PNNL-15782

Pacific Northwest
National Laboratory
Operated by Battelle for the
U.S. Department of Energy

STOMP
Subsurface Transport Over Multiple Phases

Version 4.0

User's Guide

M. D. White
M. Oostrom

June 2006

Prepared for the U.S. Department of Energy
under Contract DE-AC05-76RL01830

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor Battelle Memorial Institute, nor any of their employees, makes **any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights**. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or Battelle Memorial Institute. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

PACIFIC NORTHWEST NATIONAL LABORATORY
operated by
BATTELLE
for the
UNITED STATES DEPARTMENT OF ENERGY
under Contract DE-AC05-76RL01830

Printed in the United States of America

**Available to DOE and DOE contractors from the
Office of Scientific and Technical Information,
P.O. Box 62, Oak Ridge, TN 37831-0062;
ph: (865) 576-8401
fax: (865) 576-5728
email: reports@adonis.osti.gov**

**Available to the public from the National Technical Information Service,
U.S. Department of Commerce, 5285 Port Royal Rd., Springfield, VA 22161
ph: (800) 553-6847
fax: (703) 605-6900
email: orders@ntis.fedworld.gov
online ordering: http://www.ntis.gov/ordering.htm**

This document was printed on recycled paper.

PNNL-15782

STOMP
Subsurface Transport Over Multiple Phases

Version 4.0

User’s Guide

M. D. White
M. Oostrom

June 2006

Prepared for
the U.S. Department of Energy
under Contract DE-AC05-76RL01830

Pacific Northwest National Laboratory
Richland, Washington 99352

Preface

This guide describes the general use, input file formatting, compilation, and execution of the Subsurface Transport Over Multiple Phases (STOMP) simulator, a scientific tool for analyzing single and multiple phase subsurface flow and transport. A description of the simulator's governing equations, constitutive functions, and numerical solution algorithms are provided in a companion theory guide. In writing these guides for the STOMP simulator, the authors have assumed that the reader comprehends concepts and theories associated with multiple-phase hydrology, heat transfer, thermodynamics, radioactive chain decay, and relative permeability-saturation-capillary pressure constitutive relations. The authors further assume that the reader is familiar with the computing environment on which they plan to compile and execute the STOMP simulator.

Source codes for the sequential versions of the simulator are available in pure FORTRAN 77 or mixed FORTRAN 77/90 forms. The pure FORTRAN 77 source code form requires a *parameters* file to define the memory requirements for the array elements. The mixed FORTRAN 77/90 form of the source code uses dynamic memory allocation to define memory requirements, based on a FORTRAN 90 preprocessor STEP, that reads the input files.

The simulator utilizes a variable source code configuration, which allows the execution memory and speed to be tailored to the problem specifics, and essentially requires that the source code be assembled and compiled through a software maintenance utility. The memory requirements for executing the simulator depend on the complexity of physical system to be modeled and the size and dimensionality of the computational domain. Likewise, execution speed depends on the problem complexity, size and dimensionality of the computational domain, and computer performance.

Selected operational modes of the STOMP simulator are available for scalable execution on multiple processor (i.e., parallel) computers. These versions of the simulator are written in pure FORTRAN 90 with imbedded directives that are interpreted by a FORTRAN preprocessor. Without the preprocessor, the scalable version of the simulator can be executed sequentially on a single processor computer. The scalable versions of the STOMP modes carry the '-*Sc*' designator on the operational mode name. For example, *STOMP-WCS-Sc* is the scalable version of the *STOMP-WCS* (*Water-CO$_2$-Salt*) mode.

A separate mode containing an evaporation model as a boundary condition on the upper surface of the computation domain has also been included. This mode, *STOMP-WAE-B (Water-Air-Energy-Barriers)* can be viewed as an extension of the *STOMP-WAE (Water-Air-Energy)* mode. Details of this particular mode are outlined by Ward et al. (2005).

STOMP V4.0 includes the reactive transport module ECKEChem (Equilibrium-Conservation-Kinetic Equation Chemistry) for the *STOMP-W* (Water) and *STOMP-WCS* (*Water-CO$_2$-Salt*) modes. For this particular module, the '-*R*' designator is included in the operational mode name (e.g., *STOMP-W-R*, *STOMP-WCS-R-Sc*). This mode is described in detail by White and McGrail (2005).

For all operational modes and processor implementations, the memory requirements for executing the simulator depend on the complexity of physical system to be modeled and the size and dimensionality of the computational domain. Likewise, execution speed depends on the problem complexity, size and dimensionality of the computational domain, and computer performance.

Additional information about the simulator can be found on the STOMP webpage: http://stomp.pnl.gov. The website includes an introductory short course with problems ranging from simple one-dimensional saturated flow to complex multiphase system computations.

Summary

The STOMP (Subsurface Transport Over Multiple Phases) simulator has been developed by the Pacific Northwest National Laboratory for modeling subsurface flow and transport systems and remediation technologies. The development of the STOMP simulator started when the U.S. Department of Energy, through the Office of Technology Development, requested the demonstration of remediation technologies for the cleanup of volatile organic compounds and associated radionuclides within the soil and groundwater at arid sites. This demonstration program, called the VOC-Arid Soils Integrated Demonstration Program (Arid-ID), has been initially directed at a volume of unsaturated and saturated soil contaminated with carbon tetrachloride, on the Hanford Site near Richland, Washington. A principal subtask of the Arid-ID program involved the development of an integrated engineering simulator to evaluate the effectiveness and efficiency of various remediation technologies. The intended users of the engineering simulator include scientists and engineers who are investigating hydrologic and multifluid flow phenomena associated with remediation technologies. Principal design goals for the engineering simulator include broad applicability, verified algorithms, quality assurance controls, and validated simulations against laboratory and field-scale experiments. An important goal for the simulator development subtask involved the ability to scale laboratory and field-scale experiments to full-scale remediation technologies, and to transfer acquired technology to other arid sites.

The STOMP simulator's fundamental purpose is to produce numerical predictions of thermal and hydrogeologic flow and transport phenomena in variably saturated subsurface environments, which are contaminated with volatile or non-volatile organic compounds. The STOMP simulator is written in the FORTRAN 77 and 90 languages, following American National Standards Institute (ANSI) standards. The simulator utilizes a variable source code configuration, which allows the execution memory and speed be tailored to the problem specifics, and essentially requires that the source code be assembled and compiled through a software maintenance utility. Auxiliary applications include numerical predictions of solute transport processes including reactive transport. Quantitative predictions from the STOMP simulator are generated from the numerical solution of partial differential equations that describe subsurface environment transport phenomena. Description of the contaminated subsurface environment is founded on governing conservation equations and constitutive functions. Governing coupled flow equations are partial differential equations for the conservation of water mass, air mass, CO_2 mass, CH_4 mass, volatile organic compound mass, salt mass, and thermal energy. Constitutive functions relate primary variables to secondary variables. Solution of the governing partial differential equations occurs by the integral volume finite difference method. The governing equations that describe thermal and hydrogeological flow processes are solved simultaneously using Newton-Raphson iteration to resolve the non-linearities in the governing equations. Governing transport equations are partial differential equations for the conservation of solute mass. Solute mass conservation governing equations are solved sequentially, following the solution of the coupled flow equations.

In STOMP V4.0, a separate mode containing an evaporation model as a boundary condition on the upper surface of the computation domain has been included. This mode, *STOMP-WAE-B (Water-Air-Energy-Barriers)* can be viewed as an extension of the *STOMP-WAE (Water-Air-Energy)* mode. The extension provides the needed scientific tool to design and evaluate barriers. The model calculates water mass, air mass, and thermal energy across a boundary surface and water transport between the subsurface

and atmosphere. The STOMP-WAE-B addendum (Ward et al. 2005) provides a detailed description of this mode.

This version of STOMP includes the recently Pacific Northwest National Laboratory developed batch geochemistry solution module ECKEChem (Equilibrium-Conservation-Kinetic Equation Chemistry). The ECKEChem batch chemistry module was developed in a fashion that would allow its implementation into all operational modes of the STOMP simulator, making it a more versatile chemistry component. Additionally, this approach allows for verification of the ECKEChem module against more classical reactive transport problems involving aqueous systems. Currently, the ECKEChem package has been implemented in the *STOMP-W-R* and *STOMP-WCS-R* modes.

The fundamental objective in developing the ECKEChem module was to embody a systematic procedure for converting geochemical systems for mixed equilibrium and kinetic reactions into a system of non-linear equations. This objective has been realized through a recently developed general paradigm for modeling reactive chemicals in batch systems, which has been coded into a preprocessor for BIOGEOCHEM. To couple this processor to the STOMP simulator a conversion program, BioGeoChemTo, was written in Perl that reads the preprocessor output and converts it into STOMP simulator input format. Details of the ECKEChem module can be found in White and McGrail (2005).

A third major addition to this version of the simulator is the potential to conduct parallel simulations. These versions of the simulator are written in pure FORTRAN 90 with imbedded directives that are interpreted by a FORTRAN preprocessor.

Acknowledgments

This work was partly supported by the Remediation and Closure Science Project funded through the U.S. Department of Energy's (DOE's) Richland Operations Office. The continued development of the STOMP simulator in its sequential and parallel implementations has been funded by the Laboratory (PNNL) Directed Research and Development (LDRD) program at the PNNL. In particular, development of a scalable implementation has been funded through the Computational Science and Engineering Initiative and development of new operational modes for modeling carbon dioxide sequestration has been supported through the Carbon Management Initiative. The LDRD at the PNNL is a productive and efficient program that develops technical capabilities for solving complex technical problems that are important to DOE and to the nation. DOE Order 413.2A sets forth the DOE's LDRD policy and guidelines for DOE multiprogram laboratories and authorizes the national laboratories to allocate up to 6 percent of their operating budgets to fund the program.

The development of the STOMP-WAE-B (Water-Air-Energy-Barriers) mode was partially supported by Fluor Hanford, Inc. through its technology management in the Groundwater Remediation Project.

The development of STOMP-WCMSE (i.e., Water-CO2-CH4-Salt-Energy operational mode or STOMP-HYD) was partially supported by the Arctic Energy Technology Development Laboratory (AETDL).

Development, verification and documentation of the ECKEChem reactive transport module for the STOMP simulator by the Pacific Northwest National Laboratory (PNNL) was funded by DOE, National Energy Technology Laboratory (NETL), and Zero Emissions Research and Technology (ZERT) project.

Contents

Figures

Tables

1.0 Introduction

The initial development of the Subsurface Transport Over Multiple Phases (STOMP) simulator was supported by the VOC-Arid Soils Integration Demonstration Program, abbreviated as Arid-ID and funded by the U.S. Department of Energy (DOE), Office of Technology Development (OTD). The program was directed at the cleanup of volatile organic compounds and associated radionuclides and heavy metals in soils and groundwater at arid sites. The initial demonstration site is located within the 200 West Area on the Hanford Site near Richland, Washington. The site contains a volume of soil contaminated with the dense nonaqueous phase liquid (DNAPL) carbon tetrachloride, which includes approximately 200 vertical feet of contaminated unsaturated sediments underlying inactive disposal sites and overlying a 7-square-mile plume of contaminated groundwater. A critical component of the Arid-ID program involved assessing the impact of spatial heterogeneity of subsurface materials on remediation processes and evaluating the effectiveness and efficiency of demonstrated remedial technologies. Because of the complexity of subsurface flow and heat transport phenomena, these assessments and evaluations required complex numerical tools for their completion. Numerical tools allow scientists and engineers to integrate the current knowledge of contaminant behavior in the subsurface environment to predict and evaluate the performance of proposed remediation methods against established technologies.

A principal subtask of the Arid-ID program involved the development of an engineering simulator (a numerical tool), which is capable of numerically simulating proposed remediation processes. The design goals were that the engineering simulator would: (1) be accessible and exploitable to scientists and engineers familiar with subsurface environment phenomena, but not necessarily numerical modeling technicalities; (2) have enough general applicability to recruit a user group that is capable of supporting training, maintenance, and enhancement activities; (3) be verified by comparisons to analytical solutions and benchmarked against existing simulators; (4) be validated against germane laboratory and field experiments; and (5) have controlled configuration and documentation under an appropriate quality assurance program. An engineering simulator named STOMP, an acronym for Subsurface Transport Over Multiple Phases, has been developed by the Pacific Northwest National Laboratory, which achieves the five design goals described above. This document, the STOMP V4.0 User’s Guide, has been written to provide users of the STOMP simulator with necessary information for selecting an appropriate operational mode, understanding the code flow path and design, creating input files, dimensioning the executable, compiling and executing, and interpreting simulation outputs.

This guide has been organized in a manner that duplicates the normal approach to solving a subsurface flow and transport problem with the STOMP simulator. The STOMP simulator has been designed with a variable source code, where source code configurations are referred to as operational modes. Operational modes are classified according to the solved governing flow and transport equations, constitutive relation extensions, and implementation type (i.e., sequential vs. scalable). Therefore, prior to creating an input file or assembling the source code, the user must choose the appropriate operational mode for the particular subsurface system of interest. The selection of an operational mode requires that the user conceptualize the physical system as a computational system, which will always require making simplifying assumptions about the physical system. The complexity and execution speed of an operational mode is generally inversely related to the number of simplifying assumptions the user can justify about the physical system. Regardless of the operational mode complexity, the simulation of physical subsurface systems with the STOMP simulator always involves assumptions inherent to the founding governing equations and constitutive relations. The operational modes supported in STOMP V4.0 are:

STOMP-W (Water)
STOMP-W-R (Water w/ ECKEChem Reactive Transport)
STOMP-W-Sc (Water; Scalable)
STOMP-WA (Water-Air)
STOMP-WAE (Water-Air-Energy)
STOMP-WAE-B (Water-Air-Energy-Barriers)
STOMP-WAE-Sc (Water-Air-Energy; Scalable)
STOMP-WO (Water-Oil)
STOMP-WOA (Water-Oil-Air)
STOMP-WOA (Water-Oil-Air; Scalable)
STOMP-WOD (Water-Oil-Dissolved Oil)
STOMP-WOM (Water-Multicomponent Oil)
STOMP-WS (Water-Salt)
STOMP-WS-Sc (Water-Salt-Scalable)
STOMP-WAS (Water-Air-Salt)
STOMP-WASE (Water-Air-Salt-Energy)
STOMP-WCS (Water-CO_2-NaCl)
STOMP-WCS-R (Water-CO_2-NaCl w/ ECKEChem Reactive Transport)
STOMP-WCS-R-Sc (Water-CO_2-NaCl w/ ECKEChem Reactive Transport; Scalable)
STOMP-WCS-Sc (Water-CO_2-NaCl; Scalable)
STOMP-WCSE (Water-CO_2-NaCl-Energy)
STOMP-WCMSE (Water-CO_2-CH_4-NaCl-Energy)

Having chosen an operational mode, the user should then set about creating an *input* file. Input files, simply stated, are translations of the physical system description into a computational system. The user communicates information about the physical system to the STOMP simulator through the *input* file. The *input* file, operational mode, and linear system solver determine the source code configuration and required dimension of the executable. For those users with only access to a FORTRAN 77 compiler, the code must be configured in static memory form. In this form, the source code will include a *parameters* and a *commons* file. As FORTRAN 77 is unable to dynamically allocate memory, the user is responsible for editing the *parameters* file to define array dimensions to statically allocate memory during compilation. For users with access to a FORTRAN 90 compiler, the source code can be configured for dynamic memory allocation. In the dynamic memory form, both the *parameters* and *commons* files are replaced with a series of FORTRAN 90 modules, in the file *allo.f.* The next steps in the simulation process involve assembling the source code, creating an executable, and executing the code.

As previously mentioned, the STOMP simulator has a variable source code configuration, which is dependent on the selected operational mode. The source coding for each operational mode comprises mode specific coding and global coding from a library of routines. Assembly and compilation of the source coding for a particular operational mode has been facilitated through a UNIX utility called *make* (Talbott 1988). Instructions for using these utilities are described in this document. The STOMP simulator offers the user considerable flexibility in executing and controlling the execution of simulations. The user can create numerous execution periods within a single simulation, which differ with respect to time step advancement and convergence control. Controls of this sort are beneficial for simulations involving complex time-varying boundary conditions or sources. Moreover, the user can produce intermediate *restart* files which are essentially “snapshots” of the primary variables. Simulations can be restarted from intermediate *restart* files using the altered or unaltered descriptions of the computational system. Various techniques and approaches for executing the simulator are also described in this document.

The STOMP simulator generates results in a number of different formats, which can be controlled by the user. This guide describes these formats and the output capabilities of the simulator. Output formats include text files of simulation results and printing to the standard input/output device, which is typically a terminal or workstation screen. All *input* and *output* files generated by the simulator including *restart* files are ASCII text files, which can be read or modified by conventional text editors. Output data have been grouped in two basic formats, historic and snapshot. Historic data are records of selected variables over the simulation period. These data records are essential for creating graphs or plots showing the progression of variable values over simulation time. Snapshot data are records of selected variables at a particular moment in time over the computation domain. These data records are essential for generating images of variable values over the computational space. Snapshot data records can also be linked in a time sequence to show the evolution of variable values in both computational time and space. A series of Perl scripts for post-processing STOMP results have been developed. The scripts are discussed in Chapter 5.

Beyond providing operational type instructions, this document also contains useful information for executing the simulator with a debug utility or making modifications to the source code. The user may often be able to isolate input errors or convergence problems by tracing the execution of the simulator with a debug utility. To assist the user in performing debug executions or in making modifications to the code, critical information about the operational flow path, subroutine functions, and variable descriptions have been provided in this guide. The source coding for the STOMP simulator is suitably documented with comment statements, however, none of the variable or parameter names are defined nor is a general flow path for simulator outlined. Therefore, this guide provides essential information about subroutines and variables. Indexing of primary and secondary field variables and flux variables in the STOMP simulator is complex and depends on the operational mode. A description of the indexing patterns developed for the simulator is also provided in this guide.

2.0 Fundamentals

The STOMP simulator has been designed to solve a wide variety of non-linear, multiple-phase, flow and transport problems for variably saturated geologic media. Partial differential conservation equations for component mass, energy, and solute mass comprise the fundamental equations for the simulator. Coefficients within the fundamental equations are related to the primary variables through a set of constitutive relations. The conservation equations for component mass and energy are solved simultaneously, whereas the solute transport equations are solved sequentially after the coupled flow solution. The variable source code configuration allows the user to select the combination of solved fundamental equations. The current version of the STOMP simulator recognizes several coupled flow equation combinations (see Table 2.1). Some of the flow equation combinations allow for a scalable implementation. For these specific cases, the extension '-*Sc*' is added (e.g., *STOMP-W-Sc*). Each coupled flow equation combination includes the solution of a number of transported solutes. Reactive transport with the ECKEChem module (White and McGrail 2005) is currently an option for the *STOMP-W (Water)* and the *STOMP-WCS (Water-CO2-Salt)* modes. The names of the STOMP modes containing the ECKEChem modules have the extension '-*R*' attached (e.g., *STOMP-W-R, STOMP-WCS-R-Sc*). The extension '-*B*,' as used in the *STOMP-WAE-B (Water-Air-Energy-Barrier)* mode, indicates that the evapotranspiration model is implemented as a boundary condition in the upper surface of the computational domain. With these three extensions, the total number of recognized STOMP operational modes is 23 (Table 2.1). The associated constitutive relations for each operational mode are automatically incorporated into the source code as required.

In reading this document, it is important to distinguish between phases and components. Phases are composed of components. The terms aqueous, gas, hydrate, nonaqueous phase liquid (NAPL), and solid will be used exclusively in referring to phases. The terms water, air, CH_4, CO_2, oil, and salt refer to components. The aqueous phase is primarily composed of liquid water with lesser amounts of dissolved oil, CH_4, CO_2, air, and salt. The gas-phase composition can be highly variable, and may contain air, water vapor, and oil vapor. The hydrate phase is composed of CH_4, CO_2, and water. The NAPL phase is assumed to be composed of liquid oil only with no dissolved air or water. The current version of the STOMP simulator permits only a single oil component within the NAPL phase (i.e., a non-compositional model). Dissolved oil and transported solutes can sorb to the solid phase (i.e., the solid rock/soil matrix).

2.1 Scope and Output

The STOMP simulations are limited in application scope according to the solved fundamental equations, the associated constitutive theory, inherent assumptions, computer execution speed and memory, and the user’s creativity. A critical component to correct application of the STOMP simulator and comprehension of output results requires an understanding by the user of the assumptions taken to develop the various flow and transport algorithms. The simulator is capable of predicting flow and transport behavior for a variety of subsurface systems, however, application of the simulator to problems which violate an inherent assumption in the simulator’s design or fundamental equations could yield incorrect results. STOMP is principally limited to flow through variably saturated porous media, which can be characterized with an extended form of Darcy’s law. Additionally, the simulator is limited to a maximum of four immiscible phases referred to as the aqueous, non-aqueous liquid, hydrate, and gas phases. Low solubilities are assumed for the liquid phases and interphase mass transfer assumes

Table 2.1. Operational Mode Summary

Operational Mode with Options	Solved Coupled Equations	Primary Assumptions
STOMP-W; STOMP-W-Sc Water (STOMP1)	water mass	isothermal conditions passive gas phase no NAPL phase no dissolved oil no brine local thermodynamic equilibrium
STOMP-W-R Water (STOMP1)	water mass	isothermal conditions passive gas phase no NAPL phase no dissolved oil no brine local thermodynamic equilibrium ECKEChem transport
STOMP-WA Water-Air (STOMP2)	water mass air mass	isothermal conditions no NAPL phase no dissolved oil no brine local thermodynamic equilibrium
STOMP-WAE; STOMP-WAE-Sc Water-Air-Energy (STOMP3)	water mass air mass thermal energy	no NAPL phase no dissolved oil no brine local thermodynamic equilibrium
STOMP-WAE-B Water-Air-Energy-Barrier (STOMP3)	water mass air mass thermal energy	no NAPL phase no dissolved oil no brine local thermodynamic equilibrium inclusion of barrier extension
STOMP-WO Water-Oil (STOMP4)	water mass oil mass	isothermal conditions single component NAPL phase passive gas phase no brine local thermodynamic equilibrium
STOMP-WOM Water-Oil (STOMP4)	water mass oil mass	isothermal conditions multi-component NAPL phase passive gas phase no brine local thermodynamic equilibrium
STOMP-WOA; STOMP-WOA-Sc Water-Oil-Air (STOMP5)	water mass oil mass air mass	isothermal conditions single component NAPL phase no brine local thermodynamic equilibrium
STOMP-WOAE Water-Oil-Air-Energy (STOMP6)	water mass oil mass air mass thermal energy	single component NAPL phase no brine local thermodynamic equilibrium

Table 2.1. (contd)

Operational Mode with Options	Solved Coupled Equations	Primary Assumptions
STOMP-WOD Water-Oil-Dissolved Oil (STOMP8)	water mass oil mass dissolved oil mass	isothermal conditions single component NAPL phase passive gas phase no brine kinetic oil dissolution or local thermodynamic equilibrium
STOMP-WS; STOMP-WS-Sc Water-Salt (STOMP11)	water mass salt mass	isothermal conditions passive gas phase no NAPL phase no dissolved oil local thermodynamic equilibrium
STOMP-WAS Water-Air-Salt (STOMP12)	water mass air mass salt mass	isothermal conditions no NAPL phase no dissolved oil local thermodynamic equilibrium
STOMP-WASE Water-Air-Salt-Energy (STOMP13)	water mass air mass salt mass thermal energy	no NAPL phase no dissolved oil local thermodynamic equilibrium
STOMP-WCS; STOMP-WCS-R STOMP-WCS-R-Sc STOMP-WCS-Sc Water-CO_2-Salt (STOMP32)	water mass CO_2 mass NaCl mass	isothermal conditions no NAPL phase no dissolved oil local thermodynamic equilibrium
STOMP-WCSE Water-CO_2-Salt-Energy (STOMP33)	water mass CO_2 mass NaCl mass thermal energy	no NAPL phase no dissolved oil local thermodynamic equilibrium
STOMP-WCMSE Water-CO_2-CH_4-Salt-Energy (STOMP36)	water mass CO_2 mass NaCl mass thermal energy	no NAPL phase no dissolved oil local thermodynamic equilibrium

equilibrium conditions in most modes. The *STOMP-WOD (Water-Oil-Dissolved Oil)* Mode allows kinetic dissolution of the NAPL. Specific to the energy equation, the principal assumptions are that heat transport by gas-phase conduction and the kinetic nature of thermodynamic processes are neglected. Solute transport solutions are computed sequentially to the coupled flow equations. This approach requires the assumption that solutes are passive scalars with respect to the flow equations, which is equivalent to assuming solute concentrations are dilute. For the current implementation of ECKEChem (White and McGrail 2005), a non-iterative sequential scheme was chosen.

