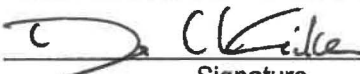
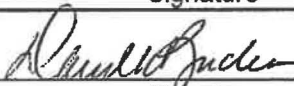
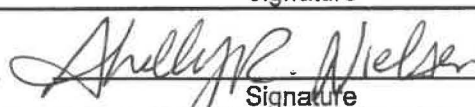

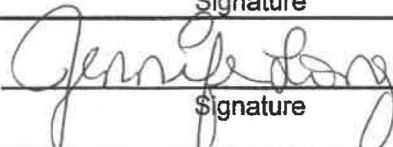


Appendix E

<p style="text-align: center;"><b>NUCLEAR WASTE MANAGEMENT PROCEDURE</b></p> <p><small>Sandia National Laboratories</small></p>	<h2 style="margin: 0;">Design Document Criteria</h2>	<p><b>Form Number: NP 19-1-4</b></p> <p><b>Page 1 of 1</b></p>
<p>1. <b>Software Name:</b> <u>DRSPALL</u></p> <p>2. <b>Software Version:</b> <u>1.22</u></p> <p>3. <b>Document Version:</b> <u>1.22</u></p> <p>4. <b>ERMS #:</b> <u>562640</u></p> <p style="font-size: small;">Prior to sign-off of the DD, all items shall be appropriately addressed by the code sponsor so that "Yes" may be checked. Include this form as part of the DD.</p>		
<p><b>Are the following appropriately defined and documented in the DD?</b></p> <p>5. Major Software Components <span style="float: right;"><input checked="" type="checkbox"/> Yes</span></p> <p>6. Technical description of the software with respect to: theoretical basis, embodied mathematical model, major control flow, control logic, and data structures <span style="float: right;"><input checked="" type="checkbox"/> Yes</span></p> <p>7. Functionalities and interfaces of object components, functions, and subroutines. <span style="float: right;"><input checked="" type="checkbox"/> Yes</span></p> <p>8. Allowable or prescribed ranges for inputs and outputs <span style="float: right;"><input checked="" type="checkbox"/> Yes</span></p> <p>9. Verifiability: Is the design verifiable through testing or other means? <span style="float: right;"><input checked="" type="checkbox"/> Yes</span></p> <p>10. Consistency and Traceability: Is the design consistent with and traceable to the software's requirements? <span style="float: right;"><input checked="" type="checkbox"/> Yes</span></p> <p>11. Technical Feasibility: Is the design technically feasible? <span style="float: right;"><input checked="" type="checkbox"/> Yes</span></p> <p>12. Implementation: Is the design presented in sufficient detail to allow for implementation as computer software? <span style="float: right;"><input checked="" type="checkbox"/> Yes</span></p>		
13.	<p><u>Dwayne C. Kicker</u> <b>Code Team/Sponsor</b> <i>(print)</i></p>	<p><u></u> Signature</p> <p><u>2/16/2015</u> Date</p>
14.	<p><u>David K. Rudeen</u> <b>Technical Reviewer</b> <i>(print)</i></p>	<p><u></u> Signature</p> <p><u>2-16-15</u> Date</p>
15.	<p><u>Shelly R. Nielsen</u> <b>QA Reviewer</b> <i>(print)</i></p>	<p><u></u> Signature</p> <p><u>4-15-15</u> Date</p>
16.	<p><u>Chris Camphouse</u> <b>Responsible Manager</b> <i>(print)</i></p>	<p><u></u> Signature</p> <p><u>4-15-15</u> Date</p>
17.	<p><u>Jennifer Long</u> <b>SCM Coordinator</b> <i>(print)</i></p>	<p><u></u> Signature</p> <p><u>4/15/15</u> Date</p>

Information Only

**WIPP PA**

**DESIGN DOCUMENT**

**for**

**DRSPALL Version 1.22**

**Document Version 1.22**

**ERMS# 562640**

**April 2015**

**Information Only**

## Table of Contents

1	Introduction.....	5
1.1	Software Identifier.....	5
1.2	Points of Contact .....	5
1.3	Description .....	5
1.4	Modifications to the DRSPALL Code .....	6
1.4.1	Boundary Conditions .....	7
1.4.2	Summary of Code Changes .....	8
2	Major Code Components.....	10
3	Theoretical Basis.....	11
3.1	Wellbore Flow Prior to Penetration.....	14
3.2	Repository Flow Prior to Penetration.....	14
3.3	Coupling of Wellbore and Repository Flow Prior to Penetration .....	15
3.4	Wellbore Flow After Penetration .....	16
3.5	Repository Flow After Penetration.....	16
3.6	Coupling of Wellbore and Repository After Penetration.....	17
3.7	Preparation for Verification and Validation Testing.....	17
4	Mathematical Model .....	17
4.1	Wellbore Equations .....	17
4.1.1	Wellbore Friction Factor.....	20
4.1.2	Viscosity of Wellbore Slurry .....	21
4.1.3	Wellbore Boundary Conditions .....	21
4.2	Wellbore-Repository Coupling Equations Prior to Penetration.....	22
4.3	Repository Equations .....	24
4.3.1	Non-Darcy Flow .....	25
4.3.2	Elastic Response and Failure .....	27
4.3.3	Fluidization .....	29
4.3.4	Repository Boundary Conditions.....	30
4.3.5	Implementation of Conceptual Submodels in DRSPALL .....	30
4.4	Wellbore-Repository Coupling Equations Following Penetration.....	32
4.5	Numerical Method – Wellbore.....	33
4.6	Numerical Method – Repository .....	35
4.6.1	Finite Difference Discretization.....	36
4.6.2	Boundary Conditions .....	38
4.6.3	Summary of Recommended Finite Difference Equations .....	39
4.6.4	Repository Stress State .....	40
4.7	Numerical Method – Coupling.....	41
4.8	Time Step Determination .....	41
5	Flow Chart with Functional Requirements Map.....	42
6	Data Structures (I/O).....	49
6.1	Input Files.....	49
6.1.1	Input Control and Parameter File.....	49

**Table of Contents (Continued)**

6.1.2	Input CDB File.....	50
6.2	Output Files .....	50
6.2.1	Output Text Files .....	50
6.2.2	CDB Output File.....	50
6.3	FORTTRAN 90 Global Variables Modules .....	51
7	Parameter Ranges.....	56
8	Test Case Control Logic .....	58
8.1	Porous Flow Test Problem .....	58
8.1.1	Constant Pressure Inner Boundary.....	59
8.1.2	Effective Bit Diameter .....	59
8.1.3	Special Output and Dimensionless Variables.....	60
8.2	Coalbed Methane Field Scale Test Case .....	61
8.3	Internal Logic Checks Test Problem .....	62
8.4	Wellbore Hydraulics Model Test Case .....	62
9	References.....	62
	Appendix A: Functional Requirements .....	65

### List of Figures

Figure 2-1. Flow chart of major code components.....	11
Figure 3-1. Schematic diagram of the flow geometry prior to repository penetration. ....	12
Figure 3-2. Schematic diagram of the flow geometry after repository penetration. Spherical case is shown. ....	13
Figure 3-3. Effective wellbore flow geometry before bit penetration. ....	14
Figure 3-4. Coupling of the repository and wellbore prior to drillbit penetration.....	15
Figure 3-5. Effective wellbore flow geometry after bit penetration.....	16
Figure 4-1. Drawing of typical radial effective stress curve in DRSPALL repository domain at some time during drill penetration. ....	31
Figure 4-2. Finite difference zoning for wellbore. ....	34
Figure 4-3. Finite difference molecules for the implicit scheme using constant zones sizes.....	40
Figure 5-1. Flow chart for program DRSPALL .....	43
Figure 5-2. Flow chart for subroutine RunLoop.....	44
Figure 5-3. Flow chart for subroutine CalculateWellBoreFlow .....	45
Figure 5-4. Flow chart for subroutines CalculateWasteStress and CalculateWasteFlow .....	46

### List of Tables

Table 4-1. Nomenclature for Wellbore Equations.....	17
Table 4-2. Nomenclature for Wellbore-Repository Coupling Equations. ....	22
Table 4-3. Nomenclature for Repository Equations. ....	24
Table 6-1. Summary of DRSPALL Input and Output Files. ....	49
Table 7-1. Input Parameters and Ranges .....	56
Table 8-1. Nomenclature for Section 8 - Test Case Control Logic . ....	58
Table 8-2. Special variables defined for Porous Flow Test Problem .....	61

## 1 INTRODUCTION

The purpose of this document is to describe the design elements of the code DRSPALL, Version 1.22. This version of DRSPALL runs on a Solaris platform. DRSPALL Version 1.22 corrects an error in the implementation of the finite difference equations in accordance with Software Problem Report (SPR) 13-001.

### 1.1 Software Identifier

DRSPALL, Version 1.22

### 1.2 Points of Contact

Code Sponsor: Dwayne C. Kicker  
Performance Assessment & Decision Analysis Dept.  
Sandia National Laboratories  
phone: (575) 628-0117  
e-mail: dckicke@sandia.gov

Code Consultant: Courtney G. Herrick  
Performance Assessment & Decision Analysis Dept.  
Sandia National Laboratories  
phone: (575) 234-0176  
e-mail: cgherri@sandia.gov

### 1.3 Description

DRSPALL (from Direct Release Spill) is written to calculate the volume of Waste Isolation Pilot Plant (WIPP) solid waste subject to material failure and transport to the surface as a result of an inadvertent drilling intrusion. The code calculates coupled repository and wellbore transient mixed-phase compressible fluid flow before, during, and after the drilling intrusion process. Mathematical models are included for bit penetration, mixed-phase (mud, salt, waste, and gas) fluid flow in the well, fluid expulsion at the surface, coupling of the well and the drilled repository, repository spalling (tensile) failure, fluidized bed transport of failed waste, and repository internal gas flow. The wellbore model is one-dimensional with linear flow, while the repository model is one-dimensional with either spherical or cylindrical radial flow.

Previous work (Hansen et al. 1997) has shown that the parameter extremes most likely to be of importance to spalling releases are high repository gas pressure, low waste mechanical strength, small waste particle size (after failure), and low waste permeability. DRSPALL includes these parameters as input variables. Since conventional drilling practices are assumed in U.S. Environmental Protection Agency (EPA) regulation of the WIPP, allowance will be made for the continuous pumping of mud during drilling, the possible stoppage of the mud pump, continuous drilling after repository penetration, and the possible stoppage of drilling, based on observations made at the surface by the driller. Of course, one possibility

is the continuance of pumping and drilling despite the intrusion into a high-pressure subsurface region.

Verification and validation of the main components of the DRSPALL model will be possible by comparison with other codes or independent solutions. However, it will be difficult to validate the full DRSPALL model in its entirety. Some validation can be achieved by comparison with selected analog models and experiments that have been developed by others, generally for use in the oil and gas industry. Parts of the ranges of behavior covered by these analogs should be applicable to waste repository conditions. Preparation for such work is included as part of the design process.

#### 1.4 Modifications to the DRSPALL Code

SPR 13-001 states that the source code file *wasteflowcalc.f90* contains an error in the implementation of the finite difference equations. This source code file contains three Forchterm equations (for the first cell, the interior cell, and the last cell), with each Forchterm equation as follows:

$$\text{Forchterm} = \frac{k'(i+1) - k'(i)}{4 k' \Delta r(i)} \quad (1.4.1)$$

where  $k'$  = permeability, and  
 $\Delta r$  = repository zone size.

However, in accordance with the previous version of the DRSPALL design document (WIPP PA 2004, Equation 4.6.3; WIPP PA 2013), which is based on a centered-difference discretization, the correct equation should be:

$$\text{Forchterm} = \frac{k'(i+1) - k'(i-1)}{4 k'(i) \Delta r(i)} \quad (1.4.2)$$

In response to SPR 13-001, the finite difference solution to the DRSPALL waste flow equation was evaluated. In addition to the error identified in SPR 13-001, it was found that the derivation of the constant zone size equations was also incorrect. Equation 4.6.2 in the previous design document (WIPP PA 2004) is incorrect because  $k'$  is shown as a constant in the denominator, despite it being a variable in the numerator. The approach for modifying the DRSPALL code was to start from Equation 4.3.10 in Section 4.3 and clarify the original Equation 4.6.1 (WIPP PA 2004), which is documented in Section 4.6. This essentially results in a simplification of the original Equation 4.6.1 to include a natural log term, as shown in Equation 4.6.4 in Section 4.6. The original Equation 4.6.2 in the previous design document was re-derived as shown by Equation 4.6.5 in Section 4.6.1.

The finite difference discretization is then performed for zones of constant size. Section 4.6 provides a simplified version of the final derivation (Equation 4.6.6), which is in fact identical to the previous design document (WIPP PA 2004, Equation 4.6.3), except that the

coefficient terms  $\alpha_1$  and  $\alpha_2$  are different due to the correction for the spatial variability of  $k'$ , which results in the natural log term.

A variable zone size implementation was described based on the previous DRSPALL design document (WIPP PA 2004, Section 4.6; WIPP PA 2013). However, this was done incorrectly, as a simple substitution of variable zone sizes into the equation derived for a constant zone size is not valid. The derivation of an equation similar to Equation 4.6.5 (Section 4.6.1) for a variable zone size would require a complete re-derivation, which was determined unnecessary because current computing resources allow for reasonably fast computational times even for a greater number of zones, such that an increasing zone size is not needed. So it was decided to run DRSPALL exclusively with a constant zone size (it will now fail in source code file *A1main\_drspall.f90* if a growth rate other than 1.0 is input).

These changes are made in the CalculateWasteFlowImplicit routine of the source code file *wasteflowcalc.f90* in the calculation of 'Forchterm' for first, interior, and last cell coefficients. In correcting the calculation of 'Forchterm', the indexing of the second permeability() term was also corrected to be 'i-1' instead of 'i'. The coefficients for the last cell ( $i = \text{numReposZones}$ ) have changed:  $aa(i)$  has been changed from '-alpha1' to '-alpha1-alpha2' and  $bb(i)$  has been changed from '1.0+alpha1' to '1.0+alpha1+alpha2'.

It should be noted that since a constant zone size is used exclusively, values for both 'reposDR(i)' (distance between zone edges) and 'reposDRH(i)' (distance between zone centers) will now be the same.

#### 1.4.1 Boundary Conditions

In the CalculateWasteFlowImplicit routine of *wasteflowcalc.f90*, the index of the "first cell coefficients" (i) has been changed from 'firstIntactZone' to 'firstIntactZone+1', since any values for the boundary ('firstIntactZone') would be constant and fixed by the specified pressure (Dirichlet) boundary condition in the cavity. That is, the boundary nodes are not included in the coefficient matrix, so there should be no  $aa(i)$ ,  $bb(i)$ ,  $cc(i)$  coefficients for 'firstIntactZone'. The effect of the boundary node (firstIntactZone) is included in the b-vector of the linear system of equations. Consequently, the indexing for the "interior cell coefficients" now begins at 'firstIntactZone+2' instead of 'firstIntactZone+1'. Also, as a consequence of this, the indexing of the matrix inversion has changed (lines 230-245). The boundary pressure is now assigned to 'reposPres(firstIntactZone)' instead of 'reposPres(0)'. Because of that, 'exitPoreVelocity' is now calculated using a centered-difference approximation, which leads to 'reposPres(firstIntactZone+1)' being used instead of 'reposPres(firstIntactZone)'.

Previously, the permeability of the 'firstIntactZone-1' zone was set to the value of the 'firstIntactZone'. This was changed by eliminating that assignment (line 32 of *wasteflowcalc.f90*) because the permeability of the 'firstIntactZone-1' is no longer used. Also, where previously the array element 'psi(firstIntactZone-1)' was calculated from the gas viscosity and boundary pressure, this assignment has been made applicable to 'psi(firstIntactZone)' (line 37 of *wasteflowcalc.f90*), since the 'firstIntactZone' is the boundary.



### 1.4.2 Summary of Code Changes

A summary of the changes in each source code file is described as follows:

- *A1main\_drspall.f90*
  - Checks of ‘growthRate’ and ‘wellGrowthRate’ were added to make sure they are set to ‘1.0’ in the input file, since the code will now be run exclusively with constant zone sizes. If not, the code exits with an error.
  - An additional output file, *fluidization\_time.dat*, was included as part of test case #4. This new output file provides the fluidization time for each zone to improve the ability to verify the DRSPALL fluidization time calculation. Specific changes to this source code file include:
    - Added file id for *fluidization\_time.dat* (line 114).
    - Added definition of filename of *fluidization\_time.dat* (line 929).
    - Added opening of *fluidization\_time.dat* (line 935).
    - Added writing of QA information to *fluidization\_time.dat* (line 969).
- *cdbcontrol.f90*

No changes.
- *cdbglobals.F90*

No changes.
- *globals.F90*

As a result of the new output file for test case #4, new definitions are required to pass values between subroutines. Specifically, *TC4FluidTimefilename* and *fluidizationTimeValidationFileID* variables were added (lines 165 and 173).
- *maincalc.f90*

A check of the timestep calculation was added for the ‘firstIntactZone’ that will skip the calculation if ‘cavityPres’ is zero. Because of the changes to the boundary condition, this change was needed to run validation test case 1.
- *parameters.f90*

The upper bound for characteristic length,  $L_t$ , was set to 10 for zone size sensitivity testing.
- *setupcalc.f90*
  - The ‘wellArea’ array definition was updated to calculate fractions 2/3 and 1/3 instead of the given decimal values of 0.6667 and 0.3333 (also done later for ‘reposVol’).
  - The ‘numReposZones’ definition was updated to work correctly with a constant zone size (added 1).
  - Updated the ‘reposZoneSize’ definition to work correctly with a constant zone size (subtracted 1 from denominator). The original version of the code failed when a constant zone size was selected.

- Added call to QAABORT that exits out of the section of code that previously recalculated the number of zones based on a growth rate greater than 1.0.
  - In the definition of ‘reposDR’, the option to calculate its values using the cell growth rate was commented out, which ensures that ‘reposDR’ array values will all be the same (constant zone size).
- *vmsfilewrite.f90*  
The following changes only affect the output for test case #4:
    - Added closing of fluidization\_time.dat (line22).
    - Added writing of header data to fluidization\_time.dat (line 231).
    - Increased output for stress validation file to indices less than 250 (line 383).
    - Redefined when information is written to the fluidization.dat file. Previously, data was only written when the first intact zone was less than or equal to zone 20 (or between 100 and 150). Based on the other changes made to the code, the relevant output is now needed when the first intact zone is less than or equal to zone 70 (or between 100 and 150), so the ‘if’ statement has been modified (line 422).
    - Added subroutine WriteToFluidizationTimeValidationFile() that writes fluidization times to fluidization\_time.dat (line 461).
  - *vmsoutputcontrol.f90*  
No changes.
  - *wasteflowcalc.f90*
    - Added definition of ‘permEnhanceFactor’ and set value to 4.0; this results in no change, since the factor of 4.0 was already used, just not defined as a variable.
    - The factor of 0.5 was eliminated in the definition of ‘curGasDen’ to be consistent with the ideal gas law.
    - The definition of ‘permeability(i)’ was updated to include the variable ‘permEnhanceFactor’ (again, no change since the value is still 4.0).
    - The definition of ‘permeability(firstIntactZone-1)’ was removed (commented out).
    - The index of ‘psi()’ was changed from ‘firstIntactZone-1’ to ‘firstIntactZone’.
    - The variable ‘temp’ (equal to ‘gasViscosity\*psi(i)’) was created to use in a test to prevent the square root of a negative number. Inserted a QAABORT call if the code is about to take the square root of a negative number.
    - The boundary pressure (‘boundaryPres’) is set equal to ‘reposPres(firstIntactZone)’ instead of ‘reposPres(0)’.
    - The variable ‘exitPoreVelocity’ is defined using ‘(reposPres(firstIntactZone+1)-boundaryPres)’ instead of ‘(reposPres(firstIntactZone)-boundaryPres)’.
    - Definition of ‘i’ is ‘firstIntactZone+1’, instead of ‘firstIntactZone’ (line 171).
    - The definition of ‘ForchTerm’ is now
 
$$\text{Forchterm} = \log(\text{permeability}(i+1)/\text{permeability}(i-1))/(4.0*\text{reposDR}(i))$$
 instead of
 
$$\text{Forchterm} = (\text{permeability}(i+1)-\text{permeability}(i))/(\text{permeability}(i)*4.0*\text{reposDR}(i))$$

This is due to a re-derivation of Equation 4.6.6 (see the definition of the coefficient terms  $\alpha_1$  and  $\alpha_2$  in Section 4.6.1) and is done in three sections of the code.

