_coef =
5	2.8987 E+00 %	4.6379 E-01
9	5.2918 E-01 %	3.3868 E-01
17	6.5718 E-02 %	1.6824 E-01
33	5.0346 E-02 %	5.1555 E-01

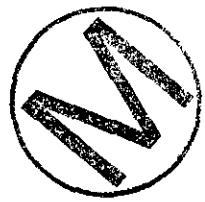


TABLE 13. 3-D TRACKER RESULTS FOR TEST #2, UNIFORM GRID, MAC = 0.

N	relerr_p	N error ratio
4	2.8786	
8	0.5292	5.4
16	0.0667	7.9
32	0.0503	1.3

Presently under development are particle tracker codes for non-orthogonal grids, which codes utilize bi-linear and tri-linear interpolations (as commonly used in isoparametric finite element methods). These codes will be subjected to the same benchmark test cases.

SUMMARY

Experience with formulating and applying several benchmark test cases in groundwater hydrology CFD problems has been described. The problems are all single phase, Darcy law flows. Three problem categories were considered: (1) full 2-D groundwater hydrology problem codes tested on a steady problem with scalar conductivities in stretched cartesian coordinates; (2) Fortran subroutines produced by computer Symbolic Manipulation for the stencil array evaluation of tensor conductivity in general non-orthogonal 2-D and 3-D coordinates; (3) particle tracking in 2-D and 3-D. Freedom from coding errors, consistency of the discretization, and order of convergence were verified. Examples were given of successful code verification, successful error detection, and unsuccessful "false negative" tests.

ACKNOWLEDGEMENTS

This work was supported under a DOE contract from Sandia National Laboratories, Albuquerque, Dr. M. Marietta, Contract Monitor.

REFERENCES

Kipp, K. L., Jr., 1987, "HST3D: A Computer Code for Simulation of Heat and Solute Transport in Three-Dimensional Ground-Water Flow Systems", U. S. Geological Survey Water Investigations Report 86-4095, USGS, Denver, Colorado.

McDonald, M. G. and Harbaugh, W., 1988, "A Modular Three-Dimensional Finite-Difference Ground-Water Flow Model", Techniques of Water-Resources Investigations of the United States Geological Survey, Book 6, Modeling Techniques, Chapter A1. Scientific Software Group, P.O. Box 23401, Washington, D.C. 20026.

Reeves, M., Ward, D. S., Johns, N. D., and Cranwell, R. M., 1986, "Theory and Implementation for SWIFT II The Sandia Waste-Isolation Flow and Transport Model for Fractured Media", SAND83-1159, Sandia National Laboratories, Albuquerque, New Mexico, 1986.

Shampine, L. F., Watts, W. A., and Davenport, S., 1976, "Solving Non-stiff Ordinary Differential Equations -The State of the Art", *Siam Review*, Vol. 18, No. 3, pp. 376-411, July 1976.

Roache, P. J. and Steinberg, S., 1984, "Symbolic Manipulation and Computational Fluid Dynamics", *AIAA Journal*, vol. 22, No. 10, October 1984, pp. 1390-1394.

Roache, P. J., Baker, B., Blaine, C. T., Blaine, R. L., Knupp, P. M., and Steinberg, S., 1990, "SECO Codes Version 1.0", Ecodynamics Research Associates, to be published.

Steinberg, S. and Roache, P. J., 1985, "Symbolic Manipulation and Computational Fluid Dynamics", *Journal of Computational Physics*, Vol. 57, pp. 251-284.

Steinberg, S. and Roache, P., 1986a, "A Toolkit of Symbolic Manipulation Programs for Variational Grid Generation", AIAA Paper 86-0241, AIAA 24th Aerospace Sciences Meeting.

Steinberg, S. and Roache, P., 1986b, "Using MACSYMA to Write FORTRAN Subroutines", *Journal of Symbolic Computation*, Vol. 2, pp. 213-216.

Steinberg, S. and Roache, P., 1990, "Symmetric Operators in General Coordinates," Ecodynamics Research Associates. To be published.

APPENDIX B. VERIFICATION OF TEMPORAL NUMERICAL ACCURACY

This appendix presents the verification of the temporal numerical accuracy of the SECOFL2D code. Since only first order time accuracy is claimed, verification is easy to achieve. Any significant coding error would result in qualitative solution errors (e.g., catastrophic instability) that would be detected by virtually any exercise of the code. For the sake of completeness, this appendix presents a more rigorous demonstration of the order of temporal accuracy, involving a straightforward calculation of a simple problem with an analytical solution. Since the time and space differencing in SECOFL2D are orthogonal, a simple constant-property problem completely exercises the time differencing method.

The constant-property problem chosen is roughly representative of the range of properties used in past WIPP PA calculations (WIPP PA, 1992), though this is not essential to the verification. The spatial domain of 25km x 25 km is resolved using a 50x50 grid of uniform mesh increments. The other physical properties are as follows.

- permeability K = 1.E-8 m^2/sec
- medium's compressibility α = 7.57E-10 Pa^{-1}
- weight density of water (set in a data statement) ρg = 9810 Newtons/m^3
- compressibility of water (set in a data statement) β = 4.4E-10 Pa^{-1}
- porosity ϕ = 0.16
- specific storativity S_S = 8.11679E-06 m^{-1}
- aquifer thickness = 1.0 m
- total simulation time = 10,000 years

The initial condition is set to

$$h(x,y,0) = 100\sin(\pi\frac{x}{x_{max}})\sin(\pi\frac{y}{y_{max}}) + 900 \qquad \text{(B-1)}$$

giving the range of initial $h \in [900,1000]$ meters. Dirichlet boundary conditions of h = 900 are fixed for all time. The exact solution is

$$h_{EXACT}(x,y,t) = 100e^{-Bt}\sin(\pi\frac{x}{x_{max}})\sin(\pi\frac{y}{y_{max}}) + 900 \qquad \text{(B-2)}$$

where

$$B = \frac{K}{S_s}\pi^2\left(\frac{1}{x_{max}^2} + \frac{1}{y_{max}^2}\right) \qquad \text{(B-3)}$$

The half-sine wave spatial structure is adequately resolved by the 50-cell grid, so that temporal discretization errors dominate spatial discretization errors at early times. A time-step

convergence test is performed, and first-order accuracy in time is demonstrated when successive halving of the time steps (or doubling of the number of time steps) reduces the numerical error (in some norm) by a factor of 1/2 asymptotically, i.e., as $\Delta t \rightarrow 0$.

Two error norms are evaluated. The commonly used $L2$ norm includes errors from all spatial grid points in a root-mean-square evaluation.

$$E^{L2} = \sqrt{\sum_{i,j} (h_{i,j} - h_{EXACT})^2} \; / \; IJ \qquad \text{(B-4)}$$

where the summation extends over all interior cells from $i = 1$ to I and $j = 1$ to J. The L∞ norm is based on the maximum error in the grid, the location of which can change at any time level of the solution.

$$E^{L\infty} = \max_{i,J} |h_{i,j} - h_{EXACT}| \qquad \text{(B-5)}$$

It is more demanding to achieve theoretical accuracy for the $L\infty$ norm, since it is sensitive to local areas of high error (e.g., singularities) and because the location of the maximum error can move as the time step is refined (until the asymptotic range is reached).

Because of the exponential decay of the exact solution, it is to be expected that the theoretical asymptotic performance will be more difficult to attain at early solution times, when the higher time derivatives of the solution are larger.

Table B-1a,b presents the results of this time-step convergence test as the time step is successively halved from 1,000 years. Both the $L2$ and $L\infty$ norms are presented, at two different solution times of 1,000 and 5,000 years. The ratio of successive errors should approach the theoretical value of 2. The Table shows this behavior. As expected, the $L2$ norm is better behaved than the $L\infty$ norm, and the later time solutions are better behaved (the ratio approaching the limit of 2 from above) than the early time solutions (which approach the ratio limit of 2 from below). Thus, the first-order temporal convergence of SECOFL2D is demonstrated.

1000 Years 50×50 Grid

Δt (years)	L2 norm	L2 ratio	L∞ norm	L∞ ratio
1000	0.1557381		15.64324 @ 25,28	
500	9.074315 E-02	1.72	9.148818 @ 25,28	1.71
250	4.956384 E-02	1.83	5.028010 @ 25,29	1.82
125	2.598208 E-02	1.91	2.674403 @ 25,30	1.88
62.5	1.335881 E-02	1.94	1.423468 @ 25,31	1.88

Table B-1a. Temporal Convergence of the SECOFL2D code at 1000 years.

5000 Years 50×50 Grid				
Δt (years)	L2 norm	L2 ratio	L∞ norm	L∞ ratio
1000	1.604517 E-02		1.603225 @ 25,26	
500	6.177462 E-03	2.59	0.6171672 @ 25,26	2.60
250	2.570141 E-03	2.40	0.2567643 @ 26,26	2.40
125	1.150919 E-03	2.23	0.1149791 @ 26,26	2.23
62.5	5.417560 E-04	2.12	0.0541224 @ 26,26	2.12

Table B-1b. Temporal Convergence of the SECOFL2D code at 5000 years.

APPENDIX C. STAND-ALONE VERSION PROGRAM STRUCTURE

This appendix contains a detailed description of the input files that are necessary to run SECOFL2D as a stand alone code (i.e., not in the CAMCON system). The information here applies to a previous version of the code that did not employ the FORTRAN NAMELIST capability. The difference in the older version and the current NAMELIST version is mostly formatting. The required input variables are the same for both versions. The older version required a set number of lines, and certain blocks of lines were designated for comments. In the current version, comments can appear anywhere and there is no required number of lines. A comment line has an exclamation point as the first character of the line. The NAMELIST version has eliminated the need for separate input files for steady and unsteady simulations. This appendix will be updated to reflect the current version of the code at future time.

1. THE SET_AQFR2 INPUT FILE

This section describes how the SECO user can create the input file aqfr2.inp for set_aqfr2. Changes in this input file are made by editing the pre-existing template or by creating a file from scratch.
The code set_aqfr2 is intended to be used only to create model aquifer data bases. In actual simulations, the data base will be obtained by other means, e.g. the CAMCON data base, but it must be output in the same format as the output of set_aqfr2 in order to be used by the SECO codes. The actual numerical values of the base aquifer parameters are set in data statements in set_aqfr2. These may be accessed by the VMS command DEPOSIT, or the FORTRAN source code can be modified, in which case the compile and link commands FOR set_aqfr2 and @LAQ2 are required before running.

1.1 INPUT FILE LINES

The following nine lines are required:

title line (line 1) - This line is the title of the input run file.
IO_SCREEN (line 2) - This value (between 0 and 4) determines the amount of data actually output to the screen. 0 is used for no screen output while 4 is used for maximum screen output.

block delineator (line 3) - This line specifies that the following block of input values are REQUIRED.

AQFR_ID (line 4) - An arbitrary aquifer identification number specified by any real number.

IDFIL_AQ (line 5) - The output file suffix; it is used to determine to which data file the aquifer grid will be saved, e.g. "10" saves aquifer grid to file FOR010.DAT. The recommended range of values is from 10-19 while 0 does not save.

blank or comment line (line 6) - This line is necessary due to the format of the input file template.

ilAQ, jlAQ (line 7) - The array dimensions of the aquifer grid. Dimensions must be less than or equal to those dimensions specified in the INCLUDE file aqfr2.par.

blank or comment line (line 8)

xlAQ, ylAQ (line 9) - The size of the aquifer grid in meters. The lengths of the grid are specified in the x and y directions.

The following lines are optional and are used only to define specific conditions within the aquifer grid:

block delineator (line 10) - This line specifies that the following block of input values are OPTIONAL. If included, the values must be listed in the given sequence. Any number of the blocks may be used.