The STOMP simulator solves transient flow and transport problems in the subsurface environment in one, two, or three dimensions. Coordinate systems must be orthogonal and currently are limited to Cartesian, tilted Cartesian, and cylindrical, where the vertical coordinate of the cylindrical system must be aligned with the gravitational vector. The STOMP simulator solves steady-state problems either directly or through false-transients starting from a user specified initial state. Direct solutions to steady-state problems are only possible for initial conditions sufficiently close to the solution; therefore, transient solutions to steady-state conditions are the recommended approach. All boundary conditions, sources, and sinks are time variant and allow the user considerable control over transient simulations. Hydrogeologic properties can be spatially varied throughout the computational domain, within the resolution of a node volume. Hydraulic and thermal transport properties for the porous medium can be anisotropic, where the diagonal tensor elements are assumed to be aligned with the principal axes of the coordinate system. Coupled flow solutions can be obtained for selected one-, two-, or three-phase systems under isothermal or non-isothermal conditions. Solute transport can be coupled to all transport solutions with the assumption of dilute solute concentrations. A variety of boundary conditions are available for each operational mode, which may be applied selectively over the boundary surfaces. Unspecified boundary surfaces are always assumed to be zero flux surfaces. Inactive nodes can be specified by the user and boundary conditions can be applied to surfaces separating active (computational) and inactive (non-computational) nodes. Selected operational modes allow the user to invoke a dynamic domain option which temporarily sets quiescent nodes to an inactive set, thus removing them from the computational domain.

The fundamental coupled flow equations are solved following an integral volume finite-difference approach with the non-linearities in the discretized equations resolved through Newton-Raphson iteration. Linear systems which result from the Newton-Raphson linearization or the solute transport solution can be solved with a direct banded matrix solver and an indirect conjugate gradient based solver. These linear system solvers are commercially available software products that have been adapted for the particulars of the STOMP simulator. The STOMP simulator allows considerable control over simulation parameters related to convergence, time stepping, solution techniques, and execution limits. A single simulation can be divided into multiple execution periods, each with a different set of solution control parameters. Restart capabilities have also been included in the simulator, which can resume a simulation from user define points with or without alterations to input parameters. Output from the simulator can be totally controlled by the user and is written both to files and to the standard input/output device (e.g., screen). Output forms included time histories of selected variables, time “snapshots” of selected variables across the computational domain, and variable integrals for sources and fluxes across boundary and nodal surfaces.

The dominant non-linear functions within the STOMP simulator are the relative permeability-saturation-capillary pressure (*k–S–P*) relations. The STOMP simulator allows the user to specify these relations through a large variety of popular and classic functions. Two-phase (gas-aqueous) *k–S–P* relations can be specified with hysteretic or non-hysteretic functions or non-hysteretic tabular data. Entrapment of air with imbibing water conditions can be modeled with the hysteretic two-phase *k–S–P* functions. Two-phase *k–S–P* relations span both saturated and unsaturated conditions. The aqueous phase is assumed to never completely disappear through extensions to the *S–P* function below the residual saturation and a vapor pressure lowering scheme. Three-phase (aqueous-NAPL-gas) *k–S–P* relations can be specified with hysteretic or non-hysteretic functions. Entrapment of non-wetting fluids (i.e., gas in aqueous, gas in NAPL, and NAPL in aqueous) with imbibing wetting fluid conditions is the only hysteretic process that is currently supported with the hysteretic three-phase *k–S–P* functions. The wettability order for fluids in the STOMP simulator is assumed to follow the descending order aqueous to NAPL to gas. Three-phase *k–S–P* relations span both total-liquid saturated and unsaturated conditions and aqueous saturated and unsaturated conditions. Phase appearances and disappearances are possible for the gas and NAPL phases, however, the aqueous phase is assumed to never completely disappear through extensions to the *S–P* function below the residual saturation and a vapor pressure lowering scheme.

2.2 Operational Modes

The STOMP simulator operational modes do not comprise all of the possible combinations of coupled governing equations. The combination sets of coupled governing equations selected for inclusion in the list of operational modes represent those with the greatest utility for physical systems. For example, a two-phase non-volatile non-isothermal operational mode could be envisioned, which solved the water mass and energy conservation equations. This operational mode would invoke the assumption of a non-participating gas phase. Two-phase flow and transport through porous media under thermal gradients, however, strongly depends on the diffusion, dispersion, and advection transport through gas phase, even for low thermal gradients. Therefore, this operational mode while capable of functioning and producing converged solutions would have limited utility, because its associate premise that gas phase transport could be neglected.

Other combinations of governing equation sets with associated assumptions have considerable utility for specific systems or problems and have been coded, but have not been selected for inclusion in the current STOMP guide. For example, an operational mode has been created that solves the water mass, air mass, oil mass, and/or energy equation with the assumption that oil concentrations remain below the aqueous solubility limit. This operational mode differs from one that models the oil as a dilute solute, because the oil mass equation is solved simultaneously with the other coupled flow equations and fluid properties have a dependence on the oil concentration. This operational mode has utility in the investigation of the remediation of oil contaminants in dissolved plumes.

3.0 Code Design

The primary design guides for the STOMP simulator have been modularity, computational efficiency, and readability. Modular code architecture is beneficial because of the ease of reading, maintaining, and modifying the algorithms and is essential to the variable configuration source code. Computational efficiency refers to both memory requirements and execution speed. The STOMP simulator has been designed with a variable configuration source code which allows the memory requirements and code algorithms to be partially customized to the computational problem. This approach offers considerable advantages with respect to achieving a computationally efficient code design. Within this source code framework, however, many design choices have been made that affect computational efficiency. Algorithm design often offers options between memory and speed. For example, to lessen memory requirements a code designer may opt to repeatedly compute commonly used variables. Conversely, execution speed may be increased at the cost of increased memory requirements, by storing commonly used variables after their initial computation. Generally, the approach in the STOMP simulator has been to favor increased memory requirements to gain computational speed. This design approach has been chosen because of current state of computer architecture and capabilities. Because the STOMP simulator has been created as a scientific tool, algorithm readability has been a primary design guide. As a scientific tool, the simulator was expected to be a constantly changing package of software tools which could be applied to new or more complex problems. This design goal makes readability an essential feature of the code. Code readability has been achieved through an extensive use of comments, a modular design, a large group of common blocks, and minimal subroutine and function arguments.

3.1 Flow Path

The general flow path for all operational modes of the STOMP simulator comprises three components, initialization, iteration, and closure. A flow chart for the initialization, iteration, and closure components of the main program is shown in Figure 3.1, where the enclosing boxes indicate either a single or group of routines. The initialization component of the program is executed once during a simulation. The routines in the initialization component are executed in sequence shown in Figure 3.1, from the program start to the start of the first time step. The iteration component of the program contains a pair of nested loops, an outer loop for time stepping, and an inner loop for Newton-Raphson linearization. Termination of the Newton-Raphson loop occurs with a successful convergence or after an iteration limit violation.

Termination of the time-stepping loop occurs after a simulation limit or after a time-step reduction limit violation. Regardless of the cause for termination during the time-stepping loop, the closure routines are executed at the simulation completion. The equilibrium transport solution is shown as a single routine on the STOMP flow diagram. It comprises, however, several transport routines within a solute loop. The flow diagram for the standard solute transport portion of the iteration component is shown in Figure 3.2. The flow diagram for the ECKEChem transport module is shown in Figure 3.3.

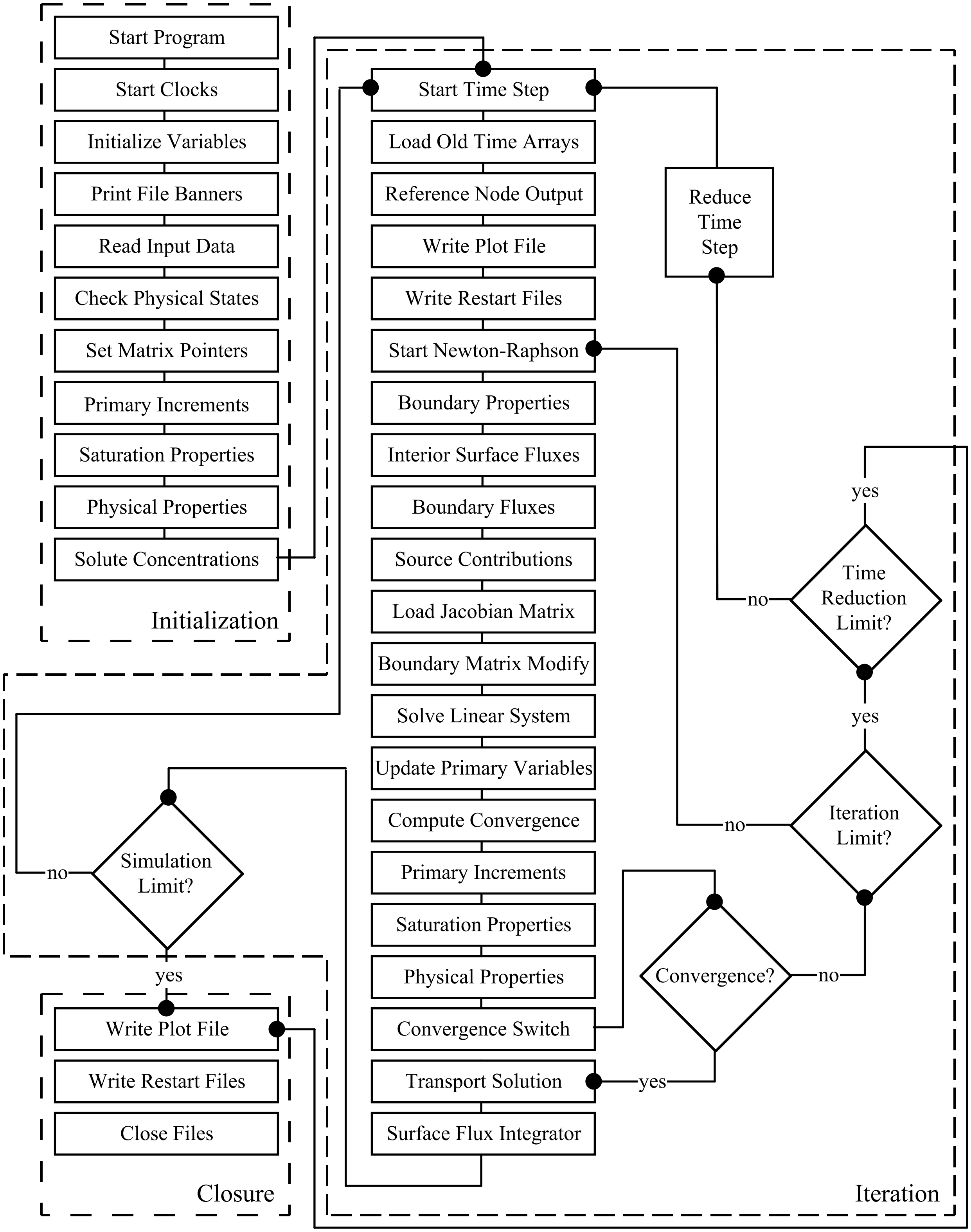

Figure 3.1. STOMP Flow Chart

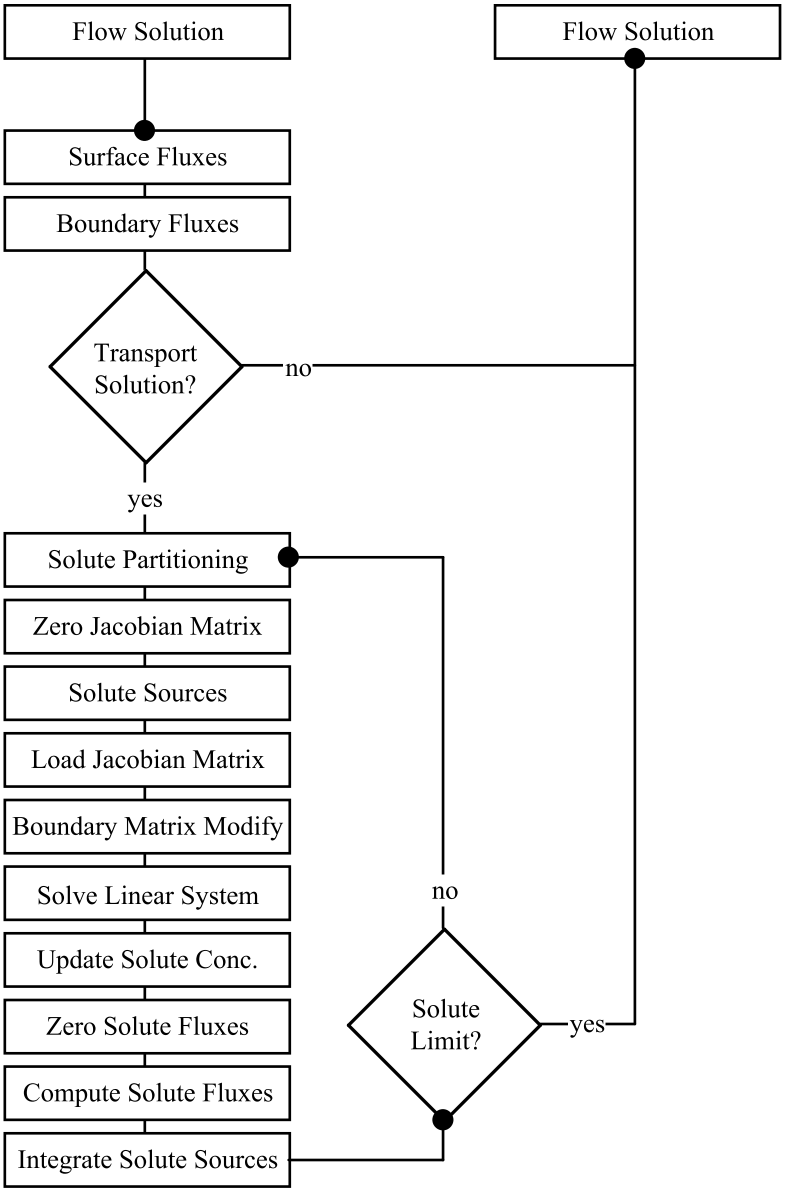

Figure 3.2. Transport Solution Flow Diagram

The initialization component (Initialize Variables) of the program flow path shown in Figure 3.1 begins by initializing all common variables. All common variables, those in the common blocks of the *commons* file, are initialized at this point either to zero or with default values. Unless specifically defaulted, integers are initialized to 0, real variables are initialized to 0.D+0, and character strings are initialized to blank. During the variable initializations, the *input* and *output* files are opened. The next routine (Print File Banners) prints the welcome statement, disclaimer, and banner to the standard output device (screen) and the *output* file. This is followed with the procedures (Read Input File) for reading the *input* file. The *input* file is read using a predefined card order. After each card has been read, the *input* file is rewound and searched from the beginning for the next card to be read. This approach allows the user to sequence input cards randomly within the *input* file. The card read order defined within the code software is critical and should not be altered. When appropriate the *restart* file is read for input data information during these procedures. Input data are then checked for saturation or thermodynamic state

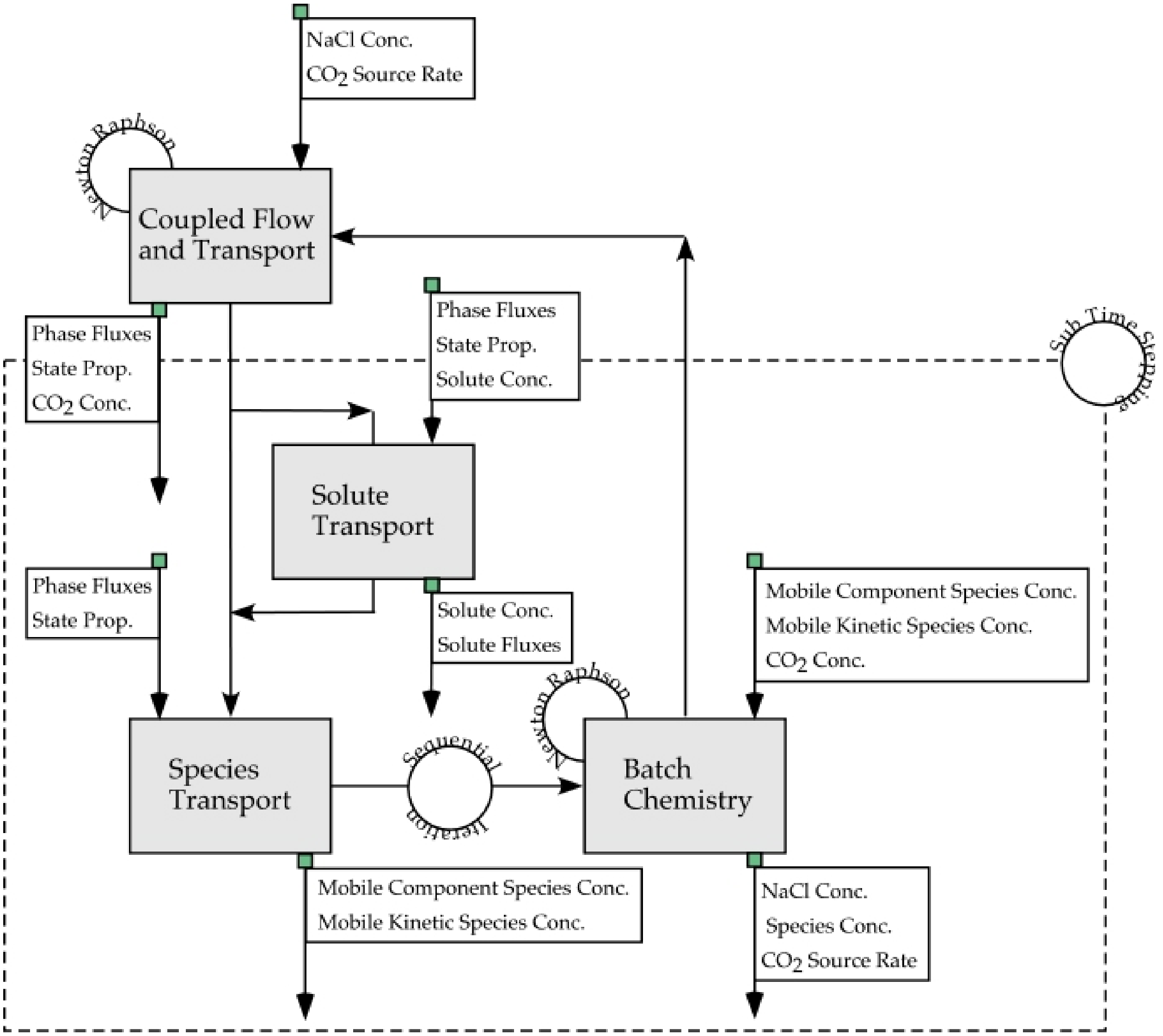

Figure 3.3. *STOMP-WCS* with ECKEChem Flow Path

consistency (Check Physical States). If an error is found in these routines the simulation will be terminated with an associated error message. If no errors in state conditions are noted then the program continues the initializing process with the Jacobian matrix pointers (Set Matrix Pointers).

The Jacobian matrix structure varies with operational mode and grid geometry. The reader is referred to the STOMP Theory Guide (White and Oostrom 2000) for a description of the numerical methods and linear system solvers. Jacobian matrix pointers are integer arrays that relate primary variables for a particular node to a location in the Jacobian matrix. If the simulation does not involve dynamic domains then these pointers are constants during the simulation and need to be computed only once. The next three initialization routines (Saturation Properties, Physical Properties and Solute Concentrations) compute initial values for the secondary variables from the initial conditions specified through the *input* or *restart* files. These routines have been divided into three components: those for computing phase saturation related variables, thermodynamic properties, and solute concentrations.

The iteration component of the program flow path contains a pair of nested loops. The outer loop time increments represent a single time step. The inner loop increments are iterations of the Newton-Raphson linearization. During a single time increment loop (time step), both the flow and transport governing equation sets are solved. The Newton-Raphson linearization loop is applicable only to the solution of the coupled mass and heat flow governing equations. Each time step loop starts with a computation of the new time step and increments to the time and time step counter. Time steps are computed with an algorithm based on the most previous complete time step, the time step acceleration factor, and the time to a transition point. Transition points occur with changes in execution period times, boundary condition times, source times, and output times. Time steps always conclude on transition points, which commonly require the time step to be temporarily reduced. The time step following a temporary reduction to meet a transition point time will resume the prior time stepping levels, except in the case of execution period transitions where a new initial or maximum time step has been declared. The next procedure (Load Old Time Arrays) involves loading the previous time step arrays for field variables. Field variables from the array location for the current field variable value are loaded into the array location for the previous time step field variable value. The next three routines (Reference Node Output, Write Plot File, and Write Restart File) prior to starting a Newton-Raphson iteration loop involves writing previous time step results to output. Simulation results are written to the *output* file, the standard input/output device (screen), one or more *plot.n* files, and a *restart.n* file, and optional *surface* files, depending on the directives made by the user on the Output Control and Surface Flux cards.

The Newton-Raphson iteration loop solves the coupled governing flow equations for component mass and energy. The first procedure (Boundary Properties) within this loop involves the calculation of boundary surface properties. Each declared boundary surface has associated field variables that are computed with the same algorithms as the node field variables, but only when the boundary condition is active. This approach eliminates the necessity for computing field variable values for disabled boundary surfaces. The next group of routines (Interior Fluxes) computes fluxes across interior surfaces (those surfaces between active nodes). Fluxes that are computed within these routines depend on the operational mode and include Darcy phase velocities, component diffusion-dispersion fluxes, thermal conductive flux, thermal advective flux, and thermal diffusion-dispersion fluxes. The same group of flux variables is computed in the next procedure (Boundary Fluxes) for each active boundary surface. As with the procedure for computing field variables on boundary surfaces the approach of computing flux variables only for active boundary surfaces eliminates computing unused boundary flux variables. The next procedure (Source Contributions) computes source and/or sink contributions to each of the governing flow equations from the user specified inputs on the Source Card. Source contributions are stored in arrays and subsequently used in computing the Jacobian matrix coefficients. At this point in the Newton-Raphson iteration loop, all field and surface flux variables have been computed, along with the source contributions. These variables compose the primary components of the governing flow equations.

The Jacobian matrix coefficients and solution vector are computed in a multiple stage sequence. The first stage involves setting all of the previous coefficient arrays to zero. This stage is necessary because the non-zero elements of the Jacobian matrix will change with time step as various boundary conditions transition between active and inactive states. The second stage (Load Jacobian Matrix) involves computing the Jacobian matrix and solution vector with the assumption of zero flux boundary conditions for all boundary surfaces, including surfaces between active and inactive nodes. The Jacobian matrix loading procedure depends on the operational mode but is sequenced according to governing partial differential equations. The resulting system of equations represents the discretized and linearized system of governing flow equations with zero flux boundary conditions imposed, where the source contributions have been incorporated. The final stage (Boundary Matrix Modify) modifies this linear system according to the active user imposed boundary conditions. Boundary conditions will alter both the coefficient matrix and solution vector. With the Jacobian matrix and solution vector elements computed, the next procedure (Solve Linear System) involves solving the linear system of equations. The linear system is solved either with a direct banded matrix solver or an iterative conjugate gradient solver. Both routines return corrections to the primary variables in the solution vector array.

Corrections to the primary variables, computed from the linear system solvers, are used to update the primary variables and determine convergence. The Newton-Raphson procedure computes corrections to the primary variable set within each iteration. The starting values for primary variables for each new time step are the previous time step values of the primary variables, as these values represent reasonable estimates of the future values. For a convergent iteration scheme, each successive iteration yields diminishing corrections to the primary variables. Phase transitions and primary variable switching schemes, however, can yield temporary increases in the correction to a particular primary variable. The two procedures (Update Primary Variables and Compute Convergence) that immediately follow the linear system solver procedures update the primary variables and determine convergence. Convergence occurs if the normalized values of the primary variable corrections for all unknowns falls below a user defined value (typically 1.0×10^{-6}). The next three (Primary Increments, Saturation Properties, and Physical Properties) are executed independent of the convergence result. If convergence occurs (Convergence Switch), then these computed values represent the current values of secondary variables at the conclusion of the time step; otherwise they represent the current iterate values of the secondary variables. If convergence occurs then the solution procedure continues with solute transport procedures. At the conclusion of non-convergent iterations two additional checks are made. If the iteration count does not exceed the user specified limit then program proceeds with a new Newton-Raphson iteration loop. If the iteration count exceeds the limit (Iteration Limit), then a check is made on the count of successive time step reductions. If convergence has failed and resulted in a time step reduction four times in succession (Time Reduction Limit), then the simulation aborts and program execution is transferred to the closure routines. Otherwise, the time step is reduced (Reduce Time Step), the program execution is transferred to the beginning of the time increment loop, and another attempt is made reach a converged solution for the time step.

The standard transport solution procedure follows the iterative solution of the mass and heat flow equations and, although a direct solution scheme, involves looping over the number of solutes. A procedure flow diagram for the transport solution routines is shown in Figure 3.2. Prior to entering the solute loop the interior-surface and boundary-surface flux procedures are called to obtain values of all flux variables at the conclusion of a time step. Flux values at the conclusion of a time step will typically vary from those computed during the last iteration of the current time step, because the primary and secondary variables will have been updated near the bottom of the last Newton-Raphson iteration loop. The surface flux calculations within the Newton-Raphson loop differ from these surface calculations in that only the current unincremented value of the surface flux is computed. Refer to Section 3.3 for a description of surface flux variable arrays.