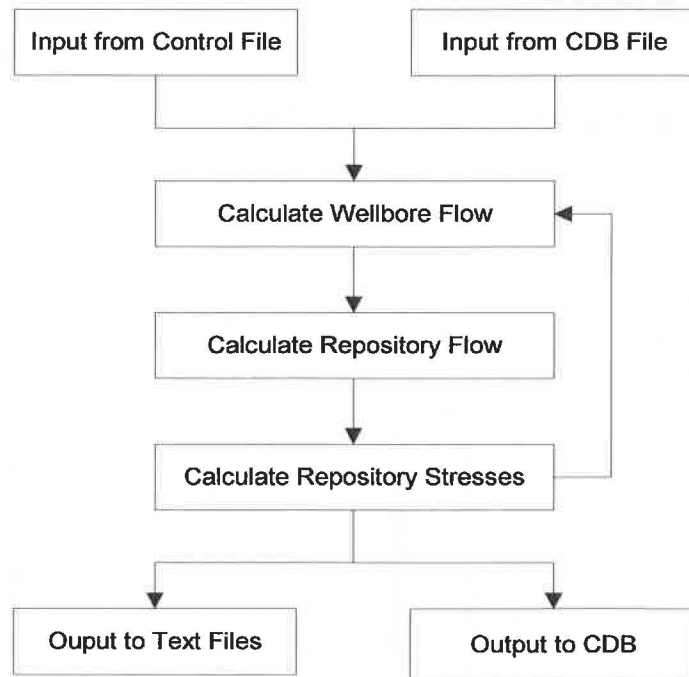
- Index of 'i' starts at 'firstIntactZone+2', instead of 'firstIntactZone+1' (line 190).
  - Definition of 'aa(i)' is now '-alpha1-alpha2' instead of '-alpha1'.
  - Definition of 'bb(i)' is now '1.0d0 + alpha1 + alpha2' instead of '1.0d0 + alpha1'.
  - Definition of 'bet' is now 'bb(firstIntactZone+1)' instead of 'bb(firstIntactZone)'.
  - The definition 'psi(firstIntactZone) = rr(firstIntactZone)/bet' has been replaced by 'psi(firstIntactZone+1) = rr(firstIntactZone+1)/bet'.
  - Index of 'i' starts at 'firstIntactZone+2', instead of 'firstIntactZone+1' (line 232).
  - Index of 'i' ends at 'firstIntactZone+1', instead of 'firstIntactZone' (line 243).
- *wastestresscalc.f90*  
In the definition of temp1, the index of reposRadius() was changed from:  
temp1 = (reposRadius(firstIntactZone-1)/reposRadius(i))\*\*geomExponent  
to  
temp1 = (reposRadius(firstIntactZone)/reposRadius(i))\*\*geomExponent.
  - *wellborecalc.f90*
    - Added 'elseif' possibility for 'fractionFluidized(i)' calculation. Previously, 'fractionFluidized(i)' was set to 1.0 only if 'i' was for the 'firstIntactZone'. Otherwise, the 'fractionFluidized' could increase to values much higher than 1.0, which are not physically reasonable, and led to problems in calculating 'permeability' in *wasteflowcalc.f90*. The 'elseif' statement was added to set 'fractionFluidized(i)' to a number slightly larger than 1.0 (1.0001), such that the 'if' statement will be satisfied for zone 'i' in a later step.
    - Added call to a new subroutine WriteToFluidizationTimeValidationFile(). This subroutine writes fluidization times to fluidization\_time.dat, which is an output file added for test case #4 to facilitate the verification of fluidization times.

## 2 MAJOR CODE COMPONENTS

The major code components consist of an input control file, an optional input CDB file, the main computational loop, and output files, as shown in Figure 2-1.

The use of CDB files for input and output is optional. Input parameters can be provided in the input control file or read from the input CDB. However, input parameters are always echoed to the diagnostic output file. For any run, all files have the same prefix name assigned by the user, while the suffix names denote the type of file.

DRSPALL makes use of the WIPP performance assessment (PA) libraries CAMCON\_LIB, CAMSUPES\_LIB and CAMDAT\_LIB to standardize common user interface and I/O functions and to simplify CDB access (WIPP PA 1996a, WIPP PA 1996b, WIPP PA 1996c). These libraries are linked to the DRSPALL executable during the build process as documented in the *Implementation Document for DRSPALL Version 1.22* (WIPP PA 2015a).



*Figure 2-1. Flow chart of major code components.*

The CAMCON\_LIB library provides a library of standardized QA and user interface routines such as a code banner with credits (authors), software ID, revision date, and current date and time; file processing routines to read and parse the command line, and to specify and open files; free-field input routines; and string processing routines.

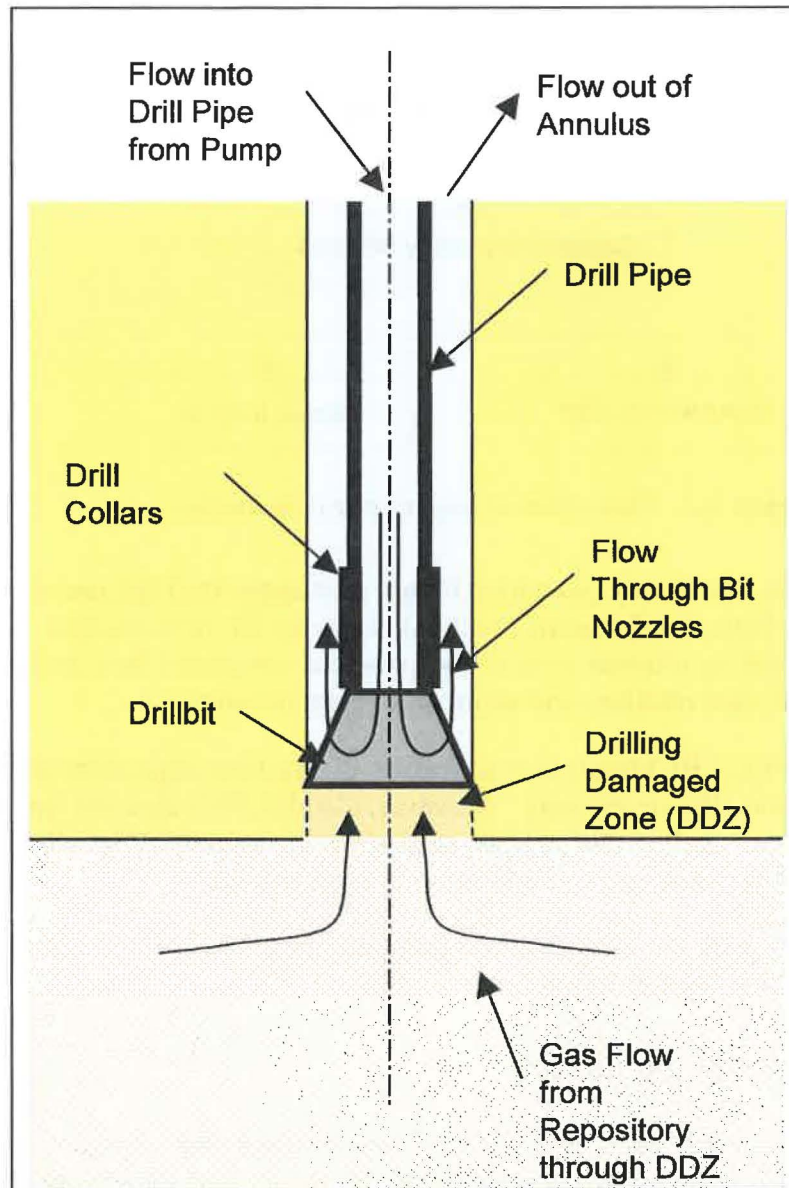
The CAMSUPES\_LIB library is a collection of machine dependent routines to access the system clock and allocate memory. However, CAMSUPES dynamic memory allocation has not been implemented in this version of DRSPALL except in the CAMDAT\_LIB library. FORTRAN 90 dynamic memory allocation is used instead.

The CAMDAT\_LIB library provides a programmer's interface through subroutine calls for accessing and creating CDB files.

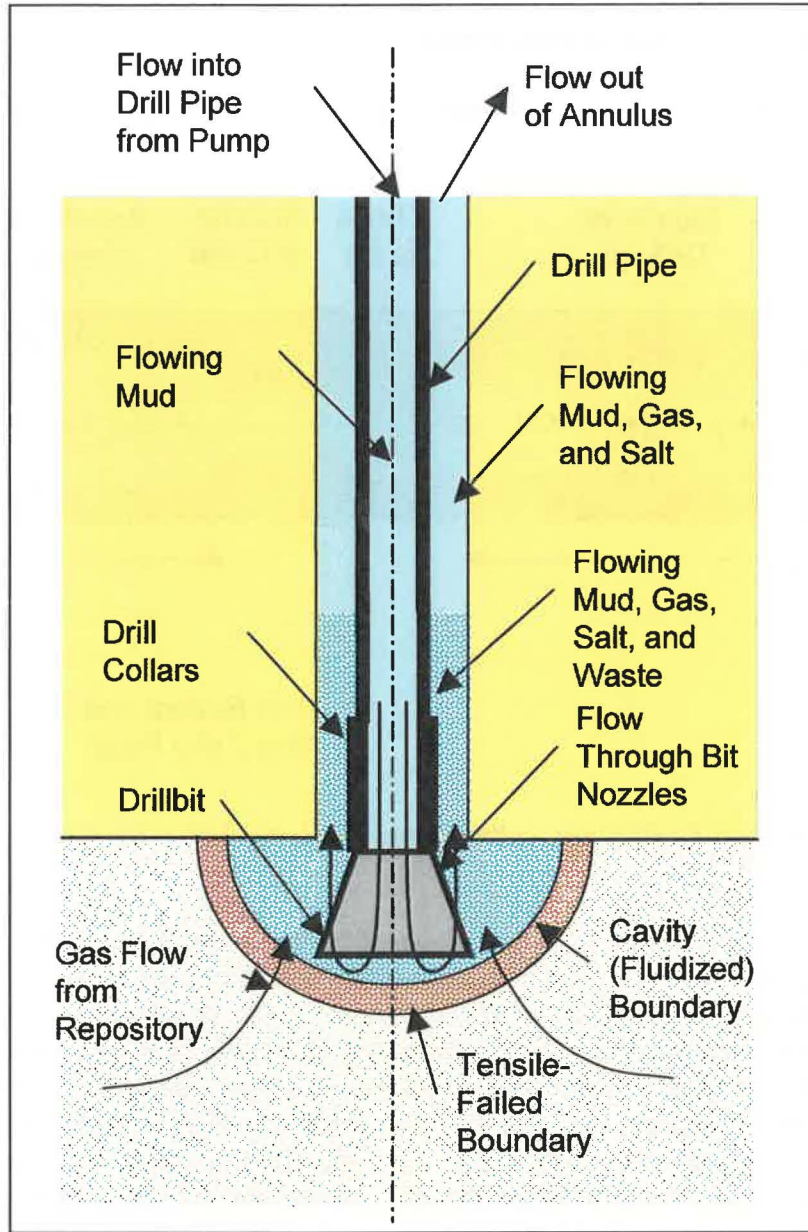
### 3 THEORETICAL BASIS

DRSPALL is based on the theory of one-dimensional time-dependent compressible isothermal fluid flow. Somewhat different forms of that theory are used, depending on whether the flow is in the wellbore or in the repository, and whether the wellbore currently penetrates the repository or not. The wellbore and repository flows are coupled at a specified boundary by a set of conditions. Schematic diagrams of the flow geometry prior to and after penetration are shown in Figure 3-1 and Figure 3-2, respectively. Throughout the process, the drillbit can move downward as a function of time, removing salt or waste material. Flow in the well is treated as a compressible, viscous, multi-phase mixture of mud, gas, salt, and

possibly waste solids. Flow in the repository is treated as viscous, compressible single-phase gas flow in a porous solid. At the cavity forming the repository-wellbore boundary (following penetration) waste solids freed by drilling, tensile failure, and associated fluidization may enter the wellbore flow stream. Between the well and the repository, flow is treated according to the state of penetration, as described later.



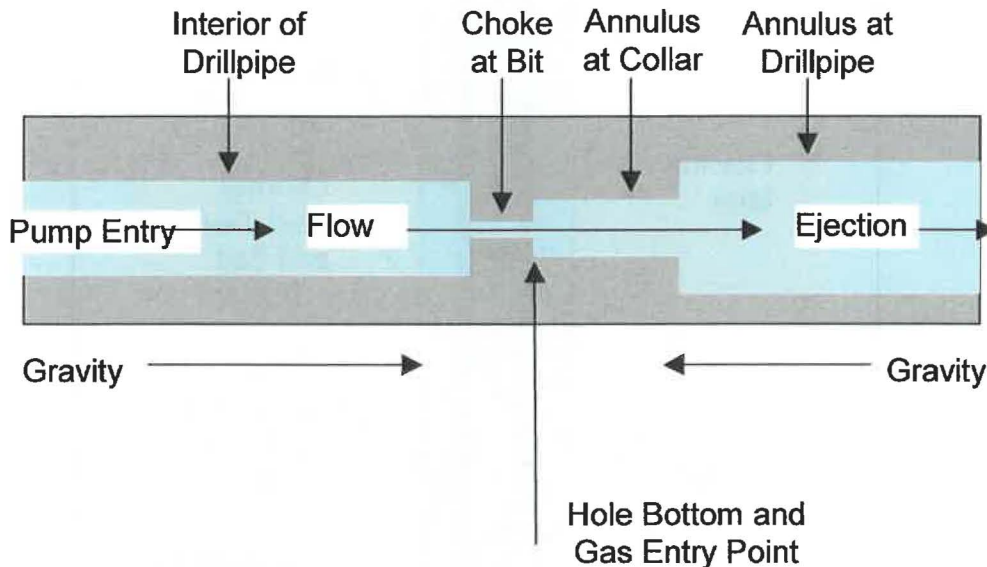
*Figure 3-1. Schematic diagram of the flow geometry prior to repository penetration.*



*Figure 3-2. Schematic diagram of the flow geometry after repository penetration. Spherical case is shown.*

### 3.1 Wellbore Flow Prior to Penetration

Flow in the well prior to drillbit penetration into the repository is treated as shown in Figure 3-3.



*Figure 3-3. Effective wellbore flow geometry before bit penetration.*

In concept this model is similar to that proposed by Podio and Yang (1986) and now in use in the oil and gas industry. The pump input is controlled by volume flow rate (with a maximum pressure), and may be stopped or started. Flow is one dimensional pipe flow with cross sectional areas corresponding to the appropriate flow area at a given position in the well. The drillbit is treated as a choke with cross-sectional area appropriate to the bit nozzle area. Immediately below the bit, gas from the repository can flow through drilling-damaged salt into the well, as described in Section 3.3. After the bit, return flow to the surface is treated as pipe flow in the annulus separating the drill string and the outer hole boundary. At the annulus output to the surface, mud ejection is to a constant atmospheric pressure. The gravitational body force acts in its appropriate direction based on position before or after the bit. The well fluid is Newtonian. Viscosity of the phase mixture is determined by an approximation as discussed in Section 4.1. Flow friction may be either laminar or turbulent, also based on an approximation given in Section 4.1.

### 3.2 Repository Flow Prior to Penetration

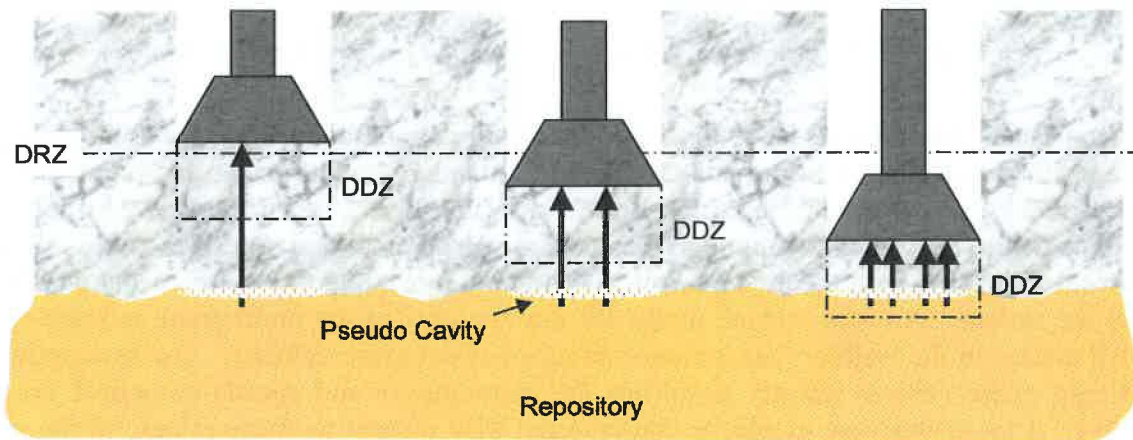
The flow in the repository is one-dimensional radial, either hemispherical or cylindrical with its origin at the centerline of the well. Darcy flow of an isothermal ideal gas in a porous medium allows the simplifying pseudopressure approach to be taken, as is commonly done in the field of petroleum reservoir engineering. A modification allowing for non-Darcy inertial effects at high flow velocities is included. Near the well, the boundary pressure is

determined as discussed in the next section on coupling. The outer boundary of the repository is set to a no flow condition, consistent with repository far-field conditions.

Solid stresses in the repository are treated as elastic and with an effective stress law. Flow-related forces are included. A homogeneous solid is assumed. Solid stresses are not relevant prior to penetration, but play a key role in determining failure and fluidization after penetration, as discussed later.

### 3.3 Coupling of Wellbore and Repository Flow Prior to Penetration

Prior to penetration, it is assumed that the drilling action creates a cylinder of altered-permeability salt material with diameter equal to the drillbit that moves ahead of the drillbit. This material allows limited porous gas flow from the repository to the wellbore. This is shown simply as the drilling damaged zone (DDZ) in Figure 3-1. Additional conceptual detail is shown in Figure 3-4. The permeability of this altered zone is greater than that of the intact salt and the disturbed rock zone (DRZ), and is based on a composite model as described in the Section 4.2.



*Figure 3-4. Coupling of the repository and wellbore prior to drillbit penetration.*

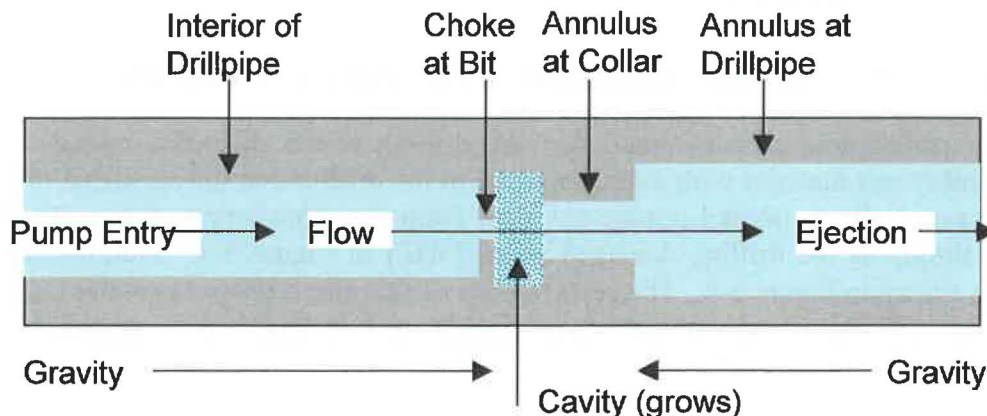
Porous flow through this cylindrical volume is determined by the boundary pressures and flow areas of the volume. The boundary pressure on the well side is the pressure immediately below the bit. The area is the bit cross-sectional area (also assumed to be the wellbore area). The boundary pressure on the repository side is the pressure in a small pseudo-cavity at the wellbore face of the waste with the same surface area as the end of the drillbit. The pseudo-cavity provides continuity between the wellbore and repository and does not play a material role in total flow.

The main effect of the DDZ is to smooth the step function in pressure and flow rate that would otherwise occur upon bit penetration of the repository.