PRECip (line 11) - This line specifies that a (net) precipitation recharge region is to be set. (only the first four characters of PRECip are required)

i1,i2, j1,j2 (line 12) - The array indexes used to define the region of precipitation recharge.

rch_rate (line 13) - The recharge rate in cm/year (>0) which is used to calculate q_well < 0.

block delineator (line 14) - This line specifies that the following block of input values are OPTIONAL. If included, the values must be listed in the given sequence. Any number of the blocks may be used.

RIVEr (line 15) - This line specifies that a river or lake recharge region is to be set. (only the first four characters of RIVEr are required)

i1,i2, j1,j2 (line 16) - The array indexes used to define the region of the river or lake.

river_conductance (line 17) - The river conductance specified in meters**2/second.

river_head (line 18) - The river head specified in meters.

river_bottom (line 19) - The river bottom specified in meters.

block delineator (line 20) - This line specifies that the following block of input values are OPTIONAL. If included, the values must be listed in the given sequence. Any number of the blocks may be used.

INACtive (line 21) - This line specifies that an inactive region (constant head AND no flux) is to be set. (only the first four characters of INACtive are required)

ABSO or INCR (line 22) - Used to set absolute head (ABSOlute) which over-writes the base data, or used to set increment in head (INCRement) which accumulates on top of the base data and previous blocks.

blank or comment line (line 23)

i1,i2, j1,j2 (line 24) - The array indexes used to define the region of the inactive cells.

con_head (line 25) - The constant head value, measured in meters.

block delineator (line 26) - This line specifies that the following block of input values are OPTIONAL. If included, the values must be listed in the given sequence. Any number of the blocks may be used.

CONHead (line 27) - This line specifies that a constant head region is to be set. (only the first four characters of CONHead are required)

ABSO or INCR (line 28) - Used to set absolute head (ABSOlute) which over-writes the base data, or used to set increment in head (INCRement) which accumulates on top of the base data and previous blocks.

blank or comment line (line 29)

i1,i2, j1,j2 (line 30) - The array indexes used to define the regions of constant head.

con_head (line 31) - The constant head value, measured in meters.

block delineator (line 32) - This line specifies that the following block of input values are OPTIONAL. If included, the values must be listed in the given sequence.

TERMinate (line 33) - This line is included if the read is to be terminated in which case the remaining cells are normal.
(only the first four characters of TERMinate are required)

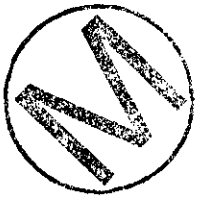
Note: in using the blocks PRECip, RIVEr, INACtive, and/or CONHead, some areas of the grid may be overlapped by consecutive blocks. In such cases, the previous block specification will be overwritten. For example, for a CONHead block with (i1,i2,j1,j2) = (5,10,2,20) followed by a RIVEr block with (i1,i2,j1,j2) = (1,8,10,30), the overlapped area (5,8,10,20) will be set as a RIVEr, not CONHead.

2. THE SET_GREG2 INPUT FILE

This section describes how the SECO user can create the input file greg2.inp for set_greg2. Changes in this input file are made by editing the pre-existing template or by creating a file from scratch.

2.1 INPUT FILE LINES

The following twenty-four lines are required:



title line (line 1) - This line is the title of the input run file.

IO_SCREEN (line 2) - This value (between 0 and 4) determines the amount of data actually output to the screen. 0 is used for no screen output while 4 is used for maximum screen output.

block delineator (line 3) - This line specifies that the following block of input values is REQUIRED.

GREG_ID (line 4) - An arbitrary regional grid identification number specified by any real number.

blank or comment line (line 5) - This line is necessary due to the format of the input file template.

blank or comment line (line 6)

blank or comment line (line 7)

IDFIL_AQ (line 8) - The input file suffix; it is used to determine from which file the aquifer grid will be read, e.g. "10" reads from the file FOR010.DAT.

blank or comment line (line 9)

x0, x1 (line 10) - The x-size of the regional grid, expressed in meters.

y0, y1 (line 11) - The y-size of the regional grid, expressed in meters.

x1_rel, y1_rel, thet1_rel (line 12) - These are the offsets of the regional grid from the aquifer-defining grid. The displacements in the x-direction, y-direction, and rotation angle are expressed in meters and degrees respectively.

blank or comment line (line 13)

L-replicate (line 14) - This refers to whether the aquifer-defining grid is to be replicated into the regional grid, e.g. ".true." WILL replicate whereas ".false." WILL NOT replicate.

IDFIL_GREG (line 15) - The output file suffix; it is used to determine to which data file the regional grid will be saved, e.g. "20" saves regional grid to file FOR020.DAT. The recommended range of values is from 20-29 while 0 does not save.

blank or comment line (line 16)

NOPT_X (line 17) - This is used to determine how the delta-x's will change throughout the grid. This is done using a numerical value from 0-3.
0 - will automatically set constant delta-x's.
1 - will automatically set delta-x's by packing around a location.
2 - will allow delta-x's to be entered by list.

3 - will allow x-coordinates to be entered by list.

For NOPT_X = 1,2 or 3 a separate block of optional lines must be entered (see below).

blank or comment line (line 18)

blank or comment line (line 19)

blank or comment line (line 20)

The following set of lines are included for NOPT_X values of 1,2 or 3:

For NOPT_X = 1, only one line is necessary:

x_loc, x_power - These specify the x-location and the power to which the delta-x's are packed; need x0 < x_loc < x1 and 0 <= x_power <= 5.0.
Larger values of x_power increase the packing density, while x_power = 0 gives constant delta-x.

For NOPT_X = 2, the number of lines necessary corresponds to the total number of x-nodes:

ilGR + 1 pairs of I, x-increment - These specify the x-increment and its change from node to node. An x-increment value must be specified for each of the ilGR + 1 pairs.

For NOPT_X = 3, the number of lines necessary corresponds to the total x-dimension:

ilGR pairs of I, x-coordinate - These specify the I array index and the corresponding x-coordinate directly. A value must be given for each of the ilGR pairs.

NOPT_Y (line 21) - This is used to determine how the delta-y's will change throughout the grid. This is done using a numerical value from 0-3.
0 - will automatically set constant delta-y's.
1 - will automatically set delta-y's by packing around a location.
2 - will allow delta-y's to be entered by list.
3 - will allow y-coordinates to be entered by list.

For NOPT_Y = 1,2 or 3 a separate block of optional lines must be entered (see below).

blank or comment line (line 22)

blank or comment line (line 23)

blank or comment line (line 24)

The following set of lines are included for NOPT_Y values of 1,2 or 3:

For NOPT_Y = 1, only one line is necessary:

y_loc, y_power - These specify the y-location and the power to which the delta-y's are packed; need y0 < y_loc < y1 and 0 <= y_power <= 5.0.
Larger values of y_power increase the packing density, while y_power = 0 gives constant delta-y.

For NOPT_Y = 2, the number of lines necessary corresponds to the total number of y-nodes:

jlGR + 1 pairs of J, y-increment - These specify the y-increment and its change from node to node. A y-increment value must be specified for each of the jlGR + 1 pairs.

For NOPT_Y = 3, the number of lines necessary corresponds to the total y-dimension:

jlGR pairs of J, y-coordinate - These specify the J array index and the corresponding y-coordinate directly. A value must be given for each of the jlGR pairs.

3. THE SET_GLOC2 INPUT FILE

This section describes how the SECO user can create the input file gloc2.inp for set_gloc2. Changes in this input file are made by editing the pre-existing template or by creating a file from scratch.

3.1 INPUT FILE LINES

The following twenty-four lines are required:

title line (line 1) - This line is the title of the input run file.

IO_SCREEN (line 2) - This value (between 0 and 4) determines the amount of data actually output to the screen. 0 is used for no screen output while 4 is used for maximum screen output.

block delineator (line 3) - This line specifies that the following block of input values is REQUIRED.

GLOC_ID (line 4) - An arbitrary local grid identification number specified by any real number.

blank or comment line (line 5) - This line is necessary due to the format of the input file template.

blank or comment line (line 6)

blank or comment line (line 7)

IDFIL_GR (line 8) - The input file suffix; it is used to determine from which file the regional grid will be read, e.g. "20" reads from the file FOR020.DAT.

blank or comment line (line 9)

x0, x1 (line 10) - The x-size of the regional grid, expressed in meters.

y0, y1 (line 11) - The y-size of the regional grid, expressed in meters.

x1_rel, y1_rel, thet1_rel (line 12) - These are the offsets of the local grid from the regional grid. The displacements in the x-direction, y-direction, and rotation angle are expressed in meters and degrees respectively.

blank or comment line (line 13)

L-replicate (line 14) - This refers to whether the regional grid is to be replicated into the local grid, e.g. ".true." WILL replicate whereas ".false." WILL NOT replicate.

IDFIL_GLOC (line 15) - The output file suffix; it is used to determine to which data file the local grid will be saved, e.g. "50" saves local grid to file FOR050.DAT. The recommended range of values is from 50-59 while 0 does not save.

blank or comment line (line 16)

NOPT_X (line 17) - This is used to determine how the delta-x's will change throughout the grid. This is done using a numerical value from 0-3.
0 - will automatically set constant delta-x's.
1 - will automatically set delta-x's by packing around a location.
2 - will allow delta-x's to be entered by list.
3 - will allow x-coordinates to be entered by list.

For NOPT_X = 1,2 or 3 a separate block of optional lines must be entered (see below).

blank or comment line (line 18)

blank or comment line (line 19)

blank or comment line (line 20)

The following set of lines are included for NOPT_X values of 1,2 or 3:

For NOPT_X = 1, only one line is necessary:

x_loc, x_power - These specify the x-location and the power to which the delta-x's are packed; need x0 < x_loc < x1 and 0 <= x_power <= 5.0. Larger values of x_power increase the packing density, while x_power = 0 gives constant delta-x.

For NOPT_X = 2, the number of lines necessary corresponds to the total number of x-nodes:

ilGR + 1 pairs of I, x-increment - These specify the x-increment and its change from node to node. An x-increment value must be specified for each of the ilGR + 1 pairs.

For NOPT_X = 3, the number of lines necessary corresponds to the total x-dimension:

ilGR pairs of I, x-coordinate - These specify the I array index and corresponding x-coordinate directly. A value must be given for each of the ilGR pairs.

NOPT_Y (line 21) - This is used to determine how the delta-y's will change throughout the grid. This is done using a numerical value from 0-3.
0 - will automatically set constant delta-y's.
1 - will automatically set delta-y's by packing around a location.
2 - will allow delta-y's to be entered by list.
3 - will allow y-coordinates to be entered by list.

For NOPT_Y = 1,2 or 3 a separate block of optional lines must be entered (see below).

blank or comment line (line 22)

blank or comment line (line 23)

blank or comment line (line 24)

The following set of lines are included for NOPT_Y values of 1,2 or 3:

For NOPT_Y = 1, only one line is necessary:

y_loc, y_power - These specify the y-location and the power to which the delta-y's are packed; need y0 < y_loc < y1 and 0 <= y_power <= 5.0. Larger values of y_power increase the packing density, while y_power = 0 gives constant delta-y.

For NOPT_Y = 2, the number of lines necessary corresponds to the total number of y-nodes:

jlGR + 1 pairs of J, y-increment - These specify the y-increment and its change from node to node. A y-increment value must be specified for each of the jlGR + 1 pairs.

For NOPT_Y = 3, the number of lines necessary corresponds to the total y-dimension:

jlGR pairs of J, y-coordinate - These specify the J array index and the corresponding y-coordinate directly. A value must be given for each of the jlGR pairs.

4. THE SECO_2 INPUT FILE

This section describes how the SECO user can create the input file seco_2.inp for seco_2. Changes in this input file are made by editing a pre-existing template or by creating a file from scratch.

4.1 INPUT FILE LINES

The following twenty lines are required:

title line (line 1) - This line is the title of the input run file.

IO_SCREEN (line 2) - This value (between 0 and 4) determines the amount of data actually output to the screen. 0 is used for no screen output while 4 is used for maximum screen output.