The transport solution procedure loops over the number of solutes in the reverse order that they are defined on the Solute/Fluid Interactions Card. A reverse looping order is used to compute progeny solutes prior to computing parent solutes. This approach allows sequential coupling between solutes that decay radioactively or chemically with first order reaction rates yielding solute products. The first step

(Solute Partitioning) of the transport solution loop involves computing the equilibrium distribution of solute between the fluid and solid phases. The second step (Zero Jacobian Matrix) is to initialize the coefficient matrix and solution vector elements to zero. As with the flow solution scheme, the possibility for boundary conditions and sources that transition makes initializing the linear system elements mandatory. The third step (Solute Sources) of the transport solution loop is to compute the solute source and/or sink contributions. Solute source contributions are incorporated directly into the coefficient matrix and solution vector elements. The fourth step (Load Jacobian Matrix) involves loading the coefficient matrix and solution vector. As with the flow solution scheme, the coefficient matrix and solution vector elements are computed assuming zero-flux conditions on all boundary surfaces, then modified for the boundary conditions (Boundary Matrix Modify). Element loading depends on the operational mode and occurs sequentially by phases, where the aqueous phase contributions are loaded first followed by the gas and NAPL phases. Solid phase contributions are loaded with the first active fluid phase. The fifth step (Solve Linear System) of the transport solution procedure is the solution of linear system of equations. Algorithms for solving the transport linear system of equations are identical to those for the flow solution, except that the returned results for the transport solution are the volumetric solute concentrations. The sixth step (Update Solute Conc.) of the transport solution is to update the solute concentrations with their newly computed values. The concluding steps (Zero Solute Fluxes, Compute Solute Fluxes and Integrate Solute Sources) for the transport solution compute solute fluxes and integrate the solute sources. Once the transport solution loop has been executed for every solute the program execution returns to the flow solution procedures. Time steps for the transport solution are by default equal to those used for the flow solution.

The closure routines are only executed once at the successful or unsuccessful conclusion of a simulation. These routines generate final *plot.n* files, *restart.n* files, and close all opened files. Upon successful conclusion to the simulation, the final *restart.n* file will contain a record of the primary variables at the conclusion of the final time step. Conversely, if a simulation concludes unsuccessfully, because of a convergence failure or otherwise, the *restart.n* file will contain a record of the primary variables at the conclusion of the previous converged time step.

An example flow path for solving multifluid subsurface flow and reactive transport problems with the STOMP simulator and ECKEChem module (White and McGrail 2005) is shown in Figure 3.3 for the coupled STOMP-WCS and ECKEChem modules. Reactive transport capabilities of the ECKEChem module have been designed to work with all operational modes of the STOMP simulator, without significantly altering the STOMP coding. A fundamental objective in developing the ECKEChem module was to create a systematic procedure for converting geochemical systems into a system of non-linear equations. This objective was realized through a recently developed general paradigm for modeling reactive chemicals in batch systems (Fang et al. 2003), which has been coded into a preprocessor for BIOGEOCHEM. A conversion program BioGeoChemTo was written in Perl to read the preprocessor's output and covert it into STOMP simulator input format. As the numerical solution approach for the reactive transport system uses an operator splitting scheme, which sequentially computes species transport and batch chemistry, the original solute transport schemes of the STOMP simulator can be used for transporting reactive species with only minor modifications. A time step begins with the solution of the coupled flow system, using Newton Raphson iteration, yielding aqueous and gas phase fluxes, and state properties (e.g., temperature, pressure, phase viscosity, phase density, phase saturation). If transport of non-reactive dilute solutes is specified for the simulation, then solute transport is computed with the option for sub-time stepping as a function of the maximum Courant number. Inputs to the solute transport solver include aqueous and gas phase fluxes, state properties, and the previous time-step or sub-time-step solute concentrations. Multiple solutes are transported sequentially. Using the operator splitting scheme, the reactive transport solution begins with transport of all the mobile total-component and total-kinetic species. The resulting concentrations of the mobile total-component and total-kinetic species are combined with their immobile counterparts to yield concentrations of the total-component and total-kinetic species. These species concentrations are then passed to the batch chemistry solver, which uses an iterative search scheme to generate initial estimates of the species concentrations for the Newton-Raphson iterative solver.

3.2 Subroutines

Each operational mode of the STOMP simulator comprises global and mode dependent subroutines. Global subroutines are those subroutines that are generally included in more than one operational mode. Mode dependent subroutines, however, are associated with a single operational mode. Subroutine names are generally descriptive abbreviations. These abbreviations frequently contain the letters G, L, or N, which respectively correspond to the gas, aqueous, or NAPL phases. Other common letters are A, O, and W, which represent air or CO_2, oil, and water components, respectively. The letters C and T often refer to the solute transport and energy equations. A bold letter **B** refers to a group of similar subroutines written for the bottom, south, west, east, north, and top surfaces, respectively. Each surface will have a corresponding subroutine with the subroutine name having a B, S, W, E, N, or T in place of the bold letter **B** to represent the bottom, south, west, east, north, or top surface, respectively. A short description of a subroutine’s function can generally be found in the heading portion of the source coding. Mode dependent subroutine names contain a numerical suffix, which corresponds to the operational mode. Refer to Table 3.1 for operational mode index definitions. Every subroutine and function begins with a series of comment statements, which include disclaimers, a short description of the coding, a creation date stamp, and a modification date stamp. The modification date stamp is used to track the last modification date for a particular subroutine. The STOMP simulator variable structure was primarily designed around large arrays held in common blocks, which are included in nearly every subroutine. Therefore, subroutines generally have few arguments.

3.3 Variables

The STOMP simulator has been designed with the principal variables defined in common blocks, which are included in nearly all subroutines. The common variables are listed in Table 3.2. This approach reduces the number of arguments that are passed between routines and increases the readability of the code, because variable names remain unchanged between routines. The STOMP simulator has been coded without equivalence statements and no variables are temporarily overwritten to save memory. Variable names generally follow the intrinsic protocols for FORTRAN 77, where integer variables begin with letters in the range 'I-N' and real valued variables begin with letters in the ranges 'A-H' and 'O-Z.' Variables that begin with the letter 'L' generally indicate integer parameters.

Field variables are defined at node points and are represented with two-dimensional arrays. Examples of field variables include primary unknowns, saturation properties, rock/soil properties, and thermodynamic properties (e.g., temperature, pressure, phase saturation, porosity, density, and viscosity). The first index of a field variable indicates the time step or increment status. The second index of a field variable indicates the node number. The dimension of the first index depends on the operational mode or

Table 3.1. Glossary of Operational Mode Indices

STOMP Mode Index	Operational Mode
1	STOMP-W; W-R; W-Sc
2	STOMP-WA
3	STOMP-WAE; WAE-B; WAE-Sc
4	STOMP-WO; WOM
5	STOMP-WOA; WOA-Sc
6	STOMP-WOAE
8	STOMP-WOD
11	STOMP-WS; WS-Sc
12	STOMP-WAS
13	STOMP-WASE
32	STOMP-WCS; WCS-R; WCS-R-Sc; WCS-Sc
33	STOMP-WCSE
36	STOMP-WCMSE

W = Water.
O = Oil.
A = Air.
E = Energy.
S = Salt.
C = CO_2.
M = CH_4.
R = Reactive.
Sc = Scalable.	

equivalently number of coupled flow equations; where the dimension will equal the number of unknowns plus two. Therefore, field variables at each node comprise a number of elements equal to the number of unknowns plus two. Each element for each field variable contains a variant on the field variable. For example, the first element or index contains the value of the field variable at the previous time step. A list of field variable variants is shown in Table 3.3.

The primary variable order and corresponding field variable variant indexing depend on the operational mode. However, primary variables are always ordered, regardless of the operational mode, according to the following equation sequence: energy, water mass, methane mass, air (or CO_2) mass, oil mass, dissolved-oil mass, and salt mass. The primary variable ordering system for each operational mode is listed in Table 3.4. Primary variables are referred to by equation in Table 3.4, because primary variables for a given equation and operational mode are dependent on the local phase condition. Refer to the STOMP Theory Guide (White and Oostrom 2000) for a description of primary variables, phase conditions, and variable switching. Consider an example for the *STOMP-WA (Water-Air)* Operational Mode, which solves the water mass and air mass conservation equations. For this operational mode, there would be four field variable variant indices. Using the aqueous saturation for an unsaturated node as the example field variable, the following variant definitions would apply: the first index would refer to the aqueous saturation at the previous time step, the second index would refer to the aqueous saturation at the current iteration or time step value, the third index would refer to the aqueous saturation with the aqueous pressure incremented, and the fourth index would refer to the aqueous saturation with the gas pressure incremented.

Flux variables are defined on node surfaces between node points or on boundary surfaces and are represented with two dimensional arrays. Examples of flux variables include heat fluxes, Darcy phase velocities, and component diffusion/dispersion fluxes. Flux variables are aligned with one of the primary orthogonal coordinate directions. Flux variables names that begin with the character 'U' are aligned with the x- or radial-direction coordinate, those which begin with the character 'V' are aligned with the y- or azimuthal-direction coordinate, and those which begin with the character 'W' are aligned with the z-direction coordinate. The first index of a flux variable indicates the increment status and the second index indicates the surface number. Unlike field variables, previous time-step fluxes are not stored. Surface numbers do not correspond directly to node numbers. The number of surfaces in any given coordinate direction equals the number of nodes in that direction plus one.

A list of the flux variable variants is shown in Table 3.5. The primary variable order and corresponding field variable variant index depend on the operational mode. In Table 3.5, the upper node refers to the node in the east, north, or top direction (positive *x*-, *y*-, or *z*-direction) with respect to the surface. Likewise, the lower node refers to the node in the west, south, or bottom direction (negative *x*-, *y*-, or *z*-direction) with respect to the surface. Consider an example for the *STOMP-WA (Water-Air)* Operational Mode, which solves the water mass and air mass conservation equations. For this operational mode, there would be five flux variable variant indices. Using the z-direction aqueous Darcy velocity for an unsaturated node as the example field variable, the following variant definitions would apply: the first index would refer to the aqueous Darcy velocity for the current iteration or time step value, the second and third indices would refer to the Darcy velocity with the aqueous pressure in the “top” and “bottom” nodes incremented, respectively; and the fourth and fifth indices would refer to the Darcy velocity with the gas pressure in the “top” and “bottom” nodes incremented, respectively.

Table 3.2. Description of Global Variables (from *commons* and *allo.F*)

Variable	Description
ACTVC	Constant Activity Coefficient for ECKEChem
AFX	X-Direction Node Area, m^2
AFY	Y-Direction Node Area, m^2
AFZ	Z-Direction Node Area, m^2
AGLF	Interfacial Area between Free Gas and Aqueous, m^2
AGLT	Interfacial Area between Trapped Gas and Aqueous, m^2
ALBEDO(6,:)	Soil Albedo Function Parameters
	(1,:) Dry-Soil Albedo
	(2,:) Wet-Soil Albedo
	(3,:) Albedo Attenuation Factor
	(4,:) Albedo @ Solar Zenith = 60°
	(5,:) Briegleb Albedo Model Parameter C
ALU	Jacobian Coefficient Matrix
ANLF	Specific Aqueous-Free NAPL Interfacial Area, 1/m
ANLT	Specific Aqueous-Trapped NAPL Interfacial Area, 1/m
ASGT	Apparent Trapped Gas Saturation
ASGTL	Apparent Trapped Gas Saturation, Aqueous
ASGTN	Apparent Trapped Gas Saturation, NAPL
ASL	Apparent Aqueous Saturation
ASLMIN	Minimum Apparent Aqueous Saturation
ASLSC	Apparent Aqueous Saturation at Reversal Point
ASNMIN	Minimum Apparent NAPL Saturation
ASNR	Apparent Residual NAPL Saturation
ASNT	Apparent Trapped NAPL Saturation
AST	Apparent Total-Liquid Saturation
ASTMAX	Maximum Apparent Total-Liquid Saturation
ASTMIN	Minimum Apparent Total-Liquid Saturation
ATMC(5)	Atmospheric Condition Measurement Parameters
	(1) Wind Speed Measurement Height, m
	(2) Air Temp./Relative Humidity Measurement Height, m
	(3) Local Longitude, deg
	(4) Local Latitude, deg
	(5) Local Meridian, deg
ATMOS(6,:)	Atmospheric Condition Parameters
	(1,:) Time, s
	(2,:) Temperature, C
	(3,:) Pressure, Pa
	(4,:) Relative Humidity, %
	(5,:) Incident Solar Radiation, W/m^2
	(6,:) Wind Speed, m/s
ATMST	Atmospheric Conditions Data Start Time, s
BC	Boundary Condition Variables

Table 3.2. (contd)

Variable	Description
BCG	LUGR Subgrid Boundary Condition Variables
BGL	Gas-Aqueous Scaling Factor
BGN	Gas-NAPL Scaling Factor
BNL	NAPL-Aqueous Scaling
BIG	Constant, 10^{20}
BIL	Ice-Aqueous Scaling Factor
BLU	Solution Vector
BNL	NAPL-Aqueous Scaling Factor
BTGL	Gas-Aqueous Scaling Factor
BTGLB	Gas-Aqueous Scaling Factor at Boundary
C	Solute Concentration, $1/m^3$
C_OBDT	Field-Observation Output Units* (64 characters)
CA_GL	Gas-Aqueous Contact Angle, deg
CA_GN	Gas-NAPL Contact Angle, deg
CA_NL	NAPL-Aqueous Contact Angle, deg
CARD	Card Name* (64 characters)
CB	Boundary Solute Concentration, $1/m^3$
CBO	Old Time Step Boundary Solute Concentration, $1/m^3$
CCL_CRN	Aqueous Cut-Off Concentration for Courant-Limit Control
CH_VRSN	User's Guide Version Number
CHDATE	Date* (10 characters)
CHDF	Solute Chain Decay Fraction
CHFDR	Hydrate Formation-Dissociation Rate Constant, 1/s
CHMER	Hydrate Molecular Exchange Rate Constant, 1/s
CHML	Characteristic Matrix Length, m
CHMSG	Error Message* (132 characters)
CHREF	Reference Node Variable Abbreviation* (4 characters)
CHSF	Surface Flux Variable Abbreviation* (4 characters)
CHTIME	Time* (8 characters)
CIMTC	Oil Interphase Mass Transfer Coefficients
CLKMX	Maximum Clock Time, s
CLKSEC	Clock Time, s
CLU	Solution Vector
CMP(3,:)	Rock/Soil Compressibility Parameters
	(1,:) Matrix Compressibility, 1/Pa
	(2,:) Fracture Compressibility, 1/Pa
	(3,:) Compressibility Reference Pressure, Pa
CMPNY	Company Name*
CMTLN	NAPL-Aqueous Interphase Mass Transfer Coefficients
CNL	Solute Concentration in NAPL, $1/m^3$
CNLO	Old Time Step Solute Concentration in NAPL, $1/m^3$
CO	Old-Time-Step Solute Concentration, $1/m^3$

Table 3.2. (contd)

Variable	Description
CPAC(4)	Isobaric Molar Specific Heat Constants
	(1) Constant a
	(2) Constant b
	(3) Constant c
	(4) Constant d
CPLY_RL	Aqueous Relative Permeability Polynomial Coefficients
CPLY_SL	Aqueous Saturation Polynomial Coefficients
CPOC	Oil Isobaric Molar Specific Heat Constants
CPS	Rock/Soil Specific Heat, J/kg K
CPUMX	Maximum CPU Time, s
CPUSEC	CPU Time, s
CRNTG	Gas Courant Number
CRNTL	Aqueous Courant Number
CRNTMX	Courant Number Upper Limit
CRNTN	NAPL Courant Number
CVS_ID	CVS Identification
DBLTIM	Mass Balance Times, s
DFEF(5,:)	Water Vapor Enhanced Diffusion Function Coefficients
	(1,:) Parameter A
	(2,:) Parameter B
	(3,:) Parameter C
	(4,:) Parameter D
	(5,:) Parameter E
DFGA	CO_2 Vapor Diffusion Coefficient, m^2/s
DFGAB	Boundary Gas Air Diffusion Coefficient, m^2/s
DFGAC	CO_2 Vapor Diffusion Coefficient, m^2/s
DFGO	Gas Oil Diffusion Coefficient, m^2/s
DFGOB	Boundary Gas Oil Diffusion Coefficient, m^2/s
DFGOC	Oil Vapor Diffusion Coefficient, m^2/s
DFGW	Gas Water Diffusion Coefficient, m^2/s
DFGWB	Boundary Gas Water Diffusion Coefficient, m^2/s
DFGWC	Water-Vapor Diffusion Coefficient, m^2/s
DFGWW	Well Water-Vapor Diffusion coefficient, m^2/s
DFGWWB	Well Boundary Water-Vapor Diffusion Coefficient, m^2/s
DFLA	Aqueous Air Diffusion Coefficient, m^2/s
DFLAB	Boundary Aqueous Air Diffusion Coefficient, m^2/s
DFLAC	Dissolved Air Diffusion Coefficient, m^2/s
DFLAW	Well Dissolved Air Diffusion Coefficient, m^2/s
DFLAWB	Well Boundary Dissolved Air Diffusion Coefficient, m^2/s
DFLO	Aqueous Oil Diffusion Coefficient, m^2/s
DFLOB	Boundary Aqueous Oil Diffusion Coefficient, m^2/s
DFLOC	Dissolved Oil Diffusion Coefficient, m^2/s

Table 3.2. (contd)

Variable	Description
DFLOW	Well Dissolved Oil Diffusion Coefficient, m^2/s
DFLOWB	Well Boundary Dissolved Oil Diffusion Coefficient, m^2/s
DFLS	Aqueous Salt Diffusion Coefficient, m^2/s
DFLSB	Bound. Aqueous Salt Diffusion Coefficient, m^2/s
DFLSC	Dissolved Salt Diffusion Coefficient, m^2/s
DFNA	Alcohol Diffusion Coefficient in NAPL, m^2/s
DFNAB	Boundary Alcohol Diffusion Coefficient in NAPL, m^2/s
DFNW	Water Diffusion Coefficient in NAPL, m^2/s
DFNWB	Boundary Water Diffusion Coefficient in NAPL, m^2/s
DISPL	Longitudinal Dispersivity, m
DISPT	Transverse Dispersivity, m
DLU	Jacobian Coefficients Matrix for SPLIB
DNR	Primary Variable Increments
DNRW	Well Primary Variable Increments
DPLD	Dissolved Oil Longitudinal Dispersivity, m
DPLGS	Salt Longitudinal Dispersivity, m
DPMA	Alcohol Dipole Moment, Debye
DPMO	Oil Dipole Moment, Debye
DPMW	Water Dipole Moment, Debye
DPTD	Dissolved Oil Transverse Dispersivity, m
DPTRS	Salt Transverse Dispersivity, m
DRLTIM	Release Plane Times, s
DRMEFF	Remediation Efficiency
DRMTIM	Remediation Time, s
DT	Time Step, s
DT_CRN	Courant-Limit Time Step, s
DTAF	Time Step Acceleration Factor
DTCF	Time Step Cut Factor
DTI	Inverse Time Step, 1/s
DTI_CRN	Courant-Limit Inverse Time Step, 1/s
DTMX	Maximum Time Step, s
DTO	Old Time Step, s
DTSO	Old Time Step for 2nd Order Time Differencing, s
DXGF	X-Direction Grid Cell Dimension, m
DXGP	X-Direction Internodal Dimension, m
DXLG	X-Direction Grid Cell Dimension, m
DYGF	Y-Direction Grid Cell Dimension, m
DYGP	Y-Direction Internodal Dimension, m
DYLG	Y-Direction Grid Cell Dimension, m
DZGF	Z-Direction Grid Cell Dimension, m
DZGP	Z-Direction Internodal Dimension, m
DZLG	Z-Direction Grid Cell Dimension, m

Table 3.2. (contd)

Variable	Description
ELC_DCF	Electrolyte Density Function Parameters
ELC_DUN	Electrolyte Density Function Parameter
ELC_SOL	Electrolyte Name* (64 characters)
ELC_VCF	Electrolyte Viscosity Function Parameters
ELC_VUN	Electrolyte Viscosity Function Parameter
EPSL	Machine Precision, 10^{-14}
EQ_C	Stoichometric Coefficient for the Conservation Equation
EQ_E	Stoichometric Coefficient for the Equilibrium Equation
EQ_K	Stoichometric Coefficient for the Kinetic Equation
FLG_EXT	Logical Flag for Creating an ".ext" File for UCODE
FLG_UNI	Logical Flag for Creating an ".uni" File for UCODE
FNBL	Balance File Name* (64 characters)
FNPL	Plot File Name* (64 characters)
FNRC	Recharge File Name* (64 characters)
FNRD	Input File Name* (64 characters)
FNRL	Release File Name* (64 characters)
FNRS	Restart File Name* (64 characters)
FNSF	Surface Flux File Name* (64 characters)
FNSR	Screen Unit Name* (64 characters)
FNVD	Remediate File Name* (64 characters)
FNWR	Output File Name* (64 characters)
FUG_LV	Fugacity on the Liquid-Vapor Saturation Line, Pa
FUG_TA	CO_2 Fugacity Coefficient Table
GAMMA(13,:)	Scaling Factor Parameters
	(1,:) Matrix Permeability
	(2,:) Matrix Porosity
	(3,:) Saturation/Relative Permeability Functions Parameter
	(4,:) Saturation/Relative Permeability Functions Parameter
	(5,:) Saturation/Relative Permeability Functions Parameter
	(6,:) Fracture Permeability
	(7,:) Fracture Porosity
	(8,:) Saturation/Relative Permeability Functions Parameter
	(9,:) Saturation/Relative Permeability Functions Parameter
	(10,:) Saturation/Relative Permeability Functions Parameter
	(11,:) Mualem Anisotropy Parameter
	(12,:) Mualem Anisotropy Parameter
	(13,:) Mualem Anisotropy Parameter
GPI	Constant, 3.1415926536
GRAV	Acceleration of Gravity, m/s^2
GRAVX	X-Direction Acceleration of Gravity, m/s^2
GRAVY	Y-Direction Acceleration of Gravity, m/s^2
GRAVZ	Z-Direction Acceleration of Gravity, m/s^2

Table 3.2. (contd)

Variable	Description
GRVPX	X-Dir. Internodal Acceleration of Gravity, m/s^2
GRVPY	Y-Dir. Internodal Acceleration of Gravity, m/s^2
GRVPZ	Z-Dir. Internodal Acceleration of Gravity, m/s^2
GRVX	X-Dir. Nodal Acceleration of Gravity, m/s^2
GRVY	Y-Dir. Nodal Acceleration of Gravity, m/s^2
GRVZ	Z-Dir. Nodal Acceleration of Gravity, m/s^2
H_TA	CO_2 Enthalpy Table, J/kg
HCAW	Henry's Constant Air-Water, Pa
HCMWE	Capillary Pressure Head for Water Entry (Triple-Curve), m
HCOW	Henry's Constant Oil-Water, Pa
HDOD	Oven-dried Capillary Head, m
HDSC	Apparent Capillary Head at Reversal Point, m
HG	Gas Enthalpy, J/kg
HGA	Air Enthalpy, J/kg
HGAB	Boundary Air Enthalpy, J/kg
HGB	Boundary Gas Enthalpy, J/kg
HGO	Oil Vapor Enthalpy, J/kg
HGOB	Boundary Oil Vapor Enthalpy, J/kg
HGW	Water Vapor Enthalpy, J/kg
HGWB	Boundary Water Vapor Enthalpy, J/kg
HH	Hydrate Enthalpy, J/kg
HH_O	Old Hydrate Enthalpy, J/kg
HHB	Boundary Hydrate Enthalpy, J/kg
HI	Ice Enthalpy, J/kg
HI_O	Old Ice Enthalpy, J/kg
HIB	Boundary Ice Enthalpy, J/kg
HKL	Interfacial Mass Transfer Coefficient in Aqueous Phase, m^2/s
HKNF	Interfacial Mass Transfer Coefficient in Free NAPL, m^2/s
HKNT	Interfacial Mass Transfer Coefficient in Trapped NAPL, m^2/s
HL	Aqueous Enthalpy, J/kg
HL_LV	Liquid Enthalpy on the Liquid-Vapor Saturation Line, Pa
HL_O	Old Aqueous Enthalpy, J/kg
HLB	Boundary Aqueous Enthalpy, J/kg
HLF	Solute Half-Life, s
HLW_P	Enthalpy of Liquid Water on Plant, J/kg
HN	NAPL Enthalpy, J/kg
HN_O	Old NAPL Enthalpy, J/kg
HNB	Boundary NAPL Enthalpy, J/kg
HSP	Precipitated Salt Enthalpy, J/kg
HSP_O	Old Precipitated Salt Enthalpy, J/kg
HV_LV	Vapor Enthalpy on the Liquid-Vapor Saturation Line, Pa
I_LV	Number of Points in Liquid-Vapor Saturation Line

Table 3.2. (contd)