### 3.4 Wellbore Flow After Penetration

After drillbit penetration, the pseudo-cavity at the waste face becomes a real cavity and is connected to the flow area of the wellbore as shown in Figure 3-5. The cavity is now representative of the material removed from the repository by drilling, material failure, and subsequent fluidization.



*Figure 3-5. Effective wellbore flow geometry after bit penetration*

After penetration, the cavity is treated as a mixing volume connecting the wellbore and the repository, and is used to provide the pressure boundary conditions at the inner wall of the repository. The volume of the cavity zone shown in Figure 3-5 is constantly adjusted so that it is equal to the growing volume of the actual hemispherical or cylindrical cavity. Fluidized waste solids mix with the mud/gas/salt flow and are moved up the wellbore and possibly out at the surface. An approximate model for the viscosity of the multi-phase mixture is used. All phases in the wellbore are assumed to move at the same velocity. The assumption of a single phase velocity greatly simplifies the mathematics and speeds numerical execution time. This assumption should be conservative with respect to waste release to the surface because a lagging waste particle velocity is expected in the wellbore, leading to a lower surface release rate and lower total release. The viscosity model causes the mixed fluid to act as a dense slurry for high solids concentrations, and there is a relative solids volume which causes complete choking. Treating the fluidized cavity as a region in one-dimensional series with the wellbore flow is a simplification. However, it is consistent with the one-dimensional nature of the flows in both wellbore and repository, and is a simple and convenient way of introducing the boundary coupling without introducing more complexity than knowledge of parameters justifies.

### 3.5 Repository Flow After Penetration

The flow model is basically the same after penetration as before. Without failure, the inner surface area of the repository is maintained at the same as the surface area of the true drilled borehole cavity. If radial tensile failure occurs as a result of pressure relief at the inner boundary and flow-related forces, the waste material is deemed susceptible to fluidization in

the region of tensile failure. Then the pore fluid flow velocity is compared to a limiting velocity based on fluidized bed theory. If this fluidization velocity is exceeded, material is assumed to disaggregate and move into the cavity, thus enlarging the cavity and adding waste solids to it. The cavity volume is determined by cavity radius. The cavity radius is the sum of two components: the drilling radius and the spallings depth. The drilling radius is the radius of the volume with surface area equivalent to the surface area of the cylinder removed by drilling, as explained in Section 4.4. The spallings depth is the depth to which fluidization extends beyond the current cavity radius, as explained in Section 4.3.3.

### 3.6 Coupling of Wellbore and Repository After Penetration

As mentioned above, the coupling of the wellbore and the repository following penetration is achieved by treating the cavity as a mixing volume (Figure 3-5). If no fluidization occurs but drilling continues, the cavity grows at a rate equivalent to the volume being drilled. If fluidization occurs, the cavity volume may be greater than the drilled volume. Waste moves into the cavity when it is fluidized. Mass balance is maintained at all times. Mud flows into the cavity through the nozzles at the bit. Gas, waste and salt are moved into the cavity as drilling and fluidization occur. Subsequently, all components progress up the well. Salt and waste solids are treated as incompressible. Volumes of the compressible phases (gas and mud) are calculated based on their equations of state and the current pressure.

### 3.7 Preparation for Verification and Validation Testing

Isolation of model components for testing requires special code logic for each test case. To facilitate testing using the production executable without modifications, the special logic is incorporated prior to the production build and is activated via user-controllable input switches. Warning messages are generated that clearly indicate that DRSPALL is being executed in the testing mode.

## 4 MATHEMATICAL MODEL

DRSPALL is implemented as a coupled system of differential equations and evaluated by time-marching integration. This section describes the system of equations used and the numerical techniques involved.

### 4.1 Wellbore Equations

Nomenclature is given in Table 4-1.

*Table 4-1. Nomenclature for Wellbore Equations.*

Symbol	Definition	Units
$A$	Flow area	$\text{m}^2$
$A_p$	Inlet area of mud pump	$\text{m}^2$
$A_{out}$	Outlet area of well	$\text{m}^2$
$c_m$	Compressibility of mud	$\text{Pa}^{-1}$

**Table 4-1. Nomenclature for Wellbore Equations (Continued).**

Symbol	Definition	Units
$d_h$	Hydraulic diameter	m
$D_i, D_o$	Inner and outer diameters	m
$g$	Acceleration of gravity	m/s <sup>2</sup>
$i$	Zone index	--
$m_w, m_s$	Mass of waste and salt	kg
$n$	Timestep index	--
$P$	Pressure	Pa
$R_{mud\ pump}$	Volumetric rate of mud pump	m <sup>3</sup> /s
$Re$	Reynold's number	--
$S$	Exponent in slurry viscosity formula	--
$S_{mass}$	Mass source term	kg / s
$S_{mom}$	Momentum source term	kg m/s <sup>2</sup>
$ss_m$	Sound speed in mud	m/s
$t$	Time	s
$u$	Velocity	m/s
$u_{out}$	Fluid velocity at outlet boundary	m/s
$V$	Volume	m <sup>3</sup>
$V_w, V_s, V_g, V_m$	Volumes of waste, salt, gas, and mud	m <sup>3</sup>
$w$	Volume fraction of waste solids	--
$w_{max}$	Maximum allowed value of $w$	--
$z$	Distance from pump inlet	m
$\varepsilon$	Roughness of wall	m
$F$	Friction loss term	kg / m <sup>2</sup> s <sup>2</sup>
$f$	Darcy-Weisbach friction factor	--
$\eta$	Viscosity	kg / m s
$\rho$	Density	kg / m <sup>3</sup>
0	Sub or superscript denoting initial value	--

Flow in the well is transient, compressible, isothermal, viscous, and multi-phase, including a mixture of mud, gas, salt, and possibly waste solids. Leakoff to the walls is ignored. All equations are adaptations of the well-known Navier-Stokes equations for compressible fluid flow. For example, see Oran and Boris (1987). The one-dimensional conservation equations that apply in this case are, for mass balance:

$$\frac{\partial}{\partial t}(\rho V) + \frac{\partial}{\partial z}(\rho V u) = S_{mass} \quad (4.1.1)$$

and for momentum balance:

$$\frac{\partial}{\partial t}(\rho V u) + \frac{\partial}{\partial z}(\rho V u^2) = -V \left( \frac{\partial P}{\partial z} - \rho g + F \right) + S_{mom} \quad (4.1.2)$$

where the value of the dissipation term in  $F$  cannot exceed the total momentum in the current flow direction. (That is, local dissipation can slow or stop the flow, but not reverse it.) To complete the system, an equation of state relating pressure and density and the friction loss term must be defined. Also, the source term and boundary conditions must be specified.

Density  $\rho$  in Equations 4.1.1 and 4.1.2 is the mixture density, computed as the sum of the masses of individual phase components in a representative volume, divided by that volume. The mixture equation of state must account for the four possible mass components in the well: mud, salt, gas, and waste. Because of known initial conditions, mud pumping rate, surface effluent rate, and the gas and solids exchange rate at the cavity-wellbore interface, the convection equations result in a known mass of each component in a given total volume. Then the individual volumes occupied by each component are determined, where

$$V = V_s + V_w + V_m + V_g \quad (4.1.3)$$

The salt and waste components are treated as incompressible, so their volumes are independent of pressure and determined from their initial user-specified densities by

$$V_s = \frac{m_s}{\rho_{0(s)}}$$

and

$$V_w = \frac{m_w}{\rho_{0(w)}} \quad (4.1.4)$$

The gas is treated as isothermal and ideal, so

$$\rho_g = \rho_{0(g)} \frac{P}{P_0} \quad (4.1.5)$$

The mud is assumed to be a compressible liquid, so

$$\rho_m = \rho_{0(m)} [1 + c_m (P - P_0)] \quad (4.1.6)$$

where  $P_0$  is atmospheric pressure. To calculate pressure at a given location, note that the masses and total volume will always be known. Also, the volumes of salt and waste will be known, since they are considered incompressible. This means the total combined volume of gas and mud are known. This allows Equations 4.1.5 and 4.1.6 to be combined, resulting in a quadratic equation for gas volume:

$$aV_g^2 + bV_g - c = 0 \quad (4.1.7)$$

where,

$$\begin{aligned} a &= 1 - c_m P^0 \\ b &= (c_m P^0 V_g^0 - a V^* + V_m^0) \\ c &= V^* c_m P^0 V_g^0 \end{aligned}$$

and

$$V^* = V_m + V_g = V - V_s - V_w$$

Once gas volume is known, densities can be determined and pressure can be evaluated from either the gas (Equation 4.1.5) or mud equation of state (Equation 4.1.6).

For friction loss, a standard formulation for pipe flow is used (Fox and McDonald 1985), where the head loss per unit length is given as:

$$F = f \frac{\rho u^2}{d_h 2} \quad (4.1.8)$$

The hydraulic diameter  $d_h$  is given by

$$d_h = \frac{4A}{\pi(D_i + D_o)} \quad (4.1.9)$$

Note that for flow in the drill pipe,  $D_i = 0$ . For annular flow up the wellbore,  $D_i$  is the outer diameter of the drill pipe and  $D_o$  is the wellbore diameter.

#### 4.1.1 Wellbore Friction Factor

The friction factor  $f$  is determined by the method of Colebrook (Fox and MacDonald 1985). In the laminar regime ( $Re < 2100$ )

$$f = \frac{64}{Re} \quad \text{where} \quad Re = \frac{u \rho d_h}{\eta} \quad (4.1.10)$$

In the turbulent regime ( $Re > 2100$ )

$$\frac{1}{\sqrt{f}} = 2.0 \log \left( \frac{\varepsilon/d_h}{3.7} + \frac{2.51}{Re \sqrt{f}} \right) \quad (4.1.11)$$

An approximate solution for Colebrook's  $f$  in the turbulent regime is given by Serghides (1984):

$$f = \left[ a - \frac{(b-a)^2}{c+a-2b} \right]^{-2} \quad (4.1.12)$$

$$a = -2 \log \left[ \frac{\varepsilon/d_h}{3.7} + \frac{12}{\text{Re}} \right] \quad (4.1.13)$$

$$b = -2 \log \left[ \frac{\varepsilon/d_h}{3.7} + \frac{2.51a}{\text{Re}} \right] \quad (4.1.14)$$

$$c = -2 \log \left[ \frac{\varepsilon/d_h}{3.7} + \frac{2.51b}{\text{Re}} \right] \quad (4.1.15)$$

#### 4.1.2 Viscosity of Wellbore Slurry

As the wellbore mixture becomes particle-laden, the viscosity of the mixture is determined from an empirical relationship developed for proppant slurry flows in channels for the oil and gas industry (Barree and Conway 1995). First, a base fluid viscosity  $\eta_0$  is determined by relative volume-weighting of the user-specified viscosities of the mud and gas. Then this viscosity is modified by an approximate slurry formula based on the volume fraction of waste solids.

$$\eta = \eta_0 \left( 1 - \frac{w}{w_{\max}} \right)^s \quad (4.1.16)$$

An important assumption is that the particle diameter of the solids is always less than the channel width. If  $w$  reaches or exceeds its maximum value  $w_{\max}$ , effective viscosity becomes large, and the flow chokes. Both  $w_{\max}$  and  $s$  are empirically determined.

#### 4.1.3 Wellbore Boundary Conditions

At the pump inlet, the source terms  $S_{\text{mass}}$  and  $S_{\text{mom}}$  are set by the volume flow rate of the pump and the density of the mud at the inlet such that

$$S_{\text{mass}} = \rho_{0(m)} R_{\text{mudpump}} \quad (4.1.17)$$

and

$$S_{\text{mom}} = \frac{\rho_{0(m)}}{A_p} R_{\text{mudpump}}^2 \quad (4.1.18)$$

The outlet boundary at the surface uses a specified pressure, usually set to atmospheric pressure. This contributes to setting  $dP/dz$  at the outlet boundary in Equation 4.1.2 and the resulting outlet boundary velocity sets the outlet source terms as

$$S_{mass} = -\rho A_{out} u_{out} \quad (4.1.19)$$

and

$$S_{mom} = -\rho A_{out} u_{out}^2 \quad (4.1.20)$$

Additionally, mass input sources, as specified in Sections 4.2 and 4.4, occur at a point corresponding to the porous flow across the drilling damaged zone prior to penetration and at a point corresponding to the cavity wall after penetration. No momentum is assumed to be contributed by these inputs.

## 4.2 Wellbore-Repository Coupling Equations Prior to Penetration

Nomenclature is given in Table 4-2.

**Table 4-2.** *Nomenclature for Wellbore-Repository Coupling Equations.*

Symbol	Definition	Units
$A_{cav}$	Outer area of pseudo cavity	m <sup>2</sup>
$d_{bit}$	Diameter of drillbit	m
$H_o$	Thickness of repository prior to closure	m
$H$	Current thickness of repository	m
$k_{eff}$	Effective permeability of cylinder	m <sup>2</sup>
$k_{DDZ}$	Permeability of DDZ	m <sup>2</sup>
$k_{DRZ}$	Permeability of DRZ	m <sup>2</sup>
$L$	Total length of cylinder	m
$L_{DDZ}$	Length of cylinder in DDZ	m
$L_{DRZ}$	Length of cylinder in DRZ	m
$m_{cav}$	Mass of gas in pseudo cavity	kg
$m_{cyl}$	Mass of gas flowing from cylinder	kg
$P_{cav}$	Pressure in repository pseudo cavity	Pa
$P_{BH}$	Pressure in bore hole just ahead of bit	Pa
$T$	Temperature	K
$R_0$	Ideal gas constant	J/kg·K
$\eta$	Gas viscosity	kg /m s
$\phi_o$	Porosity of a waste-filled room prior to closure	--
$\phi$	Current repository porosity	--
$\rho_g$	Density of gas	kg/m <sup>3</sup>

Prior to penetration, the cylinder of altered permeable salt connecting the wellbore and the repository has a permeability and gas flow rate determined by the series connection of the salt, DRZ, and DDZ permeability (Gross and Thompson 1998). Since the DDZ permeability is the greatest of these, and the DRZ boundary is fixed while the DDZ boundary advances

with drilling, and  $L$  is decreasing with time, the permeability of this zone increases with time and becomes unbounded as the repository is penetrated. However, since the cavity and borehole pressures converge as the repository is penetrated, the overall mass flow remains bounded. If we start the process at the moment the DRZ or DDZ is contacted (whichever is most above the repository) the mass flow rate thus defined is determined by an incrementally steady-state solution to Darcy's equation for compressible flow (Gross and Thompson 1998)

$$\frac{dm_{cyl}}{dt} = \frac{k_{eff} \pi \left( \frac{d_{bit}}{2} \right)^2}{2\eta R_0 TL} (P_{cav}^2 - P_{BH}^2) \quad (4.2.1)$$

where

$$k_{eff} = \frac{L}{\frac{L_{DDZ}}{k_{DDZ}} + \frac{L_{DRZ}}{k_{DRZ}}} \quad (4.2.2)$$

There is no need to start this process while there is intact salt between the DDZ and repository, as the flow is too small to have noticeable effect. In the numerical analysis, a series of steady-states (one for each timestep) is used with time-varying pressure boundary conditions. The wellbore pressure boundary is the pressure at the point of entry (just below the bit) and the repository pressure is that of the small pseudo cavity at the face of the repository. Although the flow is not truly steady-state, this approach is justified as follows: When the distance  $L$  is large (that is, drilling is at or above the DRZ) the steady state assumption will somewhat overestimate the flow, but the flow itself is so small that the error is small. When the distance  $L$  becomes small, the steady-state assumption becomes more and more accurate, since equilibrium is rapidly achieved. Thus the stepwise steady state approach becomes sufficiently accurate at the time and position of the bit where accuracy is most required. These conclusions are established by inspection of the results of (Gross and Thompson 1998).

Repository thickness  $H$  at the time of intrusion is determined from the current repository porosity  $\phi$ , and the height  $H_o$  and porosity  $\phi_o$  of a waste-filled room prior to closure:

$$H = \frac{(1 - \phi_o)H_o}{1 - \phi} \quad (4.2.3)$$

To couple to the wellbore, Equation 4.2.1 is used as the source term,  $S_{mass}$ , in Equation 4.1.1 for the wellbore, at the leading edge of the bit. To couple to the repository, the pseudo cavity is created below the bit centered at the point where penetration will eventually occur. The geometry of the pseudo cavity is a hemisphere with outer area equal to the bit area in the spherical case, or a cylinder with outer area equal to the bit area in the cylindrical case. The volume of this cavity is thus, for example, in the spherical case



$$V_{cav} = \left( \frac{\pi}{24\sqrt{2}} \right) d_{bit}^3 \quad (4.2.4)$$

and in the cylindrical case

$$V_{cav} = \left( \frac{\pi}{64H} \right) d_{bit}^4 \quad (4.2.5)$$

If the pseudo cavity volume is small compared to physical volumes in the system, it will act to provide compatible boundary conditions prior to penetration and should have no other effect on the solution. For example, once the bit has penetrated the repository by about 1/10 its diameter, the volume actually drilled in the repository exceeds that volume of the pseudo cavity. The pseudo cavity starts at repository pressure and contains gas. Mass flow out of the pseudo cavity is defined by Equation 4.2.1. Mass flow into the cavity is defined by the pressure gradient at the repository cavity wall and other permeable flow parameters

$$\frac{dm_{cav}}{dt} = \rho_{g(repos)} u_{repos} \phi A_{cav} \quad (4.2.6)$$

where the density and velocity are defined in the repository at the cavity wall boundary. Once the net mass flow into/out of the cavity is determined, the pressure in the cavity is determined from the ideal gas law.

### 4.3 Repository Equations

Nomenclature is given in Table 4-3.

**Table 4-3. Nomenclature for Repository Equations.**

Symbol	Definition	Units
$a$	Particle shape factor	--
$C_0$	Cohesion	Pa
$d_p$	Diameter of particles (mean)	m
$F_f$	Fraction of material fluidized	--
$F_0$	Forchheimer number	--
$g$	Acceleration of gravity	m/s <sup>2</sup>
$i, j$	Zone indices	--
$k$	Permeability	m <sup>2</sup>
$k'$	Velocity dependent permeability	m <sup>2</sup>
$k_f$	Permeability in failed material	m <sup>2</sup>
$L_t$	Characteristic length for tensile failure	m

**Table 4-3. Nomenclature for Repository Equations (Continued).**

Symbol	Definition	Units
$m$	Geometry exponent ( $m=3$ for spherical, $m=2$ for cylindrical)	--
$p$	Pressure in gas	Pa
$p_c$	Pressure at cavity face	Pa
$p_{ff}$	Pressure in far-field	Pa
$p_m'$	Mean normal effective stress	Pa
$r$	Radius	m
$r_{cl}$	Radius at center of first intact cell	m
$r_c, r_{cequiv}$	Radius at cavity face	m
$S_0$	Shear strength	Pa
$t$	Time	S
$t_f$	Fluidization time	s
$t_m$	Mixing time from store to cavity	s
$T_0$	Tensile strength	Pa
$U_f$	Fluidization velocity	m/s
$U_s$	Superficial gas velocity	m/s
$\beta$	Biot's constant (near 1.0)	--
$\beta_{nd}$	Non-Darcy coefficient	1/m
$\Delta H$	Current bit penetration	m
$\eta$	Viscosity of gas	kg/m s
$\psi$	Pseudopressure	Pa m <sup>2</sup> /s
$\rho$	Density of gas	kg/m <sup>3</sup>
$\rho_v$	Density of waste solids	kg/m <sup>3</sup>
$\sigma_{ff}$	Stress in far-field	Pa
$\sigma_{sr}, \sigma_{s\theta}$	Radial and tangential seepage stresses	Pa
$\sigma_r', \sigma_{\theta}'$	Radial and tangential effective stresses	Pa
$\tau$	Maximum shear stress	Pa
$\theta$	Angle of internal friction	rad
$\mu$	Friction coefficient	--
$\nu$	Poisson's ratio	--
$\zeta_m, \zeta_p$	Numerical diffusion coefficient for mass and momentum, respectively	--

#### 4.3.1 Non-Darcy Flow

Flow in the repository is transient, compressible, viscous, and single phase for a porous diffusive solid. The gas flow rates near the cavity boundary may be quite high and cause an inertial resistance to flow (Belhaj et al. 2003). (Some early authors used turbulence as an explanation for the effect but most recent authors confirm that the flow is laminar, but tortuous (Hassanizadeh and Gray 1987).) To account for the effect, we use a modification to the Darcy flow equation commonly called the Forchheimer effect, which we express as a

velocity-dependent permeability reduction. A second term in velocity squared is added to the Darcy equation:

$$-\frac{\partial p}{\partial x} = \frac{\eta}{k}u + \beta_{nd}\rho u^2 \quad (4.3.1)$$

This equation can be expressed in the form

$$-\frac{\partial p}{\partial x} = \frac{\eta}{k'}u \quad (4.3.2)$$

where

$$k' = \frac{k}{1 + F_0} \quad \text{and} \quad F_0 = \frac{\beta_{nd}\rho k u}{\eta} \quad (4.3.3)$$

The Forchheimer number  $F_0$  has sometimes been called the Reynold's number for porous flow. The non-Darcy coefficient  $\beta_{nd}$ , which depends on material properties such as the tortuosity and area of internal flow channels, is empirically determined. We choose a study by Li et al. (2001) which used high-velocity nitrogen flow through porous sandstone wafers, giving the result

$$\beta_{nd} = \frac{1.15 \times 10^{-6}}{k\phi} \quad (4.3.4)$$

Other published values and mathematical forms exist and can vary when evaluated by as much as an order of magnitude from Equation 4.3.4. This form was chosen for mathematical simplicity, because it represents neither the high nor the low end of the range, and because it is one of the studies that specifically looked at gas flow in highly porous materials with a large number of measurements. Within DRSPALL, the solution including the Forchheimer effect requires expressing Equation 4.3.2 in radial coordinates and solving the new repository flow equation.