UNSTeady or STEAdy (line 3) - This line describes whether the problem is solved as an unsteady (UNSTeady) problem, i.e. transient, time dependent, or as a steady state (STEAdy) problem.

IDFIL_GREG (line 4) - The input file suffix; it is used to determine from which data file the regional grid is read, e.g. "20" reads regional grid from file FOR020.DAT. The recommended range of values is from 20-29.

blank or comment line (line 5) - This line is necessary due to the format of the input file template.

IDFIL_WELL (line 6) - The output file suffix; it is used to determine to which data file the well data will be saved, e.g. "30" saves well data to file FOR030.DAT. The recommended range of values is from 30-39 while 0 does not save.

blank or comment line (line 7)

IDFIL_2R (line 8) - The output file suffix; it is used to determine to which data file the regional flow solution will be saved, e.g. "40" saves regional flow solution to file FOR040.DAT. The recommended range of values is from 40-49 while 0 does not save.

blank or comment line (line 9)

IDFIL_GLOC (line 10) - The input file suffix; it is used to determine from which data file the local grid is read, e.g. "50" reads local grid from the file FOR050.DAT. The recommended range of values is from 50-59.

blank or comment line (line 11)

IDFIL_2LB (line 12) - The output file suffix; it is used to determine to which data file the local grid boundary conditions will be saved, e.g. "60" saves local grid boundary conditions to file FOR060.DAT. The recommended range of values is from 60-69 while 0 does not save.

blank or comment line (line 13)

IDFIL_2L (line 14) - The output file suffix; it is used to determine to which data file the local flow solution will be saved, e.g. "70" saves local flow solution to the file FOR070.DAT. The recommended range of values is from 70-79 while 0 does not save.

blank or comment line (line 15)

L_run_seco_2r (line 16) - This line determines whether the regional grid flow solution will be run in seco_2r, e.g. ".true." WILL run a new solution and ".false." WILL NOT run a new solution. If ".false.", the subsequent runs will use whatever previous seco_2r output is available.

L_run_set_ibc_2l (line 17) - This line determines whether set_ibc_2l will be run, e.g. ".true." WILL run and ".false." WILL NOT run. The code interpolates the (possibly time-dependent) interior solution from seco_2r to (possibly time-dependent) boundary conditions for the local grid flow solution in seco_2l.

L_run_seco_2l (line 18) - This line determines whether the local grid flow solution is run in seco_2l, e.g. ".true." WILL run a new solution and ".false." WILL NOT run a new solution.

L_run_tracker_2l (line 19) - This line determines whether the local grid tracker is run in tracker_2l, e.g. ".true." WILL run tracker and ".false." WILL NOT run tracker.

L_run_tracker_2r (line 20) - This line determines whether the regional grid tracker is run in tracker_2r, e.g. ".true." WILL run tracker and ".false." WILL NOT run tracker.

5. SECO_2R_UNST.INP INPUT FILE

This section describes how the SECO user can create the input file seco_2r_UNST.inp for seco_2r. Changes in this input file are made by editing the pre-existing template or by creating a file from scratch.

5.1 INPUT FILE LINES

The following sixty-seven lines are required:

title line (line 1) - This line is the title of the input run file.

block delineator (line 2) - This line specifies that the following block of input values are REQUIRED.

FLOW_REG_ID (line 3) - An arbitrary regional flow identification number specified by using any real number.

blank or comment line (line 4) - This line is necessary due to the format of the input template file.

blank or comment line (line 5)

blank or comment line (line 6)

L_def_num (line 7) - This line determines whether seco_2r uses default values of numerical parameters OR replaces data-set default parameters for numerical methods by reading new data from the file methods_2r.inp, e.g. ".true." uses default values and ".false." replaces data-set default parameters.

blank or comment line (line 8)

blank or comment line (line 9)

L_water_table (line 10) - This line determines whether water table conditions are used, e.g. ".true." WILL use water table conditions and ".false." WILL NOT use water table conditions, i.e. the aquifer is always treated as a confined aquifer.

L_harmonic (line 11) - This line determines whether harmonic averaging is used for hydraulic conductance, e.g. ".true." WILL use harmonic averaging and ".false." WILL NOT use harmonic averaging but rather linear averaging.

L_ic_ss (line 12) - This line determines whether the initial conditions will be re-set by the steady state solution with time-dependent wells turned off, e.g. ".true." WILL re-set and ".false." WILL NOT re-set. This is a dummy input for the STEAdy option. (By "dummy" input, it is meant that some input is required by the code format, but the actual value is not used.)

blank or comment line (line 13)

block delineator (line 14) - This line specifies that the following block of input values are required for UNSTeady option only. This block is read in subroutine set_dt.

YEARs or SECOnds (line 15) - This line specifies the time units, measured in years (YEARs) or seconds (SECOnds).

ntimeL (line 16) - This line specifies the number of output time steps (max 50).

time0, timeL (line 17) - This line specifies the initial time (time0) and the final time (timeL) in units consistent with n_timeL above.

N_TIMOPT (line 18) - This value sets the change in simulation time (dt).
0 - sets the simulation times at a constant dt.
1 - sets times by packing dt around an event time; this option requires (ntime1 >= 5).
2 - used to enter dt's directly [**ntimeL pairs of n, dtime(n)**].

3 - used to enter times directly [**ntimeL + 1 pairs of n, time(n)**].

For N_TIMOPT = 1,2 or 3, the necessary input values (listed above) must be entered accordingly. Therefore, additional lines may need to be added here.

blank or comment line (line 19)

blank or comment line (line 20)

blank or comment line (line 21)

blank or comment line (line 22)

block delineator (line 23) - This line specifies that the following block of input value is REQUIRED. This block is read in the set_wells_r subroutine.

n_wells (line 24) - This line specifies the total number of wells in the simulation.

The following lines are optional and are used to define specific well conditions only if there are n_wells > 0. A block of input values must be entered as follows for EACH of the n_wells:

block delineator - This line specifies that the following block of input values is required for n_wells > 0 only. This block is read in the well_ramp subroutine.

WELL_ID - This is an arbitrary real number designator for the well scenario.

L_wells_off - This line determines whether the well entries are actually used. It allows easy turn-off of all wells, but the entries below are still read, e.g. ".true." WILL turn off all wells and ".false." WILL NOT turn off wells.

blank or comment line

n_well - Labels a specific well by using a number from 1 to n_wells.

VQMAX - This line specifies the maximum volumetric flow rate of fluid leaving the well, measured in meters**3/1000 seconds. Note that 10 gallons/min is equal to 0.631 x 10**-3 meters**3/second; i.e., VQMAX = 0.631 gives a well flow rate of about 10 gallons/min. VQMAX is >0 for a production well and <0 for an injection well.

T0f - This line specifies the start time the well will be ramped on.

T1f - This line specifies the end time of the well ramp-on time interval.

T2f - This line specifies the start time the well will be ramped off.

T3f - This line specifies the end time of the well ramp-off time interval.

x_well - This line specifies the x-location measured in meters on the regional grid, NOT on the aquifer grid.

y_well - This line specifies the y-location measured in meters on the regional grid, NOT on the aquifer grid.

z_well - this line specifies the z-location measured in meters on the regional grid, NOT on the aquifer grid.

block delineator (line 25) - This line specifies that the following block of input values are REQUIRED. The block is read in the bc_flags2 subroutine.

The following block of sixteen comment lines are included in the input file template to describe the format of the boundary condition section:

blank or comment lines (lines 26–41)

spec (line 42) - This line specifies to which boundary the following data applies; i.e., it specifies the LOWEr, UPPEr, LEFT, or RIGHt boundary of the regional grid. When these input parameters are entered, they must be entered in the given order (LOWEr, UPPEr, LEFT, RIGHt). Also, this parameter must be entered starting in column 1. (ONLY the first four characters of the word are necessary)

n_sections (line 43) - This line specifies the number of sections for a given boundary.

sec_end (line 44) - This line specifies the position in meters where a given boundary ends.

type (line 45) - This line specifies the type of boundary condition for a section n. Boundary conditions are of three types: HEAD (Dirichlet), GRADient (Neuman), and FLUX (Cauchy). Specifying the type is done by entering the desired condition preceded by four spaces. (ONLY the first four characters of the word are necessary)

where (line 46) - This line specifies where the value is determined. "I.C." uses the value from the initial condition file and "IVBC" (or any four characters other than "I.C.") uses the next input value boundary condition (vbc). This parameter must be entered with four spaces preceding it.

value (line 47) - This line specifies the value associated with a given boundary condition. The value of HEAD is measured in meters, the value of GRADient is measured in d(head)/dn (unitless), and the value of FLUX is measured in (meter**3/sec)/meter of boundary. If the value for **where** above equals "I.C.", then **value** is a dummy, such as 0.

Note that the **sec_end**, **type**, **where**, and **value** lines are repeated for each of the n_sections sections.

block delineator (line 48) - This line specifies that the following block of input values is REQUIRED. The block is read in the climate_lake subroutine and describes climate factors for interior recharge and lakes.

blank or comment line (line 49)

amplitude, cycles (line 50) - This line specifies both the amplitude and number of cycles associated with the interior recharge and lakes, e.g. "0.2, 1." would give a +- 20% sinusoidal variation of recharge factor over one cycle. A value of 0 gives no fluctuations.

ctime0, ctime1 (line 51) - This line specifies the starting and ending time of the climate cycle in years or seconds (consistent with line fifteen above).

blank or comment line (line 52)

blank or comment line (line 53)

ratio_lake_rch (line 54) - This line specifies the ratio of recharge of lakes to recharge of rivers, e.g. a value of 2.0 produces-a climate fluctuation for lakes/rivers equal to 2.0 times the recharge fluctuation.

blank or comment line (line 55)

block delineator (line 56) - This line specifies that the following block of input values is REQUIRED. The block is read in the climate_rech_bn subroutine and describes climate factors for boundary recharge.

blank or comment line (line 57)

amplitude, cycles (line 58) - This line specifies both the amplitude and number of cycles associated with boundary recharge.

ctime0, ctime1 (line 59) - This line specifies the starting and ending time of the climate cycle in years or seconds (consistent with line 14 above).

blank or comment line (line 60)

blank or comment line (line 61)

block delineator (line 62) - This line specifies that the following block of input values are REQUIRED. The block is read in the climate_head_bn subroutine and describes the climate factors for boundary heads.

blank or comment line (line 63)

amplitude, cycles (line 64) - This line specifies both the amplitude and number of cycles associated with boundary heads.

ctime0, ctime1 (line 65) - This line specifies the starting and ending time of the climate cycle in years or seconds (consistent with line 14 above).

blank or comment line (line 66)

blank or comment line (line 67)

6. SECO_2R_STEA.INP INPUT FILE

This section describes how the SECO user can create the input file seco_2r_STEA.inp for seco_2r. Changes in this input file are made by editing the pre-existing template or by creating a file from scratch.

6.1 INPUT FILE LINES

The following fifty-eight lines are required:

title line (line 1) - This line is the title of the input run file.

block delineator (line 2) - This line specifies that the following block of input values are REQUIRED.

FLOW_REG_ID (line 3) - An arbitrary regional flow identification number specified by using any real number.

blank or comment line (line 4) - This line is necessary due to the format of the input template file.

blank or comment line (line 5)

blank or comment line (line 6)

L_def_num (line 7) - This line determines whether seco_2r uses default values of numerical parameters OR replaces data-set default parameters for numerical methods by reading new data from the file methods_2r.inp, e.g. ".true." uses default values and ".false." replaces data-set default parameters.

blank or comment line (line 8)

blank or comment line (line 9)

L_water_table (line 10) - This line determines whether water table conditions are used, e.g. ".true." WILL use water table conditions and ".false." WILL NOT use water table conditions, i.e. the aquifer is always treated as a confined aquifer.