Variable	Description
I_OBDT(9,:)	Index for Field Observation Data
	(1,:) Observation Data Type
	(2,:) Observation Data Variable
	(3,:) Observed Data Statistical Index
	(4,:) Observed Data I Index
	(5,:) Observed Data J Index
	(6,:) Observed Data K Index
	(7,:) Observed Data I Index
	(8,:) Observed Data J Index
	(9,:) Observed Data K Index
IACTV	Activity Coefficient Option
IALB	Bare-Soil Albedo Option
IALB_P	Plant Albedo Option
IATM_C	Index for Cyclic Atmospheric Conditions
IBCC	Index for Cyclic Boundary Conditions
IBCCG	LUGR Subgrid Index for Cyclic Boundary Conditions
IBCD	Index for Boundary Condition Direction
IBCDFG	Aqueous Boundary Condition Diffusion Option
IBCDFL	Gas Boundary Condition Diffusion Option
IBCDG	LUGR Subgrid Index for Boundary Condition Direction
IBCIN	Boundary Condition Number Pointer
IBCM	Index for Number of Boundary Cond. Times
IBCMG	LUGR Subgrid Index for Number of Boundary Cond. Times
IBCN	Index for Adjacent Node to Boundary Cond.
IBCNG	LUGR Subgrid Index for Adjacent Node to Boundary Cond.
IBCSN	Index for Boundary Condition Surface Normal
IBCSP	Number of Boundary Condition Reactive Species
IBCT	Index for Boundary Condition Type
IBCTG	LUGR Subgrid Index for Boundary Condition Type
IBIG	Constant, 32000
IBL	Unit Number for Release File
IC_NCG	Initial Condition Option Index for STOMP-WCMSE
IC_SH	Initial Condition Option for Hydrate Saturation
IC_SP	Initial Condition Option for Reactive Species
ICAIR	Initial Condition Index for Aqueous Dissolved CO_2
ICBRN	Index Salt Concentration Initial Condition
ICD	Number of Characters in Card Name String
ICNO	Index for Reference Node Header Line
ICNS	Index for Surface Flux Header Line
ICNV	Index for Newton-Raphson Convergence
ICODE	Index for Code’s Operational Mode
ICRNT	Index for Courant Number Computation

Table 3.2. (contd)

Variable	Description
ICS	Index for Coordinate System Type
ICSN	Number of Characters in Subroutine String
ICT	Index for Aqueous Solute Concentration Initial Cond.
ICTN	Index for NAPL Solute Concentration Initial Cond.
ID	X-Direction Index of Node
IDF_ELC	Index for Electrolyte Density Function
IDFLT	Index Indicating Default Value Specified
IDFLTD	Index Indicating Default Value Used
IDG	LUGR Subgrid X-Direction Index of Node
IDISP	Index for Hydraulic Dispersion Calculation
IDMN(20)	Index for Interfacial Averaging Scheme
	(1) Thermal conductivity
	(2) Aqueous density
	(3) Gas density
	(4) Napl density
	(5) Aqueous viscosity
	(6) Gas viscosity
	(7) Napl viscosity
	(8) Aqueous relative permeability
	(9) Gas relative permeability
	(10) Napl relative permeability
	(11) Intrinsic permeability
	(12) Water vapor diffusion
	(13) Oil vapor diffusion
	(14) Dissolved air diffusion
	(15) Dissolved oil diffusion
	(16) Solute diffusion
	(17) Hydraulic dispersion
	(18) Dissolved salt diffusion
	(19) Effective permeability
	(20) Air gas diffusion
IDP	Index for Dual-Porosity Model
IDSPD	Index for Dissolved Oil Dispersion Calcu.
IDSPS	Index for Salt Dispersion Calcu.
IEDL	Index for Empirical Solute Diffusion Coefficient
IEDLS	Index for Empirical Salt Diffusion Coefficient
IEO	Index for Execution Option
IEPD	Number of Execution Periods
IEQ_C	Indices for the Conservation Equation for ECKEChem
IEQ_E	Indices for the Equilibrium Equation for ECKEChem
IEQ_K	Indices for the Kinetic Equation for ECKEChem
IEQ_S	Matrix Pointer for ECKEChem

Table 3.2. (contd)

Variable	Description
IEQA	Index for Air Equation
IEQALC	Index of Alcohol Equation
IEQC	Index for Solute Transport Equation
IEQD	Index for Dissolved-Oil Equation
IEQDA	Index for Dissolved-CO_2 Equation
IEQDO	Boundary Index for Dissolved-Oil Equation
IEQO	Index for Oil Equation
IEQS	Index for Salt/Solute Equation
IEQT	Index for Energy Equation
IEQW	Index for Water Equation
IFLD	Number of Nodes in the X-Direction
IFLDG	LUGR Subgrid Number of Nodes in the X-Direction
IFQO	Frequency of Refer. Node Prints (Output)
IFQS	Frequency of Refer. Node Prints (Screen)
IGAMMA(8)	(1) Saturated Hydraulic Conductivity Scaling Function
	(2) Diffusive Porosity Scaling Function
	(3) van Genuchten "alpha" or Brooks/Corey "psi" Scaling Function
	(4) van Genuchten "n" or Brooks/Corey "lambda" Scaling Function
	(5) Residual Saturation Scaling Function
IGL	Index for LUGR Grid Level
IHSF	Index for Surface Flux Header
IJFLD	Number of Nodes in the X-Y plane
ILES	Index for Linear Equation Solver
ILU	Index for Jacobian Matrix Pivoting
IM	Jacobian Matrix Pointer Array
IMSG	Index for Error Message
IMTC	Index for Oil Interface Mass Trans. Corr.
IMTLN	Index for Interface Mass Transfer Function
INCG	Noncondensible Gas Index
INMBLT	Number of Global Balance Check Times
INMREL	Number of SAC Release Plane Time Change, K-Node Indices
INMREM	Number of SAC Remediation Events
INPDAT	Input File Creation Date* (132 characters)
INPTIM	Input File Creation Time* (132 characters)
IO_TH	Number of CH_4 Mass Fraction Entries in Hydrate Table
IOBDEF	Unit Number for an ".ext" File for UCODE
IOBDSF	Unit Number for an ".sto" File for UCODE
IOBDUF	Unit Number for an ".uni" File for UCODE
IOM	Index for Operational Mode
IP_TA	CO_2 Property Table Pressure Index
IPCGL	Index for Gas-Aqueous Adsorption Function Option
IPCL	Index for Solute Solid-Aqueous Partition Option

Table 3.2. (contd)

Variable	Description
IPCLN	Index for Solute Aqueous-NAPL Partition Option
IPCSL	Index for Solute Dependent Solid-Aqueous Part.
IPCSLD	Index for Dissolved-Oil Sol.-Aqueous Part. Opt.
IPCSLS	Index for Salt Sol.-Aqueous Part. Opt.
IPH	Triple Curve Scanning Path Index
IPL	Unit Number of the Plot File
IPLOT	Index for Plot File Variables
IPRF	Index for Permeability Reduction Factors
IPTPS	Index for Surface Tension Correlation Function
IRC	Unit Number of Recharge File
IRC_K	Indices for the Kinetic Reaction for ECKEChem
IRCKT	Kinetic Reaction Type Index for ECKEChem
IRD	Unit Number of the Input File
IREF	Index for Reference Node Variables
IRGTBL	Start/Stop Indices for Gas Relative Permeability Table
IRHOO	Index for Oil Density Function
IRHOS	Index for Salt Density Function
IRL	Unit Number of Release File
IRLK	SAC release plane K-node index
IRLTBL	Start/Stop Indices for Aqueous Relative Permeability Table
IRLTBLT	Start/Stop Indices for Aqueous Relative Permeability Tensor Table
IRMSOL	Index for Remediated Solute
IRNTBL	Start/Stop Indices for NAPL Relative Permeability Table
IRPG	Index for Gas Relative Permeability Function
IRPL	Index for Aqueous Relative Permeability Function
IRPLT	Index for Aqueous Relative Permeability Tensor Function
IRPN	Index for NAPL Relative Permeability Function
IRS	Unit Number of the Restart File
IRSM_P	Index for the Root Stress Model for Plant Species
IS_TH	Number of Salt Mass Fraction Entries in Hydrate Table
ISALT	Index for Salt Type
ISC	Unit Number of the Screen (I.e., Standard Output)
ISCALE	Index for Scaling Group
ISCHR	Index for Saturation-Capillary Pressure Function
ISF	Unit Number of the Surface File.
ISFC(6,LSF)	Surface Flux Domain Indices
	(1) Bottom
	(2) South
	(3) West
	(4) East
	(5) North
	(6) Top

Table 3.2. (contd)

Variable	Description
ISFD	Index for the Surface Flux Direction
ISFDOM	Surface Flux Domain Indices
ISFF	Index for Surface Flux File
ISFGP	Index for Surface Flux Group
ISFT	Index for the Surface Flux Type
ISGNO	Number of Sign. Digits in the Output File
ISGNP	Number of Sign. Digits in the Plot File
ISGNS	Number of Sign. Digits on the Screen
ISIC	Index for Saturation Initial Condition
ISKP	Index for Skipping Residual Checking on Rock/Soil Types
ISLC(41)	Indices for Solution Control
	(1) Index for Solute Transport Advection/Diffusion Scheme
	(2) Index for Water-Vapor Gas Diffusion Model
	(3) Index for Tortuosity Model
	(4) Index for Aqueous-Phase Diffusion Coefficient Model
	(5) Index for Soil Freezing Conditions
	(6) Index for Salt Transport Advection/Diffusion Scheme
	(7) Index for Osmotic Pressure and Surface Tension Effects
	(8) Index for Dissolved-Oil Transport Advection/Diffusion Scheme
	(9) Index for Fluid Properties
	(10) Index for Dissolved-Air Trans. Advection/Diffusion Scheme
	(11) Index for Time-Differencing Order
	(12) Index for Diffusive-Dispersive Dissolved-Oil Transport Scheme
	(13) Index for Particle-Displacing Bubbles
	(14) Index for SPLIB Summary Output
	(15) Index for LUGR
	(16) Index for Density Dependent Solute Transport
	(17) Index for Courant Number Control
	(18) Index for Restart Files
	(19) Index for Scaling Factors
	(20) Index for Inverse (UCode)
	(21) Index for Operational Mode Restart
	(22) Index for IJK Indexing
	(23) Index for Advective Solute Transport
	(24) Index for Plants (Root-Uptake Evapotranspiration)
	(25) Index for Poynting Effect
	(26) Index for Rainfall
	(27) Index for Aqueous Molar Density Gradient
	(28) Index for Gas Molar Density Gradient
	(29) Index for NAPL Molar Density Gradient
	(30) Index for (Non)isothermal Conditions
	(31) Index for Equilibrium/First Order Hydrate Saturation

Table 3.2. (contd)

Variable	Description
	(32) Index for (Non)isobrine Conditions
	(40) Index for No Reactive/Eckechem Transport
	(41) Index for Nonlinear Solver Option for ECKEChem
ISLTBL	Start/Stop Indices for Saturation Function Table
ISM	Index for Saturation Function Extension
ISMALL	Constant, -3200
ISNR	Index for NAPL Residual Saturation Model
ISP_MN	Mineral Species Index for ECKEChem
ISP_S	Reverse Matrix Pointer for ECKEChem
ISPLK	Flow Component Link Index for ECKEChem
ISRDM	Source Domain Indices
ISRM_P	Index for Source Time
ISRM	Index for the Stomatal Resistance Model for Plant Species
ISRT	Index for Source Type
ISVC	Number of Coupled Unknowns
ISVF	Number of Flux Variable Indices
ISVT	Index for Solute Transport Solution
ISWBCF	Index for Shuttleworth-Wallace Boundary Cond. Failure
IT_TA	Index for Temperature Range in the CO_2 Property Table
IT_TH	Number of Temperature Entries in Hydrate Table
ITERDC	Index for Ternary Phase Diagram, Binodal Curve Type
ITHK	Index for Thermal Conductivity Model
ITOR	Index for Tortuosity Function
IUNK	Exponent on temperature units
IUNKG	Exponent on mass units
IUNM	Exponent on length units
IUNMOL	Exponent on molar units
IUNS	Exponent on time units
IV_TA	Index for Saturation Line in the CO_2 Property Table
IVAPO	Index for Oil Saturated Vapor Pressure Function
IVAPS	Index for Alcohol Saturated Vapor Pressure Function
IVD	Unit Number for STOMP Remediation Transfers Report
IVF_ELC	Index for Electrolyte Viscosity Function
IVISA	Index for Alcohol Viscosity Function
IVISO	Index for Oil Viscosity Function
IVISS	Index for Surfactant Viscosity Function
IVR	Number of Non-Blank Characters
IVRSN	Index for Version Number
IWCC	Index for Cyclic Well Conditions
IWL	Index for Well Number from Well Node Number
IWLDM	Well Domain Indices
IWM	Number of Well Condition Times

Table 3.2. (contd)

Variable	Description
IWN	Index for Node Number form Well Node Number
IWR	Unit Number of the Output File
IWT	Index for Well Type
IXF	Index for Fixed Nodes
IXG	LUGR Subgrid Matrix Location of Node
IXGB	LUGR Subgrid Index for Boundary Nodes
IXP	Matrix Location of Node
IXREF	Index for Curvilinear Coordinate Factor
IXW	Index for Well Node Number from Node Number
IZ	Index for Rock/Soil Type
IZG	LUGR Subgrid Index for Rock/Soil Type
JD	Y-Direction Index of Node
JDG	LUGR Subgrid Y-Direction Index of Node
JFLD	Number of Nodes in the Y-Direction
JFLDG	LUGR Subgrid Number of Nodes in the Y-Direction
JKFLD	Number of Nodes in the Y-Z Plane
JLU	Unused Banded Matrix Solver Integer Array
JM	Jacobian Matrix Pointer
KD	Z-Direction Index of Node
KDG	LUGR Subgrid Z-Direction Index of Node
KEXIT	K-Index for SAC Release Plane
KFLD	Number of Nodes in the Z-Direction
KFLDG	LUGR Subgrid Number of Nodes in the Z-Direction
KIFLD	Number of Nodes in the Z-X Plane
KLU	Coupled-Equations Conjugate Gradient Solver Pointer Array
KLUC	Transport-Equations Conjugate Gradient Solver Pointer Array
MADJ	Adjacent Node Index Pointer
MDC	Row of Diagonal Elements for Banded Sol.
MDIM	Conjugate Gradient Solver Pointer Array
MDT	Matrix Band Width
MEPD	Index for Cyclic Execution Periods
MFLX	Flux Variable Index Pointer
MKC	Number of Coupled-Equations Unknowns
MKT	Number of Transport-Equations Unknowns
MLC	Lower Half Band Width
MLT	Matrix Lower Half Band Width
MLU	Coupled-Equations Pointer for SPLIB Solver
MLUC	Transport-Equations Pointer for SPLIB Solver
MNEG	Flux Variable Index Pointer
MNEGB	Boundary Index Pointer
MNOD	Flux Variable Index Pointer
MPOS	Flux Variable Index Pointer

Table 3.2. (contd)

Variable	Description
MPOSB	Boundary Index Pointer
MUC	Upper Half Band Width
MUT	Matrix Upper Half Band Width
MXSTEP	Maximum Number of Time Steps
N_CRN	Courant Number Indexed on Solute
N_DB	Node Number for Error Messages
NATM_T	Number of Atmospheric Condition Table Entries
NBC	Number of Boundary Conditions
NBCG	Number of Boundary Conditions at the Local Grid Level
NCHEM	Number of Chemical Reactions
ND	Node Number of I, J, K Indices
NDG	Node Number of I, J, K Indices at the Local Grid Level
NDIM	Number of Active Dimensions
NDREF	Reference Node Type
NEPD	Number of Execution Periods
NEQC	Number of Conservation Equations for ECKEChem
NEQE	Number of Equilibrium Equations for ECKEChem
NEQK	Number of Kinetic Equations for ECKEChem
NFLD	Number of Active Field Nodes
NFLDG	Number of Active Field Nodes at the Local Grid Level
NGC	Node Pointer from Finer Grid to Next Coarser Grid Level
NGF	Node Pointer from Coarser Grid to Next Finer Grid Level
NGL	Number of Grid Levels
NITER	Number of Iterations
NITR	Number of Iterations at the Local Grid Level
NLU	Pointer for SPLIB Solver
NLUC	Pointer for SPLIB Solver for Solutes
NOBDP	Observed Data Point Number
NOBDS	Observed Data Sample Pointer
NOBDT	Number of Observed Data Types
NOTES	Number of Lines of Simulation Notes
NPHAZ	Phase Condition
NPHAZW	Phase Condition for Coupled Well Nodes
NPLANT	Number of Plants
NPLY_RL	Number of Aqu. Relative Permeability Polynomial Pieces
NPLY_SL	Number of Aqueous Saturation Polynomial Pieces
NPRTM	Number of Print Times
NQA	Number of Carbon Atoms in Alcohol
NQO	Number of Carbon Atoms in Oil
NRCE	Number of Equilibrium Reactions for ECKEChem
NRCK	Number of Kinetic Reactions for ECKEChem
NREF	Number of Reference Nodes

Table 3.2. (contd)

Variable	Description
NRIM	Max. Number of Newton Raphson Iterations
NRIMX	Max. Number of Newton Raphson Iterations
NROCK	Number of Rock/Soil Types
NRST	Restart Time Step Number
NRTSI	Number of Reactive Transport Sequence Iterations
NSCALE	Number of Scaling Groups
NSD	Node of Maximum Residual
NSF	Number of Surface Flux Domains
NSFDOM	Surface-Flux Domain Index
NSFGP	Number of Named Surface Flux Files
NSL_ELC	Number of Electrolyte Solutes
NSOLU	Number of Transported Solutes
NSPC	Number of Conservation-Component Reactive Species
NSPG	Number of Gas Reactive Species
NSPK	Number of Kinetic-Component Reactive Species
NSPL	Number of Aqueous Reactive Species
NSPLK	Number of Reactive Species Linked to Flow Components
NSPN	Number of NAPL Reactive Species
NSPR	Number of Reactive Species
NSPS	Number of Solid Reactive Species
NSR	Number of Sources
NSTEP	Number of Time Steps
NSX	X-Direction Lower Surface Number of Node
NSY	Y-Direction Lower Surface Number of Node
NSZ	Z-Direction Lower Surface Number of Node
NSZW	Z-Direction Lower Surface Number of Coupled Well Node
NTBL	Number of Table Entries
NTSR	Num. of Successive Time Step Reductions
NVPLOT	Number of Plot File Variables
NVREF	Number of Reference Node Variables
NWLN	Number of Coupled Well Nodes
NWLS	Number of Coupled Wells
NXP	Number of Inactive Nodes
OEC	Osmotic Efficiency Coefficient
OECB	Boundary Osmotic Efficiency Coefficient
ONE	Constant, 1
P_LV	CO_2 or CH_4 Liquid-Vapor Saturation Pressure, Pa
P_TA	Pressure Array for CO_2 Property Data Table, Pa
P_TH	Pressure Array of Saturation Line for CO_2 Property Data Table, Pa
PAF	Component Pitzer Acentric Factor
PAFA	Alcohol Pitzer Acentric Factor
PAFO	Oil Pitzer Acentric Factor

Table 3.2. (contd)

Variable	Description
PAFS	Surfactant Pitzer Acentric Factor
PAFW	Water Pitzer Acentric Factor
PARMS_P	Plant Species Parameters
	(1,LPLANT) Maximum Root Depth, m
	(2,LPLANT) Stomatal Res.: Minimum Stomatal Resistance, s/m
	(3,LPLANT) Stomatal Res.: Light Response Coefficient, W/m^2
	(4,LPLANT) Null Root Depth, m
	(5,LPLANT) Solar Albedo: Initial Stage Start
	(6,LPLANT) Solar Albedo: Crop Development Start
	(7,LPLANT) Root Depth Fit Parameter
	(8,LPLANT) Solar Albedo: Mid-Season Start
	(9,LPLANT) Solar Albedo: Late-Season Start
	(10,LPLANT) Solar Albedo: Late-Season Stop
	(11,LPLANT) Canopy Height, m
	(16,LPLANT) Maximum Condensate Depth, m
	(12,LPLANT) Root Stress-Point 1 (Capillary Head), m
	(13,LPLANT) Root Stress-Point 2 (Capillary Head), m
	(14,LPLANT) Root Stress-Point 3 (Capillary Head), m
	(15,LPLANT) Root Stress-Point 4 (Capillary Head), m
	(12,LPLANT) Wilting-Point Water Content
	(13,LPLANT) Norm. Soil Water Content: Critical Point 1
	(14,LPLANT) Norm. Soil Water Content: Critical Point 2
	(15,LPLANT) Saturated Water Content
	(12,LPLANT) Cap. Head when Root Uptake Reduced 50%, m
	(17,LPLANT) Crop Coeff.: Initial Stage Start
	(18,LPLANT) Crop Coeff.: Initial Stage Start Day, s
	(19,LPLANT) Crop Coeff.: Crop Development Start
	(20,LPLANT) Crop Coeff.: Crop Development Start Day, s
	(21,LPLANT) Crop Coeff.: Mid-Season Start
	(22,LPLANT) Crop Coeff.: Mid-Season Start Day, s
	(23,LPLANT) Crop Coeff.: Late-Season Start
	(24,LPLANT) Crop Coeff.: Late-Season Start Day, s
	(25,LPLANT) Crop Coeff.: Late-Season Stop
	(26,LPLANT) Crop Coeff.: Late-Season Stop Day, s
	(27,LPLANT) Stomatal Res.: Minimum Temperature, C
	(28,LPLANT) Stomatal Res.: Maximum Temperature, C
	(29,LPLANT) Stomatal Res.: Optimum Temperature, C
PATM	Atmospheric Pressure, Pa
PC	Component Critical Pressure, Pa
PCCVA	Alcohol Pure Comp. Characteristic Volume, L/mol
PCCVO	Oil Pure Comp. Characteristic Volume, L/mol
PCGL	Gas-Aqueous Partition Coefficient, m^3/m^3

Table 3.2. (contd)

Variable	Description
PCLN	Aqueous-NAPL Partition Coefficient, m^3/m^3
PCMP	Bulk Compressibility Reference Pressure, Pa
PCRA	CO_2 Critical Pressure, Pa
PCRO	Oil Critical Pressure, Pa
PCRW	Water Critical Pressure, Pa
PCSL	Solid-Aqueous Partition Coefficient, m^3/kg
PCSLD	Dissolved-Oil Solid-Aqueous Part. Coefficient, m^3/kg
PCSLS	Salt Solid-Aqueous Part. Coefficient, m^3/kg
PERM	Intrinsic Permeability, m^2
PERMRF	Permeability Reduction Factor
PG	Gas Pressure, Pa
PGB	Boundary Gas Pressure, Pa
PGW	Well Gas Pressure, Pa
PGW_CO	Old Water-Vapor Partial Press. at Mean Crop Flow Height, m
PH	Hydrate Pressure, Pa
PHB	Boundary Hydrate Pressure, Pa
PHDL	Aqueous Ponding Height, m
PHDN	NAPL Ponding Height, m
PI	Ice Pressure, Pa
PIB	Boundary Ice Pressure, Pa
PL	Aqueous Pressure, Pa
PL_SO	Old Aqueous Pressure at Ground Surface, Pa
PLANT	Plant Name
PLB	Boundary Aqueous Pressure, Pa
PLWB	Well Bottom Pressure, Pa
PMDD	Mean Particle Diameter, m
PMN	Minimum Gas Pressure, Pa
PMX	Maximum Gas Pressure, Pa
PN	NAPL Pressure, Pa
PNB	Boundary NAPL Pressure, Pa
POR(4,:)	Porosity
	(1) Total Porosity
	(2) Diffusive Porosity
	(3) Fracture Total Porosity
	(4) Diffusive Porosity
POR_M	Mineral Volume Fraction for ECKEChem
PORD	Diffusive Porosity
PORD_O	Old Diffusive Porosity
PORDB	Boundary Diffusive Porosity
PORT	Total Porosity
PORT_O	Old Total Porosity
PORTB	Boundary Total Porosity

Table 3.2. (contd)

Variable	Description
POSB	Boundary Osmotic Pressure, Pa
POSM	Osmotice Pressure, Pa
PRTM	Plot File Print Time, s
PSO	Oil Saturated Vapor Pressure, Pa
PSOB	Boundary Oil Saturated Vapor Pressure, Pa
PSW	Water Saturated Vapor Pressure, Pa
PSWB	Water Saturated Vapor Pressure, Pa
PTPS	Surface Tension Function Parameters
PVA	Air Vapor Pressure, Pa
PVA_O	Old CO_2 Partial Pressure, Pa
PVAB	Boundary Air Vapor Pressure, Pa
PVO	Oil Vapor Pressure, Pa
PVO_O	Old CH_4 Partial Pressure, Pa
PVOB	Boundary Oil Vapor Pressure, Pa
PVW	Water Vapor Pressure, Pa
PVWB	Boundary Water Vapor Pressure, Pa
PW	Coupled Well Total Pressure, Pa
PWB	Coupled Well Bottom Pressure, Pa
PWGW	Coupled Well Gas Pressure, Pa
PWLW	Coupled Well Aqueous Pressure, Pa
PWNW	Coupled Well NAPL Pressure, Pa
QL_W	Coupled Well Aqueous Volumetric Flow Rate, m^3/s
QLW	Well Aqueous Volumetric Flow Rate, m^3/s
QN_W	Coupled Well NAPL Volumetric Flow Rate, m^3/s
QNW	Well NAPL Volumetric Flow Rate, m^3/s
QT_W	Coupled Well Total-Liquid Volumetric Flow Rate, m^3/s
QTW	Well Total-Liquid Volumetric Flow Rate, m^3/s
R_OBDS	Observed Data Time and Value
R_OBDT	Observed Data Location and Weighting Factors
RC_E	Equilibrium Reaction Constant Parameters
RC_K	Kinetic Reaction Constant Parameters
RCA	Air Gas Constant, 287.0 J/kg K
RCHDF	Chemical Reaction Decay Factor
RCHDFL	Chemical Reaction Decay Factor in Aqueous
RCHDFN	Chemical Reaction Decay Factor in NAPL
RCNME	Equilibrium Reaction Name for ECKEChem
RCNMK	Kinetic Reaction Name for ECKEChem
RCO	Oil Gas Constant, J/kg K
RCS	Surfactant Gas Constant, J/kg K
RCU	Universal Gas Constant, 8314.34 J/kg K
RCW	Water Gas Constant, 461.52 J/kg K
RFIM_P	Intercepted Rainfall on Plant, kg/ m^2 ground