The equation is derived by using Equation 4.3.2, the ideal gas equation of state:

$$p = p_0 \frac{\rho}{\rho_0} \quad (4.3.5)$$

and the conservation of mass:

$$\phi \frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x}(\rho u) = 0 \quad (4.3.6)$$

which gives:

$$\frac{\partial p}{\partial t} = \frac{k'}{2\phi\eta} \nabla^2 p^2 + \frac{1}{2\phi\eta} (\nabla p^2) \nabla(k') \quad (4.3.7)$$

where the operator is given by

$$\nabla^2 = \frac{1}{r^{m-1}} \frac{\partial}{\partial r} \left( r^{m-1} \frac{\partial}{\partial r} \right) \quad (4.3.8)$$

For an isothermal ideal gas, the pseudopressure is defined as

$$\psi = \frac{p^2}{\eta} \quad \text{or} \quad p = \sqrt{\eta\psi} \quad (4.3.9)$$

Combining Equations 4.3.7, 4.3.8, and 4.3.9 leads to

$$\frac{\partial \psi}{\partial t} = \frac{D(\psi)}{r^{m-1}} \frac{\partial}{\partial r} \left( r^{m-1} \frac{\partial \psi}{\partial r} \right) + \frac{D(\psi)}{k'} \frac{\partial k'}{\partial r} \frac{\partial \psi}{\partial r} \quad (4.3.10)$$

where

$$D(\psi) = \frac{k'}{\phi} \sqrt{\frac{\psi}{\eta}} = \frac{k'p}{\phi\eta}$$

Since the equation of state is embedded in the pseudopressure definition as used here, no additional equation relating gas pressure and density is needed.

The singularity as  $r$  approaches zero in Equation 4.3.10 is not a concern since the cavity/repository boundary always occurs at finite  $r$  and the repository equation never has to be solved for  $r$  approaching zero. The boundary conditions at the inner radius of the repository are discussed in Sections 4.2 and 4.4. The condition at the outer boundary is based on no flow with an initial pressure equal to the repository pressure.

### 4.3.2 Elastic Response and Failure

To determine failure and fluidization, solid elastic effective stresses are required. While the fluid calculations using Equation 4.3.10 are fully transient, the solid stress calculations are assumed to be quasi-static. (That is, we ignore sound-speed phenomena in the solid.) The stress equations are thus found as follows.

Elastic effective stresses are (Jaeger and Cook 1969):

$$\sigma_r'(r) = \sigma_{sr}(r) + \sigma_{ff} \left[ 1 - \left( \frac{r_c}{r} \right)^m \right] + p_c \left( \frac{r_c}{r} \right)^m - \beta p(r) \quad (4.3.11)$$

$$\sigma_{\theta}'(r) = \sigma_{s\theta}(r) + \sigma_{ff} \left[ 1 + \frac{1}{(m-1)} \left( \frac{r_c}{r} \right)^m \right] - \frac{p_c}{(m-1)} \left( \frac{r_c}{r} \right)^m - \beta p(r) \quad (4.3.12)$$

where the flow-related stresses are given by (in analogy to differential thermal expansion, Timoshenko and Goodier 1970):

$$\sigma_{sr}(r) = (m-1) \beta \left( \frac{1-2\nu}{1-\nu} \right) \frac{1}{r^m} \int_{r_c}^r [p(r) - p_{ff}] r^{m-1} dr \quad (4.3.13)$$

$$\sigma_{s\theta}(r) = -\beta \left( \frac{1-2\nu}{1-\nu} \right) \left( \frac{1}{r^m} \int_{r_c}^r [p(r) - p_{ff}] r^{m-1} dr - [p(r) - p_{ff}] \right) \quad (4.3.14)$$

Since stresses are calculated as quasi-static, an initial stress reduction caused by an instantaneous pressure drop at the cavity face propagates instantaneously through the waste. The result of calculating Equation 4.3.11 can be an instantaneous early-time tensile failure of the entire repository if the boundary pressure is allowed to change suddenly. This is non-physical and merely a result of the quasi-static stress assumption combined with the true transient pore pressure and flow-related stress equations. To prevent this behavior, tensile failure is not allowed until the propagation distance determined by a tensile propagation limit (“tensile failure velocity”, Hansen et al. 1997), beginning at the cavity face, occurs. This should have no quantitative effect on results other than to prevent a non-physical tensile failure occurring at initial wellbore penetration. It is possible that allowing pre-penetration flow through the DDZ as described in Section 4.2 renders the introduction of this procedure unnecessary.

For tensile failure, the radial effective stress is averaged from the cavity boundary into the waste over a characteristic length  $L_t$ . If this average stress is tensile and exceeds the material tensile strength,  $T_0$ , the waste is assumed to be no longer capable of supporting radial stress, and the material from the cavity out to the characteristic length  $L_t$  is marked as failed. After failure, fluidization is allowed in the failed material. The cavity boundary does not move outward until after the failed material fluidizes, as discussed in Sections 4.3.3 and 4.3.5.

Potential shear failure locations are determined (flagged), but no relaxation associated with shear failure is used. For shear failure, the maximum shear stress

$$\tau(r) = \frac{1}{2} |(\sigma_r'(r) - \sigma_{\theta}'(r))| \quad (4.3.15)$$

and mean normal effective stress

$$p_m'(r) = \frac{[\sigma_r'(r) + (m-1)\sigma_{\theta}'(r)]}{m} \quad (4.3.16)$$

are used to determine if maximum shear stress exceeds the linear Mohr-Coulomb shear strength

$$\tau(r) > S_0 + \mu p_m'(r) \quad (4.3.17)$$

where

$$S_0 = \frac{C_0}{2} \left[ \mu + (\mu^2 + 1)^{\frac{1}{2}} \right] \quad (4.3.18)$$

$$\mu = \tan(\theta) \quad (4.3.19)$$

and  $\theta$  is the angle of internal friction. Again, singularities as  $r$  approaches zero are avoided because of the always-finite cavity. The solid stress boundary conditions are, for the inner boundary, no shear stress and radial stress equal to cavity pressure. For the outer boundary, the far field confining (radial) stress is set as an input parameter. There is no far-field shear stress.

### 4.3.3 Fluidization

In a region where tensile failure has occurred, the waste material is first assumed to be disaggregated, but not in motion. It remains as porous, bedded material lining the cavity face, and is treated as a continuous part of the repository from the perspective of the porous flow calculations. The bedded material is subject to mobilization and removal to the flow stream in the cavity by fluidized bed theory. As such, solids are allowed to flow into the cavity if the superficial gas velocity,  $U_s$ , at the cavity face exceeds a minimum fluidization velocity,  $U_f$ . Superficial gas velocity is defined as the average velocity across the entire surface area of the cavity ( $U_s = \text{pore velocity} \times \phi$ ). The cavity radius will increase corresponding to the volume of solid material removed by fluidization. The minimum fluidization velocity is determined by solving the following quadratic equation (Cherimisinoff and Cherimisinoff 1984; Ergun 1952).

$$\frac{1.75}{a\phi^3} \left( \frac{d_p U_f \rho}{\eta} \right)^2 + 150 \left( \frac{1-\phi}{a^2 \phi^3} \right) \left( \frac{d_p U_f \rho}{\eta} \right) = \frac{d_p^3 \rho (\rho_w - \rho) g}{\eta^2} \quad (4.3.20)$$

If fluidization occurs, the cavity is allowed to grow until the cavity radius equals the depth of the first intact (non tensile-failed) material. New tensile failure is allowed from the boundary into the intact repository material. Once the entire region of length  $L_t$  nearest boundary fluidizes, a new region of length  $L_t$  is defined and evaluated for failure as discussed above. With cavity growth, the gas and waste particles in the newly fluidized region must mix into the cavity in such a way as to conserve mass. To account for the fact that this mixing cannot be instantaneous, which would be non-physical and lead to numerical instability later (much as allowing instantaneous tensile failure propagation), a small artificial relaxation time, equal to the cavity radius divided by the superficial gas velocity, is imposed upon the mixing

phenomenon. As fluidization occurs, the permeability of the failed waste in that region is allowed to grow according to

$$k'_f = k'(1 + 4F_f) \quad (4.3.21)$$

where  $F_f$  is the fraction fluidized and is based on the fluidization relaxation time. This approximately accounts for the bulking of material as it fluidizes. It is interesting to note that the Ergun equation, which leads to Equation 4.3.20, can also be used to derive a non-Darcy coefficient for particulate beds (Narayanaswamy et al. 1999). This takes a similar form to Equation 4.3.8 (but not the same value). This suggests, as would be expected, that there is continuous transition from intact to failed to fluidized material. However, the level of accuracy inherent in the present work would not benefit from further investigation of this transition.

#### 4.3.4 Repository Boundary Conditions

The porous flow equations are solved with a pressure (flow) boundary condition at the inner (cavity) wall and a zero pressure gradient (no-flow) condition at the outer wall. For the inner wall, the cavity pressure (either pseudo or real) sets a pressure outside the porous solid. The inner cavity wall propagates radially outward as material is drilled or failed and fluidized and moved from the repository to the cavity. Thus, the boundary condition is subsequently applied to the adjacent non-fluidized but possibly failed material. The pressure inside the porous solid then develops a gradient, based on the pressure difference, which is used to calculate a pore velocity at the boundary using Darcy's Law:

$$u_{repos} = \frac{k'}{\eta\phi} \frac{\partial p}{\partial r} \Big|_{r=r_c} \quad (4.3.22)$$

Material (gas) loss across the boundary as a function of time can then be calculated using Equation 4.2.6.

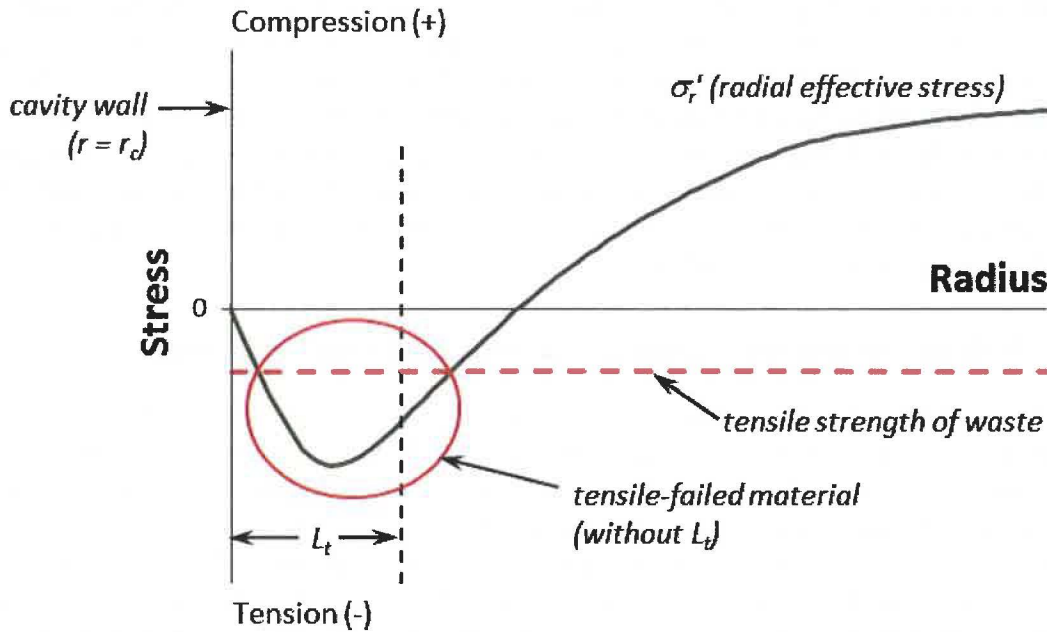
#### 4.3.5 Implementation of Conceptual Submodels in DRSPALL

Some of the features of the conceptual and mathematical model discussed earlier require special implementation procedures in a finite difference code. The purpose of this section is to elucidate the implementation steps that are not necessarily clear in a discussion of the conceptual model alone.

##### 4.3.5.1 Characteristic Length and Tensile Failure

A review of the effective stress formulation (Equation 4.3.11) reveals several issues that require careful consideration when applying this model to a repository domain divided into discrete computational zones. A typical radial effective stress curve is drawn in Figure 4-1. The tensile strength of the waste is shown in the figure as well. Effective radial tensile stresses (negative values on the stress axis) are shown to develop near to, but not exactly at the cavity wall. In fact, the effective stress exactly at the cavity wall is always  $\sigma_r'(r_c) = 0$ .

As such, a region of tension develops that may exceed the tensile strength at radii greater than, but not at, the cavity wall. The region of material subject to tensile failure, without consideration of the characteristic length concept, is circled in Figure 4-1. Note that material between the failed region and the cavity wall is still intact according to the current stress model.



**Figure 4-1.** Drawing of typical radial effective stress curve in DRSPALL repository domain at some time during drill penetration.

Failed regions trapped between intact shells cannot, by themselves, be removed from the repository domain without first removing the intact cavity wall. It is proposed here that once a tensile-failed region exists, the cavity wall is likely to eventually succumb to buckling failure. A simple approach to this is implemented in DRSPALL by introducing a characteristic failure length  $L_t$  in which the mean tensile stress is evaluated over a region that extends from the cavity wall ( $r = r_c$ ) to a depth of one characteristic length ( $r = r_c + L_t$ ) into the domain. This characteristic length must be large enough to encompass several computational zones, but not so large that it masks the tensile region. If the mean tensile stress over  $L_t$  exceeds the tensile strength, all the zones in  $L_t$  fail, and they are flagged as “failed” in DRSPALL. Zones outside of  $L_t$  are not allowed to fail until all the disaggregated, bedded waste is fluidized.

#### 4.3.5.2 Cavity Growth by Drilling

During normal drilling, when the bit penetration depth passes the outer edge of a computational zone, that zone (gas and solids) is removed from the repository domain and put into “store.” From store, the drilled mass is then released to the cavity over a mixing time  $t_m = (r_{cl}/U_s)$  where  $r_{cl}$  is the center of the cell that forms the cavity wall ( $r_{cl} = r_c + 1/2\Delta r$ ). This is done because instantaneously adding the entire contents of one computational



zone to the cavity introduces high frequency spikes that can lead to an increase in numerical error. The controlled release from store dampens the numerical shock to the system.

#### 4.3.5.3 Cavity Growth by Fluidization

In the event that a group of zones comprising  $L_t$  fails, the zones are all tested for fluidization potential using the Ergun (1952) model given in Equation 4.3.20. If the superficial gas velocity exceeds the minimum fluidization velocity ( $U_s > U_f$ ), all failed zones are flagged for fluidization. The fluidization process requires a finite fluidization time  $t_f = r_{ci}/U_s$ . Zones must begin fluidization sequentially so that the zone comprising the cavity wall starts first, after which the next zone can start fluidization on the very next timestep. This proceeds until all zones flagged for fluidization are removed from the repository domain and thus incorporated into the wellbore domain. The cavity thus grows in zone-by-zone increments as subsequent zones complete fluidization.

### 4.4 Wellbore-Repository Coupling Equations Following Penetration

Upon penetration by the drillbit, the gas flow Equation 4.2.1 at the drillbit/cavity inner boundary is no longer used. Instead, the cavity volume is assumed to be included in the wellbore flow path as shown in Figure 3-5. If the cavity subsequently grows due to drilling, tensile failure and fluidization, the volume of this part of the flow path increases to be consistent with the growing cavity volume. The equivalent physical assumption is that mud flow passes through the bit nozzles, mixes fully with the gas and waste in the cavity and then the resulting mixture flows up the annulus around the drill collars. Thus no new equations are needed, other than to be sure that mass balance is maintained and that cavity volume is correct. Equation 4.2.6 is still used to determine gas mass flow into the cavity from the repository. Waste flows into the cavity by simply mixing in fluidized waste as discussed in Section 4.3. The mass input provided by gas and waste flowing into the cavity replace the gas mass source provided by the DDZ prior to penetration, thus mathematical consistency and continuity are maintained throughout the drilling and gas/waste flow process.

The true cylindrical cavity of increasing height formed during the drilling process must be mapped each computational step to an equivalent cavity radius in the specified one-dimensional spherical or cylindrical geometry. This mapping assumes conservation of surface area in order to preserve the early time repository pressure gradients and thus gas flux from the repository to the wellbore. Note that the drilled volume will not be conserved.

The surface area of the true cylinder cut from the waste takes the shape of a soup can with one end removed. The surface area of this cut cylinder is therefore:

$$A_{cut} = \frac{\pi d_b^2}{4} + \pi d_b \Delta H \quad (4.4.1)$$

In spherical geometry the equivalent radius is determined by equating  $A_{cut}$  with the surface area of a hemisphere ( $A = 2\pi r_{eq}^2$ ). Solving for  $r_{eq}$  gives:

$$r_{eq} = \left( \frac{A_{cut}}{2\pi} \right)^{\frac{1}{2}} \quad (4.4.2)$$

In cylindrical geometry the equivalent radius is determined by equating  $A_{cut}$  with the surface of a cylinder with fixed height  $H$  and open ends ( $A = \pi d_b H$ ). Solving for  $r_{eq}$  in this case gives:

$$r_{eq} = \left( \frac{A_{cut}}{2\pi H} \right) \quad (4.4.3)$$

For small penetration depths this implies a very small diameter equivalent cylinder with a height equal to the repository thickness.

Equivalent volumes are calculated from the equivalent radius using the volume equation appropriate for the assumed geometry. When the bit reaches the floor of the repository, drilling is stopped and in cylindrical geometry the actual drilled and equivalent volumes will be the same. But, in spherical geometry the equivalent volume will be greater than the drilled volume.

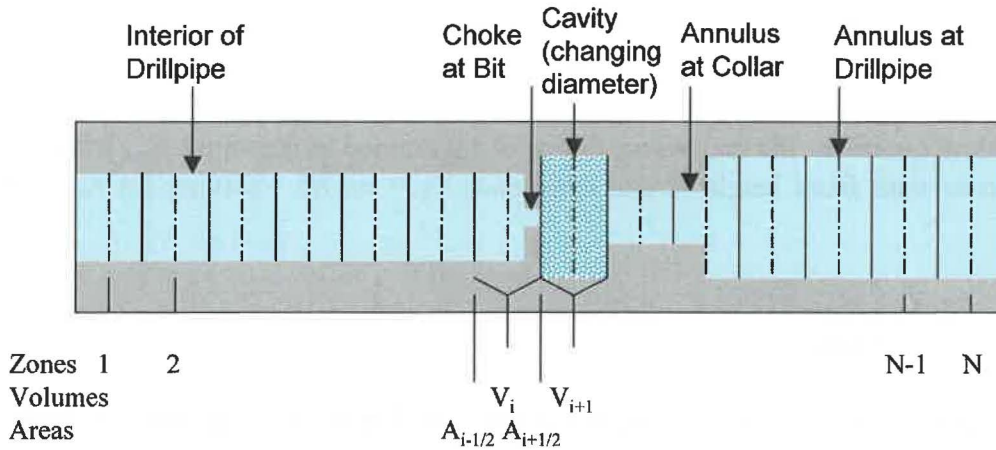
#### 4.5 Numerical Method – Wellbore

Both the wellbore and the repository calculations use time-marching finite differences. These are part of a single computational loop and therefore use the same time step. The differencing schemes for the wellbore and repository calculations are similar but not identical.