L_harmonic (line 11) - This line determines whether harmonic averaging is used for hydraulic conductance, e.g. ".true." WILL use harmonic averaging and ".false." WILL NOT use harmonic averaging but rather linear averaging.

L_ic_ss (line 12) - This line determines whether the initial conditions will be re-set by the steady state solution with time-dependent wells turned off, e.g. ".true." WILL re-set and ".false." WILL NOT re-set. This is a dummy input for the STEAdy option. (By "dummy" input, it is meant that some input is required by the code format, but the actual value is not used.)

blank or comment line (line 13)

block delineator (line 14) - This line specifies that the following block of input vlaue is REQUIRED. This block is read in the set_wells_r subroutine.

n_wells (line 15) - This line specifies the total number of wells in the simulation.

The following lines are optional and are used to define specific well conditions only if there are n_wells > 0. A block of input values must be entered as follows for EACH of the n_wells:

block delineator - This line specifies that the following block of input values is required for n_wells > 0 only. This block is read in the well_ramp subroutine.

WELL_ID - This is an arbitrary real number designator for the well scenario.

L_wells_off - This line determines whether the well entries are actually used. It allows easy turn-off of all wells, but the entries below are still read, e.g. ".true." WILL turn off all wells and ".false." WILL NOT turn off wells.

blank or comment line

n_well - Labels a specific well by using a number from 1 to n_wells.

VQMAX - This line specifies the maximum volumetric flow rate of fluid leaving the well, measured in meters**3/1000 seconds. Note that 10 gallons/min is equal to 0.631 x 10**-3 meters**3/second; i.e., VQMAX = 0.631 gives a well flow rate of about 10 gallons/min. VQMAX is >0 for a production well and <0 for an injection well.

T0f - This line specifies the start time the well will be ramped on.

T1f - This line specifies the end time of the well ramp-on time interval.

T2f - This line specifies the start time the well will be ramped off.

T3f - This line specifies the end time of the well ramp-off time interval.

x_well - This line specifies the x-location measured in meters on the regional grid, NOT on the aquifer grid.

y_well - This line specifies the y-location measured in meters on the regional grid, NOT on the aquifer grid.

z_well - This line specifies the z-location measured in meters on the regional grid, NOT on the aquifer grid.

block delineator (line 16) - This line specifies that the following block of input values are REQUIRED. The block is read in the bc_flags2 subroutine.

The following block of sixteen comment lines are included in the input file template to describe the format of the boundary condition section:

blank or comment lines (lines 17-32)

spec (line 33) - This line specifies to which boundary the following data applies; i.e., it specifies the LOWEr, UPPEr, LEFT, or RIGHt boundary of the regional grid. When these input parameters are entered, they must be entered in the given order (LOWEr, UPPEr, LEFT, RIGHt). Also, this parameter must be entered starting in column 1. (ONLY the first four characters of the word are necessary)

n_sections (line 34) - This line specifies the number of sections for a given boundary.

sec_end (line 35) - This line specifies the position in meters where a given boundary ends.

type (line 36) - This line specifies the type of boundary condition for a section n. Boundary conditions are of three types: HEAD (Dirichlet), GRADient (Neuman), and FLUX (Cauchy). Specifying the type is done by entering the desired condition preceded by four spaces. (ONLY the first four characters of the word are necessary)

where (line 37) - This line specifies where the value is determined. "I.C." uses the value from the initial condition file and "IVBC" (or any four characters other than "I.C.") uses the next input value boundary condition (vbc). This parameter must be entered with four spaces preceding it.

value (line 38) - This line specifies the value associated with a given boundary condition. The value of HEAD is measured in meters, the value of GRADient is measured in d(head)/dn (unitless), and the value of FLUX is measured in (meter**3/sec)/meter of boundary. If the value for **where** above equals "I.C.", then **value** is a dummy, such as 0.

Note that the **sec_end**, **type**, **where**, and **value** lines are repeated for each of the n_sections sections.

block delineator (line 39) - This line specifies that the following block of input values is REQUIRED. The block is read in the climate_lake subroutine and describes climate factors for interior recharge and lakes.

blank or comment line (line 40)

amplitude, cycles (line 41) - This line specifies both the amplitude and number of cycles associated with the interior recharge and lakes, e.g. "0.2, 1." would give a +- 20% sinusoidal variation of recharge factor over one cycle. A value of 0 gives no fluctuations.

ctime0, ctime1 (line 42) - This line specifies the starting and ending time of the climate cycle in years or seconds (consistent with line fourteen above).

blank or comment line (line 43)

blank or comment line (line 44)

ratio_lake_rch (line 45) - This line specifies the ratio of recharge of lakes to recharge of rivers, e.g. a value of 2.0 produces a climate fluctuation for lakes/rivers equal to 2.0 times the recharge fluctuation.

blank or comment line (line 46)

block delineator (line 47) - This line specifies that the following block of input values is REQUIRED. The block is read in the climate_rech_bn subroutine and describes climate factors for boundary recharge.

blank or comment line (line 48)

amplitude, cycles (line 49) - This line specifies both the amplitude and number of cycles associated with boundary recharge.

ctime0, ctime1 (line 50) - This line specifies the starting and ending time of the climate cycle in years or seconds (consistent with line 14 above).

blank or comment line (line 51)

blank or comment line (line 52)

block delineator (line 53) - This line specifies that the following block of input values are REQUIRED. The block is read in the climate_head_bn subroutine and describes the climate factors for boundary heads.

blank or comment line (line 54)

amplitude, cycles (line 55) - This line specifies both the amplitude and number of cycles associated with boundary heads.

ctime0, ctime1 (line 56) - This line specifies the starting and ending time of the climate cycle in years or seconds (consistent with line 14 above).

blank or comment line (line 57)

blank or comment line (line 58)

7. SECO_2L_UNST.INP INPUT FILE

This section describes how the SECO user can create the input file seco_2l_UNST.inp for seco_2l. Changes in this input file are made by editing the pre-existing template or by creating a file from scratch.

7.1 INPUT FILE LINES

The following forty-four lines are required:

title line (line 1) - This line is the title of the input run file.

block delineator (line 2) - This line specifies that the following block of input values are REQUIRED.

FLOW_LOC_ID (line 3) - An arbitrary local flow identification number specified by using any real number.

blank or comment line (line 4) - This line is necessary due to the format of the input template file.

blank or comment line (line 5)

blank or comment line (line 6)

L_def_num (line 7) - This line determines whether seco_2l uses default values of numerical parameters OR replaces data-set default parameters for numerical methods by reading new data from the file methods_2l.inp, e.g. ".true." uses default values and ".false." replaces data-set default parameters.

blank or comment line (line 8)

blank or comment line (line 9)

L_water_table (line 10) - This line determines whether water table conditions are used, e.g. ".true." WILL use water table conditions and ".false." WILL NOT use water table conditions, i.e. the aquifer is always treated as a confined aquifer.

L_harmonic (line 11) - This line determines whether harmonic averaging is used for hydraulic conductance, e.g. ".true." WILL use harmonic averaging and ".false." WILL NOT use harmonic averaging but rather linear averaging.

block delineator (line 12) - This line specifies that the following block of input values are required for UNSTeady option only. This block is read in subroutine set_dt.

YEARs or SECOnds (line 13) - This line specifies the time units, measured in years (YEARs) or seconds (SECOnds).

ntimeL (line 14) - This line specifies the number of output time steps (max 50).

time0, timeL (line 15) - This line specifies the initial time (time0) and the final time (timeL) in units consistent with n_timeL above.

N_TIMOPT (line 16) - This value sets the change in simulation time (dt).
0 - sets the simulation times at a constant dt.
1 - sets times by packing dt around an event time; this option requires (ntime1 >= 5).
2 - used to enter dt's directly [**ntimeL pairs of n, dtime(n)**]. 3 - used to enter times directly [**ntimeL + 1 pairs of n, time(n)**].

For N_TIMOPT = 1,2 or 3, the necessary input values (listed above) must be entered accordingly. Therefore, additional lines may need to be added here.

blank or comment line (line 17)

blank or comment line (line 18)

blank or comment line (line 19)

blank or comment line (line 20)

block delineator (line 21) - This line specifies that the following block of input values are REQUIRED. The block is read in the bc_flags2 subroutine.

The following block of eleven comment lines are included in the input file template to describe the format of the boundary condition section:

blank or comment lines (lines 22-32)

spec (line 33) - This line specifies to which boundary the following data applies; i.e., it specifies the LOWEr, UPPEr, LEFT, or RIGHt boundary of the local grid. When these input parameters are entered, they must be entered in the given order (LOWEr, UPPEr, LEFT, RIGHt). Also, this parameter must be entered starting in column 1. (ONLY the first four characters of the word are necessary)

n_sections (line 34) - This line specifies the number of sections for a given boundary.

sec_end (line 35) - This line specifies the position in meters where a given boundary ends.

type (line 36) - This line specifies the type of boundary condition for a section n. Boundary conditions are of three types: HEAD (Dirichlet), GRADient (Neuman), and FLUX (Cauchy). Specifying the type is done by entering the desired condition preceded by four spaces. (ONLY the first four characters of the word are necessary)

Note that the **sec_end** and **type** lines are repeated for each of the n_sections sections.

block delineator (line 37) - This line specifies that the following block of input values is REQUIRED. The block is read in the climate_lake subroutine and describes the climate factors for interior recharge and lakes.

blank or comment line (line 38)

amplitude, cycles (line 39) - This line specifies both the amplitude and number of cycles associated with the interior recharge and lakes, e.g. "0.2, 1." would give a +- 20% sinusoidal variation of recharge factor over one cycle. A value of 0 gives no fluctuations.

ctime0, ctime1 (line 40) - This line specifies the starting and ending time of the climate cycle in years or seconds (consistent with line thirteen above).

blank or comment line (line 41)

blank or comment line (line 42)

ratio_lake_rch (line 43) - This line specifies the ratio of recharge of lakes to recharge of rivers, e.g. a value of 2.0 produces a climate fluctuation for lakes/rivers equal to 2.0 times the recharge fluctuation.

blank or comment line (line 44)

8. SECO_2L_STEA.INP INPUT FILE

This section describes how the SECO user can create the input file seco_2l_STEA.inp for seco_2l. Changes in this input file are made by editing the pre-existing template or by creating a file from scratch.

8.1 INPUT FILE LINES

The following thirty-five lines are required:

title line (line 1) - This line is the title of the input run file.

block delineator (line 2) - This line specifies that the following block of input values are REQUIRED.

FLOW_LOC_ID (line 3) - An arbitrary local flow identification number specified by using any real number.

blank or comment line (line 4) - This line is necessary due to the format of the input template file.

blank or comment line (line 5)

blank or comment line (line 6)

L_def_num (line 7) - This line determines whether seco_2l uses default values of numerical parameters OR replaces data-set default parameters for numerical methods by reading new data from the file methods_2l.inp, e.g. ".true." uses default values and ".false." replaces data-set default parameters.

blank or comment line (line 8)

blank or comment line (line 9)

L_water_table (line 10) - This line determines whether water table conditions are used, e.g. ".true." WILL use water table conditions and ".false." WILL NOT use water table conditions, i.e. the aquifer is always treated as a confined aquifer.

L_harmonic (line 11) - This line determines whether harmonic averaging is used for hydraulic conductance, e.g. ".true." WILL use harmonic averaging and ".false." WILL NOT use harmonic averaging but rather linear averaging.

block delineator (line 12) - This line specifies that the following block of input values are REQUIRED. The block is read in the bc_flags2 subroutine.