Table 3.2. (contd)

Variable	Description
RHLF	First-Order Reaction Rate Constant, s
RHLFL	Aqueous First-Order Reaction Rate Constant, s
RHLFN	NAPL First-Order Reaction Rate Constant, s
RHO_TA	CO_2 Property Table Values for Density, kg/m^3
RHOG	Gas Density, kg/m^3
RHOG_O	Old Gas Density, kg/m^3
RHOGB	Boundary Gas Density, kg/m^3
RHOGW	Well Gas Density, kg/m^3
RHOGW_O	Old Well Gas Density, kg/m^3
RHOGWB	Well Boundary Gas Density, kg/m^3
RHOH	Hydrate Density, kg/m^3
RHOH_O	Old Hydrate Density, kg/m^3
RHOHB	Boundary Hydrate Density, kg/m^3
RHOI	Ice Density, kg/m^3
RHOI_O	Old Ice Density, kg/m^3
RHOIB	Boundary Ice Density, kg/m^3
RHOL	Aqueous Density, kg/m^3
RHOL_LV	CO_2 or CH_4 Saturated Liquid Density, kg/m^3
RHOL_O	Old Aqueous Density, kg/m^3
RHOLB	Boundary Aqueous Density, kg/m^3
RHOLI	Density for Invariant Fluids, kg/m^3
RHOLW	Well Aqueous Density, kg/m^3
RHOLW_O	Old Well Aqueous Density, kg/m^3
RHOLWB	Well Boundary Aqueous Density, kg/m^3
RHOMG	Gas Molar Density, kmol/m^3
RHOMGB	Boundary Gas Molar Density, kmol/m^3
RHOML	Aqueous Molar Density, kmol/m^3
RHOMLB	Boundary Aqueous Molar Density, kmol/m^3
RHOMN	NAPL Molar Density, kmol/m^3
RHOMNB	Boundary NAPL Molar Density, kmol/m^3
RHON	NAPL Density, kg/m^3
RHON_O	Old NAPL Density, kg/m^3
RHONB	Boundary NAPL Density, kg/m^3
RHONW	Well NAPL Density, kg/m^3
RHONW_O	Old Well NAPL Density, kg/m^3
RHONWB	Well Boundary NAPL Density, kg/m^3
RHORA	Alcohol Liquid Reference Density, kg/m^3
RHORG	Gas Reference Density, kg/m^3
RHORL	Aqueous Reference Density, kg/m^3
RHORN	NAPL Reference Density, kg/m^3
RHORO	Oil Liquid Reference Density, kg/m^3
RHOS	Rock/Soil Particle Density, kg/m^3

Table 3.2. (contd)

Variable	Description
RHOSP	Precipitated Salt Density, kg/m^3
RHOSP_O	Old Precipitated Salt Density, kg/m^3
RHOV_LV	CO_2 or CH_4 Saturated Vapor Density, kg/m^3
RKG	Gas Relative Permeability
RKGB	Boundary Gas Relative Permeability
RKL(3,:,:)	Aqueous Relative Permeability
	(1,:,:) X-Direction
	(2,:,:) Y-Direction
	(3,:,:) Z-Direction
RKLB(3,:,:)	Boundary Aqueous Relative Permeability
	(1,:,:) X-Direction
	(2,:,:) Y-Direction
	(3,:,:) Z-Direction
RKN	NAPL Relative Permeability
RKNB	Boundary NAPL Relative Permeability
RLMSG	Real Number Component of Error Message
RLXF	Relaxation Factor
ROCK	Rock/Soil Name* (64 characters)
RP	Radial Coordinate Radius, m
RPG	Radial Coordinate Radius, m
RPGC	Gas Relative Permeability Function Coefficients
RPLC	Aqueous Relative Permeability Function Coefficients
RPLT	Aqueous Relative Permeability Function Coefficients
RPNC	NAPL Relative Permeability Function Coefficients
RS_S	Mineral Specific Area for ECKEChem, m^2
RSD	Maximum Residual of Equation
RSDL	Residual
RSDM	Maximum Residual of Execution Period
RSDMX	Maximum Residual
S	Salt Concentration, kg/m^3 node volume
S_O	Old Salt Concentration, kg/m^3 node volume
SATAC	Alcohol Saturated Vapor Pressure Function Coefficients
SATOC	Oil Saturated Vapor Pressure Function Coefficients
SB	Boundary Salt Concentration, kg/m^3 node volume
SCALNM	Scaling Group Name
SCHR	Soil Moisture Retention Charac. Coefficient
SDCL	Solute Aqueous Molecular Diffusion Coefficient, m^2/s
SDCLS	Salt/Surf. Aqueous Molecular Diffusion Coefficient, m^2/s
SDPF	Fracture Saturation (Dual Porosity)
SDPFB	Bound. Fracture Saturation (Dual Porosity)
SDPM	Matrix Saturation (Dual Porosity)
SDPMB	Bound. Matrix Saturation (Dual Porosity)

Table 3.2. (contd)

Variable	Description
SF	Surface Flux Rate and Integral
SFCSF	Aqueous-Gas Surface Tension Coefficients
SG	Gas Saturation
SG_O	Old Gas Saturation
SGB	Boundary Gas Saturation
SGT	Trapped Gas Saturation
SGTL	Aqueous Trapped Gas Saturation
SGTN	NAPL Trapped Gas Saturation
SH	Hydrate Saturation
SH_O	Old Hydrate Saturation
SHB	Boundary Hydrate Saturation
SI	Ice Saturation
SI_O	Old Ice Saturation
SIB	Boundary Ice Saturation
SIG_GL	Gas-Aqueous Interfacial Tension, N/m
SIG_GN	Gas-NAPL Interfacial Tension, N/m
SIG_HL	Hydrate-Aqueous Interfacial Tension, N/m
SIG_IL	Ice-Aqueous Interfacial Tension, N/m
SIG_NL	NAPL-Aqueous Interfacial Tension, N/m
SL	Aqueous Saturation
SL_O	Old Aqueous Saturation
SLB	Boundary Aqueous Saturation
SLSC	Aqueous Scanning Path Saturation
SLW	Well Aqueous Saturation
SLW_O	Old Well Aqueous Saturation
SMALL	Constant, 1^{-20}
SMDEF	Macrodispersivity Enhancement Factor
SMDG	Solute Gas Molecular Diffusion Coefficient, m^2/s
SMDL	Solute Aqueous Molecular Diffusion Coefficient, m^2/s
SMDLS	Salt Aqueous Molecular Diffusion Coefficient, m^2/s
SMDN	Reactive Species NAPL Molecular Diffusion Coefficient, m^2/s
SN	NAPL Saturation
SN_O	Old NAPL Saturation
SNB	Boundary NAPL Saturation
SNR	Residual NAPL Saturation
SNT	Entrapped NAPL Saturation
SNW_O	Old Well NAPL Saturation
SNW	Well NAPL Saturation
SOLML	Solute Aqueous Solubility Limit, $1/m^3$
SOLUT	Solute Name* (64 characters)
SP_C	Reactive Species Volumetric Concentration, mol/m^3
SP_CBO	Old Reactive Species Volumetric Concentration, mol/m^3

Table 3.2. (contd)

Variable	Description
SP_CO	Old Boundary Reactive Species Volumetric Conc., mol/m^3
SP_L(3,:)	Aqueous Species Parameters
	(1,:) Aqueous Species Charge
	(2,:) Aqueous Species Atomic Diameter, m
	(3,:) Aqueous Species Molecular Weight, kg/kmol
SP_MDG	Reactive Species Gas Molecular Diffusion Coefficient, m^2/s
SP_MDL	Reactive Species Aqueous Molecular Diffusion Coefficient, m^2/s
SP_MDN	Reactive Species NAPL Molecular Diffusion Coefficient, m^2/s
SP_S(2,:)	Solid Species Parameters
	(1,:) Solid Species Mass Density, kg/m^3
	(2,:) Solid Species Molecular Weight, kg/kmol
SPCMON	Scaled Nodal Space Monitor for Local Grid Refinement
SPCTOL	Scaled Nodal Space Tolerance for Local Grid Refinement
SPNMC	Conservation-Component Species Name for ECKEChem
SPNMG	Gas Species Name for ECKEChem
SPNML	Kinetic-Component Species Name for ECKEChem
SPNMK	Aqueous Species Name for ECKEChem
SPNMN	NAPL Species Name for ECKEChem
SPNMS	Solid Species Name for ECKEChem
SRC	Source Quantity
SRCA	Air Mass Source, kg/s
SRCD	Dissolved-Oil Mass Source, kg/s
SRCIA	Air Mass Source Integral, kg
SRCIC	Solute Source Integral, 1/s
SRCID	Dissolved-Oil Mass Source Integral, kg
SRCIO	Oil Mass Source Integral, kg
SRCIS	Salt Mass Source Integral, kg
SRCIT	Energy Source Integral, J
SRCIW	Water Mass Source Integral, kg
SRCO	Oil Mass Source, kg/s
SRCO_W	Well Oil Mass Source, kg/s
SRCP(2)	SVE Well Parameters
	(1) SVE Well Minimum Gas Pressure, Pa
	(2) SVE Well Diameter, m
SRCS	Salt Mass Source, kg/s
SRCT	Energy Source, W
SRCW	Water Mass Source, kg/s
SRCW_W	Well Water Mass Source, kg/s
SS	Precipitated Salt Saturation
SS_O	Old Precipitated Salt Saturation
SSB	Boundary Precipitated Salt Saturation
STW	Well Total-Liquid Saturation

Table 3.2. (contd)

Variable	Description
STW_O	Old Well Total-Liquid Saturation
SUBNM	Subroutine Path* (512 characters)
SUBNMX	Subroutine Path* (512 characters)
SUFA	Surface Tension of Pure Alcohol @ 25 C, N/m
SUFO	Surface Tension of Pure Oil @ 25 C, N/m
SUFW	Surface Tension of Pure Water @ 25 C, N/m
SUGL	Gas-Aqueous Interfacial Tension, N/m
SUGN	Gas-NAPL Interfacial Tension, N/m
SUNL	NAPL-Aqueous Interfacial Tension, N/m
SWBCD	Shuttleworth-Wallace Boundary-Condition Depth, m
T	Temperature, C
T_CO	Old Crop Temperature, C
T_LV	CO_2 or CH_4 Liquid-Vapor Saturation Temperature, C
T_O	Old Temperature, C
T_PO	Old Plant Temperature, C
T_SO	Old Ground Surface Temperature, C
T_TA	Temperature Array for CO_2 Property Data Table
T_TH	Temperature Array of Saturation Line for CO_2 Property Data Table
TABS	Absolute Temperature, 273.15 K
TB	Boundary Temperature, C
TBA	Air Normal Boiling Temperature, K
TBLDDX	Array of Second Derivatives for Spline Interpolation
TBLDDY	Array of Second Derivatives for Spline Interpolation
TBLX	Array of Tabulated Values for Spline Interpolation
TBLY	Array of Tabulated Values for Spline Interpolation
TBO	Oil Normal Boiling Temperature, K
TBW	Water Normal Boiling Temperature, K
TC	Component Critical Temperature, K
TCRA	Alcohol or CO_2 Critical Temperature, K
TCRO	Oil Critical Temperature, K
TCRS	Surfactant Critical Temperature, K
TCRW	Water Critical Temperature, K
TDRA	Alcohol Density Reference Temperature, C
TDRO	Oil Density Reference Temperature, C
TENTH	Constant, 0.1
TERDC(11)	Ternary Phase Diagram Constants
	(1) Binodal Curve Constant: Ah
	(2) Binodal Curve Constant: Bh
	(3) Binodal Curve Constant: Eh
	(4) Binodal Curve Constant: Fh
	(5) Intermediate Value
	(6) Intermediate Value

Table 3.2. (contd)

Variable	Description
	(7) Intermediate Value
	(8) Solubility Limit of Water in Alcohol-Free Liquid Oil
	(9) Solubility Limit of Water in Alcohol-Free Liquid Oil
	(10) Solubility Limit of Oil in Alcohol-Free Liquid Water
	(11) Solubility Limit of Oil in Alcohol-Free Liquid Water
TFPA	Alcohol Freezing Point Temperature, C
TFPO	Oil Freezing Point Temperature, C
THKG	Gas Thermal Conductivity, W/m K
THKGB	Boundary Gas Thermal Conductivity, W/m K
THKH	Hydrate Thermal Conductivity, W/m K
THKHB	Boundary Hydrate Thermal Conductivity, W/m K
THKI	Ice Thermal Conductivity, W/m K
THKIB	Boundary Ice Thermal Conductivity, W/m K
THKL	Aqueous Thermal Conductivity, W/m K
THKLB	Boundary Aqueous Thermal Conductivity, W/m K
THKN	NAPL Thermal Conductivity, W/m K
THKNB	Boundary NAPL Therm. Conductivity, W/m K
THKRA	Reference Air Thermal Conductivity, W/m K
THKRW	Reference Water Thermal Conductivity, W/m K
THKS	Thermal Conductivity Model Parameters, W/m K
TITLE	Simulation Title* (132 characters)
TLTX	X-Direction Surface Tilt, deg
TLTY	Y-Direction Surface Tilt, deg
TLTZ	Z-Direction Surface Tilt, deg
TM	Time, s
TM_CRN	Cournant Number Limiting Time Step, s
TMBL	Mass Balance Check Time, s
TMMX	Maximum Time, s
TMN	Minimum Temperature, C
TMOB	Observed Date Output Time, s
TMPA	Time Step Acceleration Factor
TMPC	Time Step Cut Factor
TMPD	Execution Period Initial Time Step, s
TMPE	Execution Period Stop Time, s
TMPR	Plot/Restart File Print Time, s
TMPS	Execution Period Start Time, s
TMPX	Execution Period Maximum Time Step, s
TMRL	Release Plane Time, s
TMRM	Remediation Event Time, s
TMX	Maximum Temperature, C
TOLN	Log to Ln Conversion Factor
TOLS	Adaptive Grid Spatial Tolerance

Table 3.2. (contd)

Variable	Description
TOR	Tortuosity Function Constants
TORG	Gas Tortuosity
TORGB	Boundary Gas Tortuosity
TORL	Aqueous Tortuosity
TORLB	Boundary Aqueous Tortuosity
TORN	NAPL Tortuosity
TORNB	Boundary NAPL Tortuosity
TRPGL	Gas-Aqueous Trapping Number
TRPNL	NAPL-Aqueous Trapping Number
TSPRF	Solute Diffusion Reference Temperature, C
U_TA	CO_2 Property Table Values for Internal Energy, J/kg
UC	X-Direction Solute Flux, $1/m^2$ s
UCN	X-Direction Solute Flux in NAPL, $1/m^2$ s
UDGA	X-Direction Air Vapor Diffusive Flux, kg/m^2 s
UDGO	X-Direction Oil Vapor Diffusive Flux, kg/m^2 s
UDGW	X-Direction Water Vapor Diffusive Flux, kg/m^2 s
UDGW_W	Well Radial Water Vapor Diffusive Flux, kg/m^2 s
UDLA	X-Direction Dissolved-Air Diffusive Flux, kg/m^2 s
UDLA_W	Well Radial Dissolved-Air Diffusive Flux, kg/m^2 s
UDLO	X-Direction Dissolved-Oil Diffusive Flux, kg/m^2 s
UDLO_W	Well Radial Dissolved-Oil Diffusive Flux, kg/m^2 s
UDS	X-Direction Salt Diffusion Flux, kg/m^2 s
UEG	Gas Internal Energy, J/kg
UEG_O	Old Gas Internal Energy, J/kg
UEGA	CO_2 Gas Internal Energy, J/kg
UEGAB	Boundary CO_2 Gas Internal Energy, J/kg
UEGB	Boundary Gas Internal Energy, J/kg
UG	X-Direction Darcy Gas Flux, m/s
UG_W	Well X-Direction Darcy Gas Flux, m/s
UGA	X-Direction Mass Flux of Air or CO_2 in Gas, kg/s
UGO	X-Direction Mass Flux of CH_4 in Gas, kg/s
UL	X-Direction Darcy Aqueous Flux, m/s
UL_LV	Liquid Internal Energy on the Liquid-Vapor Saturation Line, J/kg
UL_W	Well X-Direction Darcy Aqueous Flux, m/s
ULA	X-Direction Mass Flux of Air or CO_2 in Aqueous, kg/s
ULO	X-Direction Mass Flux of CH_4 in Aqueous, kg/s
UN	X-Direction Darcy NAPL Flux, m/s
UN_W	Well X-Direction Darcy NAPL Flux, m/s
UNA	X-Direction Mass Flux of Air or CO_2 in NAPL, kg/s
UNAR	Units for Radiants
UNLN	Output/Reference Node Length Units* (64 characters)
UNPLOT	Plot File Variable Index

Table 3.2. (contd)

Variable	Description
UNREF	Reference Node Variable Index
UNSF	Surface Flux Rate and Integral Units* (64 characters)
UNTM	Output/Reference Node Time Units* (64 characters)
UQ	X-Direction Energy Flux, W/m^2
US	X-Direction Salt Flux, kg/m^2 s
USER	User Name* (64 characters)
UV_LV	Vapor Internal Energy on the Liquid-Vapor Saturation Line, J/kg
VARB	Input Variable Name* (64 characters)
VC	Y-Direction Solute Flux, 1/m^2 s
VCN	Y-Direction Solute Flux in NAPL, 1/m^2 s
VCRA	Air Critical Molar Volume, cm^3/mol
VCRO	Oil Critical Molar Volume, cm^3/mol
VCRS	Surfactant Critical Molar Volume, cm^3/mol
VCRW	Water Critical Molar Volume, cm^3/mol
VDGA	Y-Direction Air Vapor Diffusive Flux, kg/m^2 s
VDGO	Y-Direction Oil Vapor Diffusive Flux, kg/m^2 s
VDGW	Y-Direction Water Vapor Diffusive Flux, kg/m^2 s
VDLA	Y-Direction Dissolved-Air Diffusive Flux, kg/m^2 s
VDLO	Y-Direction Dissolved-Oil Diffusive Flux, kg/m^2 s
VDS	Y-Direction Salt Diffusion Flux, kg/m^2 s
VG	Y-Direction Gas Darcy Flux, m/s
VGA	Y-Direction Mass Flux of Air or CO_2 in Gas, kg/s
VGO	Y-Direction Mass Flux of CH_4 in Gas, kg/s
VISCA	Alcohol Viscosity Function Constants
VISCO	Oil Viscosity Function Constants
VISCS	Surfactant Viscosity Function Constants
VISG	Gas Viscosity, Pa s
VISGB	Boundary Gas Viscosity, Pa s
VISGW	Well Gas Viscosity, Pa s
VISGWB	Well Boundary Gas Viscosity, Pa s
VISL	Aqueous Viscosity, Pa s
VISLB	Boundary Aqueous Viscosity, Pa s
VISLI	Viscosity of Invariant Nodes, Pa s
VISLW	Well Aqueous Viscosity, Pa s
VISLWB	Well Boundary Aqueous Viscosity, Pa s
VISN	NAPL Viscosity, Pa s
VISNB	Boundary NAPL Viscosity, Pa s
VISNW	Well NAPL Viscosity, Pa s
VISNWB	Well Boundary NAPL Viscosity, Pa s
VISRG	Gas Reference Viscosity, Pa s
VISRL	Aqueous Reference Viscosity, Pa s
VISRN	NAPL Reference Viscosity, Pa s

Table 3.2. (contd)

Variable	Description
VL	Y-Direction Water Darcy Flux, m/s
VLA	Y-Direction Mass Flux of Air or CO_2 in Aqueous, kg/s
VLO	Y-Direction Mass Flux of CH_4 in Aqueous, kg/s
VMC	Component Critical Molar Volume, cm^3/mol
VN	NAPL Darcy Flux, m/s
VNA	Y-Direction Mass Flux of Air or CO_2 in NAPL, kg/s
VOL	Node Volume, m^3
VQ	Y-Direction Energy Flux, W/m^2
VS	Y-Direction Salt Flux, kg/m^2 s
WBR	Well Bore Radius, m
WBRS	Well Bore Radius for Storage, m
WC	Z-Direction Solute Flux, $1/m^2$ s
WCN	Z-Direction Solute Flux in NAPL, $1/m^2$ s
WDGA	Z-Direction Air Vapor Diffusive Flux, kg/m^2 s
WDGO	Z-Direction Oil Vapor Diffusive Flux, kg/m^2 s
WDGW	Z-Direction Water Vapor Diffusive Flux, kg/m^2 s
WDGW_W	Well Z-Direction Water Vapor Diffusive Flux, kg/m^2 s
WDLA	Z-Direction Dissolved-Air Diffusive Flux, kg/m^2 s
WDLA_W	Well Z-Direction Dissolved-Air Diffusive Flux, kg/m^2 s
WDLO	Z-Direction Dissolved-Oil Diffusive Flux, kg/m^2 s
WDLO_W	Well Z-Direction Dissolved-Oil Diffusive Flux, kg/m^2 s
WDS	Z-Direction Salt Diffusion Flux, kg/m^2 s
WFMN	Interfacial Average Weighting Factor
WG	Z-Direction Gas Darcy Flux, m/s
WG_W	Well Z-Direction Gas Darcy Flux, m/s
WGA	Z-Direction Mass Flux of Air or CO_2 in Gas, kg/s
WGO	Z-Direction Mass Flux of CH_4 in Gas, kg/s
WHBTA	Alcohol HBT Accentric Factor
WHBTO	Oil HBT Accentric Factor
WHP	Initial Well-Head Pressure, Pa
WIDA	Initial Well Dissolved-Air Saturation
WIDO	Initial Well Dissolved-Oil Saturation
WL	Z-Direction Aqueous Darcy Flux, m/s
WL_W	Well Z-Direction Aqueous Darcy Flux, m/s
WLA	Z-Direction Mass Flux of Air or CO_2 in Aqueous, kg/s
WLO	Z-Direction Mass Flux of CH_4 in Aqueous, kg/s
WLVR	Well Head Parameters
WN	Z-Direction NAPL Darcy Flux, m/s
WN_W	Well Z-Direction NAPL Darcy Flux, m/s
WNA	Z-Direction Mass Flux of Air or CO_2 in NAPL, kg/s
WQ	Z-Direction Energy Flux, W/m^2
WRH	Initial Well Relative Humidity

Table 3.2. (contd)

Variable	Description
WS	Z-Direction Salt Flux, kg/m^2 s
WTM	Component Molecular Weight, kg/kgmol
WTMA	Air or CO_2 Molecular Weight, kg/kgmol
WTMO	Oil Molecular Weight, kg/kgmol
WTMS	Surfactant Molecular Weight, kg/kgmol
WTMW	Water Molecular Weight, kg/kgmol
WWD	Well Total-Liquid Depth, m
WWDL	Well Aqueous Depth, m
WWDN	Well NAPL Depth, m
X	X-Direction Lower Node Surface Position, m
XGA	Air Mass Fraction in Gas
XGA_O	Old Air Mass Fraction in Gas
XGAB	Boundary Air Mass Fraction in Gas
XGAW	Well Air Mass Fraction in Gas
XGAW_O	Old Well Air Mass Fraction in Gas
XGAWB	Well Boundary Air Mass Fraction in Gas
XGO	Oil Mass Fraction in Gas
XGO_O	Old Oil Mass Fraction in Gas
XGOB	Boundary Oil Mass Fraction in Gas
XGW	Water Mass Fraction in Gas
XGW_O	Old Well Oil Mass Fraction in Gas
XGWB	Boundary Water Mass Fraction in Gas
XGWW	Well Water Mass Fraction in Gas
XGWW_O	Old Well Water Mass Fraction in Gas
XGWWB	Well Boundar Water Mass Fraction in Gas
XHA	CO_2 Mass Fraction in Hydrate
XHA_O	Old CO_2 Mass Fraction in Hydrate
XHAB	Boundary CO_2 Mass Fraction in Hydrate
XHO	CH_4 Mass Fraction in Hydrate
XHO_O	Old CH_4 Mass Fraction in Hydrate
XHOB	Boundary CH_4 Mass Fraction in Hydrate
XHW	Water Mass Fraction in Hydrate
XHW_O	Old Water Mass Fraction in Hydrate
XHWB	Boundary Water Mass Fraction in Hydrate
XLA	Dissolved-Air Mass Fraction in Aqueous
XLA_O	Old Dissolved-Air Mass Fraction in Aqueous
XLAB	Boundary Dissolved-Air Mass Fraction in Aqueous
XLAW	Well Dissolved-Air Mass Fraction in Aqueous
XLAW_O	Well Old Dissolved-Air Mass Fraction in Aqueous
XLAWB	Well Boundary Dissolved-Air Mass Fraction in Aqueous
XLCA_TH	CO_2 Large-Cage Occupancy Number
XLCO_TH	CH_4 Large-Cage Occupancy Number