The wellbore is zoned for finite differencing as shown in Figure 4-2. This shows zones, zone indices, grid boundaries, volumes, and interface areas. The method is Eulerian. That is, zone boundaries are fixed, and fluid moves through the interfaces by convection. Quantities are zone-centered and integration is explicit in time. In the following, detailed discussion will be specific to the mass transport given by Equation 4.1.1. However, the differencing method for the momentum transport given by Equation 4.1.2 is the same.

The equations shown below are for constant zone size. All of the materials (mud, salt, gas, and waste) are assumed to move together as a mixture. Since fluid moves through the grid boundaries, calculation requires values for the flow through these boundaries during a timestep. This is obtained from difference between the fluid velocities at the zone centers, given by

$$u_{i+1/2} = \frac{1}{2} (u_{i+1}^{n-1} + u_i^{n-1}) \quad (4.5.1)$$



**Figure 4-2.** Finite difference zoning for wellbore.

The mass transport equation, prior to any volume change, becomes

$$V_i \rho_i^* = V_i \rho_i^{n-1} - \Delta t \left( A_{i+1/2} \rho_{i+1/2}^{n-1} u_{i+1/2} - A_{i-1/2} \rho_{i-1/2}^{n-1} u_{i-1/2} \right) + \Delta t S_{m,i} \quad (4.5.2)$$

Here the source terms ( $S_{m,i}$ ) are set to correspond to material entering or exiting at the pump, cavity, and surface. The “upwind” zone centered densities are used for the interfaces values,  $\rho_{i+1/2}^{n-1}$  and  $\rho_{i-1/2}^{n-1}$ . Finally we incorporate any changed volumes and add numerical mass diffusion for stability:

$$V_i \rho_i^n = V_i \rho_i^* + \Delta z_i \sum_{m=1}^{N_c} \zeta_m \left[ A_{i+1/2} \left( (\rho f_m)_{i+1}^{n-1} - (\rho f_m)_i^{n-1} \right) - A_{i-1/2} \left( (\rho f_m)_i^{n-1} - (\rho f_m)_{i-1}^{n-1} \right) \right] \quad (4.5.3)$$

The densities,  $\rho f_m$ , are for the particular constituent,  $m$ , being diffused and are calculated from the mixture density,  $\rho$ , times the mass fraction,  $f_m$ , of the constituent in the referenced cell. The numerical diffusion coefficient  $\zeta_m$  is chosen empirically for stability. Separate diffusion coefficients could be used for the different materials (mud, gas, etc.). However, sufficient stability is obtained by only diffusing mud and salt using the same coefficient. This better preserves the accuracy of the location of gases and waste in the well.

Momentum is differenced as

$$V_i (\rho u)_i^* = V_i (\rho u)_i^{n-1} - \Delta t \left( A_{i+1/2} (\rho u)_{i+1/2}^{n-1} u_{i+1/2} - A_{i-1/2} (\rho u)_{i-1/2}^{n-1} u_{i-1/2} \right) - V_i \left( \frac{P_{i+1}^{n-1} - P_{i-1}^{n-1}}{2\Delta z} - \rho_i^{n-1} g + F_i^{n-1} \right) + \Delta t S_{mom,i} \quad (4.5.4)$$

where the dissipation term  $F_i^{n-1}$  is obtained from Equation 4.1.8 and is constrained as follows:

$$\left| F_i^{n-1} \right| \leq \left| \frac{P_{i+1}^{n-1} - P_{i-1}^{n-1}}{2\Delta z} - \rho_i^{n-1} g \right| \quad (4.5.5)$$

and the sign  $F_i^{n-1}$  is chosen such that it opposes flow.

Finally, numerical momentum diffusion is added without distinguishing between phases in the mixture ( $\rho$  is the mixture density).

$$V_i (\rho u)_i^n = V_i (\rho u)_i^* - \zeta_p \Delta z_i \left[ A_{i+1/2} \left( (\rho u)_{i+1}^{n-1} - (\rho u)_i^{n-1} \right) - A_{i-1/2} \left( (\rho u)_i^{n-1} - (\rho u)_{i-1}^{n-1} \right) \right] \quad (4.5.6)$$

#### 4.6 Numerical Method – Repository

The time integration method for the repository flow is implicit, with spatial derivatives determined after the time increment. This method requires the inversion of a matrix for the entire repository, which is usually straightforward. The implicit scheme is unconditionally stable. However, it is still necessary to use small time steps to ensure gradient accuracy.

We start with Equation 4.3.10, and note that

$$\frac{\partial \ln k'}{\partial r} = \frac{1}{k'} \frac{\partial k'}{\partial r} \quad (4.6.1)$$

Substituting Equation 4.6.1 into Equation 4.3.10 yields:

$$\frac{\partial \psi}{\partial t} = \frac{D(\psi)}{r^{m-1}} \frac{\partial}{\partial r} \left( r^{m-1} \frac{\partial \psi}{\partial r} \right) + D(\psi) \frac{\partial \ln(k')}{\partial r} \frac{\partial \psi}{\partial r} \quad (4.6.2)$$

Expand first term on right-hand side of Equation 4.6.2:

$$\begin{aligned} \frac{D(\psi)}{r^{m-1}} \frac{\partial}{\partial r} \left( r^{m-1} \frac{\partial \psi}{\partial r} \right) &= D(\psi) \left[ \frac{1}{r^{m-1}} \frac{\partial}{\partial r} \left( r^{m-1} \frac{\partial \psi}{\partial r} \right) \right] \\ &= D(\psi) \left[ \frac{1}{r^{m-1}} r^{m-1} \frac{\partial}{\partial r} \left( \frac{\partial \psi}{\partial r} \right) + \frac{\partial \psi}{\partial r} \frac{1}{r^{m-1}} \frac{\partial (r^{m-1})}{\partial r} \right] \\ &= D(\psi) \left[ \frac{\partial^2 \psi}{\partial r^2} + \frac{\partial \psi}{\partial r} \frac{1}{r^{m-1}} (m-1) (r^{m-2}) \frac{\partial r}{\partial r} \right] \\ &= D(\psi) \left[ \frac{\partial^2 \psi}{\partial r^2} + \frac{(m-1)}{r} \frac{\partial \psi}{\partial r} \right] \end{aligned} \quad (4.6.3)$$

Substituting Equation 4.6.3 into Equation 4.6.2 gives:

$$\frac{\partial \psi}{\partial t} = D(\psi) \left[ \frac{\partial^2 \psi}{\partial r^2} + \frac{(m-1)}{r} \frac{\partial \psi}{\partial r} + \frac{\partial \ln(k')}{\partial r} \frac{\partial \psi}{\partial r} \right] \quad (4.6.4)$$

Equation 4.6.4 is nonlinear due to the dependence of the parameter  $D$  on the state variable  $\psi$ . Hence, its numerical solution requires use of an iterative scheme such as the Newton-Raphson method. However, as explained in the next section,  $D$  is treated as independent of  $\psi$  in DRSPALL calculations, so an iterative Newton-Raphson scheme is not necessary here.

#### 4.6.1 Finite Difference Discretization

Using an implicit scheme, Equation 4.6.4 can be represented in finite difference form by using the central difference method to discretize the right-hand side and the forward difference method to discretize the left-hand side (Özişik 1993, Chapter 12, “Implicit Method”).

This gives:

$$\frac{\psi_j^{n+1} - \psi_j^n}{\Delta t} = D(\psi_j^{n+1}) \left[ \frac{\psi_{j+1}^{n+1} - 2\psi_j^{n+1} + \psi_{j-1}^{n+1}}{(\Delta r)^2} + \frac{(m-1)(\psi_{j+1}^{n+1} - \psi_{j-1}^{n+1})}{r_j 2\Delta r} + \frac{(\ln(k'_{j+1}^{n+1}) - \ln(k'_{j-1}^{n+1}))(\psi_{j+1}^{n+1} - \psi_{j-1}^{n+1})}{2\Delta r 2\Delta r} \right] \quad (4.6.5)$$

As discussed in Lord et al. (2006),  $D(\psi)$  is assumed constant over a zone, which simplifies the numerical implementation. Using its zone centered value at the current time, the linearizing approximation  $D(\psi) \approx D(\psi_j^{n+1}) \approx D(\psi_j^n) \approx D_j^n$  is made. Equation 4.6.5 becomes.

$$\frac{\psi_j^{n+1} - \psi_j^n}{\Delta t} = D_j^n \left[ \frac{\psi_{j+1}^{n+1} - 2\psi_j^{n+1} + \psi_{j-1}^{n+1}}{(\Delta r)^2} + \frac{(\psi_{j+1}^{n+1} - \psi_{j-1}^{n+1})}{2\Delta r} \left( \frac{(m-1)}{r_j} + \frac{(\ln(k'_{j+1}^{n+1}) - \ln(k'_{j-1}^{n+1}))}{2\Delta r} \right) \right]$$

Solve for  $\psi_j^n$  and collect similar terms

$$\begin{aligned}
\psi_j^n &= \psi_j^{n+1} - D_j^n \Delta t \left[ \frac{\psi_{j+1}^{n+1} - 2\psi_j^{n+1} + \psi_{j-1}^{n+1}}{(\Delta r)^2} + \frac{(\psi_{j+1}^{n+1} - \psi_{j-1}^{n+1})}{2\Delta r} \left( \frac{(m-1)}{r_j} + \frac{(\ln(k_{j+1}^{n+1}) - \ln(k_{j-1}^{n+1}))}{2\Delta r} \right) \right] \\
&= -D_j^n \Delta t \left[ \frac{1}{(\Delta r)^2} - \frac{1}{2\Delta r} \left( \frac{(m-1)}{r_j} + \frac{(\ln(k_{j+1}^{n+1}) - \ln(k_{j-1}^{n+1}))}{2\Delta r} \right) \right] \psi_{j-1}^{n+1} + \left[ 1 + \frac{2D_j^n \Delta t}{(\Delta r)^2} \right] \psi_j^{n+1} \\
&\quad - D_j^n \Delta t \left[ \frac{1}{(\Delta r)^2} + \frac{1}{2\Delta r} \left( \frac{(m-1)}{r_j} + \frac{(\ln(k_{j+1}^{n+1}) - \ln(k_{j-1}^{n+1}))}{2\Delta r} \right) \right] \psi_{j+1}^{n+1} \\
&= -\frac{D_j^n \Delta t}{\Delta r} \left[ \frac{1}{\Delta r} - \frac{(m-1)}{2r_j} - \frac{(\ln(k_{j+1}^{n+1}) - \ln(k_{j-1}^{n+1}))}{4\Delta r} \right] \psi_{j-1}^{n+1} + \left[ 1 + \frac{2D_j^n \Delta t}{(\Delta r)^2} \right] \psi_j^{n+1} \\
&\quad - \frac{D_j^n \Delta t}{\Delta r} \left[ \frac{1}{\Delta r} + \frac{(m-1)}{2r_j} + \frac{(\ln(k_{j+1}^{n+1}) - \ln(k_{j-1}^{n+1}))}{4\Delta r} \right] \psi_{j+1}^{n+1}
\end{aligned}$$

Noting that  $\ln(k_{j+1}^{n+1}) - \ln(k_{j-1}^{n+1}) = \ln\left(\frac{k_{j+1}^{n+1}}{k_{j-1}^{n+1}}\right)$

$$\begin{aligned}
\psi_j^n &= -\frac{D_j^n \Delta t}{\Delta r} \left( \frac{1}{\Delta r} - \frac{(m-1)}{2r_j} - \frac{\ln\left(\frac{k_{j+1}^{n+1}}{k_{j-1}^{n+1}}\right)}{4\Delta r} \right) \psi_{j-1}^{n+1} + \left( 1 + \frac{2D_j^n \Delta t}{(\Delta r)^2} \right) \psi_j^{n+1} \\
&\quad - \frac{D_j^n \Delta t}{\Delta r} \left( \frac{1}{\Delta r} + \frac{(m-1)}{2r_j} + \frac{\ln\left(\frac{k_{j+1}^{n+1}}{k_{j-1}^{n+1}}\right)}{4\Delta r} \right) \psi_{j+1}^{n+1}
\end{aligned}$$

Rewriting the equation:

$$\psi_j^n = -\alpha_1 \psi_{j-1}^{n+1} + (1 + 2\alpha) \psi_j^{n+1} - \alpha_2 \psi_{j+1}^{n+1} \quad (4.6.6)$$

where

$$\alpha = \frac{D_j^n \Delta t}{(\Delta r)^2}$$

$$\alpha_1 = \frac{D_j^n \Delta t}{\Delta r} \left( \frac{1}{\Delta r} - \frac{(m-1)}{2r_j} - \frac{\ln\left(\frac{k_{j+1}^{n+1}}{k_{j-1}^{n+1}}\right)}{4\Delta r} \right)$$

$$\alpha_2 = \frac{D_j^n \Delta t}{\Delta r} \left( \frac{1}{\Delta r} + \frac{(m-1)}{2r_j} + \frac{\ln\left(\frac{k_{j+1}^{n+1}}{k_{j-1}^{n+1}}\right)}{4\Delta r} \right)$$

and  $j$  and  $n$  are the cell and timestep indices, respectively.

Also:

$$\alpha_1 + \alpha_2 = \frac{D_j^n \Delta t}{\Delta r} \left( \frac{1}{\Delta r} - \frac{(m-1)}{2r_j} - \frac{\ln\left(\frac{k_{j+1}^{n+1}}{k_{j-1}^{n+1}}\right)}{4\Delta r} \right) + \frac{D_j^n \Delta t}{\Delta r} \left( \frac{1}{\Delta r} + \frac{(m-1)}{2r_j} + \frac{\ln\left(\frac{k_{j+1}^{n+1}}{k_{j-1}^{n+1}}\right)}{4\Delta r} \right)$$

$$= \frac{D_j^n \Delta t}{\Delta r} \left( \frac{1}{\Delta r} + \frac{1}{\Delta r} \right)$$

$$= 2 \frac{D_j^n \Delta t}{(\Delta r)^2}$$

$$\alpha_1 + \alpha_2 = 2\alpha$$

#### 4.6.2 Boundary Conditions

For the boundary condition at the inner radius we use the method given in Thomas (1995, Section 1.6.3, "Cell Centered Grids") for cell centered grids. Using this method the difference equation is derived on the second cell in the usual, here central difference, manner. The first intact zone is the zone closest to the borehole and is indexed as 1. The boundary condition is implemented by noting that for the first intact cell ( $j - 1 = 1$ ),  $\psi_1^{n+1}$  is the cavity pseudo pressure,  $\psi_{cav}^{n+1}$ , which is known. Therefore,  $\psi_1^{n+1} = \psi_{cav}^{n+1}$  can be moved to the left-hand side of Equation 4.6.6.

Using  $j = 2$ , the second cell, Equation 4.6.6 gives:

$$\psi_2^n + \alpha_1 \psi_{cav}^{n+1} = (1 + 2\alpha) \psi_2^{n+1} - \alpha_2 \psi_3^{n+1} \quad (4.6.7)$$

The far field boundary condition is a zero gradient condition. Mathematically this is:

$$\left. \frac{\partial \psi}{\partial r} \right|_{r=\infty} = 0 \quad (4.6.8)$$

Discretizing using a second-order central difference formulation:

$$\begin{aligned} \frac{\psi_{N+1}^{n+1} - \psi_{N-1}^{n+1}}{2\Delta r} &= 0 \\ \psi_{N+1}^{n+1} &= \psi_{N-1}^{n+1} \end{aligned} \quad (4.6.9)$$

Using this to eliminate the fictitious point in the domain discretization of Equation 4.6.6, at node point  $j = N$  (i.e., the final cell):

$$\begin{aligned} \psi_N^n &= -\alpha_1 \psi_{N-1}^{n+1} + (1+2\alpha) \psi_N^{n+1} - \alpha_2 \psi_{N+1}^{n+1} \\ &= -\alpha_1 \psi_{N-1}^{n+1} + (1+2\alpha) \psi_N^{n+1} - \alpha_2 \psi_{N-1}^{n+1} \\ &= -(\alpha_1 + \alpha_2) \psi_{N-1}^{n+1} + (1+2\alpha) \psi_N^{n+1} \end{aligned}$$

Using  $\alpha_1 + \alpha_2 = 2\alpha$ , the final condition can be simplified to:

$$\psi_N^n = -(2\alpha) \psi_{N-1}^{n+1} + (1+2\alpha) \psi_N^{n+1} \quad (4.6.10)$$

In the previous version of the DRSPALL design document (WIPP 2004, which was prepared for DRSPALL Version 1.10), a forward difference formulation was applied to the boundary condition Equation 4.6.8. Özişik (1993) shows that the error involved with the central difference representation is second-order accurate, i.e.,  $O((\Delta r)^2)$ ; whereas the error involved with the forward difference representation is first-order accurate, i.e.,  $O(\Delta r)$ . Therefore, the central difference formulation, Equation 4.6.10, is used because it decreases numerical discretization errors and provides a more accurate numerical approximation to the exact solution.

#### 4.6.3 Summary of Recommended Finite Difference Equations

The recommended tri-diagonal linear system of equations is:

$$\begin{aligned} \psi_j^n &= -\alpha_1 \psi_{cav}^{n+1} + (1+2\alpha) \psi_j^{n+1} - \alpha_2 \psi_{j+1}^{n+1} & \text{for } j = 2 \\ \psi_j^n &= -\alpha_1 \psi_{j-1}^{n+1} + (1+2\alpha) \psi_j^{n+1} - \alpha_2 \psi_{j+1}^{n+1} & \text{for } j = 3, \dots, N-1 \\ \psi_j^n &= -(2\alpha) \psi_{j-1}^{n+1} + (1+2\alpha) \psi_j^{n+1} & \text{for } j = N \end{aligned} \quad (4.6.11)$$

where

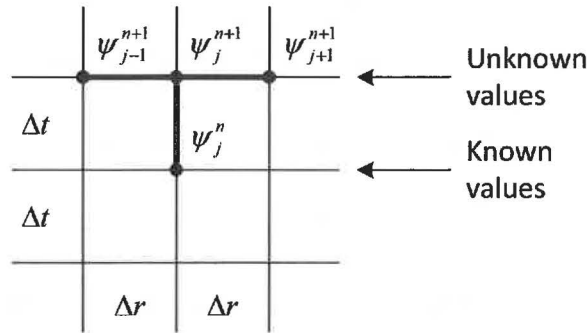


$$\alpha = \frac{D_j^n \Delta t}{(\Delta r)^2}$$

$$\alpha_1 = \frac{D_j^n \Delta t}{\Delta r} \left( \frac{1}{\Delta r} - \frac{(m-1)}{2r_j} - \frac{\ln\left(\frac{k_{j+1}^{n+1}}{k_{j-1}^{n+1}}\right)}{4\Delta r} \right)$$

$$\alpha_2 = \frac{D_j^n \Delta t}{\Delta r} \left( \frac{1}{\Delta r} + \frac{(m-1)}{2r_j} + \frac{\ln\left(\frac{k_{j+1}^{n+1}}{k_{j-1}^{n+1}}\right)}{4\Delta r} \right)$$

Figure 4-3 illustrates the expansion point (j, n+1) and the surrounding finite-difference molecules.