The following block of eleven comment lines are included in the input file template to describe the format of the boundary condition section:

blank or comment lines (lines 13-23)

spec (line 24) - This line specifies to which boundary the following data applies; i.e., it specifies the LOWEr, UPPEr, LEFT, or RIGHt boundary of the local grid. When these input parameters are

entered, they must be entered in the given order (LOWEr, UPPEr, LEFT, RIGHt). Also, this parameter must be entered starting in column 1. (ONLY the first four characters of the word are necessary)

n_sections (line 25) - This line specifies the number of sections for a given boundary.

sec_end (line 26) - This line specifies the position in meters where a given boundary ends.

type (line 27) - This line specifies the type of boundary condition for a section n. Boundary conditions are of three types: HEAD (Dirichlet), GRADient (Neuman), and FLUX (Cauchy). Specifying the type is done by entering the desired condition preceded by four spaces. (ONLY the first four characters of the word are necessary)

Note that the **sec_end** and **type** lines are repeated for each of the n_sections sections.

block delineator (line 28) - This line specifies that the following block of input vales is REQUIRED. The block is read in the climate_lake subroutine and describes climate factors for interior recharge and lakes.

blank or comment line (line 29)

amplitude, cycles (line 30) - This line specifies both the amplitude and number of cycles associated with the interior recharge and lakes, e.g. "0.2, 1." would give a +- 20% sinusoidal variation of recharge factor over one cycle. A value of 0 gives no fluctuation.

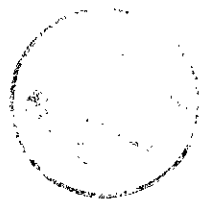
ctime0, ctime1 (lime 31) - This line specifies the starting and ending time of the climate cycle in years or seconds.

blank or comment line (line 32)

blank or comment line (line 33)

ratio_lake_recharge (line 34) - This line specifies the ratio of recharge of lakes to recharge of rivers, e.g. a value of 2.0 produces a climate fluctuation for lakes/rivers equal to 2.0 times the recharge fluctuation.

blank or comment line (line 35)



9. THE CHANGE_SIZE_2 INPUT FILE

This section describes how the SECO user can create the input file change_size_2.inp for change_size_2. Changes in this input file are made by editing the pre-existing template or by creating a file from scratch.

9.1 INPUT FILE LINES

The following six lines are required:

title line *(line 1) - This line is the title of the input run file.*

IO_SCREEN (line 2) - This value (between 0 and 4) determines the amount of data actually output to the screen. 0 is used for no screen output while 4 is used for maximum screen output.

L_d_GREG (line 3) - This determines whether the size of the regional grid is to be changed, e.g. ".true." WILL change the size of the regional grid and ".false." WILL NOT change the size of the regional grid.

il, jl (line 4) - This line specifies the new array dimensions of the regional grid. For L_d_greg = ".false.", a dummy input is required. (By "dummy" input, it is meant that some input is required by the code format, but the actual value is not used.)

L_d_gloc (line 5) - This refers to whether the size of the local grid is to be changed, e.g. ".true." WILL change the size of the local grid and ".false." WILL NOT change the size of the local grid.

il, jl *(line 6) - This line specifies the new array dimensions of the local grid. For L_d_gloc = ".false.", a dummy input is required.*

10. THE METHODS_2R INPUT FILE

This section describes how the SECO user can create the input file methods_2r.inp for read_methods called by the regional grid flow solution code seco_2r. Changes in this input file are made by editing the pre-existing template or by creating a file from scratch.

10.1 INPUT FILE LINES

The following twenty-five lines are required:

blank or comment line (line 1) - This line is necessary due to the format of the input file template.

n_symbc (line 5) - This integer value (either 0 or 1) determines whether the ghost cell boundary stencils are symmetrized. 1 is used for the multigrid (mg) solver and is required for n_sor = 0 in seco_2r. The value will be re-set to 1 for the mg solver.

itermx (line 6) - This integer value specifies the maximum number of intra-time-step iterations.

lpr (line 7) - This integer value (either 1, >=0 or <0) specifies the flag that controls the format and amount of information about arrays that will be written to the screen.
1 - prints maximum information.
>=0 - uses the FORTRAN F format.
<0 - uses the FORTRAN E format.

lprc10 (line 8) - This integer value (either 0 or 1) specifies the flag that controls if the array that contains the right hand side of the equation is to be written to the screen.
0 - does not print.
1 - prints the array to the screen.

npauser (line 9) - This integer value (either 0 or 1) determines if the simulation pauses after each time step is complete.
0 - will pauses the simulation.
1 - will complete all the time steps.

n_shift (line 10) - This integer value (either -1 or 1) determines whether a shift in the level of head will be performed before the iterative solution. This reduces machine round-off error.
-1 - will not perform the shift.
1 - will perform the shift. The shift is only possible when
there are no Robin conditions and the steady state option has been selected.

n_sor (line 11) - This integer value (either 0,+-1 or 2) specifies which solver is to be used.
0 - selects the symmetric multigrid solver (default).
1 - selects the point sor solver.
-1 - selects the symmetric point sor solver.
2 - selects the line sor solver (uses nzebra = 0 or 1, see line 18 below).

lim1 (line 12) - This line specifies the input lower limit on the number of iterations.

lim2f (line 13) - This line specifies the factor that determines the upper limit (lim2) on the number of iterations.

conv (line 14) - This line specifies the convergence test on resmax = rsx/bscale. If resmax < conv and nit >= lim1, a return is made even if nit < lim2. (Typical values for conv are 1.e-03 to 1.e-06.)

rf (line 15) - This line specifies the relaxation factor for sor iterations.

bscale (line 16) - This line specifies the scaling factor used for convergence testing. It should approximate the expected range of solution values for head. If input bscale .lt. 0.0, the code calculates a new value based on the range of input boundary values of head. This could be inadequate for a single Dirichlet bc or for non-Dirichlet problems or for large source terms c10(i,j). If input bscale = 0.0, the

code calculates a new value based on the actual range of the solution. The code will re-set a minimum of 0.01, needed to avoid indeterminacy.

ngrind (line 17) - This line gives an intermediate printout of resmax during the sor iterative process, at every ngrind-th iteration. It is useful for long interactive runs. For ngrind .lt. 0, the code sets a new ngrind = 1 + 5000/(il*jl).
nzebra (line 18) - This value (either 0 or 1) specifies the type of j-line sor solver.
0 - means a j-line sor solver (implicit in i) is used.
1 - means a zebra j-line sor solver in i is used. It starts
at j = 1,3,5,7,9,... and then at j = 2,4,6,8,10,....

init_lsor (line 19) - This value (either <0,0 or >0) controls initialization of certain arrays in the sor solving routine. <0 - initializes only without solving.
0 - initializes and solves.
>0 - solves only using the previous initialization.

The following six variables with suffix _mg relate to the multigrid solver:

ifd59_mg (line 20) - This integer value (either 5 or 9) indicates the stencil size. 5 - means a five-point finite difference stencil (ac, aw and as) is defined on the finest grid by the user.
9 - means a nine-point finite difference stencil (ac, aw, as, asw and ase) is defined on the finest grid by the user.

ifmg_mg (line 21) - This integer value (either 0 or 1) determines the "full multigrid" algorithm.
0 - means the full multigrid algorithm is not used to obtain a good initial guess on the fine grid.
1 - means the full multigrid algorithm is used to obtain a good initial guess on the fine grid.

ncyc_mg (line 22) - This value specifies the maximum number of multigrid "v"-cycles to be used. If the maximum norm of the residual is not less than tol at the end of ncyc cycles, the multigrid algorithm is terminated.

tol_mg (line 23) - This real value (either >0,0,-1 or -2) is related to convergence criteria.
>0 - means the maximum norm of the residual is calculated at the end of each multigrid cycle. The algorithm is terminated when this maximum becomes less than tol or when the maximum number of iterations (see ncyc_mg) is exceeded. It is up to the user to provide a meaningful tolerance criterion for the particular problem being solved.
0 - performs ncyc_mg multigrid cycles. It calculates and prints the maximum norm of the residual after each cycle.
-1 - performs ncyc_mg multigrid cycles. The maximum norm of the final residual is calculated and returned in the variable rmax in the calling list of mgss2.
-2 - performs ncyc_mg multigrid cycles. The maximum norm of the residual is never calculated.

nman_mg (line 24) - This integer value (either 0 or 1) flags the singular (all Neuman boundary condition) problem.
0 - usually chosen.

1 - signals that the fine grid equations are singular for the case when homogeneous Neumann boundary conditions are applied along the entire boundary. In this case, the difference equations are singular and the condition that the integral of q over the domain be zero is added to the set of difference equations. This condition is satisfied by adding the appropriate constant vector to q on the fine grid. It is assumed, in this case, that a well-defined problem has been given to mgss2, i.e. the integral of f over the domain is zero.

iw_mg (line 25) - This integer value (either 0 or 1) determines a multigrid option. Usually iw_mg = 1 is best. However, if the boundary condition equations have been absorbed into the interior equations then iw_mg = 0 can be used which results in a slightly more efficient algorithm.

11. THE METHODS_2L INPUT FILE

This section describes how the SECO user can create the input file methods_2l.inp for read_methods called by the regional grid flow solution code seco_2l. Changes in this input file are made by editing the pre-existing template or by creating a file from scratch.

11.1 INPUT FILE LINES

The following twenty-five lines are required:

blank or comment line (line 1) - This line is necessary due to the format of the input file template.

n_symbc (line 5) - This integer value (either 0 or 1) determines whether the ghost cell boundary stencils are symmetrized. 1 is used for the multigrid (mg) solver and is required for n_sor = 0 in seco_2r. The value will be re-set to 1 for the mg solver.

itermx (line 6) - This integer value specifies the maximum number of intra-time-step iterations.

lpr (line 7) - This integer value (either 1, >=0 or <0) specifies the flag that controls the format and amount of information about arrays that will be written to the screen.
1 - prints maximum information.
>=0 - uses the FORTRAN F format.
<0 - uses the FORTRAN E format.

lprc10 (line 8) - This integer value (either 0 or 1) specifies the flag that controls if the array that contains the right hand side of the equation is to be written to the screen.
0 - does not print.
1 - prints the array to the screen.

npauser (line 9) - This integer value (either 0 or 1) determines if the simulation pauses after each time step is complete.
0 - will pauses the simulation.
1 - will complete all the time steps.

n_shift (line 10) - This integer value (either -1 or 1) determines whether a shift in the level of head will be performed before the iterative solution. This reduces machine round-off error.
-1 - will not perform the shift.
1 - will perform the shift. The shift is only possible when
there are no Robin conditions and the steady state option has been selected.

n_sor (line 11) - This integer value (either 0,+-1 or 2) specifies which solver is to be used.
0 - selects the symmetric multigrid solver (default).
1 - selects the point sor solver.
-1 - selects the symmetric point sor solver.
2 - selects the line sor solver (uses nzebra = 0 or 1, see line 18 below).

lim1 (line 12) - This line specifies the input lower limit on the number of iterations.

lim2f (line 13) - This line specifies the factor that determines the upper limit (lim2) on the number of iterations.

conv (line 14) - This line specifies the convergence test on resmax = rsx/bscale. If resmax < conv and nit >= lim1, a return is made even if nit < lim2. (Typical values for conv are 1.e-03 to 1.e-06.)

rf (line 15) - This line specifies the relaxation factor for sor iterations.

bscale (line 16) - This line specifies the scaling factor used for convergence testing. It should approximate the expected range of solution values for head. If input bscale .lt. 0.0, the code calculates a new value based on the range of input boundary values of head. This could be inadequate for a single Dirichlet bc or for non-Dirichlet problems or for large source terms c10(i,j). If input bscale = 0.0, the code calculates a new value based on the actual range of the solution. The code will re-set a minimum of 0.01, needed to avoid indeterminacy.

ngrind (line 17) - This line gives an intermediate printout of resmax during the sor iterative process, at every ngrind-th iteration. It is useful for long interactive runs. For ngrind .lt. 0, the code sets a new ngrind = 1 + 5000/(il*jl).
nzebra (line 18) - This value (either 0 or 1) specifies the type of j-line sor solver.
0 - means a j-line sor solver (implicit in i) is used.
1 - means a zebra j-line sor solver in i is used. It starts
at j = 1,3,5,7,9,... and then at j = 2,4,6,8,10,....

init_lsor (line 19) - This value (either <0,0 or >0) controls initialization of certain arrays in the sor solving routine. <0 - initializes only without solving.
0 - initializes and solves.
>0 - solves only using the previous initialization.