Table 3.2. (contd)

Variable	Description
XLO	Dissolved-Oil Mass Fraction in Aqueous
XLO_O	Oil Dissolved-Oil Mass Fraction in Aqueous
XLOB	Boundary Dissolved-Oil Mass Fraction in Aqueous
XLOW	Well Dissolved-Oil Mass Fraction in Aqueous
XLOW_O	Old Dissolved-Oil Mass Fraction in Aqueous
XLOWB	Well Boundary Dissolved-Oil Mass Fraction in Aqueous
XLS	Dissolved-Salt Mass Fraction in Aqueous
XLS_O	Old Dissolved-Salt Mass Fraction in Aqueous
XLS_TH	Salt Aqueous Mass Fraction Array for CO_2 Property Data Table
XLSB	Boundary Dissolved-Salt Mass Fraction in Aqueous
XLW	Water Mass Fraction in Aqueous
XLW_O	Old Well Dissolved-Salt Mass Fraction in Aqueous
XLWB	Boundary Water Mass Fraction in Aqueous
XLWW	Well Water Mass Fraction in Aqueous
XLWW_O	Old Well Water Mass Fraction in Aqueous
XLWWB	Well Boundary Water Mass Fraction in Aqueous
XMGA	Air Mole Fraction in Gas
XMGAB	Boundary Air Mole Fraction in Gas
XMGO	Oil Mole Fraction in Gas
XMGOB	Boundary Oil Mole Fraction in Gas
XMGW	Water Mole Fraction in Gas
XMGWB	Boundary Oil Mole Fraction in Gas
XMGWW	Well Water-Vapor Mole Fraction in Gas
XMGWWB	Well Boundary Water-Vapor Mole Fraction in Gas
XMLA	Air Mole Fraction in Aqueous
XMLAB	Boundary Air Mole Fraction in Aqueous
XMLAW	Well Dissolved-Air Mole Fraction in Aqueous
XMLAWB	Well Boundary Dissolved-Air Mole Fraction in Aqueous
XMLO	Oil Mole Fraction in Aqueous
XMLOB	Boundary Oil Mole Fraction in Aqueous
XMLOW	Well Dissolved-Oil Mole Fraction in Aqueous
XMLOWB	Well Boundary Dissolved-Oil Mole Fraction in Aqueous
XMLS	Salt Mole Fraction in Aqueous
XMLSB	Bound. Salt Mole Frac. in Aqueous
XMLW	Water Mole Fraction in Aqueous
XMLWB	Bound. Water Mole Fraction in Aqueous
XMNA	Alcohol Mole Fraction in NAPL
XMNAB	Boundary Alcohol Mole Fraction in NAPL
XMNO	Oil Mole Fraction in NAPL
XMNO	Boundary Oil Mole Fraction in NAPL
XMNW	Water Mole Fraction in NAPL
XMNWB	Boundary Water Mole Fraction in NAPL

Table 3.2. (contd)

Variable	Description
XNA	Alcohol Mass Fraction in NAPL
XNA_O	Old Alcohol Mass Fraction in NAPL
XNAB	Boundary Alcohol Mass Fraction in NAPL
XNO	Oil Mass Fraction in NAPL
XNO_O	Old Oil Mass Fraction in NAPL
XNOB	Boundary Oil Mass Fraction in NAPL
XNW	Water Mass Fraction in NAPL
XNW_O	Old Water Mass Fraction in NAPL
XNWB	Boundary Water Mass Fraction in NAPL
XP	X-Direction Node Centroid Position, m
XPG	X-Direction Node Centroid Position at Local Grid Level, m
XREF	Generalized Curvilinear Coordinate Factor
XREFU	Generalized Curvilinear Coordinate Factor
XSCA_TH	CO_2 Small-Cage Occupancy Number
XSCO_TH	CH_4 Small-Cage Occupancy Number
XSO_O	Sorbed Oil Mass Fraction
XSO_O	Old Sorbed Oil Mass Fraction
XSOB	Boundary Sorbed Oil Mass Fraction
Y	Y-Direction Lower Node Surface Position, m
YG	Fraction of Solute in Gas
YGB	Boundary Fraction of Solute in Gas
YL	Fraction of Solute in Aqueous
YLB	Boundary Fraction of Solute in Aqueous
YLS	Fraction of Dissolved-Salt in Aqueous
YLS_O	Old Fraction of Dissolved-Salt in Aqueous
YLSB	Boundary Fraction of Dissolved-Salt in Aqueous
YMGO_TH	CH_4 Mass Fraction of Hydrate Formers in Gas
YMHO_TH	CH_4 Mass Fraction of Hydrate Formers in Hydrate
YMLOW	Well Dissolved-Oil Mole Fraction in Aqueous
YN	Fraction of Solute in NAPL
YNB	Boundary Fraction of Solute in NAPL
YP	Y-Direction Node Centroid Position, m
YPG	Y-Direction Node Centroid Position at Local Grid Level, m
Z	Z-Direction Lower Node Surface Position, m
ZCRA	Alcohol Critical Compressibility Factor
ZCRO	Oil Critical Compressibility Factor
ZCRS	Surfactant Critical Compressibility Factor
ZCRW	Water Critical Compressibility Factor
ZERO	Constant, 0.
ZLAB	Micro-Emulsion Volume Fraction

Table 3.2. (contd)

Variable	Description
ZP	Z-Direction Node Centroid Position, m
ZPG	Z-Direction Node Centroid Position at Local Grid Level, m
ZRA	Component Critical Compressibility Factor
ZRAA	Alcohol Rackett Compressibility Factor
ZRAO	Oil Rackett Compressibility Factor

Table 3.3. Primary Variable Sequencing Scheme

Primary Variable Sequence					
STOMP Mode	Equation 1	Equation 2	Equation 3	Equation 4	Equation 5
STOMP-W; W-R; W-Sc	Water				
STOMP-WA	Water	Air			
STOMP-WAE; WAE-B; WAE-Sc	Energy	Water	Air		
STOMP-WO; WOM	Water	Oil			
STOMP-WOA; WOA-Sc	Water	Air	Oil		
STOMP-WOAE	Energy	Water	Air	Oil	
STOMP-WOD	Water	Oil	Dissolved-Oil		
STOMP-WS; WS-Sc	Water	Salt			
STOMP-WAS	Water	Air	Salt		
STOMP-WASE	Energy	Water	Air	Salt	
STOMP-WCS; WCS-R; WCS-R-Sc; WCS-Sc	Water	CO_2	NaCl		
STOMP-WCSE	Energy	Water	CO_2	Salt	
STOMP-WCMSE	Energy	Water	CH_4	CO_2	Salt

Table 3.4. Field Variable Indexing Scheme

Index	Description
1	Previous time step value
2	Current iteration or time step value
3	First primary variable incremented
4	Second primary variable incremented
5	Third primary variable incremented
6	Fourth primary variable incremented

Table 3.5. Flux Variable Indexing Scheme

Index	Description
1	Current iteration or time step value
2	First primary variable in the upper node incremented
3	First primary variable in the lower node incremented
4	Second primary variable in the upper node incremented
5	Second primary variable in the lower node incremented
6	Third primary variable in the upper node incremented
7	Third primary variable in the lower node incremented
8	Fourth primary variable in the upper node incremented
9	Fourth primary variable in the lower node incremented

4.0 Input File

The STOMP simulator is controlled through a text file, which must be titled *input* for proper execution. This input file has a structured format composed of cards, which contain associated groups of input data. Depending on the operational mode, input cards maybe required, optional, or unused. Required cards must be present in an input file. Optional cards are not strictly required to execute the simulator, but may be required to execute a particular problem. Unused cards are treated as additional text that is unrecognized by the simulator, but will not hinder a proper execution. Cards may appear in any order within the input file. However, the data structure within a card is critical and must follow the formatting directives shown in Appendix A and B. Data structures within cards vary with operational mode, which requires the user to follow a series of logic type statements to construct a readable input file. The simulator contains logic to capture and report some input errors. These capabilities are primarily limited to indicating syntax or formatting type errors and will generally not reveal errors such as those associated with ill-posed problems, atypical parameters, or inappropriate grid structures, for example. Because of its text format, STOMP input files may be generated with text editors, word processors, spreadsheet programs, or graphical user interfaces and are portable between computing environments. The simulator's read routines are case insensitive (e.g., SAND and sand are equivalent) and allow considerable flexibility in specifying the simulation directives; however, close attention to the formatting instructions in this section will be necessary to prepare an executable input file.

4.1 Input File Structure

A STOMP input file is composed of cards, some of which are required and others which are optional or unused. The number of required cards depends on the operational mode. If an attempt is made to execute the simulator on an *input* file with an incomplete set of required cards, an error message will be generated and the code execution will stop. Optional cards are used to specify STOMP capabilities that may be required to execute a particular problem or generate desired output data. These cards are considered optional, because the capabilities accessed through these cards are not necessarily required to execute the code. Execution of the simulator on input files with an incomplete set of optional cards yields messages, which will note the missing optional cards but allow the execution to continue. A summary of the required and optional cards which compose a STOMP input file for each operational mode is shown in Table 4.1.

Each card begins with a header, which must contain a tilde symbol in the first column followed by the card name (e.g., ~Simulation Title Card). Cards may be arranged in any order within an input file; however, the input format within a card is structured. If a card type is repeated within an input file only the first card will be read by the simulator; the other cards of similar type will be ignored. Blank lines or additional comment lines may be included in the input file outside of the card structures.

Table 4.1. Required and Optional Input Cards

Operational Mode	Card Name	Card Status
All	Aqueous Relative Permeability Boundary Conditions Directional Aqueous Permeability Grid Hydraulic Properties Inactive Nodes Initial Conditions Mechanical Properties Output Control Rock/Soil Zonation Saturation Function Simulation Title Surface Flux Solute/Fluid Interactions Solute/Porous Media Interactions Solution Control Source Surface	Required Optional Optional Required Required Optional Optional Required Optional Required Required Required Optional Optional Optional Required Optional Optional
STOMP-W, STOMP-W-Sc	Observed Data Scaling Factor UCODE Control	Optional Optional Optional
STOMP-W-R	Aqueous Species Conservation Equation Equilibrium Equations Equilibrium Reactions Gas Species Kinetic Equations Kinetic Reactions Lithology Observed Data Scaling Factor Solid Species Species Link UCODE Control	Optional Optional Optional Optional Optional Optional Optional Optional Optional Optional Optional Optional Optional
STOMP-WA	Gas Relative Permeability	Required
STOMP-WAE, STOMP-WAE-Sc	Gas Relative Permeability Thermal Properties	Required Required
STOMP-WAE-B	Atmospheric Conditions Gas Relative Permeability Observed Data Plant Thermal Properties	Required Required Optional Optional Required
STOMP-WO	Dissolved-Oil Transport NAPL Relative Permeability Function Oil Properties Well	Optional Required Required Optional

Table 4.1. (contd)

Operational Mode	Card Name	Card Status
STOMP-WOA	Dissolved-Oil Transport Gas Relative Permeability Function NAPL Relative Permeability Function Oil Properties	Optional Required Required Required
STOMP-WOAE	Dissolved-Oil Transport Gas Relative Permeability Function NAPL Relative Permeability Function Oil Properties Thermal Properties	Optional Required Required Required Required
STOMP-WOD	Dissolved-Oil Transport NAPL Relative Permeability Function Oil Properties	Required Required Required
STOMP-WS STOMP-WS-Sc	Salt Transport	Required
STOMP-WAS	Gas Relative Permeability Function Salt Transport	Required Required
STOMP-WASE	Gas Relative Permeability Function Salt Transport Thermal Properties	Required Required Required
STOMP-WCS, STOMP-WCS-Sc	Gas Relative Permeability Function Salt Transport	Required Required
STOMP-WCS-R, STOMP-WCS-R-Sc	Gas Relative Permeability Function Aqueous Species Conservation Equation Equilibrium Equations Equilibrium Reactions Gas Species Kinetic Equations Kinetic Reactions Lithology Salt Transport Solid Species Species Link	Required Optional Optional Optional Optional Optional Optional Optional Optional Required Optional Optional
STOMP-WCSE	Gas Relative Permeability Function Salt Transport Thermal Properties	Required Required Required
STOMP-WCMSE	Gas Relative Permeability Function Salt Transport Thermal Properties	Required Required Required

4.2 Formatting and Notation

Every input card has formatting specifications, which must be followed to create an input file that is readable by the simulator. The format structure of certain cards varies with the operational mode and/or other input options, and the user must strictly follow the formatting instructions for the particular operational mode of interest. As may be expected, the complexity of the input increases with the number of solved equations. Guides for formatting input cards and card lines are given in Appendix B. The formatting guides shown in the Appendix were written in a compact format using special notations. Definitions of the formatting notation used in Appendix B are given in Appendix A.

Input files are organized into three hierarchical structures, cards, lines, and data. Cards are delimited by a tilde (~) in the first column of the card title. Lines are delimited by hard returns and data are delimited by commas. The simulator recognizes three different types of data: integers, reals and character strings. Integers are used primarily to indicate indices or integral numbers. Integer data must be entered without decimal points or exponential notation. Real data are used to indicate dimensional parameters and can contain decimal points and/or exponential notation. Character string data are primarily used to indicate names, options and units, and are limited to 64 characters unless otherwise noted. Card lines comprise a series of input data items delimited by commas. A comma at the end of the card line is required to close the last data item. The format structure for each line in a card is shown in Appendix B using a format guide (e.g., **Format**: *Integer*^{a}, *Real*^{b}, *Char*^{c},). Data types in the format statements are indicated as *Integer*, *Real*, and *Char* (i.e., integer, real and character string) with lowercase letter superscripts. The lettered superscripts are used to make correspondence between the data item and its position in the input line. The total length a single input line cannot exceed 132 characters. Additional text (comments or notes) may appear after the closing comma of an input line. Many input variables contain default values (indicated with an underline in Appendix B). Default values can be accessed by using a *null* entry. The *null* entry requires that the closing comma for the input data item immediately follow the previous closing comma or only blank spaces fill the space between the closing commas. To start a line with a *null* entry, the closing comma can occur in column 1 or after a number of blank spaces. Example input cards are shown in Appendix B after the formatting instructions for each card. A considerable portion of the input file formatting and creation work can be eliminated by developing input files from previously generated files or through an input generator.

4.3 Units

The simulator offers considerable flexible in specifying units. The user can declare units for both input or output data. Unspecified units will be assumed to be in standard Systeme International (SI) units for the data item. The simulator operates internally almost exclusively in SI units, with pressures expressed in gauge, relative to 1 atm. Unless specifically stated, all input and output pressures are expressed in absolute values. Unit variables are read by the simulator as character strings, translated into primary unit form, and compared against the standard unit form for each data item. During the translation to primary unit form a conversion factor to SI units is generated. A unit character string comprises a combination of the recognized units delimited by spaces and/or a single divisor symbol (i.e., /). Only one divisor may appear in a unit character string. Spaces should not be used to separate the units immediately prior to or following the divisor symbol. The unit strings prior to the divisor symbol are considered as part of the numerator; conversely, the unit strings following the divisor symbol are considered as part of the denominator. Units recognized by the simulator are listed in Table 4.2.

Table 4.2. Recognized Units

Notation	Description	SI Equivalent	Conversion to SI	Base Units
1	one	N/A	N/A	N/A
aqu	aqueous phase	N/A	N/A	N/A
aqueous	aqueous phase	N/A	N/A	N/A
atm	atmosphere	Pa	101325.0	$M / L T^2$
bar	bar	Pa	$1. \times 10^5$	$M / L T^2$
btu	BTU	J	1054.4	$M L^2 / T^2$
c	Celsius	C	1.	K
cal	calorie	J	4.184	$M L^2 / T^2$
ci	Curie	N/A	N/A	N/A
cm	centimeter	m	$1. \times 10^{-2}$	M / L T
cp	centipoise	Pa s	$1. \times 10^{-3}$	M / L T
d	day	s	1/86400.	T
darcy	darcy	m^2	9.8697×10^{-13}	L^2
day	day	s	1/86400.	T
debyes	Debyes	N/A	N/A	$(M/L T^2)^{1/2}$
deg	degrees	rad	0.017453	N/A
degree	degree	rad	0.017453	N/A
dyn	dynes	N	$1. \times 10^{-5}$	$M L / T^2$
dynes	dynes	N	$1. \times 10^{-5}$	$M L / T^2$
f	Fahrenheit	C	(f-32) x (5/9)	K
ft	foot	m	0.3048	L
furlong	furlong	m	2.01168×10^2	L
g	gm	kg	$1. \times 10^{-3}$	M
gal	gallon	m^3	3.7854×10^{-3}	L^3
gas	gas phase	N/A	N/A	N/A
gm	gm	kg	$1. \times 10^{-3}$	M
gram	gm	kg	$1. \times 10^{-3}$	M
hc	hydraulic cond.	N/A	1.03910×10^{-7}	N/A
hour	hour	s	1/3600.	T
hp	horsepower	W	745.7	$M L^2 / T^3$
hr	hour	s	1/3600.	T
in	inch	m	2.54×10^{-2}	L
j	Joule	J	1.	$M L^2 / T^2$
k	kelvin	C	1.	K
kg	kilogram	kg	1.	M
kgmol	kilogram-mole	kgmol	1.	Mo
l	liter	m3	$1. \times 10^{-3}$	L^3
langley	Langley	J/m^2	4.186×10^{-6}	M / T^2
lb	pound	kg	0.45359	M
lbf	pound-mass force	N	4.4482	$M L / T^2$
lbm	pound mass	kg	0.45359	M
lbmol	pound-mol	kgmol	0.45359	Mo

Table 4.2. (contd)

Notation	Description	SI Equivalent	Conversion to SI	Base Units
liq	liquid	N/A	N/A	N/A
liter	liter	m^3	$1. \times 10^{-3}$	L^3
m	meter	m	1.	L
mg	milligram	kg	1.0×10^{-6}	M
min	minute	s	1/60.	T
ml	milliliter	m^3	1.0×10^{-6}	L^3
mm	millimeter	m	1.0×10^{-3}	L
mol	mole	kgmol	1.0×10^{-3}	Mo
mole	mole	kgmol	1.0×10^{-3}	Mo
n	Newton	N	1.	$M L / T^2$
napl	NAPL	N/A	N/A	N/A
newton	Newton	N	1.	$M L / T^2$
oil	oil	N/A	N/A	N/A
p	Poise	Pa s	1.0×10^{-1}	M / L T
pa	Pascal	Pa	1.0	$M / L T^2$
pci	picoCuries	N/A	N/A	N/A
plant	plant	N/A	N/A	N/A
psf	lbs/ft^2	Pa	4.7880556×10^1	$M / L T^2$
psi	lbs/in^2	Pa	6894.8	$M / L T^2$
r	Rakine	K		K
rad	radian	rad	1.	N/A
radian	radian	rad	1.	N/A
rod	rod	m	5.0292	L
s	second	s	1.	T
sec	second	s	1.	T
slug	slug	kg	14.594	M
sol	solid phase	N/A	N/A	N/A
solid	solid phase	N/A	N/A	N/A
voc	VOC phase	N/A	N/A	N/A
w	Watt	W	1.	$M L^2 / T^3$
water	water	N/A	N/A	N/A
week	week	s	1/604800.	T
wh	water head	Pa	9.79353×10^3	$M / L T^2$
wk	week	s	1/604800.	T
yd	yard	m	0.9144	L
year	year	s	1/31557600.	T
yr	year	s	1/31557600.	T

The *hc* unit is uniquely reserved for specifying rock/soil hydraulic conductivities, which are normally expressed in velocity units. Without the *hc* unit the input for rock/soil permeability will be interpreted as an intrinsic permeability value. The simulator does not apply hydraulic conductivity values directly, but instead first converts hydraulic conductivity values into intrinsic permeability values using the density and viscosity of water at 20°C. Therefore, when rock/soil permeability values are read as hydraulic conductivities the associated unit character string should include the *hc* unit (e.g., hc cm/hr). The unit *wh* indicates water equivalent head and can be combined with a length unit for a pressure unit (e.g., wh ft).

As an example, the standard units for thermal conductivity are W/K m, however, a user may prefer to specify thermal conductivity in the standard English units of Btu in/hr ft^2 F. In this case, the user would enter the character string 'Btu in/hr ft^2 F' for the units data item. Note that the carat (i.e., ^) symbol is used to indicate an exponential. During simulator execution, the string of English units for thermal conductivity would be translated into the primary unit form 'm/s^3 kg K' and compared against the standard unit form. If the primary unit form does not agree with the standard unit form, then an error message is generated by the simulator and the program execution stops. Otherwise, if the comparison were successful, then the input data for thermal conductivity, which precedes its unit string, would be converted to SI units.

4.4 Card Descriptions

Formatting instructions for the input cards are provided in Appendix B. This section provides a brief synopsis of each of the 39 input cards with emphasis on its purpose and application. Italicized words refer to specific files, cards, options and data entries shown in the card formats in the Appendix. Input cards are listed in alphabetical order. The cards are:

1. Atmospheric Conditions Card
2. Aqueous Relative Permeability Card
3. Aqueous Species Card
4. Boundary Conditions Card
5. Conservation Equation Card
6. Directional Aqueous Relative Permeability Card
7. Dissolved Oil Transport Card
8. Equilibrium Equations Card
9. Equilibrium Reactions Card
10. Gas Relative Permeability Card
11. Gas Species Card
12. Grid Card
13. Hydraulic Properties Card
14. Inactive Nodes Card
15. Initial Conditions Card
16. Kinetic Equations Card
17. Kinetic Reactions Card
18. Lithology Card
19. Mechanical Properties Card
20. NAPL Relative Permeability Card
21. Observed Data Card
22. Oil Properties Card
23. Output Control Card
24. Plant Properties Card
25. Rock/Soil Zonation Card
26. Salt Transport Card
27. Saturation Function Card
28. Scaling Factor Card
29. Simulation Title Card
30. Solid Species Card
31. Solute/Fluid Interactions Card
32. Solute/Porous Media Interactions Card
33. Solution Control Card
34. Source Card
35. Species Link Card
36. Surface Flux Card
37. Thermal Properties Card
38. UCODE Control Card
39. Well Card

4.4.1 Atmospheric Conditions Card

This card allows the user to specify time-varying atmospheric conditions for temperature, pressure, relative humidity, net solar radiation, and wind speed. At present, data from this card are only used in conjunction with the *Plant Card* and the *STOMP-WAE-B (Water-Air-Energy-Barrier)* operational mode. Detailed information on this mode can be found in Ward et al. (2005). This card consists of two input sections: (1) reference data and (2) time-varying atmospheric conditions data. The reference data inputs include: atmospheric conditions start time; wind speed, air temperature, relative humidity measurement heights, local longitude and meridian; and roughness parameters. The *Atmospheric Start Time*, *Local Longitude*, and *Local Meridian* inputs are used to split total horizontal solar radiation into direct-beam and indirect components. The *Atmospheric Start Time* is also used as the reference time for specifying the plant growth cycle and time dependent albedo. The *Wind Speed Measurement Height* input is used to convert wind speed at the measurement height to wind speed near the ground surface and plant canopy heights. As with the wind speed, the *Air Temperature* and *Relative Humidity Measurement Height* inputs are used to convert their respective data to ground surface and plant canopy heights. The surface roughness of a barrier at multiple scales is determined from the roughness associated with soil particles and grass and shrubs protruding into the air. As the wind flows over the surface, it relates the heat flux to the difference between the aerodynamic temperature at the thermal roughness height and the temperature in the surface layer. Therefore, the *Aerodynamic Roughness Length* and the *Thermal Roughness Lengths* are required parameters.

The time-varying inputs are read in tabular form where, each line of input represents the atmospheric conditions at that point in time. The *Atmospheric Condition Time* input is the first entry in each time-varying input line and is relative to the *Atmospheric Start Time* declared in the reference data input section. The time entry is followed in succession by the *Atmospheric Condition Temperature*, *Atmospheric Condition Pressure, Atmospheric Conditions Relative Humidity*, *Atmospheric Conditions Net Solar Radiation*, and *Atmospheric Conditions Wind Speed* inputs. The Net Solar Radiation input should be total solar radiation incident on a horizontal ground surface. The time-varying inputs can be read directly from the input file or from an external file. Weather data are read as hourly or average daily measurements, which for Hanford-specific simulations can be obtained from the Hanford Meterological Station (HMS). In cases where required atmospheric data are not available, the missing data can be calculated from available data using published theoretical relationships.