**Figure 4-3.** Finite difference molecules for the implicit scheme using constant zones sizes.

#### 4.6.4 Repository Stress State

Once pore pressure is known, evaluation of the repository stress state, Equations 4.3.11 and 4.3.12, is algebraic except for the integral in the seepage stress calculations in Equations 4.3.13 and 4.3.14. The integrals are evaluated using trapezoidal integration with trapezoids formed between the cell centers where pore pressure and stress components are evaluated. The first increment is over the half-cell between the cavity wall and the center of the first non-fluidized cell. Radii are evaluated at cell centers, except for the left side in the first increment, which is the cavity wall radius:

$$\int_{r_c}^{r_j} [p(r_j) - p_{ff}] r_j^{m-1} dr = (1 - F_f(j)) \sum_{i=0}^j 0.5((p_{i-1} - p_n) r_{i-1}^{m-1} + (p_i - p_n) r_i^{m-1})(r_i - r_{i-1}) \quad (4.6.12)$$

where  $i=1$  is the first intact cell (non-fluidized);  $j$  is index of the cell for which stress components are being evaluated;  $r_j$  is radius to the center of cell  $j$ ;  $p_0$  is the cavity pressure;  $r_0$

is the radius of cavity cell next to the interface;  $F_f(j)$  is the fraction of cell  $j$  that has fluidized and is based on the fluidization relaxation time; and  $n$  is the index of the last computational cell. The fraction fluidized term is used to relax the seepage stresses in cells that have failed and are in the process of fluidizing.

#### 4.7 Numerical Method – Coupling

The calculation of gas flux from the repository to the wellbore (Equation 4.2.6) requires the gas velocity at the cavity interface,  $u_{repos}$  (Equation 4.3.22). The velocity is evaluated numerically as follows:

$$u_{repos} = \frac{k_1'}{\eta\phi} \left( \frac{P_1 - P_{cav}}{\Delta r_1} \right) \quad (4.7.1)$$

The mass of gas flowing from the repository to the cavity at the bottom of the wellbore is then evaluated from:

$$\Delta m_{cav} = \rho_{g(repos)} u_{repos} \phi A_{cav} \Delta t \quad (4.7.2)$$

#### 4.8 Time Step Determination

For the wellbore, which uses a conditionally stable explicit method, the Courant condition

$$ss_m \frac{\Delta t}{\Delta z} < constant \quad (4.8.1)$$

is the basis of time step determination, where  $\Delta z$  is the zone size and  $ss_m$  is the sound speed in the mud and  $\rho_m$  is the mud density:

$$ss_m = \sqrt{\frac{1}{c_m \rho_m}} \quad (4.8.2)$$

The wellbore timestep is calculated by replacing the inequality in Equation 4.8.1 with equality, solving for  $\Delta t$  using a user specified *constant* and finding the minimum over all computational cells

$$\Delta t = \min \left( constant \frac{\Delta z_i}{ss_{m_i}} \right) \quad (4.8.3)$$

Normally, the value of the *constant* is taken as 1.0 for the “standard” Courant condition. However, due to the addition of the numerical diffusion terms and the extreme conditions in the present problem, especially where mass feeds from the repository into the wellbore, *constant* might be much less than 1 for a stable solution and is found by trial and error. A

value that produces stable results (free of cell-to-cell oscillations) will typically work across a range of model parameter values.

For the repository, which uses an unconditionally stable implicit method, a constant timestep would typically be specified by the user. DRSPALL actually implements the timestep selection as a user specified factor,  $F_{rep}$ , that is multiplied by the Courant criteria that would normally be used by an explicit algorithm.

$$\Delta t = \min \left( F_{rep} \frac{\Delta r_i^2}{\frac{k'_i}{\phi} \left( \frac{\psi_i}{\mu_i} \right)^{1/2}} \right) \quad (4.8.4)$$

Where  $\Delta t$  is the minimum over all cells in the repository domain. Trial and error or convergence testing can be used to find a timestep factor (usually greater than 1) that gives suitable results and gradient definitions.

The final timestep used is the lesser of the wellbore-determined and repository-determined timestep.

## 5 FLOW CHART WITH FUNCTIONAL REQUIREMENTS MAP

The major flow and control logic for DRSPALL is presented in Figure 5-1 in the form of a flow chart. In the flow chart subroutine names are underlined and descriptive information is in ‘Sentence case’. Also shown in the figure is a mapping of Functional Requirements. The requirement numbers (i.e., R.1) corresponds to those presented in the *Requirements Document for DRSPALL Version 1.00* (WIPP PA 2003) and summarized in Appendix A. Figure 5-1 shows the overall flow logic of DRSPALL and roughly corresponds to the DRSPALL main routine. Subsequent figures show details of modules that are first introduced in pentagonal elements in earlier figures. The modules (groups of subroutines) detailed in Figures 5-2 through 5-4 are:

- DRSPALL – main program , input and initialization
- RunLoop – main calculational control including time-stepping loop
- CalculateWellBoreFlow – controls and calculates multi-component fluid flow in the wellbore and fluidization of failed waste.
- CalculateWasteFlow – controls and calculates gas flow through the waste in the repository and fluidization of the failed waste
- CalculateWasteStress – controls and calculates the effective stresses and spalling (tensile) failure in the waste

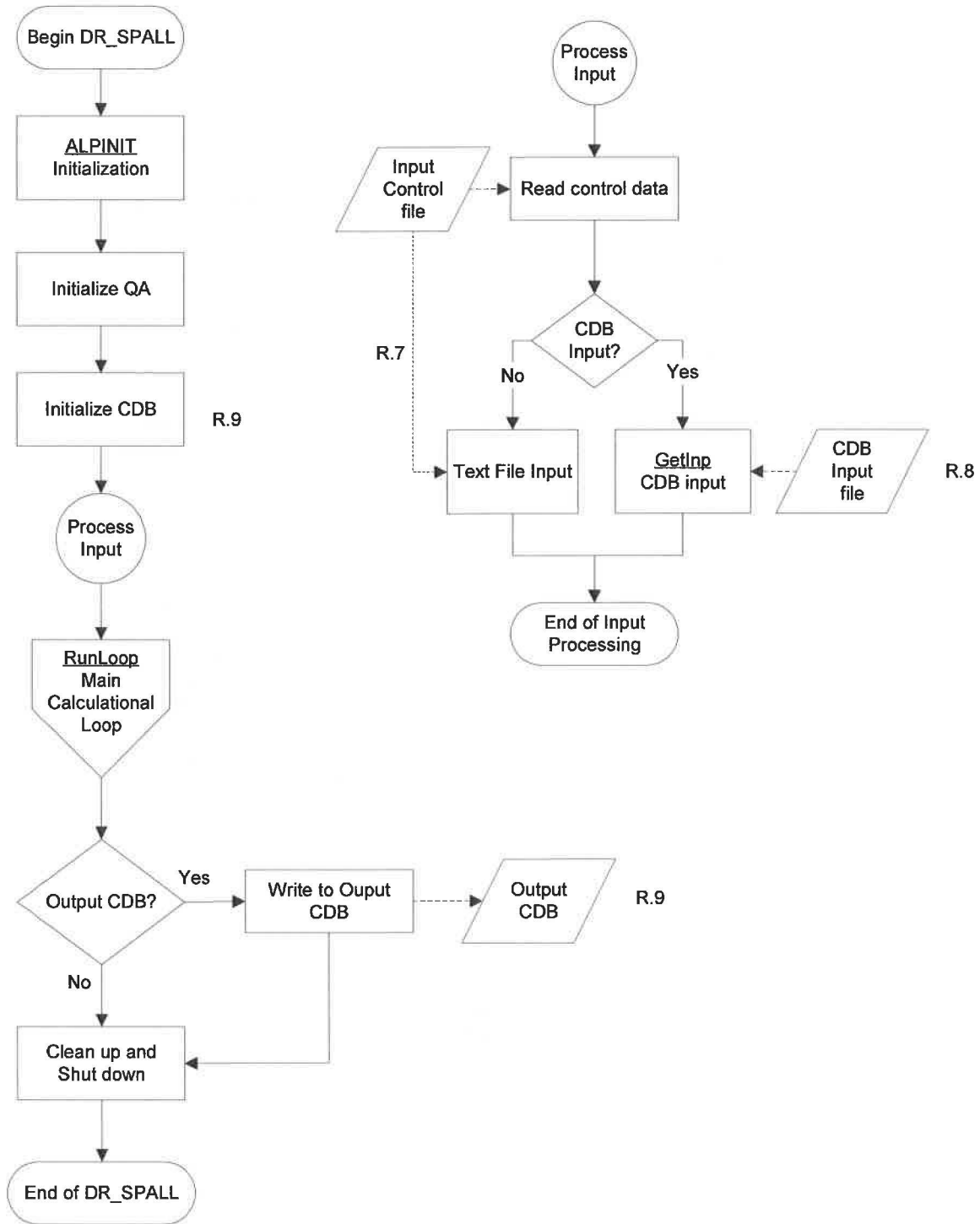


Figure 5-1. Flow chart for program DRSPALL

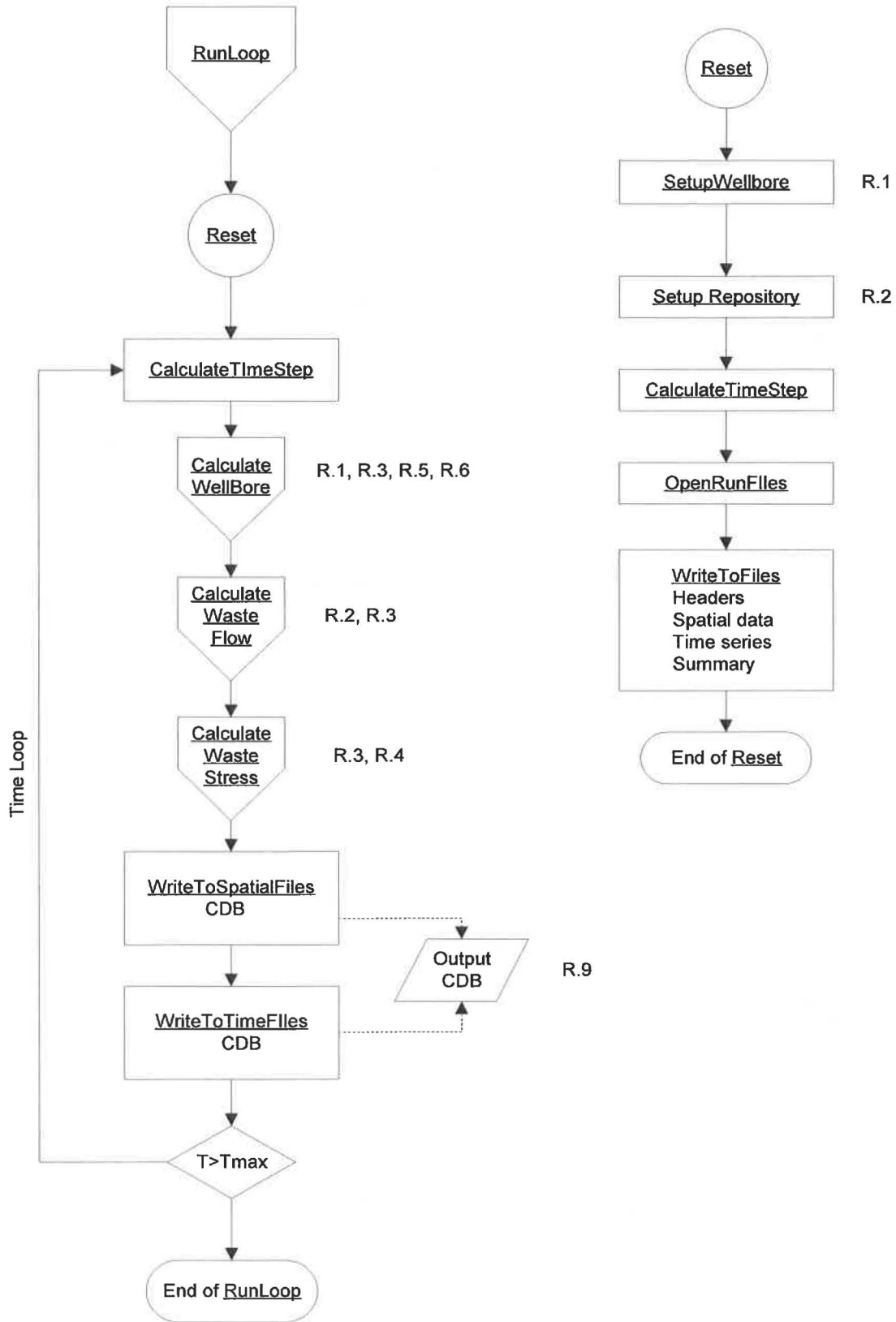


Figure 5-2. Flow chart for subroutine RunLoop

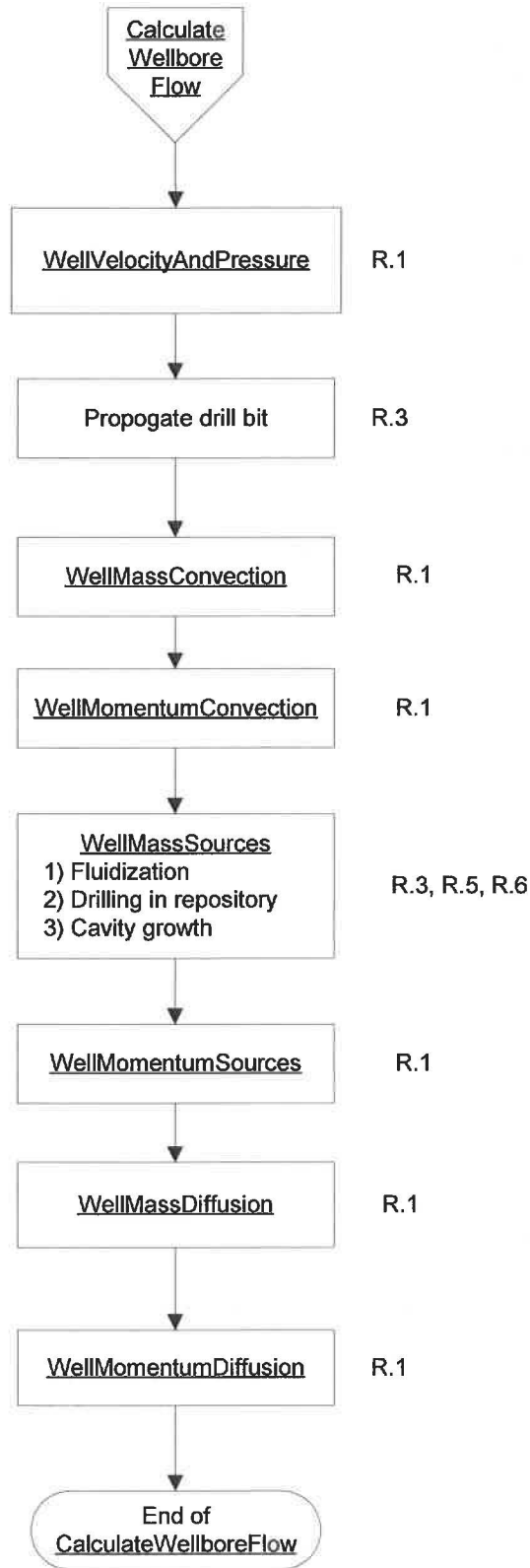
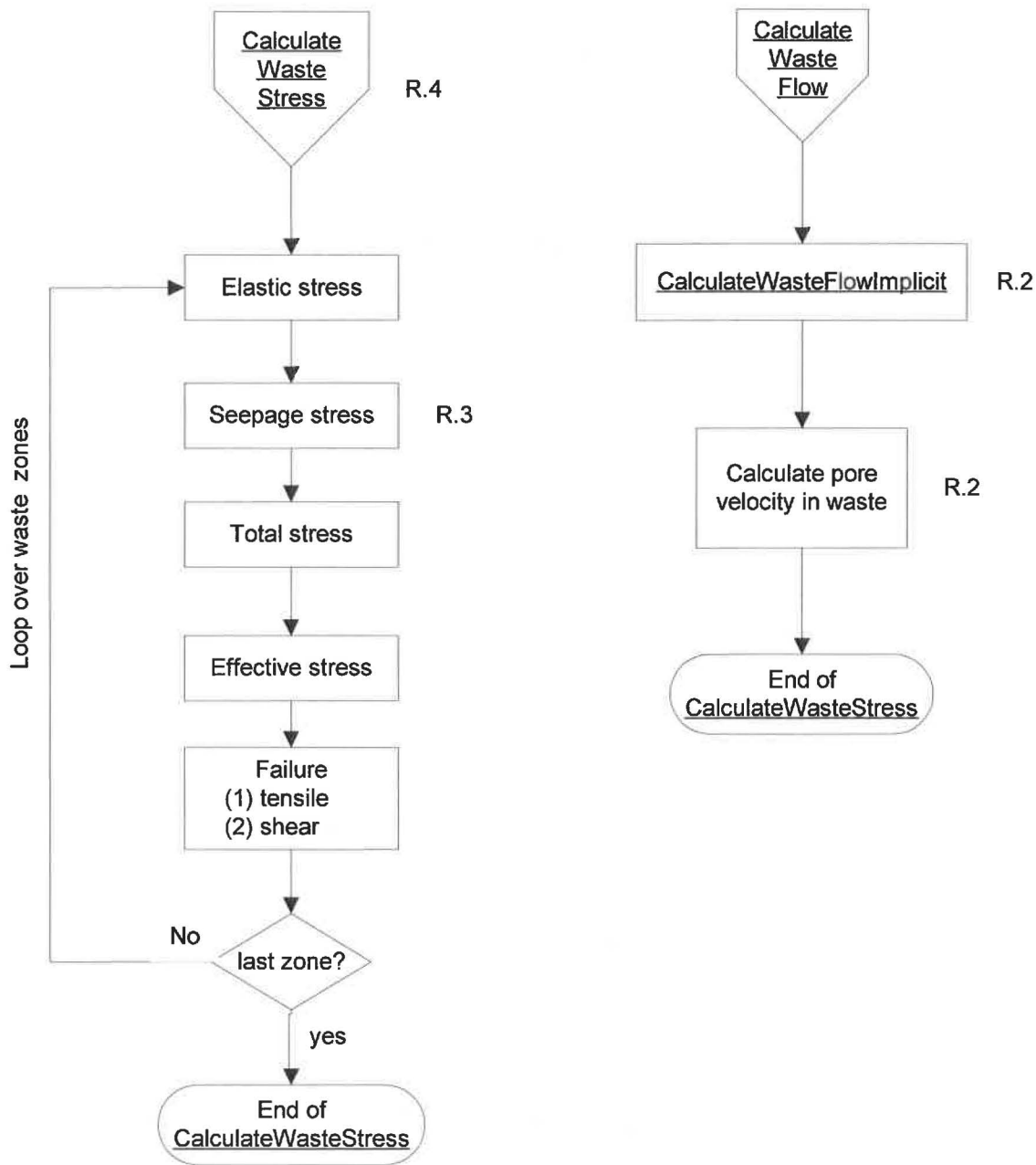


Figure 5-3. Flow chart for subroutine CalculateWellBoreFlow



**Figure 5-4.** Flow chart for subroutines CalculateWasteStress and CalculateWasteFlow

The following is the DRSPALL call tree that includes a brief routine (subroutine or function) description. An indentation indicates a call to the named routine. The combination of the flow charts (Figures 5-1 through 5-4) and the following call tree provides a complete description of the DRSPALL flow logic:

DRSPALL	Main program
. ALPINIT	Initialization
. . ReadControlFile	Read input control parameters, properties from control file or input CAMDAT file
. . CheckParameterBounds	Checks that input parameters are within valid range
. . InitializeTextOutput	Opens output files and writes headers
. . Reset	Problem setup control
. . . SetupWellbore	Setup wellbore flow model
. . . . ZoneWellbore	Setup grid in wellbore
. . . SetupRepository	Setup Repository waste model
. . . . CalculateWasteStresses	Initialize stress in repository
. . . CalculateTimestep	Calculate initial time step
. . . SetSpatialSaveTimes	Setup times for saving spatial variables
. . . SetTimeSaveTimes	Setup times for saving time series data
. . . OpenRunFiles	Open output text files
. . . WriteFileHeaders	Write header record to output files
. . . CloseRunFiles	If calculation finished, close files
. . CreateCDBHeader	Initializes and output CAMDAT file
. . CreateCDBVariables	Defines output variables for output CAMDAT file
. RunLoop	Calculation control

*Begin Time Loop*

. . CalculateTimestep	Calculate time increment for current step
. . CalculateWellboreFlow	Wellbore flow model control
. . . WellVelocityAndPressure	Fluid velocity and pressure in wellbore
. . . WellMassConvection	Mass flow (balance) in wellbore
. . . WellMomentumConvection	Momentum balance in wellbore
. . . WellMassSources	Determine mass sources (mud, gas, waste...)
. . . WellMomentumSources	Determine momentum sources
. . . WellMassDiffusion	Mass diffusion
. . . WellMomentumDiffusion	Momentum diffusion
. . CalculateWasteFlow	Gas flow in repository control
. . . CalculateWasteFlowImplicit	Implicit solver for gas flow in repository



- . . CalculateWasteStresses                    Effective stress model with seepage forces
- . . OutputControl                            Controls output to text and CAMDAT files

*End of time loop*

- . . RunShutDown                              Clean up for termination

DRSPALL is written in FORTRAN 90 for compilation under Solaris on the Oracle Sun cluster, but is designed to be relatively portable. The subroutines and functions that make up DRSPALL have been logically grouped by functionality into the compilable FORTRAN 90 (.f90) source files that follows:

<i>cdbcontrol.f90</i>	Collection of routines that read from and write to CAMDAT files.
<i>cdbglobals.F90</i>	Global variables related to CAMDAT files are declared here.
<i>A1main_drspall.f90</i>	This is the main program source file, which includes input control and initialization, QA functions and CAMDAT file initialization.
<i>vmsfilewrite.f90; vmsoutputcontrol.f90</i>	Collection of subroutines that write data to the output text file.
<i>globals.F90</i>	Global variables are declared here.
<i>maincalc.f90</i>	Collection of subroutines that include the main run loop and timestep calculations.
<i>parameters.f90</i>	Collection of subroutines that control parameter initialization and writes to the input parameter data echo.
<i>setupcalc.f90</i>	Collection of subroutines that initialize the wellbore, repository and other values required to start a run.
<i>wasteflowcalc.f90</i>	Collection of subroutines that calculate pressure and fluid flow in the repository.
<i>wastestresscalc.f90</i>	Subroutine that calculates solid stresses and failure in the repository.
<i>wellborecalc.f90</i>	Collection of subroutines that calculate fluid flow in the well, including pumping, expulsion and the coupling to the repository via the DDZ or cavity.