The following six variables with suffix _mg relate to the multigrid solver:

ifd59_mg (line 20) - This integer value (either 5 or 9) indicates the stencil size.
5 - means a five-point finite difference stencil (ac, aw and as) is defined on the finest grid by the user.
9 - means a nine-point finite difference stencil (ac, aw, as, asw and ase) is defined on the finest grid by the user.

ifmg_mg (line 21) - This integer value (either 0 or 1) determines the "full multigrid" algorithm.
0 - means the full multigrid algorithm is not used to obtain a good initial guess on the fine grid.
1 - means the full multigrid algorithm is used to obtain a good initial guess on the fine grid.

ncyc_mg (line 22) - This value specifies the maximum number of multigrid "v"-cycles to be used. If the maximum norm of the residual is not less than tol at the end of ncyc cycles, the multigrid algorithm is terminated.

tol_mg (line 23) - This real value (either >0,0,-1 or -2) is related to convergence criteria.
>0 - means the maximum norm of the residual is calculated at the end of each multigrid cycle. The algorithm is terminated when this maximum becomes less than tol or when the maximum number of iterations (see ncyc_mg) is exceeded. It is up to the user to provide a meaningful tolerance criterion for the particular problem being solved.
0 - performs ncyc_mg multigrid cycles. It calculates and prints the maximum norm of the residual after each cycle.
-1 - performs ncyc_mg multigrid cycles. The maximum norm of the final residual is calculated and returned in the variable rmax in the calling list of mgss2.
-2 - performs ncyc_mg multigrid cycles. The maximum norm of the residual is never calculated.

nman_mg (line 24) - This integer value (either 0 or 1) flags the singular (all Neuman boundary condition) problem.
0 - usually chosen.
1 - signals that the fine grid equations are singular for the case when homogeneous Neumann boundary conditions are applied along the entire boundary. In this case, the difference equations are singular and the condition that the integral of q over the domain be zero is added to the set of difference equations. This condition is satisfied by adding the appropriate constant vector to q on the fine grid. It is assumed, in this case, that a well-defined problem has been given to mgss2, i.e. the integral of f over the domain is zero.

iw_mg (line 25) - This integer value (either 0 or 1) determines a multigrid option. Usually iw_mg = 1 is best. However, if the boundary condition equations have been absorbed into the interior equations then iw_mg = 0 can be used which results in a slightly more efficient algorithm.

<u>Guidelines for using various SECO templates</u>

METHODS_2R.INP - [used w.r.t. regional grid] Primarily will be used when there is a need to change the type of solver used by SECO (either multigrid, point sor solver, or line sor solver). Most other values seem to be set at a particular value. This, of course, may change.

METHODS_2L.INP - [used w.r.t. local grid] Same as above.

SECO_2r_UNST.inp - [used w.r.t. regional, unsteady solution] Used to set a variety of specifications before running the SECO code. For instance, the length of time and change in time can be set. Also, the number of wells and related well information can be set. The boundary conditions for the regional grid (upper, lower, left, and right) can be set according to type (head, gradient, mixed, flux, and adaptive). Finally, climate conditions regarding lakes, recharge, and head can be set.

SECO_2l_UNST.inp - [used w.r.t. local, unsteady solution] Similar to above description only local grid boundaries are used.

SECO_2r_STEA.inp - [used w.r.t. regional, steady-state solution] Similar to above description only a steady-state condition exists.

SECO_2l_STEA.inp - [used w.r.t. local, steady-state solution] Similar to above description only a steady-state condition exists.

seco_2.inp - [used w.r.t. unsteady or steady-state solution - user determined] Used with **run seco_2** command. Main driver where specific FORTRAN data files are read from and saved to. Also determines whether flow solutions are to be regional, local, or both. A tracker for both regional and local grids can also be specified here.

greg2.inp - Used with the **run set_greg2** command. This template is used to specify all of the conditions representing the regional grid. Grid dimensions, cell size, and delta-x's and delta-y's can all be specified here by the user.

gloc2.inp - Used with the **run set_gloc2** command. This template is used to specify all of the conditions representing the local grid. Grid dimensions, cell-size, and delta-x's and delta-y's can all be specified here by the user.

Appendix II: Sample Diagnostics/Debug File

THE SAMPLE DIAGNOSTICS/DEBUG FILE INCLUDED IN THIS USER'S MANUAL *IS NOT* THE DIAGNOSTICS/DEBUG FILE PROVIDED IN THE 1996 WIPP PA CALCULATION. IT IS INCLUDED *SOLELY FOR ILLUSTRATIVE PURPOSES.*

```
       SSSSSS   EEEEEEE   CCCCC     00000    FFFFFFF  LL          2222    DDDDDD
      SS        EE       CC   CC  OO   OO  FF       LL         2    2   DD   DD
      SS        EE       CC       OO   OO  FF       LL             2   DD   DD
       SSSSS    EEEEE    CC       OO   OO  FFFFF    LL            2    DD   DD
           SS   EE       CC       OO   OO  FF       LL           2     DD   DD
           SS   EE       CC   CC  OO   OO  FF       LL          2      DD   DD
      SSSSSS    EEEEEEE   CCCCC     00000    FF       LLLLLLL  222222  DDDDDD
SECOFL2D [12H                                  Version C-3.01VV
                                  Revised 08/09/93
                              Written by P. J. ROACHE
                            Sponsored by rebecca blaine                    Run on 08/18/95
at 16:16:36
                                Run on VAX ALPH VMS 6.1 SECOFL2D C-3.01VV (08/09/93)
08/18/95 16:16:Input file does not have standard text file format
```

```
**************************************************************

Prepared for
Sandia National Laboratories
Albuquerque, New Mexico 87185-5800
for the United States Department of Energy
under Contract DE-AC04-76DP00789

**************************************************************

                          Disclaimer
                          ----------

This computer program was prepared as an account of work
sponsored by an agency of the United States Government.
Neither the United States Government nor any agency thereof,
nor any of their employees, nor any of their contractors,
subcontractors, or their employees, makes any warranty,
express or implied, or assumes any legal liability or
responsibility for the accuracy, completeness, or usefulness
of any information, apparatus, product, or process disclosed
or represents that its use would not infringe privately
owned rights. Reference herein to any specific commercial
product, process, or service by trade name, trademark,
manufacturer, or otherwise, does not necessarily constitute
or imply its endorsement, recommendation, or favoring by the
United States Government, any agency thereof or any of their
contractors or subcontractors. The views and opinions
expressed herein do not necessarily state or reflect those
of the United States Government, any agency thereof or any
of their contractors or subcontractors.
```

```
          ************************************************************

 ***************************************************************************

 FILE ASSIGNMENTS:
 -----------------
 SECOFL2D input file:
    T4:[DSCOTT.E1E2_UG1]SECOFL2D.INP;2
 SECOFL2D regional property data file:
    T4:[DSCOTT.E1E2_UG1]SECO_REG_BINPROP_R001.INP;2
 SECOFL2D local property data file:
    T4:[DSCOTT.E1E2_UG1]SECO_LOC_BINPROP_R001.INP;2
 SECOFL2D regional output data file:
    T4:[DSCOTT.E1E2_UG1]SECO_REG_R001.OUT
 SECOFL2D local output data file:
    T4:[DSCOTT.E1E2_UG1]SECO_LOC_R001.OUT
 SECOFL2D local boundary condition file:
    T4:[DSCOTT.E1E2_UG1]SECO_LOC_BC.DAT
 SECOFL2D output file:
    T4:[DSCOTT.E1E2_UG1]SECOFL2D.DBG
  io_screen =           1
  Time-dependent solution will be calculated.
  Now calling SECO_2R ...
  seco_2R (2D Regional grid) run...
 Initial conditions will be re-set by steady state solution
  IDFIL_GREG =         - 20
  Now reading 9 2D arrays and 4 1D arrays.
  (Requires 39 seconds on MicroVAX for 40 x 34 grid.)
 From SET_TOLK: tolK =     1.537000062E-09
      WELL_ID, L_wells_on, n_wells =   0.000000000000000E+000 F           0

 Now setting b.c. flags and values.

  boundary #, n_sections(nb) =           1           3
  n_sec, sec_end, type, where, value_A, value_B, cbcf =
           1   4000.00000000000
    HEAD    I.C.
 0.000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000
           2   18595.0000000000
    GRAD    I.C.
 0.000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000
           3   30000.0000000000
    HEAD    I.C.
 0.000000000000000E+000  0.000000000000000E+000   1.00000000000000

  boundary #, n_sections(nb) =           2           2
  n_sec, sec_end, type, where, value_A, value_B, cbcf =
           1   27240.0000000000
    HEAD    I.C.
 0.000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000
           2   30000.0000000000
    GRAD    I.C.
 0.000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000

  boundary #, n_sections(nb) =           3           1
  n_sec, sec_end, type, where, value_A, value_B, cbcf =
           1   25000.0000000000
    HEAD    I.C.
 0.000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000

  boundary #, n_sections(nb) =           4           3
  n_sec, sec_end, type, where, value_A, value_B, cbcf =
           1   10000.0000000000
    HEAD    I.C.
 0.000000000000000E+000  0.000000000000000E+000   1.00000000000000
           2   17300.0000000000
    HEAD    I.C.
 0.000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000
           3   25000.0000000000
```