4.4.2 Aqueous Relative Permeability Function Cards

This collection of cards allows the user to declare and define a relative permeability-saturation function for the aqueous phase for each defined rock/soil type. Every rock/soil type defined on the *Rock/Soil Zonation Card* must be referenced. With the *IJK Indexing* option, node dependent parameters are entered via external files and node independent parameters are entered directly on the card. Aqueous relative permeability function types and the required input items are dependent on the operational mode. The Mualem and Burdine relative permeability functions are also dependent on the saturation function type and are strictly applicable to the van Genuchten (1980) and Brooks and Corey (1964) functions. For these functions, either the van Genuchten '*m*' parameter or the Brooks and Corey 'λ' parameter can be defaulted to the values entered or defaulted with the saturation function. Functional forms for the relative permeability-saturation functions are preferred. Tabular input is acceptable for some operational modes. By default, tabular data will be interpolated using linear interpolation, whereas values beyond the table limits will be assigned either the table minimum or maximum values appropriately. Refer to the STOMP

Theory Guide (White and Oostrom 2000) for a complete description of aqueous relative permeability-saturation functions. For some operational modes, aqueous relative permeability functions can be specified as a diagonal tensor (i.e., unique aqueous relative permeability functions can be declared for the principal grid-coordinate directions). With this option, a default aqueous relative permeability can be declared for all coordinate directions via the *Aqueous Relative Permeability Card*, then individual directional components can be specified via the directional cards (i.e., *X Aqueous Relative Permeability Card, Y Aqueous Relative Permeability Card,* or *Z Aqueous Relative Permeability Card*).

4.4.3 Aqueous Species Card

This card, used for the STOMP modes with reactive transport, defines the aqueous species to be considered in the simulation. Required input includes the species name, aqueous molecular diffusion coefficient for all species, activity coefficient model option, species charge, species diameter, and species molecular weight. The species name must be unique and distinct from gas and solid species names (e.g., CO2(aq), CO2_aqueous, dissolved CO2, CO2a). Because only component and kinetic species are transported, transport properties, such as diffusion coefficients, are species independent. Currently, the activity coefficient models include Davies, B-Dot, Pitzer, and a constant coefficient option. If the constant coefficient option is chosen then the species charge, diameter, and molecular weight inputs are not required. The reactive transport module ECKEChem is described by White and McGrail (2005).

4.4.4 Boundary Conditions Card

This card allows the user to control the simulation by defining time varying boundary conditions. This card is optional, but is generally necessary to simulate a particular problem. Boundary conditions may be applied to any boundary surface or surface dividing active and inactive nodes. By default, all undeclared boundary surfaces have zero flux boundary conditions for both flow and transport. Boundary conditions may be applied only to surfaces of active nodes. To apply a boundary condition to a boundary surface, the surface is referenced by the adjacent active node and a direction with respect to the adjacent node. To apply a boundary condition to a surface dividing an active and inactive node, the surface is referenced by the active node. Boundary conditions are time varying. The user is not allowed to assign multiple boundary conditions to a boundary surface during the same time period, but multiple boundary conditions can be applied to a boundary surface over different time periods. The simulator controls time steps to agree with time transitions in boundary conditions.

Application of boundary conditions requires an appropriate conceptualization of the physical problem and translation of that conceptualization into boundary condition form. The variety of boundary condition types available in the simulator should afford the user the flexibility to solve most subsurface flow and transport problems. The boundary condition card reader within the simulator performs limited error checking on the boundary condition inputs. An error free boundary condition card does not guarantee the user has not created an ill-posed problem or an execution that will successfully converge. For example, a mistake frequently made by users is to specify infiltration rates at the top of a column with positive fluxes. While this input would be perfectly acceptable to the boundary condition input reader, the specified condition would actually withdraw flux from the top of the column since the *z*-axis and *z*-direction flux are positive in the upward direction.

The *Boundary Surface Direction* is specified with respect to the active node adjacent to a boundary surface. For the Cartesian coordinate system, the terms west, south, and bottom refer to the negative *x*-, *y*-, and *z*-directions, respectively, and the terms east, north, and top refer to the positive *x*-, *y*-, and *z*-directions, respectively. For the cylindrical coordinate system, the terms west, south, and bottom refer to the negative *r*-, *θ*-, and *z*-directions, respectively, and the terms east, north, and top refer to the positive *r*-, *θ*-, and *z*-directions, respectively.

Boundary condition inputs depend on the operational mode, with the required inputs becoming more complex with an increasing number of solved equations. Generally, the number of boundary types that must be declared for each boundary equals the number of solved equations governing flow and transport. The *Dirichlet* boundary type is used to specify a field value (e.g., pressure, temperature, or solute concentration) at the boundary surface. The *Neumann* boundary type allows the user to specify a flux (e.g., liquid phase flux, heat flux, or solute flux) at the boundary surface. The *Zero Flux* boundary type is used to impose no flow and/or transport conditions across the boundary. The *Saturated* boundary type is available only for two-phase conditions and imposes total-liquid saturation conditions (e.g., water table) at the boundary surface. The *Unit Gradient* boundary type imposes hydrostatic conditions across the boundary surface for the specified phase. The *Hydraulic Gradient* boundary type should be applied only to a column or plane of vertical surfaces. With this boundary type, the user specifies a fluid phase pressure at the lowest surfaces of a column or row and the simulator then computes fluid phase pressure for the remaining boundary surfaces assuming hydrostatic conditions for the fluid phase. The *Seepage Face* boundary type is similar to a *Hydraulic Gradient* boundary, but is limited to pressure boundaries of the local gas pressure. This boundary type is designed to model an exposed vertical face that seeps liquids. Liquid can enter a seepage face only for phase pressures that exceed the local gas pressure. The *Initial Conditions* boundary type fixes the boundary field variables (e.g., pressure, temperature, or solute concentration) to the initial value of the field variables of the node adjacent to the boundary surface. This boundary type is invariant with time. *Inflow* and *Outflow* boundary types are applicable only to solute and energy boundary conditions. These boundary types consider only advectively transported solute or energy, while diffusion transport across the boundary surface is neglected. The solute concentration boundary types (e.g., *Volumetric Concentration*, *Aqueous Conc.*, *Gas Conc.*, and *NAPL Conc.*) are equivalent to Dirichlet boundary types for solute transport. These boundary types differ by their definitions of solute concentration.

Time variations of the boundary conditions are controlled through declaring multiple boundary times. All *Boundary Time* inputs are referenced against the *Initial Time* specified in the *Solution Control* card or obtained from a *restart* file. A boundary condition declared with a single *Boundary Time* implies that the boundary condition is time invariant, and the specified *Boundary Time* represents the start time for the boundary condition. Prior to the start time the boundary surface will be assumed to be of type *Zero Flux*. The specified boundary condition will remain in effect from the start time until the execution is completed. If a boundary condition is declared with multiple *Boundary Times*, then the first time listed equals the start time, the last time listed equals the stop time, and the intermediate times are transition points. For simulation times outside of the start and stop time limits, *Zero Flux* boundary conditions apply. For simulation times between two *Boundary Times*, linear interpolation of the boundary conditions is applied. Step boundary condition changes can be simulated by defining duplicate *Boundary Times.* The first time would indicate the completion of the previous boundary condition and the second time would indicate the start of the new boundary condition. At the completion of the step boundary condition, another set of duplicate *Boundary Time* declarations would be used. Step boundary conditions are convenient methods for introducing slugs of fluids, heat, or solute in conjunction with the *Neumann* boundary type.

Regardless of the boundary type, the boundary condition inputs are used to compute phase saturations, phase relative permeabilities, and physical properties at the boundary surfaces. For the more complex operational modes, especially those modes involving three-phase conditions, it is critical for the user to specify boundary conditions that yield appropriate secondary field variables at the boundary surface. For example, a three-phase system with infiltrating NAPL under ponded conditions could be declared as type *Zero Flux* for the aqueous phase, *Zero Flux* for the gas phase, and *Dirichlet* for the NAPL. For this boundary condition scenario, the aqueous and gas pressures requested for the *Zero Flux* type boundary are used with the specified NAPL pressure to compute liquid saturation conditions at the boundary surface. If the pressures specified yield zero NAPL saturation, then the NAPL relative permeability would equal zero and no NAPL would infiltrate across the surface, in spite of the ponding conditions.

For the Reactive Transport modules, this card has been modified to allow for boundary conditions to be specified for reactive species. Chemical reaction systems generally involve large numbers of reactive species, which has necessitated some formatting changes for this card. Reactive species boundary condition types include: *Aqueous Concentration*, *Inflow Aqueous*, *Outflow Aqueous*, *Inflow-Outflow Aqueous Concentration*, *Zero-Flux*, and *Initial Condition*. Equilibrium conditions are assumed between gas and aqueous species. The three principal departures from the conventional formatting for the *Boundary Conditions Card* are (1) a single boundary condition type is applied to all reactive species, (2) not all reactive species concentrations need to be specified, and 3) line returns are permitted to allow for large numbers of reactive species. Required inputs, specific to the reactive species, include the boundary condition type, and then at each time point the number of species, species name, species concentration, and species concentration units. Line returns are allowed during the specification of boundary condition values at each time point. Although individual species concentrations are specified on the boundaries, these concentrations are converted to component and kinetic species before being transported across the boundary surface.

Unique to the SVAT scheme of the *STOMP-WAE-B (Water-Air-Energy-Barrier)* mode is the Bare-Shuttleworth-Wallace and Shuttleworth-Wallace boundary types (Shuttleworth and Wallace 1985) that may be imposed as an energy boundary on the upper surfaces of boundary nodes. The Bare-Shuttleworth-Wallace boundary condition is used to calculate potential evaporation from the bare soil surfaces and the Shuttleworth-Wallace boundary conditions is used to calculate the PET from the canopy and surface. Details of these boundary types are described by Ward et al. (2005). Note that in *STOMP-W (Water)*, the *Potential Evapotranspiration Aqueous* boundary condition is available for flow simulation with evapotranspiration at the surface. This condition requires calculating the PET values externally. This is typically done by applying the Penman equation.

4.4.5 Conservation Equations Card

This card, used for the STOMP modes with reactive transport, specifies the conservation equations to be considered in the simulation. Conservation equations have the following general form:

$$\frac{d\sum\left(a_i\,C_i\right)}{dt} = 0 \tag{4.1}$$

where C_i = concentration of species *i* (expressed as aqueous molar concentration)

a_i = stochiometric coefficient of species *i*

$\sum \left(a_i \, C_i \right)$ = component species concentration (expressed as aqueous molar concentration).

Required inputs include the component species name, number of species in the conservation equation, species names, and species stochiometric coefficients. The component species name must begin with '*Total_*' followed by the species name of a reactive species in the conservation equation (e.g., Total_CO2, Total_H2CO3, Total_H+). This name specification is critical because it links the named species with the conservation equation, making the concentration for that species the primary unknown for the conservation equation. The reactive transport module ECKEChem is described by White and McGrail (2005).

4.4.6 Directional Aqueous Relative Permeability Card

For some operational modes, aqueous relative permeability functions can be specified as a diagonal tensor (i.e., unique aqueous relative permeability functions can be declared for the principal grid-coordinate directions). With this option, a default aqueous relative permeability can be declared for all coordinate directions via the *Aqueous Relative Permeability Card*, then individual directional components can be specified via the directional cards (i.e., *X Aqueous Relative Permeability Card, Y Aqueous Relative Permeability Card,* or *Z Aqueous Relative Permeability Card*). Every rock/soil type defined on the *Rock/Soil Zonation Card* must be referenced. With the *IJK Indexing* option, node dependent parameters are entered via external files and node independent parameters are entered directly on the card.

4.4.7 Dissolved-Oil Transport Card

This card allows the user to declare functions and assign values required for dissolved oil transport. If the operational mode is *STOMP-WOD (Water-Oil-Dissolved Oil)*, then an *Interphase Mass Transfer Function Option* must be declared using either the Welty or Parker function and the associated parameters specified. Properties for dissolved oil dispersion and adsorption must be specified with respect to each rock/soil type. With the *IJK Indexing* option, node dependent parameters are entered via external files and node independent parameters are entered directly on the card. For each, the *Longitudinal Dispersivity* and *Transverse Dispersivity* of dissolved oil must be specified with respect to each rock/soil type, and a *Dissolved-Oil Adsorption Function* declared from among the choices of Linear, Linear $K_{d,}$ Freundlich, or Langmuir functions, with associated parameters for the chosen function specified. Refer to Appendix B for details on the *Interphase Mass Transfer Function* and *Dissolved-Oil Adsorption Function* options and required input parameters specified through this card.

4.4.8 Equilibrium Equations Card

This card, used for the STOMP modes with reactive transport, specifies the equilibrium equations to be considered in the simulation. Equilibrium equations have the following general form:

$$\left(C_i \right) = K_{eq} \prod_{j \neq i} \left(C_j \right)^{e_j} \tag{4.2}$$

where K_{eq} = equilibrium reaction constant specified via the *Equilibrium Reactions Card*

$\left(C_i\right)$ = activity of the equilibrium species *i*

$\left(C_j\right)$ = activity of the species *j* in the equilibrium equation.

Activity coefficients and model options are specified via the *Aqueous Species Card*. The species exponents e_j can be positive or negative. Required inputs include the number of species in the equilibrium equation (including the equilibrium species), species names, equilibrium reaction name, and the species exponents. The equilibrium species (indicated with the subscript *i*) is distinguished from the other species in the equilibrium equation (indicated with the subscript *j*) by being the first species listed for the equilibrium equation. The reactive transport module ECKEChem is described by White and McGrail (2005).

4.4.9 Equilibrium Reactions Card

This card, used for the STOMP modes with reactive transport, specifies the equilibrium reaction constants to be considered in the simulation. Species are associated with the defined equilibrium reaction constants via the *Equilibrium Equations Card*. This card is only used to specify the parameters used in the temperature dependent equations for equilibrium constants. Required inputs include the equilibrium reaction name and equation coefficients for the temperature dependent equilibrium constant. The equilibrium reaction name must be unique and distinct from kinetic reaction names (e.g., EqRc-1, E1, Equil-Reac-1, er-1). The reactive transport module ECKEChem is described by White and McGrail (2005).

4.4.10 Gas Relative Permeability Function Card

This card is used to declare and define a relative permeability-saturation function for the gas phase for each defined rock/soil type. Every rock/soil type defined on the *Rock/Soil Zonation Card* must be referenced. With the *IJK Indexing* option, node dependent parameters are entered via external files and node independent parameters are entered directly on the card. This card is required only for simulations involving flow and/or transport through the gas phase. Gas relative permeability functions and the required input items depend on the operational mode. The Mualem and Burdine relative permeability functions are also dependent on the saturation function type and are strictly applicable to the van Genuchten and Brooks and Corey functions. For these functions, either the van Genuchten '*m*' parameter or the Brooks and Corey 'λ' parameter can be defaulted to the values entered or defaulted with the saturation function. Functional forms for the relative permeability-saturation functions are preferred, however tabular input is acceptable. By default, tabular data will be interpolated using linear interpolation; cubic spline interpolation is optional; where, values beyond the table limits will be assigned either the table minimum or maximum values appropriately. Refer to the STOMP Theory Guide (White and Oostrom 2000) for a complete description of gas relative permeability and saturation functions.

4.4.11 Gas Species Card

This card, used for the STOMP modes with reactive transport, defines the gas species to be considered in the simulation. Required input includes the species name, gas molecular diffusion coefficient for all species, associate aqueous species name, and gas-aqueous partitioning parameters. The species name must be unique and distinct from aqueous and solid species names (e.g., CO2(g), CO2_gas, gaseous CO2, CO2g). All gas species must be associated with an aqueous species, with which it is in equilibrium. Equilibrium partitioning between the gas and aqueous species is specified via a temperature dependent function, whose parameters are entered on this card. The reactive transport module ECKEChem is described by White and McGrail (2005).

4.4.12 Grid Card

The STOMP simulator's finite-difference formulation is based on orthogonal grid systems. Currently, three orthogonal grid systems are recognized: Cartesian, cylindrical, and boundary fitted (orthogonal). The Cartesian coordinate system is a "right-handed" system with the longitudinal axis (*z*–direction) aligned with the negative gravitational vector. Cartesian coordinate systems may be defined that are tilted with respect to the gravitational vector. The cylindrical coordinate system has the longitudinal axis (*z*-direction) aligned with the negative gravitational vector. The radial (*r*–direction) and azimuthal (θ-direction) axes are constrained to a horizontal plane. The boundary fitted (orthogonal) coordinate system allows for curvilinear boundaries. This coordinate system requires the number of grid points in each direction as input variables. The coordinates are read in via a user defined text file.

For the Cartesian coordinate system, the terms west, south, and bottom refer to the negative *x*-, *y*-, and *z*-directions, respectively, and the terms east, north, and top refer to the positive *x*-, *y*-, and *z*-directions, respectively. For the cylindrical coordinate system, the terms west, south, and bottom refer to the negative *r*-, θ-, and *z*-directions, respectively, and the terms east, north, and top refer to the positive *r*-, θ-, and *z*-directions, respectively. Negative dimensional values are not recognized, and axes are defined positive towards increasing node numbers. The grid dimensions that are specified on the *Grid Card* refer to node surfaces; therefore, for grids with non-uniform spacing one plus the number of nodes entries are required for each grid direction. Node volumes are defined by their bounding surfaces. Cylindrical coordinates systems are restricted to azimuthal axes which are less than or equal to 360 degrees. Refer to the STOMP Theory Guide (White and Oostrom 2000) for graphical descriptions of the Cartesian and cylindrical coordinate systems.

4.4.13 Hydraulic Properties Card

This card allows the user to assign values to the intrinsic permeability of each defined rock/soil type. Every rock/soil type defined on the Rock/Soil Zonation Card must be referenced. With the *IJK Indexing* option, node dependent parameters are entered via external files and node independent parameters are entered directly on the card. Intrinsic permeability can be declared directly or through entering the hydraulic conductivity at reference conditions, where reference conditions refer to atmospheric pressure and 20°C. By default, the simulator reads the permeability values on this card as intrinsic permeabilities, unless the character string *hc* is included in the associated units. Default units of m^2 are applied to null entries for the units associated with permeability values. A primary assumption of the simulator is that principal components of the intrinsic permeability tensor are aligned with the principal coordinate directions. For cylindrical coordinate systems the radial, azimuthal, and vertical permeabilities correspond with the *x*-, *y*-, and *z*-direction values, respectively. Refer to the STOMP Theory Guide (White and Oostrom 2000) for a description of the conversion of hydraulic conductivity at reference conditions to intrinsic permeability. The key words 'dp,' 'dual porosity,' or 'fractured' in the rock/soil name indicate a dual porosity medium, which triggers the reading of both matrix and fracture properties (e.g., *Fracture and Matrix Intrinsic Permeability*). Reduction in intrinsic permeability with precipitation of salt, as computed in the *STOMP-WCS (Water-CO_2-NaCl)* and *STOMP-WCSE (Water-CO_2-Salt-Energy)* operational modes requires two parameters, which are entered via this card: *Pore-Body Fractional Length* and *Fractional Critical Porosity*.

4.4.14 Inactive Nodes Card

This card allows the user to declare nodes, within the computational domain, as inactive or non-computational. Inactive nodes are those nodes that will remain permanently excluded from the computational domain. Because inactive nodes reduce the number of unknowns and, therefore, increase execution speed, their use is recommended. Inactive nodes can be used to simulate irregular boundaries or define impervious zones. Inactive nodes can also be used to define internal boundary surfaces. Boundary conditions may be applied to any surface between an active and inactive node. Boundary condition definitions, however, must refer to an active node. Examples of appropriate uses for inactive nodes include representation of the atmosphere above a sloping surface, or conversely a non-horizontal water table along the bottom of a domain. To facilitate user input, a number of features have been incorporated into this card: explicit declaration, external files, rock/soil type association, and multiple entries. Using explicit declaration the user declares domains of nodes as being inactive. Declaring inactive nodes via external files can be useful when the inactive nodes domains are automatically generated. As with the external files option, the rock/soil type association can be useful when the rock/soil zonation has been entered via external files. As no rock/soil types are defined for *IJK Indexing*, rock/soil type association is not recognized in conjunction with *IJK Indexing*. The multiple entries option allows a combination of input options on a single *Inactive Nodes Card* (e.g., explicit declaration and rock/soil type association).

4.4.15 Initial Conditions Card

This card allows the user to assign starting values to both primary and secondary field variables. The current version of the simulator does not allow the user to initialize flux and/or surface variables. Such initializations would be useful for solving transport problems with steady flow fields. For *restart* simulations, initial conditions are obtained from the *restart* file, unless the word *"Overwrite"* appears with the initial condition variable name. Not all initial condition variables can be overwritten. Those variables that can be overwritten are indicated in Appendix B with the optional text string [Overwrite] show in the *Variable Name Option.* Transport problem simulations with the simulator require the solution of the flow fields. The variables that may be initialized depend on the operational mode. For all operational modes, default values for the initial conditions have been specified within STOMP as shown in Table 4.3. For the two-phase, gas-aqueous operational modes (e.g., *STOMP-WA (Water-Air), STOMP-WCS (Water-CO_2-Salt)*, STOMP allows the user to specify the initial aqueous saturation by assigning initial conditions for any two of the following three variables: gas pressure, aqueous pressure, and aqueous saturation. The unassigned variable will be computed from the other two through the declared saturation-capillary pressure functions.

Table 4.3. STOMP Initial Conditions Default Values

Initial Field Variable	Symbol	Default Value
Temperature	T	20.D+0
Pressure of phase *j*	P_j	101325.D+0
Saturation of phase *j*	s_j	0.D+0
Mole fraction of component *i* in phase *j*	χ^i_j	0.D+0
Solute concentration in phase γ	C_γ	0.D+0
Salt concentration in phase γ	S_γ	0.D+0

No initial saturation options are allowed for the three-phase, aqueous-NAPL-gas, operational modes (e.g., *STOMP-WO (Water-Oil), STOMP-WOA(Water-Oil-Air)*). For these three-phase operational modes, initial saturations must be declared by specifying initial gas, aqueous, and NAPL pressures. If the initial NAPL pressure is assigned a value below the critical point, which signifies no NAPL, then the initial NAPL pressure is reset within the simulator to the critical pressure. Therefore, conditions without NAPL can be specified by initializing the NAPL pressure to any value below the aqueous pressure. Initial conditions may be declared repeatedly for a node with the last definition being applied. Initial conditions for solutes are expressed in terms of solute per unit volume where the volume can refer to the total node volume (*Volumetric*), the aqueous-phase volume (*Aqueous Volumetric*), the gas-phase volume (*Gas Volumetric*), or the NAPL volume (*NAPL Volumetric*). Solute units are undefined and can be expressed as the user chooses (e.g., Ci, pCi, gm, kg, mol, kgmol). Although units for expressing solute quantity may vary among solutes, units must be consistent for a single solute among all input data entries.

The gradient utilities of the initial condition card are invaluable and should be used to advantage when possible. Gradients to the initial conditions allow the user to specify that the initial value of field variables varies along one or more directions in the physical domain. The initial condition variable assigned to an initial condition domain applies to the node with the lowest *x*-, *y*-, and *z*-direction indices. If non-zero gradient values are specified, then the initial condition values will vary according to the gradients specified for each physical direction. Default values for the initial condition gradients are zero, indicating no variation. Gradient values are applicable only over the state initial condition domain. An example application of the initial condition gradient utility occurs for problems which start with hydrostatic conditions. For these problems, the pressure at the lowest *z*-direction node positions would be entered with a *Z-Dir. Gradient* that is equal to the product of the fluid's density times gravitational acceleration. For water at 20°C, a *z*–direction gradient of -9793.5192 1/m will yield constant head conditions, which would be equivalent to hydrostatic conditions. This approach could be used to locate the position of a water table under no-recharge equilibrium conditions given the rock/soil saturation function properties and knowledge of a single pressure using an *Initial Condition* execution mode simulation.

For the Reactive Transport modules, this card has been modified to allow for reactive species concentrations to be initialized. Initial aqueous or solid species concentrations can be entered as node volume molar, aqueous molar or aqueous molal. Initial gas species concentrations can be entered as node volume molar, gas molar, aqueous molar or aqueous molal. Required inputs are the species name, concentration type specification, concentration value, concentration units, concentration gradients, and domain region. To distinguish reactive species and transported solutes, the keyword *Species* needs to be included in the concentration type specification (e.g., *Species Aqueous Volumetric Concentration*, *Species Aqueous Molal*). Formatting requirements for specifying initial species concentrations are identical to those for specifying initial solute concentrations. Internally, all species concentrations, including gas and solid species, are stored as aqueous molar concentrations (i.e., mole/m$^3_{\text{aqueous}}$).

4.4.16 Kinetic Equations Card

This card, used for the STOMP modes with reactive transport, specifies the kinetic equations to be considered in the simulation. Kinetic equations have the following general form:

$$\frac{d \sum \left(b_i \, C_i \right)}{dt} = \sum \left(c_j \, R_j \right) \tag{4.3}$$

where C_i = concentration of species *i* (expressed as aqueous molar concentration)

b_i = stochiometric coefficient of species *i*

$\sum \left(b_i \, C_i \right)$ = kinetic species concentration (expressed as aqueous molar concentration)

R_j = kinetic-reaction rate

c_j = kinetic-reaction-rate coefficient.