## 6 DATA STRUCTURES (I/O)

The primary data structures used by DRSPALL are the input files, output files and two FORTRAN 90 modules that contain most of the problem variables and parameters that must be shared between subroutines. Each of these structures is detailed in the following subsections.

Files are named with the prefix of the run name provided by the user and are suffixed according to the data contained in the file. The suffix and the corresponding data file description are summarized in Table 6-1.

*Table 6-1. Summary of DRSPALL Input and Output Files.*

Suffix	File contents
DRS	Input control and problem specification parameters
CDB	Input and output CAMDAT Database files
DBG	Input data echo and runtime diagnostics

### 6.1 Input Files

#### 6.1.1 Input Control and Parameter File

The input control and parameter file, prefixed 'drs' (suffixed 'inp'), lists input control and problem specification parameters. The following problem specification parameters may be specified in this file:

Land Elevation	Gas Viscosity
Repository Top	Mud Density
Total Thickness	Mud Viscosity
DRZ Thickness	Wall Roughness, Pipe
DRZ Permeability	Wall Roughness, Annulus
Outer Radius	Max. Solids Vol. Fraction
Initial Gas Pressure	Solids Viscosity Exponent
Far-Field In-Situ Stress	Choke Efficiency
Waste Porosity	Bit Diameter
Waste Permeability	Pipe Diameter
Forchheimer Beta	Collar Diameter
Biot Beta	Pipe Inside Diameter
Waste Poisson's Ratio	Collar Length
Waste Cohesion	Exit Pipe Length
Waste Friction Angle	Exit Pipe Diameter
Waste Tensile Strength	Drilling Rate
Characteristic Length (Lt)	Bit Above Repository (Initial)
Particle Diameter	Mud Pump Rate

Max Pump Pressure	Radius, Growth Rate
DDZ Thickness	Wellbore Cell Length
DDZ Permeability	Wellbore Zone Growth Rate
Stop Drilling Exit Vol. Rate	First Wellbore Zone
Stop Pumping Exit Vol. Rate	Well Stability Factor
Stop Drilling Time	Repository Stability Factor
Spherical/Cylindrical Geometry Flag	Mass Diffusion Factor
Fluidization Flag	Momentum Diffusion Factor
Maximum Run Time	Validation Test Case
Repository Cell Length	Initial Cavity Radius

### 6.1.2 *Input CDB File*

The user will be able to optionally read the problem specification data listed in Section 6.1.1 from an input CDB file. The details of the self-describing file structure can be found in Rechar et al. (1993) and in the *WIPP PA User's Manual for CAMDAT\_LIB* (WIPP PA 1996a). CAMDAT\_LIB provides a simplified programmers interface for accessing the data contained in a CDB file through subroutine calls.

## 6.2 **Output Files**

### 6.2.1 *Output Text Files*

A single output text file is created (suffix .DBG) which contains

- An echo of the input .DRS file.
- A listing of the input parameters, after processing, that reflects the actual parameters used in the current analysis. This includes values of parameters that were retrieved from the input CAMDAT file.
- A several line summary of results and timing information for each timestep that data is written to the output CAMDAT file.
- Run time diagnostic messages.
- A results summary at the end of the calculation.

### 6.2.2 *CDB Output File*

The optional CDB output file will contain, at a minimum, all input problem specification parameters listed in Section 6.1.1 and all the output data listed in Section 6.2.1. The CDB file is flexible enough to handle both spatial and temporal variables. The self-describing file structure is documented Rechar et al. (1993) and the *WIPP PA User's Manual for CAMDAT\_LIB* (WIPP PA 1996a). CAMDAT\_LIB provides a simplified programmers interface for creating the CDB file through subroutine calls.

### 6.3 FORTRAN 90 Global Variables Modules

Two FORTRAN 90 modules (*globals.F90*, *cdbglobals.F90*) have been defined for specifying all global variables. The module *cdbglobals.F90* contains shared variables related to reading and writing CAMDAT files. The module *globals.F90* contains all other shared variables. The modules include variable type declarations (integer, real, character, logical), precision (double, single), array dimensioning, records, structures definitions and parameter value setting. All the information contained in the module is made accessible to a routine by including a “*USE globals*” or “*USE cdbglobals*” statements in the declaration section of the routine. This is typically the first executable statement in the routine. DRSPALL uses variable names that are self-descriptive and therefore, the *globals.F90* and *cdbglobals.F90* listing shown below are, essentially, self-documenting. Note that some of the lines maybe wrapped.

```

Module Globals
!-----
!
! File Globals.f90 -- contains global parameters, variables, and constants
!
!-----

!Use DFWin
!Use DFLib
!Use GraphicsRecords
Implicit NONE

! Input Parameters
! -----

! Repository
Real(8) surfaceElevation, repositoryTop, repositoryThickness, dRZThickness, &
  dRZPerm, uncompactedWastePorosity, repositoryHeight, repositoryThickness0, &
  repositoryOuterRadius, repositoryInitialPressure, farFieldPorePressure, &
  farfieldStress, reposRadius1, growthRate

! Waste
Real(8) repositoryInitialPorosity, repositoryInitialPerm, forchBeta
Real(8) biotBeta, poissonsRatio, cohesion, frictionAngle, tensileStrength, &
  particleDiameter, fluidizationVelocity, Lt

! Gas
Real(8) gasBaseDensity, gasViscosity

! Mud
Real(8) initialMudDensity, mudViscosity, wallRoughness(2), mudSolidsMax, &
  mudSolidsViscosityExponent

! Wellbore/Drilling
Real(8) bitDiameter, pipeDiameter, collarDiameter, pipeInsideDiameter, collarLength, &
  drillingRate, initialBitAboveRepository, mudPumpRate, dDZThickness, dDZPerm, &
  stopDrillingExitVolRate, stopPumpingExitVolRate, stopDrillingTime, timeToFloor, &
  maxPumpPressure, initialPumpRate, pitGain

! Computational
Character(1) geometry, allowFluidization
Real(8) maxTime, initialReposZoneSize, initialWellZoneSize, &
  wellStabilityFactor, reposStabilityFactor

! APG Plot axes minimums / maximums (left here for compatibility with PC version)
Real(8) minPlotRadius, maxPlotRadius, minPlotWellPos, maxPlotWellPos, &
  minPlotPressure, maxPlotPressure, minPlotStress, maxPlotStress, &

```

```

minPlotVelocity, maxPlotVelocity, &
minPlotWellPressure, maxPlotWellPressure, &
minPlotWellVelocity, maxPlotWellVelocity, &
minPlotWellRho, maxPlotWellRho, &
minPlotWellFraction, maxPlotWellFraction
Logical newPlotRadius, newPlotWellPos, &
newPlotPressure, newPlotStress, newPlotVelocity, &
newPlotWellPressure, newPlotWellVelocity, newPlotWellRho, newPlotWellFraction

! Internal Parameters and Variables
! -----

! Repository array
!dkr Logical(1)
Logical , DIMENSION(:), ALLOCATABLE :: &
tensileFailureStarted, tensileFailureCompleted, shearFailed, &
fluidizationStarted, fluidizationCompleted
REAL(8), DIMENSION(:), ALLOCATABLE :: &
reposRadius, reposRadiusH, reposPres, porosity, permeability, forchratio, &
poreVelocity, superficialVelocity, reposFactor, pL, &
psi, dCoeff, radEffStress, tanEffStress, shearStress, &
fractionTensileFailed, tensileFailureTime, meanEffStress, shearStrength, &
fractionFluidized, fluidizationTime, FluidStartTime, FluidStopTime, FailStartTime, &
radElasticStress, tanElasticStress, radSeepageStress, tanSeepageStress, &
reposDR, reposDRH, reposVol, reposGasMass, drillingFailure
Real(8), DIMENSION(:), ALLOCATABLE :: aa, bb, cc, rr, gam

! Repository non-array
!dkr Integer(2)
Integer geomExponent
!dkr Integer(4)
Integer numReposZones, maxTensileFailedIndex, maxShearFailedIndex, firstIntactZone, &
interface1, interface2, lastFailedZone, firstInternalFailedZone, &
minNumLt
Real(8) reposZoneSize, wasteBoundaryPoreVelocity, sumReposGasMass, &
totalGasFromWaste, totalWasteFromRepos, maxForchRatio

!JFS3
!dkr Integer(4), Logical(1)
Integer fluidizationWaitZone
Logical surfaceFailureAllowed

! Cavity
Real(8) cavityPres, initialCavityRadius, initialCavityArea, initialCavityVol,
initialCavityGasVol, &
cavityGasMass, cavityWasteVol, cavityWasteMass, deltaGasIntoWell, deltaWasteIntoWell, &
deltaSaltIntoWell, deltaGasFromWaste, exitPoreVelocity, exitPoreArea, &
inputCavityRadius, cutMass, splMass, totMass, splvoleq, cutvoleq, totvoleq, &
cutVol, splVol, splvol2, TotVol, cutTrueVol, splmass2, beddepth, minCharVel

! Wellbore array
REAL(8), DIMENSION(:), ALLOCATABLE :: &
wellPos, wellPres, wellRhoV, wellVol, WellArea, wellZoneSize, &
wellV, wellRho, wellAreaInt, wellDeltaVInt, wellFactor, &
wellMudMass, wellGasMass, wellWasteMass, wellSaltMass, &
wellMudVol, wellGasVol, wellWasteVol, wellSaltVol, hydraulicDia
REAL(8), DIMENSION(:), ALLOCATABLE :: &
wellMassOrig, wellPresOrig, wellMudMassOrig, &
wellGasMassOrig, wellWasteMassOrig, wellSaltMassOrig
REAL(8), DIMENSION(:), ALLOCATABLE :: &
wellGRhoStar, wellGRhoVStar, wellGRhoT, wellGRhoVT, &
wellVolInt, momDiff, massDiff

! Wellbore non-array
!dkr Logical(1), Integer(4)
Logical repositoryPenetrated, firstPenetration, drilling, pumping, exitPipe, &
internalFailure

```

```

Integer numWellZones, wellBottomIndex, numExitZones, firstWellZone, &
  numWellZones1, numWellZones2, numWellZones3, numWellZones4

Real(8) wellLength, bitAboveRepository, dRZDiameter, dRZArea, wellGrowthRate, &
  bitArea, bitFlowArea, pipeArea, collarAnnulusArea, pipeAnnulusArea, &
  exitPipeLength, exitPipeDiameter, exitPipeArea, wellDepth, &
  mudMassEjected, gasMassEjected, wasteMassEjected, saltMassEjected, &
  totalMudInWell, totalGasInjected, totalGasInWell, totalWasteInWell, totalSaltInWell, &
  cuttingsMassMax, cuttingsRadiusMax, equivDrilledCavityRadius, &
  volStore, wasteStore, gasStore, wasteInjected, originalBitDiameter, &
  sumWellGasMass, sumWellWasteMass, Z1, Z2, Z3, Z4

! General array
REAL(8), DIMENSION(:), ALLOCATABLE :: invPorosity

! General non-array
Real(8) invGasViscosity, invWellZoneSize, invWasteDensity, invSaltDensity,
invWaterCompressibility, &
  invInitialMudDensity, invGasBaseDensity, invInitialCavityGasVol, invPi, invAtmosphericPressure,
&
  invReposZoneSize, timeOfPenetration

! Computational Parameters and Constants
!-----

! Run management
Real(8) runTime, deltaTime
Double Precision cpuTime, cpuBegin, cpuEnd, & !apg REAL
  cpuT0, cpuT1, cpuT2, cpuT3, cpuWBF, cpuWSF, cpuWSS

!dkr Integer(4), Logical(1)
Integer runIndex
Logical stopRun, isRunning, pauseRun, CDBOutput, textOutput
Character*3 platform, machine
Character*132 root

! Window unit constants
!dkr Integer(4)
Integer ParameterUnit, TimeUnit, RepositorySpatialPressureUnit, RepositorySpatialStressUnit, &
  RepositorySpatialVelocityUnit, WellboreSpatialPressureUnit, WellboreSpatialMassUnit, &
  ResultsSummaryUnit

! File logic
!Type(T_OpenFilename) fName
!Logical(1) fileLoaded, fileSaved, fileNamed
Character(255) fullParameterFilename, pressureFileName, radEffStressFileName, &
  tanEffStressFileName, poreVelocityFileName, timeFileName, summaryFileName, outParamsFileName, &
  diagnosticFileName, validationFilePrefix
!apg Added validationFilePrefix for files below
Character(255) TC1chanFileName, TC4couplefilename, TC4stressfilename, &
  TC4Fluidfilename, TC4Ejectfilename, TC4FluidTimefilename, TC5Wellfilename

!dkr Integer(4)
Integer parameterFileID, pressureFileID, radEffStressFileID, tanEffStressFileID, &
  poreVelocityFileID, timeFileID, summaryFileID, outParamsFileID, diagnosticFileID, &
  chanValidationFileID, &
  !test case #4
  couplingValidationFileID, stressValidationFileID, fluidizationvalidationFileID, &
  expulsionValidationFileID, fluidizationTimeValidationFileID, &
  !test case #5
  wellboreValidationFileID

! Character-based user input parameter values
Character(30) gridItems(2,44)
Character(10) gridUnits(2,44)
Character(14) gridValues(2,44)

```

```

! Parameter input grids
!dkr Integer(2)
Integer numGridRows, numGridCols, rowHeight, margin
Integer gridWidth, colWidth(3)

! DKR Graphics records moved to separate module for portability with VMS

! Plot and save times
!dkr Real(4), Integer(4)
Real(8) spatialPlotTime(1000), spatialSaveTime(0:1000), & !apg Real
    timeSaveTime(0:2000), chanSaveTime(10)
Integer spatialPlotIndex, spatialSaveIndex, timeSaveIndex, &
    chanSaveIndex

! APG for time plots
Integer ixtplt
!dkr Real(4) times 3
Real(8) finaltplt !apg Real
Integer, Parameter :: maxtplt = 1000
Real(8) tpltTime(0:maxtplt) !apg Real
Real(8) tpltPumpPressure(0:maxtplt), tpltWellBottomPressure(0:maxtplt),
tpltCavityPressure(0:maxtplt), & !apg Real
    tpltDrilledRadius(0:maxtplt), tpltCavityRadius(0:maxtplt), tpltTensileRadius(0:maxtplt), &
    tpltWasteBoundaryPoreVelocity(0:maxtplt), tpltFluidizationVelocity(0:maxtplt),
tpltMudEjectionVelocity(0:maxtplt), &
    tpltWasteInWell(0:maxtplt), tpltWasteEjected(0:maxtplt), &
    tpltGasInjected(0:maxtplt), tpltGasInWell(0:maxtplt), tpltGasEjected(0:maxtplt), &
    tpltGasPosInWell(0:maxtplt), tpltWastePosInWell(0:maxtplt)

! for plotting and saving
! time plots
!dkr Real(4)
Real(8) curTime, & !apg Real
    curPumpPressure, curWellBottomPressure, curCavityPressure, &
    curDrilledRadius, curCavityRadius, curTensileRadius, &
    curWasteBoundaryPoreVelocity, curFluidizationVelocity, curMudEjectionVelocity, &
    curWasteInWell, curWasteEjected, &
    curGasInjected, curGasInWell, curGasEjected, &
    curGasPosInWell, curWastePosInWell

!dkr Real(4) times 2
Real(8) prevTime !apg Real
Real(8) prevPumpPressure, prevWellBottomPressure, prevCavityPressure, & !apg Real
    prevDrilledRadius, prevCavityRadius, prevTensileRadius, &
    prevWasteBoundaryPoreVelocity, prevFluidizationVelocity, prevMudEjectionVelocity, &
    prevWasteInWell, prevWasteEjected, &
    prevGasInjected, prevGasInWell, prevGasEjected, &
    prevGasPosInWell, prevWastePosInWell

! summary display
!dkr Real(4)
Real(8) displayedTime, & !apg Real
    displayedPumpPressure, displayedBitAbove, &
    displayedWellBottomPressure, displayedCavityPressure, displayedWasteBoundaryPoreVelocity, &
    displayedFluidizationVelocity, displayedEquipDrilledCavityRadius, displayedCavityRadius, &
    displayedTensileFailedRadius, displayedShearFailedRadius, displayedGasInjected, &
    displayedMudInWell, displayedGasInWell, displayedWasteInWell, displayedSaltInWell, &
    displayedMudVelocity, displayedMudVolRate, displayedMudEjected, displayedGasEjected,
displayedWasteEjected, displayedSaltEjected, &
    displayedMudExitFraction, displayedGasExitFraction, displayedWasteExitFraction,
displayedSaltExitFraction, &
    displayedGasPosInWell, displayedWastePosInWell, displayedradEffStress, displayedFirstIntactZone

Integer cellControl(5), oldCellControl, stepControl, oldStepControl
Integer DisplayedCell
Character*8 stepControlName(5), displayedControl

```

```

! Validation Test Case specific parameters
!dkr Integer(2)
Integer validationTestCase, validationSubcase
Real(8)    characteristicTime

!test case #4
Real(8) stressSaveTime, stressSaveDelta, fluidizationSaveTime, fluidizationSaveDelta,&
        initialGasInRepository

!DKR - parameters switched to variables. set in reset
REAL(8) gravity, MassDiffusionFactor, MomentumDiffusionFactor, chokeEfficiency, &
        WaterCompressibility, GasConstant, ReposTemp, WasteDensity, saltDensity, &
        BitNozzleNumber, BitNozzleDiameter, Pi, AtmosphericPressure, shapeFactor, &
        TensileVelocity
!dkr Integer(2)
Integer FIX, ENG

! Constants
!dkr Integer(2) times 3
Integer, Parameter :: FULL_RESET = 1
Integer, Parameter :: RUNNING     = 2
Integer, Parameter :: REOPENING   = 3

end

!-----

MODULE CDBGlobals
!-----
! This module conatins variables required to read or write a CDB
!
! By:    David Rudeen
!       GRAM, Inc.
!
! Modification History
! 12/19/02 - DKR: Original version
!-----
IMPLICIT NONE
INTEGER, PARAMETER :: nev=26, nhv=47
CHARACTER*8    HisName(nhv), NAMXYZ(3), EleName(nev), blockName
CHARACTER*80   HEAD
CHARACTER*255  outCDBfilename, incDBfilename
INTEGER        IDBout, NOUT, NumHisVar, LDUM, IERRDB, NDIM, NUMNP, &
        NELBLK, NELX, NELY, NELZ, MAXLNK, NEB, NUMPRP, NUMATR, &
        NUMEL, NumEleVar, outCDBfileID, incDBfileID, &
        NINFO, numCDBsteps, numParam
INTEGER        IDBin, NQAREC, NELBLKin, NUMATRin, NUMPRPin, NUMNPS, LNPSNL, &
        NUMESS, LGELEL, LGNODL, DRsblock, reposBlock, wellBlock1, wellBlock2
LOGICAL,       DIMENSION(:, :), ALLOCATABLE :: ISEVOK, ISPROK
LOGICAL,       DIMENSION(:),   ALLOCATABLE :: ISATOK
INTEGER,       DIMENSION(:, :), ALLOCATABLE :: Link
INTEGER,       DIMENSION(:),   ALLOCATABLE :: EleMap, IDELB, NUMLNK, NUMELB
REAL(4),      DIMENSION(:),   ALLOCATABLE :: Xn, Yn, ELEVAT, THICK, DEL_Y, DEL_X
REAL(4),      DIMENSION(:, :), ALLOCATABLE :: EleVal, PROP
CHARACTER*80, DIMENSION(:),   ALLOCATABLE :: INFO
CHARACTER*8,  DIMENSION(:),   ALLOCATABLE :: NAMELB, PRPname, ParamName, NAMATR
CHARACTER*8,  DIMENSION(:, :), ALLOCATABLE :: QAREC

REAL(4)       RMEM(1), HisVal(nhv), ZN(1)
CHARACTER*1   CMEM(1)
LOGICAL       WHOLE, CDBinput

END

```



## 7 PARAMETER RANGES

Many of the parameters must be within certain acceptable ranges to represent physical reality or true repository and/or wellbore conditions or for compatibility with other input parameters. Where possible, a limitation scheme involving parameter range checking at the time parameters are read in will flag unacceptable parameter values and stop execution. But, because DRSPALL has been designed for any self-consistent set of input parameters and any single parameter value can vary significantly depending on the system of units, it is not possible to provide upper bounds for many of the input parameters. Table 7-1 lists all input variables and provides parameter bounds when they can be reasonably defined.