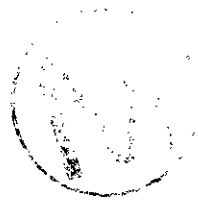
```
   GRAD    I.C.
 0.000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000
 Boundary #            1

Left Boundary at i = 1, range of y =   0.000E+00  3.000E+04
 Section #, sec_end=              1   4000.00000000000
 nbc, value_A, value_B, cbcf =            1  0.000000000000000E+000
 0.000000000000000E+000  0.000000000000000E+000
 Section #, sec_end=              2   18595.0000000000
 nbc, value_A, value_B, cbcf =            2  0.000000000000000E+000
 0.000000000000000E+000  0.000000000000000E+000
 Section #, sec_end=              3   30000.0000000000
 nbc, value_A, value_B, cbcf =            1  0.000000000000000E+000
 0.000000000000000E+000   1.00000000000000
 Boundary #            2

Boundary at i = iL, range of y =   0.000E+00  3.000E+04
 Section #, sec_end=              1   27240.0000000000
 nbc, value_A, value_B, cbcf =            1  0.000000000000000E+000
 0.000000000000000E+000  0.000000000000000E+000
 Section #, sec_end=              2   30000.0000000000
 nbc, value_A, value_B, cbcf =            2  0.000000000000000E+000
 0.000000000000000E+000  0.000000000000000E+000
 Boundary #            3

Lower Boundary at j = 1, range of x =   0.000E+00  2.500E+04
 Section #, sec_end=              1   25000.0000000000
 nbc, value_A, value_B, cbcf =            1  0.000000000000000E+000
 0.000000000000000E+000  0.000000000000000E+000
 Boundary #            4
 Section #, sec_end=              1   10000.0000000000
 nbc, value_A, value_B, cbcf =            1  0.000000000000000E+000
 0.000000000000000E+000   1.00000000000000
 Section #, sec_end=              2   17300.0000000000
 nbc, value_A, value_B, cbcf =            1  0.000000000000000E+000
 0.000000000000000E+000  0.000000000000000E+000
 Section #, sec_end=              3   25000.0000000000
 nbc, value_A, value_B, cbcf =            2  0.000000000000000E+000
 0.000000000000000E+000  0.000000000000000E+000
new time, clim_lake_fac, clim_rchb_fac, clim_hedb_fac =
 3.1557E+10  1.0000E+00  1.0000E+00  9.9998E-01
wusun_mg, wun_mg, rmax_mg =  0.335E+01  0.806E+02  0.661E-06
sums of boundary fluxes out [m**3/sec] (i1,iL,j1,jL,total) =
-2.0226E+06 -1.7363E+04  2.0450E+06 -5.0473E+03  1.2508E-03
total boundary fluxes in, out, [m**3/sec], %difference =
 4.1842E+06  4.1842E+06  2.9894E-08

 Now writing 6 2D arrays for head, etc.
             to file FOR0          40
 (Requires 8 seconds on MicroVAX for 40 x 34 grid.)
 Now writing head...
 Now writing con_salt...
 Now writing uscf...
 Now writing vscf...
 Now writing spec_stor_dhdt...
new time, clim_lake_fac, clim_rchb_fac, clim_hedb_fac =
 3.1557E+10  1.0000E+00  1.0000E+00  9.9998E-01
wusun_mg, wun_mg, rmax_mg =  0.335E+01  0.806E+02  0.295E-09
sums of boundary fluxes out [m**3/sec] (i1,iL,j1,jL,total) =
-2.0226E+06 -1.7363E+04  2.0450E+06 -5.0473E+03  2.0958E-04
total boundary fluxes in, out, [m**3/sec], %difference =
 4.1842E+06  4.1842E+06  5.0087E-09

 Now writing 6 2D arrays for head, etc.
             to file FOR0          40
 (Requires 8 seconds on MicroVAX for 40 x 34 grid.)
 Now writing head...
 Now writing con_salt...
 Now writing uscf...
 Now writing vscf...
```

```
 Now writing spec_stor_dhdt...
new time, clim_lake_fac, clim_rchb_fac, clim_hedb_fac =
 6.3114E+10  1.0000E+00  1.0000E+00  1.0086E+00
wusun_mg, wun_mg, rmax_mg =  0.000E+00  0.806E+02  0.217E-07
sums of boundary fluxes out [m**3/sec] (i1,iL,j1,jL,total) =
-2.4933E+06 -1.7109E+04  2.5151E+06 -5.1782E+03 -4.7824E+02
total boundary fluxes in, out, [m**3/sec], %difference =
 5.1980E+06  5.1975E+06  9.2009E-03

 Now writing 6 2D arrays for head, etc.
             to file FOR0          40
 (Requires 8 seconds on MicroVAX for 40 x 34 grid.)
 Now writing head...
 Now writing con_salt...
 Now writing uscf...
 Now writing vscf...
 Now writing spec_stor_dhdt...
new time, clim_lake_fac, clim_rchb_fac, clim_hedb_fac =
 9.4671E+10  1.0000E+00  1.0000E+00  1.0086E+00
wusun_mg, wun_mg, rmax_mg =  0.000E+00  0.806E+02  0.197E-09
sums of boundary fluxes out [m**3/sec] (i1,iL,j1,jL,total) =
-2.4938E+06 -1.7106E+04  2.5160E+06 -5.1354E+03 -1.4980E+01
total boundary fluxes in, out, [m**3/sec], %difference =
 5.1996E+06  5.1996E+06  2.8811E-04

 Now writing 6 2D arrays for head, etc.
             to file FOR0          40
 (Requires 8 seconds on MicroVAX for 40 x 34 grid.)
 Now writing head...
 Now writing con_salt...
 Now writing uscf...
 Now writing vscf...
 Now writing spec_stor_dhdt...
new time, clim_lake_fac, clim_rchb_fac, clim_hedb_fac =
 1.2623E+11  1.0000E+00  1.0000E+00  1.0000E+00
wusun_mg, wun_mg, rmax_mg =  0.000E+00  0.806E+02  0.217E-07
sums of boundary fluxes out [m**3/sec] (i1,iL,j1,jL,total) =
-2.0250E+06 -1.7359E+04  2.0478E+06 -5.0026E+03  4.7087E+02
total boundary fluxes in, out, [m**3/sec], %difference =
 4.1897E+06  4.1902E+06  1.1238E-02

 Now writing 6 2D arrays for head, etc.
             to file FOR0          40
 (Requires 8 seconds on MicroVAX for 40 x 34 grid.)
 Now writing head...
 Now writing con_salt...
 Now writing uscf...
 Now writing vscf...
 Now writing spec_stor_dhdt...
new time, clim_lake_fac, clim_rchb_fac, clim_hedb_fac =
 1.5778E+11  1.0000E+00  1.0000E+00  1.0001E+00
wusun_mg, wun_mg, rmax_mg =  0.000E+00  0.806E+02  0.364E-09
sums of boundary fluxes out [m**3/sec] (i1,iL,j1,jL,total) =
-2.0281E+06 -1.7360E+04  2.0505E+06 -5.0457E+03  7.8729E+00
total boundary fluxes in, out, [m**3/sec], %difference =
 4.1960E+06  4.1960E+06  1.8763E-04

 Now writing 6 2D arrays for head, etc.
             to file FOR0          40
 (Requires 8 seconds on MicroVAX for 40 x 34 grid.)
 Now writing head...
 Now writing con_salt...
 Now writing uscf...
 Now writing vscf...
 Now writing spec_stor_dhdt...
new time, clim_lake_fac, clim_rchb_fac, clim_hedb_fac =
 1.8934E+11  1.0000E+00  1.0000E+00  1.0088E+00
wusun_mg, wun_mg, rmax_mg =  0.000E+00  0.806E+02  0.219E-07
sums of boundary fluxes out [m**3/sec] (i1,iL,j1,jL,total) =
-2.5023E+06 -1.7104E+04  2.5241E+06 -5.1794E+03 -4.7895E+02
```

```
 total boundary fluxes in, out, [m**3/sec], %difference =
  5.2176E+06  5.2171E+06  9.1799E-03

  Now writing 6 2D arrays for head, etc.
              to file FOR0          40
  (Requires 8 seconds on MicroVAX for 40 x 34 grid.)
  Now writing head...
  Now writing con_salt...
  Now writing uscf...
  Now writing vscf...
  Now writing spec_stor_dhdt...
 new time, clim_lake_fac, clim_rchb_fac, clim_hedb_fac =
  2.2090E+11  1.0000E+00  1.0000E+00  1.0089E+00
 wusun_mg, wun_mg, rmax_mg =  0.000E+00  0.806E+02  0.295E-09
 sums of boundary fluxes out [m**3/sec] (i1,iL,j1,jL,total) =
 -2.5065E+06 -1.7099E+04  2.5287E+06 -5.1377E+03 -1.7358E+01
 total boundary fluxes in, out, [m**3/sec], %difference =
  5.2270E+06  5.2269E+06  3.3209E-04

  Now writing 6 2D arrays for head, etc.
              to file FOR0          40
  (Requires 8 seconds on MicroVAX for 40 x 34 grid.)
  Now writing head...
  Now writing con_salt...
  Now writing uscf...
  Now writing vscf...
  Now writing spec_stor_dhdt...
 new time, clim_lake_fac, clim_rchb_fac, clim_hedb_fac =
  2.5246E+11  1.0000E+00  1.0000E+00  1.0003E+00
 wusun_mg, wun_mg, rmax_mg =  0.000E+00  0.806E+02  0.215E-07
 sums of boundary fluxes out [m**3/sec] (i1,iL,j1,jL,total) =
 -2.0412E+06 -1.7350E+04  2.0640E+06 -5.0057E+03  4.6788E+02
 total boundary fluxes in, out, [m**3/sec], %difference =
  4.2247E+06  4.2251E+06  1.1074E-02

  Now writing 6 2D arrays for head, etc.
              to file FOR0          40
  (Requires 8 seconds on MicroVAX for 40 x 34 grid.)
  Now writing head...
  Now writing con_salt...
  Now writing uscf...
  Now writing vscf...
  Now writing spec_stor_dhdt...
 new time, clim_lake_fac, clim_rchb_fac, clim_hedb_fac =
  2.8401E+11  1.0000E+00  1.0000E+00  1.0004E+00
 wusun_mg, wun_mg, rmax_mg =  0.000E+00  0.806E+02  0.525E-09
 sums of boundary fluxes out [m**3/sec] (i1,iL,j1,jL,total) =
 -2.0478E+06 -1.7349E+04  2.0702E+06 -5.0495E+03  4.6209E+00
 total boundary fluxes in, out, [m**3/sec], %difference =
  4.2385E+06  4.2385E+06  1.0902E-04

  Now writing 6 2D arrays for head, etc.
              to file FOR0          40
  (Requires 8 seconds on MicroVAX for 40 x 34 grid.)
  Now writing head...
  Now writing con_salt...
  Now writing uscf...
  Now writing vscf...
  Now writing spec_stor_dhdt...
 new time, clim_lake_fac, clim_rchb_fac, clim_hedb_fac =
  3.1557E+11  1.0000E+00  1.0000E+00  1.0092E+00
 wusun_mg, wun_mg, rmax_mg =  0.000E+00  0.806E+02  0.221E-07
 sums of boundary fluxes out [m**3/sec] (i1,iL,j1,jL,total) =
 -2.5255E+06 -1.7091E+04  2.5473E+06 -5.1839E+03 -4.8231E+02
 total boundary fluxes in, out, [m**3/sec], %difference =
  5.2677E+06  5.2672E+06  9.1565E-03

  Now writing 6 2D arrays for head, etc.
              to file FOR0          40
  (Requires 8 seconds on MicroVAX for 40 x 34 grid.)
```

```
 Now writing head...
 Now writing con_salt...
 Now writing uscf...
 Now writing vscf...
 Now writing spec_stor_dhdt...
 BACK IN SECO_FLOW_2.
 Now calling SET_IBC_2L ...
 IN SET_IBC_2L,
FILE SUFFIX OF REGIONAL GRID =  20
FILE SUFFIX OF LOCAL GRID =  50
FILE SUFFIX OF REGIONAL GRID SOLUTIONS =  40
FILE SUFFIX TO WRITE INTERPOLATED LOCAL GRID
INITIAL AND BOUNDARY CONDITIONS =  60
 IDFIL_GREG =           20
 Now reading 2 2D arrays and 4 1D arrays.
 (Requires 8 seconds on MicroVAX for 40 x 34 grid.)
 IDFIL_GLOC =           50
 Now reading 4 1D arrays and 2 2D arrays.
 (Requires 8 seconds on MicroVAX for 40 x 34 grid.)
 For time level #           0
 Now reading 6 2D arrays for head, etc.
             from file FOR0          40
 (Requires 8 seconds on MicroVAX for 40 x 34 grid for each time level)
 Now writing 4 2D arrays for head, salt and velocities.
             to file FOR0          60
 (Requires 4 seconds on MicroVAX for 40 x 34 grid for each time level)
 Finished in SET_IBC_2L with time level =          10
 BACK IN SECO_FLOW_2.
 Now calling SECO_2L ...
 seco_2L (2D Local grid) run...
 IDFIL_GLOC =           50
 Now reading 4 1D arrays and 11 2D arrays.
 (Requires 45 seconds on MicroVAX for 40 x 34 grid.)
IN READ_2LB, READ INTERPOLATED LOCAL GRID