*Table 7-1. Input Parameters and Ranges.*

<b>Parameter Description</b>	<b>Range</b>
Land Elevation	> Repository Top
Repository Top	Any real
Total Thickness	$\geq 0$
DRZ Thickness	> 0
DRZ Permeability	> 0
Outer Radius	> 0.5 * Bit Diameter
Initial Gas Pressure	> 0, < Overburden ( $\rho_s g h$ ); h = Land Elevation – Repository Top
Far-Field In-Situ Stress	> 0, < Overburden
Waste Porosity	0 < Porosity < 1.0
Waste Permeability	> 0
Forchheimer Beta	> 0
Biot Beta	$\geq 0, \leq 1$
Waste Poisson's Ratio	0 < Poisson's Ratio < 0.5
Waste Cohesion	> 0
Waste Friction Angle	0 < Friction Angle < 90°
Waste Tensile Strength	< 0
Failure Characteristic Length (Lt)	> 0
Waste Particle Diameter	< Bit Diameter
Gas Density at STP	> 0
Gas Viscosity	> 0
Mud Density	> 0
Mud Viscosity	> 0
Wall Roughness Pipe	> 0
Wall Roughness Annulus	> 0
Max. Solids Vol. Fraction	> 0, < 1
Solids Viscosity Exponent	< 0
Bit Diameter	> 0
Pipe Diameter	> 0, < Bit Diameter
Collar Diameter	> Pipe Diameter, < Bit Diameter

<b>Parameter Description</b>	<b>Range</b>
Pipe Inside Diameter	> 0, < Pipe Diameter
Collar Length	> 0, < h
Exit Pipe Length	> 0
Exit Pipe Diameter	> 0
Drilling Rate	> 0
Bit Above Repository (Initial)	> 0, < h
Mud Pump Rate	> 0
Max Pump Pressure	> 0
DDZ Thickness	> 0
DDZ Permeability	> 0
Stop Drilling Exit Vol. Rate	> 0
Stop Pumping Exit Vol. Rate	> 0

*Table 7-1. Input Parameters and Ranges (Continued).*

<b>Parameter Description</b>	<b>Range</b>
Stop Drilling Time	> 0
Spherical/Cylindrical Geometry Flag	S or C
Fluidization Flag	Y or N
Maximum Run Time	> 0
Repository Cell Length	> 0
Radius, Growth Rate	> 0, 1.0
Wellbore Cell Length	> 0
Wellbore Cell Growth Rate	1.0
First Wellbore Zone	> 0
Well Stability Factor	> 0
Repository Stability Factor	> 0
Mass Diffusion Factor	> 0
Momentum Diffusion Factor	> 0
Initial Cavity Radius	> 0
Minimum Characteristic Velocity	> 0
Minimum Number Zones/Lt	≥ 1
Pi	3.141592653589793d0
Atmospheric Pressure	> 0
Gravity	9.8067d0
Gas Constant	4124.0d0
Repository Temperature	> 0
Water Compressibility	> 0
Waste Density	> 0
Salt Density	> 0
Shape Factor	> 0, ≤ 1
Tensile Velocity	> 0
Bit Nozzle Number	> 0
Bit Nozzle Diameter	> 0
Choke Efficiency	> 0, < 1

## 8 TEST CASE CONTROL LOGIC

Each of the validation test cases described in the *Verification and Validation Plan / Validation Document for DRSPALL Version 1.22* (VVP/VD, WIPP PA 2015b), requires some code control logic specific to the test case. This involves either turning off functionality not required in the test case, setting special boundary conditions, or creating formatted ASCII output files that are analyzed and evaluated against the acceptance criteria outlined in the VVP/VD. The input control file to DRSPALL contains a parameter that identifies the specific test case running, i.e., "1.2." The subsequent discussion identifies the code logic specific to these test cases, and includes rationale for why the logic is used.

The nomenclature used in this section is shown in Table 8-1.

**Table 8-1. Nomenclature for Section 8 - Test Case Control Logic.**

Symbol	Definition	Units
$a$	Cavity radius	m
$A$	Surface area of cavity forming interface between repository domain and wellbore domain	m <sup>2</sup>
$D_b$	True diameter of drill bit	m
$D_E$	Effective diameter of drill bit required to simulate full penetration at early time for the porous flow test problem	m
$\eta$	Gas viscosity	Pa·sec
$H$	Height of repository at time of penetration	m
$k$	Repository permeability	m <sup>2</sup>
$\phi$	Repository porosity	--
$P(r)$	Gas pressure as a function of radius at a given time	Pa
$P_I$	Initial repository pressure	Pa
$\tau$	Dimensionless time	--
$t_o$	Characteristic time	sec
$t$	Time	sec
$\Psi$	Dimensionless pseudopressure	--
$z$	Dimensionless radius	--
$\zeta$	Dimensionless plotting parameter	--

### 8.1 Porous Flow Test Problem

The purpose of the gas flow test problem is to determine whether DRSPALL can accurately calculate transient gas pressures in the repository during the first few seconds after a borehole intrusion. The problem is designed to simulate the repository domain only. As such, the wellbore domain is decoupled from the repository and inactive during execution of the test problem. Input parameters pertaining to the wellbore are therefore irrelevant in this test case, as are any code output describing wellbore conditions.

### 8.1.1 Constant Pressure Inner Boundary

The test problem calls for a constant pressure inner boundary. While normal execution of DRSPALL couples the repository and wellbore with a common pressure boundary, this test case circumvents the coupling by assigning an inner boundary gas pressure of  $P = 0$  MPa.

### 8.1.2 Effective Bit Diameter

Drilling does not actually progress in this test case. Rather, the problem starts with a “cavity” cut out of the repository that has a surface area equivalent to the area of the cavity face that would result from the penetration of the true bit diameter  $D_b$  through a repository of height  $H$ . The surface area  $A$  of this cylinder representing the borehole wall is evaluated as:

$$A = \pi D_b H \quad (8.1.1)$$

where the flat ends of the cylinder are not considered because these do not transfer gas between the repository and the wellbore at full penetration. For the purpose of this test case, an effective bit diameter  $D_E$  must be defined that represents a hypothetical bit that creates a cavity of area  $A$  the instant it touches the repository. The initial contact geometry of the bit with the repository is a circle, with diameter  $D_E$ . The effective diameter is therefore related to the cavity area by the formula for the area of a circle:

$$A = \frac{\pi D_E^2}{4} \quad (8.1.2)$$

Setting the surface areas in Equations 8.1.1 and 8.1.2 equal to one another yields:

$$A = \frac{\pi D_E^2}{4} = \pi D_b H \quad (8.1.3)$$

and rearranging to solve for  $D_E$  yields:

$$D_E = 2\sqrt{D_b H} \quad (8.1.4)$$

The effective diameter is calculated by the code only if this test case flag is turned on. While the actual bit diameter  $D_b$  is read as an input variable for this test case,  $D_E$  is calculated according to Equation 8.1.4 and used in the porous flow calculations.

The cavity radius,  $a$ , used for the characteristic time shown in Table 8-2 below, is defined depending on the problem geometry. For the cylindrical case,  $a$  is equal to the true bit radius:

$$a = D_E / 2 \quad (8.1.5)$$

and for the spherical case,  $a$  is defined as the radius of a hemisphere with area  $A$ :

$$\alpha = \sqrt{\frac{A}{2\pi}} \quad (8.1.6)$$

### **8.1.3 *Special Output and Dimensionless Variables***

The methodology outlined in Chan et al. (1993) calls for the definition of several special variables defined in Table 8-2.

**Table 8-2. Special variables defined for Porous Flow Test Problem.**

Variable name	Definition	Equation number
Dimensionless pseudopressure	$\Psi = \frac{P(r)^2}{P_1^2}$	8.1.7
Dimensionless time	$\tau = \frac{t}{t_0}$	8.1.8
Characteristic time (sec)	$t_0 = \frac{\phi\eta a^2}{kP_1}$	8.1.9
Dimensionless radius	$z = \ln(r/a)$ cylindrical $z = a/r$ spherical	8.1.10
Dimensionless plotting parameter	$\zeta = \frac{(e^z - 1)}{\tau^{1/2}}$ cylindrical $\zeta = \frac{(1/z - 1)}{\tau^{1/2}}$ spherical	8.1.11

Relevant variables are defined in the nomenclature table for Section 8 (Table 8-1). The scaling of output to dimensionless space allows us to compare DRSPALL results to different solutions without matching specific input conditions (i.e., values for variables in Table 8-1). As such, selection of input parameter values is not instrumental to proper solution of this test case. Logic is built into the code such that upon activating this test case, the code will:

- calculate the characteristic time  $t_0$  for the particular set of input variables
- calculate dimensional times  $t=t_0\tau$  corresponding to dimensionless times  $\tau = 0.01, 0.1, 1.0,$  and  $10.0$
- write to an output ASCII file the following data at  $\tau = 0.01, 0.1, 1.0,$  and  $10.0$ 
  - $t$
  - $\tau$
  - $r, P(r,t), \zeta, \Psi$  ...for ~25 computational cells in the domain starting at the cavity wall

## 8.2 Coalbed Methane Field Scale Test Case

This test case is designed to validate the total spallings solids release model against field observations of coalbed methane cavity completion (“cavitation”) operations. The problem does not include drilling or mud. Only gas plus coal solids flow in the well. Therefore, some special logic is included in the problem setup modules to properly define the problem. The time step algorithm is modified to use the sound speed in gas rather than mud. The wellbore pressure logic uses the mixture equation-of-state, but without the presence of mud and salt.

### 8.3 Internal Logic Checks Test Problem

This test case demonstrates that DRSPALL accurately calculates:

1. Coupling of flows in the wellbore and the repository
2. Tensile failure of homogenous waste material using effective stress and seepage laws
3. Fluidized bed transport of disaggregated waste material
4. Expulsion of disaggregated waste material at the land surface.

There is no special code logic for this test case that affects the calculations. There is coding that creates and formats four ASCII output files that are used in the post-processing of the test case data. The format and content of these files are discussed in VVP/VD for DRSPALL (WIPP PA 2015b).

### 8.4 Wellbore Hydraulics Model Test Case

The objective of wellbore hydraulics test case is to validate the DRSPALL wellbore calculations against an independent computational fluid dynamics model FLUENT. This problem is designed to study the wellbore only. As such, the repository domain is decoupled from the wellbore and inactive during execution of the test problem. Input parameters pertaining to the repository are therefore irrelevant in this test case, as are any code output describing repository conditions.

The lower boundary to the wellbore domain corresponds to the very bottom of the borehole where repository gases and solids may enter the mud flow stream. In normal DRSPALL execution, the bottomhole pressure acts as a common boundary condition for the inner boundary of the repository and the bottom of the wellbore. Gas and potentially solids are transferred from the repository domain to the wellbore domain at this boundary. For this test case, the lower boundary requires specific control over the mass loading rate of gas and/or solids in order to set up analogous runs with the computational tool FLUENT. These mass loading rates are pre-determined and described in the VVP/VD for DRSPALL (WIPP PA 2015b). They are implemented in DRSPALL by circumventing the normal coupling function with the repository and building the mass loading directly into the wellbore mass transfer algorithm.

## 9 REFERENCES

- Barree, R.D. and M.W. Conway 1995 "Experimental and Numerical Modeling of Convective Proppant Transport," *Journal of Petroleum Technology*, March, 216-222.
- Belhaj, H.A., K.R. Agha, A.M. Nouri, S.D. Butt, H.F. Vaziri, and M.R. Islam. 2003. "Numerical Simulation of Non-Darcy Flow Utilizing the New Forchheimer's Diffusivity Equation," SPE 81499, presented at the SPE 13<sup>th</sup> Middle East Oil Show & Conference, Bahrain, April, 2003.

- Chan, D.Y.C., B.D. Hughes, and L. Patterson 1993. *Transient Gas Flow Around Boreholes, Transport in Porous Media*, 10, 137-152. (Copy of document on file in the Sandia WIPP Central Files as WPO#46902). Albuquerque, NM: Sandia National Laboratories.
- Cherimisinoff, N.P. and P.N. Cherimisinoff 1984. *Hydrodynamics of Gas-Solids Fluidization*. Houston, TX: Gulf Publishing Company.
- Ergun, S. 1952. Fluid Flow Through Packed Columns, *Chemical Engineering Progress*, 48(2):89-94.
- Fox, R.W. and A.T. McDonald 1985. *Introduction to Fluid Mechanics, 3<sup>rd</sup> Ed.* New York, NY: John Wiley and Sons.
- Gross, M.B. and T.W. Thompson 1998 "Analysis Package for the Semi-Analytical Calculations Conducted In Support of Spalling Release Estimates for the 1998 Annual Sensitivity Study," (ASA98) WPO# 49423, CAO Technical Assistance Contract Carlsbad, New Mexico, for Waste Isolation Pilot Plant (WIPP) Project. Albuquerque, NM: Sandia National Laboratories.
- Hassanizadeh, S.M. and W.G. Gray 1987. "High Velocity Flow in Porous Media," *Transport in Porous Media*. Vol 2:521-531.
- Hansen, F.D., M.K. Knowles, T.W. Thompson, M. Gross, J.D. McLennan and J.F. Schatz 1997. *Description and Evaluation of a Mechanistically Based Conceptual Model for Spall*. SAND97-1369. Albuquerque, NM: Sandia National Laboratories.
- Jaeger, J.C. and N.G.W. Cook 1969. *Fundamentals of Rock Mechanics*, Chapman and Hall Ltd., London, England.
- Li, D., R.K. Svec, T.W. Engler, and R.B. Grigg 2001. "Modeling and Simulation of the Wafer Non-Darcy Flow Experiments, paper SPE 68822, SPE Western Regional Meeting, Bakersfield, CA, March 26-30.
- Lord, D.L., D.K. Rudeen, J.F. Schatz, A.P. Gilkey, and C.W. Hansen 2006. *DRSPALL: Spallings Model for the Waste Isolation Pilot Plant 2004 Recertification*. SAND2004-0730. Albuquerque, NM: Sandia National Laboratories.
- Narayanaswamy, G., M.K. Sharma, and G.A. Pope 1999. "Effect of Heterogeneity on the Non-Darcy Flow Coefficient," *SPE Reservoir Eval. & Eng.* 2 (3) June.
- Oran, E.S. and J.P. Boris 1987. *Numerical Simulation of Reactive Flow*. New York, NY: Elsevier.
- Özişik, M.N. 1993. *Heat Conduction*. Second Edition. New York, NY: John Wiley & Sons.
- Podio, A.L. and A.P. Yang 1986. "Well Control Simulator for IBM Personal Computer," IADC/SPE 14737. International Association of Drilling Engineers/Society of



Petroleum Engineers paper presented at the IADC/SPE Drilling Conference held in Dallas, TX. February 10-12.

- Rechard, R.P., A.P. Gilkey, H.J. Iuzzolino, D.K. Rudeen, and K.A. Byle 1993. *Programmer's Manual for CAMCON: Compliance Assessment Methodology Controller*. SAND90-1984. Albuquerque, NM: Sandia National Laboratories.
- Serghides, T.K. 1984. "Estimate Friction Factor Accurately," *Chemical Engineering*, March 5, 63-64.
- Thomas, J.W. 1995. *Numerical Partial Differential Equations: Finite Difference Methods*. Texts in Applied Mathematics 22. New York, NY: Springer-Verlag.
- Timoshenko, S.P. and J.N. Goodier 1970. *Theory of Elasticity*, Third Ed. McGraw-Hill: New York, NY.
- WIPP PA 1996a. *WIPP PA User's Manual for CAMDAT\_LIB, Version 1.22*. Albuquerque, NM: Sandia National Laboratories. ERMS# 527727.
- WIPP PA 1996b. *WIPP PA User's Manual for CAMCON\_LIB, Version 2.16*. Albuquerque, NM: Sandia National Laboratories. ERMS# 527738.
- WIPP PA 1996c. *WIPP PA User's Manual for CAMSUPES\_LIB, Version 2.18*. Albuquerque, NM: Sandia National Laboratories. ERMS# 527745.
- WIPP PA. 2003. *Requirements Document for DRSPALL Version 1.00*. Carlsbad, NM: Sandia National Laboratories. ERMS# 531278.
- WIPP PA 2004. *Design Document for DRSPALL Version 1.10*. Carlsbad, NM: Sandia National Laboratories. ERMS# 533149.
- WIPP PA 2013. *Addendum to Design Document (ERMS# 533149) for DRSPALL Version 1.21*. Carlsbad, NM: Sandia National Laboratories. ERMS# 560045.
- WIPP PA. 2015a. *Implementation Document for DRSPALL Version 1.22*. Carlsbad, NM: Sandia National Laboratories. ERMS# 562641.
- WIPP PA. 2015b. *Verification and Validation Plan / Validation Document for DRSPALL Version 1.22*. Carlsbad, NM: Sandia National Laboratories. ERMS# 562643.

## APPENDIX A: FUNCTIONAL REQUIREMENTS

In general DRSPALL shall calculate the volume of WIPP waste subject to material failure and transport to the surface as a result of an inadvertent drilling intrusion into the repository. More specifically DRSPALL will calculate the following:

- R.1 Compressible, viscous, isothermal, multiphase mixture flow (mud, salt, waste, repository gas) in the wellbore using one dimensional linear geometry and assuming a Newtonian fluid. Either laminar or turbulent flow shall be modeled depending on wellbore and fluid properties.
- R.2 Repository gas flow as single-phase Darcy porous flow using either one dimensional cylindrical or spherical geometry
- R.3 Coupling of the wellbore and the repository flow models prior to and after penetration
- R.4 Spalling (tensile) failure of the homogeneous waste material using an effective stress law with seepage forces
- R.5 Fluidized bed transport of failed (disaggregated) waste material.
- R.6 Mixture expulsion at the surface

### External Interface Requirements

- R.7 DRSPALL shall read an input control file, which may be pre-generated using a text processor. It will contain numerical control parameters and, optionally, material properties and problem geometry.
- R.8 Properties and non-numerical control parameters will, optionally, be read from a CDB.
- R.9 Grid, properties, parameters and spatial and time dependent results will be written to an output CDB.

DRSPALL will link to the standard WIPP code libraries, namely, CAMDAT\_LIB, CAMCON\_LIB, and CAMSUPES\_LIB (Rechard et al. 1993). User interactions will consist of command line execution, batch files, and input files.