INITIAL AND BOUNDARY CONDITIONS
from file FOR0 60.DAT)

 Now reading 4 2D arrays for head, con_salt, u, v.
 for time-dependent boundary conditions
             from file FOR0          60
 (Requires 4 seconds on MicroVAX for 40 x 34 grid.)

 Now reading 4 2D arrays for head, con_salt, u, v.
 for time-dependent boundary conditions
             from file FOR0          60
 (Requires 4 seconds on MicroVAX for 40 x 34 grid.)
From SET_TOLK: tolK =     2.923202038E-06
     WELL_ID, L_wells_on, n_wells =   0.000000000000000E+000 F           0

Now setting b.c. flags and values.

 boundary #, n_sections(nb) =           1           1
 n_sec, sec_end, type, where, value_A, value_B, cbcf =
         1   6625.00000000000
   HEAD
 0.000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000

 boundary #, n_sections(nb) =           2           1
 n_sec, sec_end, type, where, value_A, value_B, cbcf =
         1   6625.00000000000
   HEAD
 0.000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000

 boundary #, n_sections(nb) =           3           1
 n_sec, sec_end, type, where, value_A, value_B, cbcf =
         1   5750.00000000000
   HEAD
 0.000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000

 boundary #, n_sections(nb) =           4           1
```

```
 n_sec, sec_end, type, where, value_A, value_B, cbcf =
         1   5750.00000000000
   HEAD
 0.000000000000000E+000  0.000000000000000E+000  0.000000000000000E+000
 Boundary #            1

Left Boundary at i = 1, range of y =   0.000E+00  6.625E+03
 Section #, sec_end=            1   6625.00000000000
 nbc =            1
 Boundary #            2

Boundary at i = iL, range of y =   0.000E+00  6.625E+03
 Section #, sec_end=            1   6625.00000000000
 nbc =            1
 Boundary #            3

Lower Boundary at j = 1, range of x =   0.000E+00  5.750E+03
 Section #, sec_end=            1   5750.00000000000
 nbc =            1
 Boundary #            4
 Section #, sec_end=            1   5750.00000000000
 nbc =            1
new time, clim_lake_fac, clim_rchb_fac, clim_hedb_fac =
 3.1557E+10  1.0000E+00  1.0000E+00  1.0000E+00
sums of boundary fluxes out [m**3/sec] (i1,iL,j1,jL,total) =
 0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00
total boundary fluxes in, out, [m**3/sec], %difference =
 0.0000E+00  0.0000E+00  0.0000E+00

 Now writing 6 2D arrays for head, etc.
            to file FOR0          70
 (Requires 8 seconds on MicroVAX for 40 x 34 grid.)
 Now writing spec_stor_dhdt...
new time, clim_lake_fac, clim_rchb_fac, clim_hedb_fac =
 3.1557E+10  1.0000E+00  1.0000E+00  1.0000E+00
wusun_mg, wun_mg, rmax_mg =  0.335E+01  0.806E+02  0.229E-10
sums of boundary fluxes out [m**3/sec] (i1,iL,j1,jL,total) =
-4.9564E+02 -6.0403E+01  3.1756E+03 -2.6191E+03  4.7865E-01
total boundary fluxes in, out, [m**3/sec], %difference =
 3.7007E+03  3.7012E+03  1.2933E-02

 Now writing 6 2D arrays for head, etc.
            to file FOR0          70
 (Requires 8 seconds on MicroVAX for 40 x 34 grid.)
 Now writing spec_stor_dhdt...

 Now reading 4 2D arrays for head, con_salt, u, v.
 for time-dependent boundary conditions
            from file FOR0          60
 (Requires 4 seconds on MicroVAX for 40 x 34 grid.)
new time, clim_lake_fac, clim_rchb_fac, clim_hedb_fac =
 6.3114E+10  1.0000E+00  1.0000E+00  1.0000E+00
wusun_mg, wun_mg, rmax_mg =  0.000E+00  0.806E+02  0.373E-08
sums of boundary fluxes out [m**3/sec] (i1,iL,j1,jL,total) =
-7.3572E+02 -5.4391E+01  4.0462E+03 -3.2770E+03 -2.0960E+01
total boundary fluxes in, out, [m**3/sec], %difference =
 4.6824E+03  4.6615E+03  4.4863E-01

 Now writing 6 2D arrays for head, etc.
            to file FOR0          70
 (Requires 8 seconds on MicroVAX for 40 x 34 grid.)
 Now writing spec_stor_dhdt...

 Now reading 4 2D arrays for head, con_salt, u, v.
 for time-dependent boundary conditions
            from file FOR0          60
 (Requires 4 seconds on MicroVAX for 40 x 34 grid.)
new time, clim_lake_fac, clim_rchb_fac, clim_hedb_fac =
 9.4671E+10  1.0000E+00  1.0000E+00  1.0000E+00
wusun_mg, wun_mg, rmax_mg =  0.000E+00  0.806E+02  0.266E-10
```



```
 sums of boundary fluxes out [m**3/sec] (i1,iL,j1,jL,total) =
-7.2847E+02 -6.1917E+01  4.0621E+03 -3.2722E+03 -4.1427E-01
 total boundary fluxes in, out, [m**3/sec], %difference =
  4.6797E+03  4.6793E+03  8.8529E-03

  Now writing 6 2D arrays for head, etc.
              to file FOR0          70
  (Requires 8 seconds on MicroVAX for 40 x 34 grid.)
  Now writing spec_stor_dhdt...

  Now reading 4 2D arrays for head, con_salt, u, v.
  for time-dependent boundary conditions
              from file FOR0          60
  (Requires 4 seconds on MicroVAX for 40 x 34 grid.)
 new time, clim_lake_fac, clim_rchb_fac, clim_hedb_fac =
  1.2623E+11  1.0000E+00  1.0000E+00  1.0000E+00
 wusun_mg, wun_mg, rmax_mg =  0.000E+00  0.806E+02  0.407E-10
 sums of boundary fluxes out [m**3/sec] (i1,iL,j1,jL,total) =
-4.8903E+02 -6.8963E+01  3.1954E+03 -2.6166E+03  2.0860E+01
 total boundary fluxes in, out, [m**3/sec], %difference =
  3.7023E+03  3.7231E+03  5.6185E-01

  Now writing 6 2D arrays for head, etc.
              to file FOR0          70
  (Requires 8 seconds on MicroVAX for 40 x 34 grid.)
  Now writing spec_stor_dhdt...

  Now reading 4 2D arrays for head, con_salt, u, v.
  for time-dependent boundary conditions
              from file FOR0          60
  (Requires 4 seconds on MicroVAX for 40 x 34 grid.)
 new time, clim_lake_fac, clim_rchb_fac, clim_hedb_fac =
  1.5778E+11  1.0000E+00  1.0000E+00  1.0000E+00
 wusun_mg, wun_mg, rmax_mg =  0.000E+00  0.806E+02  0.266E-10
 sums of boundary fluxes out [m**3/sec] (i1,iL,j1,jL,total) =
-4.9803E+02 -6.1660E+01  3.1864E+03 -2.6265E+03  2.4663E-01
 total boundary fluxes in, out, [m**3/sec], %difference =
  3.7126E+03  3.7129E+03  6.6427E-03

  Now writing 6 2D arrays for head, etc.
              to file FOR0          70
  (Requires 8 seconds on MicroVAX for 40 x 34 grid.)
  Now writing spec_stor_dhdt...

  Now reading 4 2D arrays for head, con_salt, u, v.
  for time-dependent boundary conditions
              from file FOR0          60
  (Requires 4 seconds on MicroVAX for 40 x 34 grid.)
 new time, clim_lake_fac, clim_rchb_fac, clim_hedb_fac =
  1.8934E+11  1.0000E+00  1.0000E+00  1.0000E+00
 wusun_mg, wun_mg, rmax_mg =  0.000E+00  0.806E+02  0.373E-08
 sums of boundary fluxes out [m**3/sec] (i1,iL,j1,jL,total) =
-7.4016E+02 -5.4710E+01  4.0634E+03 -3.2896E+03 -2.1107E+01
 total boundary fluxes in, out, [m**3/sec], %difference =
  4.7014E+03  4.6803E+03  4.4996E-01

  Now writing 6 2D arrays for head, etc.
              to file FOR0          70
  (Requires 8 seconds on MicroVAX for 40 x 34 grid.)
  Now writing spec_stor_dhdt...

  Now reading 4 2D arrays for head, con_salt, u, v.
  for time-dependent boundary conditions
              from file FOR0          60
  (Requires 4 seconds on MicroVAX for 40 x 34 grid.)
 new time, clim_lake_fac, clim_rchb_fac, clim_hedb_fac =
  2.2090E+11  1.0000E+00  1.0000E+00  1.0000E+00
 wusun_mg, wun_mg, rmax_mg =  0.000E+00  0.806E+02  0.373E-08
 sums of boundary fluxes out [m**3/sec] (i1,iL,j1,jL,total) =
-7.3473E+02 -6.2073E+01  4.0860E+03 -3.2897E+03 -5.7431E-01
```

```
 total boundary fluxes in, out, [m**3/sec], %difference =
  4.7061E+03   4.7055E+03   1.2204E-02

  Now writing 6 2D arrays for head, etc.
              to file FOR0          70
  (Requires 8 seconds on MicroVAX for 40 x 34 grid.)
  Now writing spec_stor_dhdt...

  Now reading 4 2D arrays for head, con_salt, u, v.
  for time-dependent boundary conditions
              from file FOR0          60
  (Requires 4 seconds on MicroVAX for 40 x 34 grid.)
 new time, clim_lake_fac, clim_rchb_fac, clim_hedb_fac =
  2.5246E+11   1.0000E+00   1.0000E+00   1.0000E+00
 wusun_mg, wun_mg, rmax_mg =   0.000E+00   0.806E+02   0.212E-10
 sums of boundary fluxes out [m**3/sec] (i1,iL,j1,jL,total) =
 -4.9707E+02 -6.9072E+01   3.2259E+03 -2.6391E+03   2.0699E+01
 total boundary fluxes in, out, [m**3/sec], %difference =
  3.7361E+03   3.7568E+03   5.5251E-01

  Now writing 6 2D arrays for head, etc.
              to file FOR0          70
  (Requires 8 seconds on MicroVAX for 40 x 34 grid.)
  Now writing spec_stor_dhdt...

  Now reading 4 2D arrays for head, con_salt, u, v.
  for time-dependent boundary conditions
              from file FOR0          60
  (Requires 4 seconds on MicroVAX for 40 x 34 grid.)
 new time, clim_lake_fac, clim_rchb_fac, clim_hedb_fac =
  2.8401E+11   1.0000E+00   1.0000E+00   1.0000E+00
 wusun_mg, wun_mg, rmax_mg =   0.000E+00   0.806E+02   0.373E-08
 sums of boundary fluxes out [m**3/sec] (i1,iL,j1,jL,total) =
 -5.0781E+02 -6.1757E+01   3.2235E+03 -2.6539E+03   8.7451E-02
 total boundary fluxes in, out, [m**3/sec], %difference =
  3.7537E+03   3.7538E+03   2.3297E-03

  Now writing 6 2D arrays for head, etc.
              to file FOR0          70
  (Requires 8 seconds on MicroVAX for 40 x 34 grid.)
  Now writing spec_stor_dhdt...

  Now reading 4 2D arrays for head, con_salt, u, v.
  for time-dependent boundary conditions
              from file FOR0          60
  (Requires 4 seconds on MicroVAX for 40 x 34 grid.)
 new time, clim_lake_fac, clim_rchb_fac, clim_hedb_fac =
  3.1557E+11   1.0000E+00   1.0000E+00   1.0000E+00
 wusun_mg, wun_mg, rmax_mg =   0.000E+00   0.806E+02   0.404E-10
 sums of boundary fluxes out [m**3/sec] (i1,iL,j1,jL,total) =
 -7.5165E+02 -5.4810E+01   4.1070E+03 -3.3218E+03 -2.1264E+01
 total boundary fluxes in, out, [m**3/sec], %difference =
  4.7497E+03   4.7284E+03   4.4870E-01

  Now writing 6 2D arrays for head, etc.
              to file FOR0          70
  (Requires 8 seconds on MicroVAX for 40 x 34 grid.)
  Now writing spec_stor_dhdt...
  BACK IN SECO_FLOW_2.

 SECOFL2D CPU time is 0:12 (minute:second)


 *** END OF SECOFL2D ***
 SECOFL2D C-3.01VV (08/09/93)                    08/18/95 16:16:
```

Appendix III: Review Forms

This appendix contains review forms for the SECOFL2D User's Manual.

NOTE: Copies of the User's Manual Reviewer's Forms are available in the Sandia WIPP Central Files.