Required inputs include the number of species, species names, species stochiometric coefficients, number of kinetic-reaction rates, kinetic reaction names, and kinetic-reaction-rate coefficients. Kinetic-reaction names and associated rate functions are defined on the *Kinetic Reactions Card*. The reactive transport module ECKEChem is described by White and McGrail (2005).

4.4.17 Kinetic Reactions Card

This card, used for the STOMP modes with reactive transport, specifies the kinetic reactions to be considered in the simulation. Required inputs include the kinetic reaction name, kinetic reaction type, number of reactant species, reactant species names, reactant species stochiometric coefficients, number of product species, product species names, product species stochiometric coefficients, and kinetic reaction parameters. The kinetic reaction name must be unique and distinct from equilibrium reaction names (e.g., KnRc-1, K1, Kin-Reac-1, kr-1). The required kinetic reaction parameters depend on the specified *Kinetic Reaction Type* option. For the *Precipitation-Dissolution* kinetic reaction type, the required reaction parameters include the reference forward and backward reaction rates, reaction activation energy, reference temperature, and equation coefficients for the temperature dependent equilibrium constant.

For the *Forward-Backward* kinetic reaction type, the required reaction parameters include the forward and backward rate exponents. For the *Valocchi-Monod* kinetic reaction type, the required reaction parameters include the acceptor half-saturation constant, maximum specific rate of substrate utilization, microbial yield coefficient, and first-order microbial decay coefficient. For the *Valocchi-*

Sorption kinetic reaction type, the required reaction parameters include the mass transfer and distribution coefficients. The reactive transport module ECKEChem is described by White and McGrail (2005).

4.4.18 Lithology Card

This card, used for the STOMP modes with reactive transport, specifies the mineral specific areas and species volume fractions for every rock/soil type defined on the *Rock/Soil Zonation Card*. This card is only required when specific kinetic reactions types are specified on the *Kinetic Reactions Card*. Currently, this card is only required when the *Precipitation-Dissolution* kinetic reaction type is specified for the *Kinetic Reaction Type* option. Required input for each rock/soil type includes the primary and secondary mineral specific area, and species volume fractions for every solid species defined on the *Solid Species Card*. The reactive transport module ECKEChem is described by White and McGrail (2005).

4.4.19 Mechanical Properties Card

This card allows the user to assign values to the particle density, porosity, specific storativity, compressibility, and tortuosity function for each defined rock/soil type. Every rock/soil type defined on the *Rock/Soil Zonation Card* must be referenced. With the *IJK Indexing* option, node dependent parameters are entered via external files and node independent parameters are entered directly on the card. *Particle Density* represents the rock grain density. This value will default to 2650 kg/m^3 by using a *null* entry for both the particle density and its associated unit. *Total Porosity* refers to total connected and unconnected pore volumes. *Diffusive Porosity* refers to only the connected pore volume. As with *Particle Density*, the *Specific Storativity* will be the default value by using a *null* entry for both the *Specific Storativity* and its associated *Units*. Default specific storativity is computed from the *Diffusive Porosity* and a default value of 1.x10^{-7} 1/Pa for the compressibility. Alternatively, the *Pore Compressibility* can be read directly along with a *Pore Compressibility Reference Pressure*. Tortuosity functions are required for simulations that involve solute transport or diffusion of components through phases (e.g., water vapor diffusing through the gas phase or dissolved oil diffusing through the aqueous phase). Tortuosities can be computed either as constants, which require input values, or as functions of the phase saturation and diffusive porosity through the Millington and Quirk function. Refer to the STOMP Theory Guide (White and Oostrom 2000) for a description of the Millington and Quirk tortuosity function. The key words 'dp,' 'dual porosity,' or 'fractured' in the rock/soil name indicate a dual porosity medium, which triggers the reading of both matrix and fracture properties (e.g., *Fracture and Matrix Diffusive Porosity*, *Fracture and Matrix Specific Storativity*).

4.4.20 NAPL Relative Permeability Function Card

This card allows the user to declare and define a relative permeability-saturation function for the NAPL for each defined rock/soil type. Every rock/soil type defined on the *Rock/Soil Zonation Card* must be referenced. With the *IJK Indexing* option, node dependent parameters are entered via external files, and node independent parameters are entered directly on the card. This card is required only for simulations involving flow and/or transport through the NAPL phase. The Mualem and Burdine relative permeability functions for the NAPL phase depend on the saturation function type and are strictly applicable to the van Genuchten and Brooks and Corey functions. For these functions, either the van Genuchten '*m*' parameter or the Brooks and Corey 'λ' parameter can be defaulted to the values entered or defaulted with the saturation function. Refer to the STOMP Theory Guide (White and Oostrom 2000) for a complete description of the gas relative permeability-saturation functions.

4.4.21 Observed Data Card

Parameter estimation with the STOMP simulator is executed using inverse modeling techniques in conjunction with UCODE (Poeter and Hill 1998), a computer code for universal inverse modeling. In the current version of the STOMP, simulator inverse modeling is only an option for the *STOMP-W (Water)* and *STOMP-WAE (Water-Air-Energy)* operational modes. Inverse modeling involves repetitive forward modeling, whereas each forward realization differs by increments in the parameters of interest. Simulation results are compared against observed data to compute error.

This card is divided into two sections: one for specifying observed data parameters and the other for entering field (laboratory) observation data. Observed data can be field parameters (i.e., aqueous pressure, aqueous saturation, aqueous moisture content, solute concentrations); flux rate parameters (i.e., aqueous volumetric flux, solute flux); or flux integral parameters (i.e., aqueous volumetric flux integral, solute flux integral). Observed data is defined by the parameter type, physical location, statistical index, and time and space weighting factors. Physical locations for field parameters can be specified as *x*, *y*, *z* (*r*, *θ, z*) coordinates or an *i*, *j*, *k* node index, whereas the node indexing refers to the centroid of the grid cell. Flux rate and integral parameters can only be specified by defining surfaces as with the *Surface Flux Card* (i.e., a surface direction and a domain of *i*, *j*, *k* indexed nodes). As with the *Surface Flux Card*, surfaces can span over a range of nodes and the reported surface flux values will be the summed quantity (i.e., the flux rate or integral across the entire surface). In the first section of the card, observed data parameters are defined and located within the computational domain. In the second section of the card, experimental data for the defined parameter is read, either directly from the input file or from an external file. The observed experimental data includes a time and observed data value, whereas the time corresponds with the simulation time. STOMP does not compute or assess the error between the computed and experimental observed data, but it simply records these data to a file. For reference data, specified as *x*, *y*, *z* (*r*, *θ, z*) coordinates, tri-linear interpolation is used to compute the observed data parameter at the specified physical location.

4.4.22 Oil Properties Card

This card allows the user to declare functions and assign values required to compute the physical properties for an oil or dissolved oil. This card is required only for simulations involving flow and/or transport of an oil or dissolved oil. Properties for oils are computed from critical properties and functional parameters following the Corresponding States Method as presented by Reid et al. (1987). Because the oil property relationships within the simulator were primarily based on the formulations in the reference book by Reid et al. (1987), the user is encouraged to obtain or borrow a copy of this reference book. Many of the requested input parameters for numerous oils may be found in the appendix of this reference. Unless specifically stated as an optional input, all parameters requested by this card should be considered as required. The *Freezing Point Temperature* is an optional input and will be the default value with a *null* entry for the parameter and its associated units. The *Boiling Point Temperature* is used in computing oil component diffusion coefficients, oil vapor enthalpy, and liquid oil thermal conductivity; therefore, its value and associated units will be the default value with *null* entries for simulations, which do not involve oil component diffusion or solution of the energy conservation equation. The *Critical Molar Volume* and its associated units are used only to calculate oil component diffusion coefficients and can be defaulted with *null* entries for simulations not involving oil component diffusion. The *Critical Compressibility* and *Dipole Moment* are used only to calculate the gas viscosity and can be defaulted with *null* entries for simulations that do not solve the gas flow equations. The *Pitzner Accentric Factor* and *Isobaric Molar Specific Heat Constants* are used only to calculate the liquid oil enthalpy and can be defaulted with *null* entries for simulations which do not involve heat transfer or solution of the energy conservation equation. Refer to the STOMP Theory Guide (White and Oostrom 2000) for a complete description of the Corresponding States Method functions used to compute the oil property functions.

4.4.23 Output Control Card

This card allows the user to control output written to the *output* file, *plot* file, and *screen* (i.e., standard input/output device (STDIO)). The *output* file contains an interpreted and reformatted version of the *input* and simulation results for selected variables at selected *reference nodes* over the simulation period. The *plot.n* file contains values of geometric parameters and selected variables for the entire computational domain (both active and inactive nodes) at selected simulation times. A *plot.n* file will always be generated at the conclusion of an execution. The output to the STDIO primarily comprises the reference node variable results versus simulation time and/or time step. It is recommended that the user request *screen* output, because well chosen output maybe invaluable in tracking the simulation progress and identifying possible input errors. If a suite of repetitive simulations is being performed, then *screen* output can be reduced to minimum values. As with other input cards, output options depend on the operational mode. With respect to this card, output options primarily refer to computed field and flux variables.

Reference node output is generated by selecting reference nodes and output variables. The user may request any number of reference nodes, but reference node output was primarily designed for tracking the time evolution of selected variables at key nodes of interest. Reference nodes are defined with three indices, which indicate the *x-, y-,* and *z*-direction coordinates of the node. Node numbering in the simulator increments in the order i, j, and k, where the indices refer to the *x-, y-,* and *z*-directions, respectively, for Cartesian coordinate systems, and *r-, θ-,* and *z*-directions, respectively, for cylindrical coordinate systems. *Reference Node Screen Output Frequency* and *Reference Node Output File Frequency* are parameters that indicate how often, with respect to time step reference node, output will be written to STDIO and the *output* file, respectively. A frequency value of 1 indicates reference node output occurs every time step, whereas a frequency value of 10 indicates reference node output occurs every 10 time steps. The user has control over the output time and length units and the number of significant digits reported to the various output media. Unless declared through the *Output Time Units* or *Output Length Units* input items, values for time and lengths recorded to the output media will be expressed in units of seconds and meters, respectively. These inputs allow the user customize time- and length-scale units to those most appropriate to the solved problem. *The Screen Significant Digits*, *Output File Significant Digits*, and *Plot File Significant Digits* input items allow the user to customize the number of significant digits which appear in field and flux variable results written to the STDIO, *output* file, and *plot.n* file, respectively. The default value for the number of significant digits is 5, and the minimum number of significant digits is 4. Field and flux variables output for the reference nodes are selected from the list shown for each operational mode in Appendix B. The same list of variables for each operational mode is available for output to the *plot.n* files. Output units for all variables with units can be specified immediately following the variable name. Variables without units require a *null* entry for the variable units. *Null* entries for variables with units yield default output units, which are expressed in SI units.

Plot files are written at the conclusion of an execution, by default, and at each requested *Plot File Output Tim*e. Plot files contain geometry data and selected field and flux variable results for every node in the computational domain. These files represent a "snapshot" of the simulation at a certain point in time. Requests for *Plot File Output Times* can be specified with user defined units. A *restart.n* file is generated with every *plot.n* file. Both *restart.n* and *plot.n* files are suffixed with a file name extension of a dot followed by an integer (e.g., plot.567, restart.32). The extension integer corresponds with the time step for which the file was written. Both *plot.n* and *restart.n* files are written at the conclusion of a time step. Field and flux variables recorded to *plot.n* files are selected from the list shown for each operational mode in Appendix B. Output units for all variables with units can be specified in the input item immediately following the variable name. Variables without units require a *null* entry for the variable units. *Null* entries for variables with units yield default output units, which are expressed in SI units.

For Reactive Transport modules, this card has been modified to allow for reactive species output to the *output* and *plot.n* files. Reactive species output in *Reference Node* or *Plot File* forms is specified in the same manner as solute output, with the exception that the keyword *Species* is used instead of *Solute*. Required input includes the output variable, species name, and output units.

4.4.24 Plant Card

This card is used to specify plant varietals and characteristics growing on the ground surface, their time-invariant characteristics, and how the canopy microclimate is handled. Under the current configuration of the simulator, data from this card is only used in conjunction with the *Atmospheric Conditions Card* and the *STOMP-WAE-B (Water-Air-Energy-Barrier)* operational mode. Detailed information on this mode can be found in Ward et al. (2005). Time varying plant species data are specified through the *Boundary Conditions Card*. With this card, the user defines the number of plant species and their characteristics with respect to the canopy structure and its rainfall interception characteristics, the root distribution, its stress functions, and albedo for each plant species. The soil-vegetation-atmosphere-transfer (SVAT) scheme in *STOMP-WAE-B (Water-Air-Energy-Barrier)* provides three options for describing the canopy structure and simulating the effects of plants: (1) *Bare Surface*, (2) *Single Plant Temperature*, and (3) *Multiple Plant Temperature*. The option selected determines the nature of the equations to be determined. Rainfall that strikes the plant is either intercepted or travels directly to the soil surface. There are two options for simulating how precipitation is handled: (1) *No Rainfall Interception* in the canopy (default) and (2) *Rainfall Interception*. The spatial location and ground-coverage density of plants is defined through the *Boundary Conditions Card*, using the Shuttleworth-Wallace boundary condition. The Shuttleworth-Wallace boundary condition computes transpiration and evaporation losses from the ground surface and plant canopy in response to atmospheric conditions, specified on the *Atmospheric Conditions Card*.

Plant species names must be unique and contain no more than 64 characters. For each defined *Plant Species*, the user must specify characteristics about the plant, including vertical root extent, canopy height, stress characteristics, and crop coefficients. For each Plant Species, the root stress function must be specified. The extent and spatial distribution of plant roots in the vertical direction is defined through three input parameters: *Max. Root Depth*, *Null Root Depth*, *Root Depth Fit Parameter*. The *Max. Root Depth* input represents the maximum rooting depth, and the *Null Root Depth* and *Root Depth Fit Parameter* inputs are empirical parameters. These parameters provide for zero root water uptake at the maximum rooting depth, account for asymmetrical root water uptake with depth, and allow for maximum root water uptake at any depth from the ground surface to the maximum rooting depth. The *Short-Wave*

Albedo is the required for calculating the amount of incoming radiation that is reflected by the surface vegetation. In the *Bare-Surface* option, albedo is essentially a measure of the reflectivity of the bare surface and is specified in the *Thermal Properties Card*. The *Plant Canopy Height* input is used to compute shading of taller plants on shorter plants, and used to compute canopy height, wind speed, and temperature. A stress function is available to reduce the root water uptake as a function of capillary pressure and uses four coefficient inputs: *Water Stress Point 1*, *Water Stress Point 2*, *Water Stress Point 3*, and *Water Stress Point 4*. Variations in the transpiration rate, root water uptake, and foliage on a yearly basis are handled via a crop-coefficient function, which requires four crop coefficients and associated times: *Crop Coefficient (Start)*, *Crop Coefficient (Mature Stage 1)*, *Crop Coefficient (Mature Stage 2)*, and *Crop Coefficient (Die-Off)* where, linear interpolation based on the time of year is used between crop coefficients.

4.4.25 Rock/Soil Zonation Card

In the STOMP simulator, hydrologic flow and transport properties are associated with rock/soil types. The zonation or distribution of rock/soil types across the computational domain is declared via the *Rock/Soil Zonation Card*. Several options are available through this card for declaring the distribution of rock/soil types: explicit declaration, external files, or *IJK Indexing.* Explicit declaration is a good choice for simulations involving a limited number of rock/soil types. Under this option the user associates rock/soil type names with portions of the computational domain. Rock/soil names must be unique and contain no more than 64 characters. The following key words have special meanings when used within a rock/soil name: '*skip conv*,' '*dp*, '*dual porosity*,' '*fractured*;' where, '*skip conv*' is used to skip convergence checks on nodes with associated rock/soil types, and '*dp*,' '*dual porosity*,' and '*fractured*' is used to declare the rock/soil type as having dual porosity characteristics. When explicitly declaring the rock/soil zonations, rock/soil names can be repeatedly applied to a grid cells or domain of grid cells, with only the last definition being applied. For example, to simplify the zonation of a problem with a dominant rock/soil type and isolated pockets or bands of another rock/soil type, the user should initially declare the dominant rock/soil type as covering the entire problem domain. With subsequent input lines, the user can overwrite the original rock/soil type for selected nodes with another rock/soil type. This layering approach is recommended and will often avoid leaving nodes of undeclared rock/soil types. External files can be useful when the rock/soil zonation data are automatically generated (e.g., geologic visualization programs). *IJK Indexing* should be used when every grid cell is associated with a unique rock/soil type, as common with statistically generated rock/soil distributions. With *IJK Indexing* the concept of rock/soil types is abandoned for a system where hydrologic flow and transport properties are associated with the grid cell. The rock/soil type of a grid cell is extended to boundary surfaces adjacent to the grid cell (i.e., boundary surface hydrologic properties are computed using the rock/soil type property or *IJK Indexing* property descriptions for the node adjacent to the boundary surface). Hydrogeologic properties for rock/soil types are specified through the *Mechanical Properties Card*, *Hydraulic Properties Card*, *Thermal Properties Card*, *Saturation Function Card*, *Aqueous Relative Permeability Function Card*, *Dissolved-Oil Transport Card, Salt Transport Card, Surfactant Transport Card, Gas Relative Permeability Function Card*, *NAPL Relative Permeability Function Card* and *Solute/Porous Media Interaction Card*.

4.4.26 Salt Transport Card

This card allows the user to define porous media dependent dispersivities for salt. This card is required only for simulations involving salt transport. For operational modes involving salt transport, the salt transport equation is fully coupled with the flow and energy transport equations. For every porous media defined on the *Rock/Soil Zonation Card,* dispersivities must be referenced. With the *IJK Indexing* option, node dependent parameters are entered via external files, and node independent parameters are entered directly on the card. Computation of effective diffusivity may be either by a conventional or empirical equation. If the empirical equation is selected, a *Molecular Diffusion Rate* must also be specified for each porous media. *Longitudinal Dispersivity* is defined with respect to dispersion along the flow path and is assumed to be independent of the flow direction with respect to the porous media structure. Likewise, *Transverse Dispersivity* is defined with respect to dispersion transverse to the flow path, independent of the flow direction. Refer to the STOMP Theory Guide (White and Oostrom 2000) for a more complete description of the salt mass conservation equation.

4.4.27 Saturation Function Card

This card allows the user to declare and define a saturation-capillary pressure function for each defined rock/soil type. Every rock/soil type defined on the *Rock/Soil Zonation Card* must be referenced. With the *IJK Indexing* option, node dependent parameters are entered via external files and node independent parameters are entered directly on the card. Saturation function types and the required input items vary greatly among the operational modes. Functional forms for the saturation-capillary pressure functions are preferred; however, for some operational modes, tabular input is acceptable. By default, tabular data will be interpolated using linear interpolation, whereas values beyond the table limits will be assigned either the table minimum or maximum values appropriately. The available function forms and form options vary among the operational modes, with all operational modes offering options to use van Genuchten and Brooks and Corey functions. For the van Genuchten function, the *'m' Parameter* can be defaulted (with a *null* entry) or assigned a value.

Default values will depend on which porosity distribution model (Mualem or Burdine) is chosen on the relative permeability function cards for the aqueous phase. Common options to these functional forms include: non-wetting fluid entrapment and non-wetting fluid residual formation. Fluid entrapment functions consider the hysteresis between wetting and draining paths due only to entrapment of the non-wetting fluids by imbibing wetting fluids. Fluid entrapment functions allow the specification of initial trapped non-wetting fluid saturations without specifying detailed saturation path histories. These functions additionally allow dissolution of the non-wetting phase into the wetting phase (e.g., dissolution of air and oil into the aqueous phase). Non-hysteretic functions require fewer input items than their hysteretic counterparts and use a single scanning path for both drainage and imbibition events. Operational modes that consider NAPL flow, require interfacial tension parameters (i.e., gas-NAPL, NAPL-aqueous, and gas-aqueous) for scaling the saturation-capillary pressure functions. Allowances are made for spreading and non-spreading NAPLs. Refer to the STOMP Theory Guide (White and Oostrom 2000) for a complete description of allowed saturation-capillary pressure functions.

4.4.28 Scaling Factor Card

The *STOMP-W (Water)* operational modes include an option for lumping rock/soil types into scaling groups, where hydraulic parameters for individual rock/soil types are computed from those declared for the scaling group and scaling factors declared for the rock/soil type. The premise behind the scaling factor card is that like rock/soil types can be scaled to a common set of hydrologic parameters. When applied to inverse calculations, this concept can reduce the number of unknown hydrologic parameters by a factor equal to the number of rock/soil types in the scaling group. For example, an inverse problem requiring 5 hydrologic parameters per rock/soil type in a domain with 7 different rock/soil types would have 35 unknowns. Using the scaling factor approach, the number of unknowns could be reduced to five hydrologic parameters for the scaling group.

The association between rock/soil types and scaling groups is made on the *Rock/Soil Zonation Card,* and this card is used to declare scaling factors. If an association is made between a rock/soil type and a scaling group, then the hydrologic parameters for the scaling group and not the individual rock/soil types are declared through the hydrologic property and function cards (i.e., *Mechanical Properties Card*, *Hydraulic Properties Card*, *Saturation Function Card*, and *Aqueous Relative Permeability Function Card*). An input file can simultaneously contain scaling groups and individual rock/soil types. When scaling groups are applied, individual rock/soil parameters for Saturated Hydraulic Conductivity, Diffusive Porosity, van Genuchten 'alpha' and 'n,' Brooks-Corey 'psi' and 'lambda,' and Residual Saturation are computed from the scaling group parameters and scaling factors through a scaling function. Scaling factors are dependent on the scaling group, the rock/soil type, and hydrologic parameter. Scaling functions are either linear or logarithmic.

4.4.29 Simulation Title Card

This card primarily provides a means to document a simulation. Information recorded in this card is rewritten on the *output* file, which then serves as a permanent record of the simulation. The user is encouraged to use descriptive titles and to briefly record the specifics and purpose of the simulation in the *Simulation Notes* section of the card. This becomes especially valuable when the user is making repeated simulations with small modifications to the input parameters. The time required to indicate these changes in either the *Simulation Title* or *Simulation Notes* will be invaluable when reviewing archived *output* files.

4.4.30 Solid Species Card

This card, used for the STOMP modes with reactive transport, defines the solid species to be considered in the simulation. Required input includes the species name, species mass density, and species molecular weight. The species name must be unique and distinct from aqueous and gas species names (e.g., FeCO3(s), FeCO3_solid, solid FeCO3, FeCO3s). The reactive transport module ECKEChem is described by White and McGrail (2005).

4.4.31 Solute/Fluid Interactions Card

This card allows the user to define solutes, solute-fluid interactions, and solute radioactive decay paths. This card is required only for simulations involving transport of solutes. The simulator is capable of simulating any number of solutes with the assumption that solute concentrations remain dilute (solute concentrations do not vary the physical properties of the transporting fluid phases). Solutes can decay radioactively to produce other solutes. For the loose coupling between parent solutes and progeny solutes to function properly, parent solutes must be defined on this card prior to their progeny. The simulator actually solves the transport equation for each solute sequentially in reverse order from the definition list on this card. Solutes are defined by a unique solute name, which must contain no more than 64 characters. *Chain Decay Fraction* relates a parent decay member with a progeny and refers to the fraction of the decaying parent that produces a particular progeny. *Chain Decay Fraction* inputs should be fractional values between 0.0 and 1.0, inclusively.

Solute/fluid interaction parameters such as molecular diffusion coefficients and interface partition coefficients depend on the operational mode. Molecular diffusion coefficients refer to the diffusion of the solute through the transporting fluid phase outside of the porous media. Corrections for transport through porous media are handled within the simulator. The interface partition coefficients define the equilibrium distribution of solute between the active liquid phases. Partition coefficients that define the equilibrium adsorption of solute onto the solid phase are defined on the *Solute/Porous Media Interaction Card*.

The user should carefully note the definitions and requested units for each interphase partition coefficient. The *Aqueous-Gas Partition Coefficient* is the ratio of the concentration of solute in the aqueous phase per unit mass of aqueous phase to the concentration of solute in the gas phase per unit volume of gas phase. The *Aqueous-NAPL Partition Coefficient* is the ratio of the concentration of solute in the aqueous phase (per unit mass of aqueous phase) to the concentration of solute in the NAPL phase (per unit mass of NAPL). Refer to the STOMP Theory Guide (White and Oostrom 2000) for a more complete description of the interface partition coefficients, solute diffusion coefficients, radioactive decay rate equations, and radioactive chain decay fractions.

4.4.32 Solute/Porous Media Interaction Card

This card allows the user to define solid-aqueous phase partition coefficients and porous media dependent hydraulic dispersivities. This card is required only for simulations involving transport of solutes. This card differs from the *Solute/Fluid Interaction Card* because the input parameters declared are dependent on both the solute and rock/soil type. For every solute defined on the *Solute/Fluid Interaction Card*, every rock/soil type defined on the *Rock/Soil Zonation Card* must be referenced. With the *IJK Indexing* option, node dependent parameters are entered via external files, and node independent parameters are entered directly on the card. The *Solid-Aqueous Partition Coefficient* defines the interface equilibrium of a solute adsorbed on the solid and dissolved in the aqueous phase, and refers to the concentration of solute adsorbed on the solid phase (per unit mass of solid phase) over the concentration of solute dissolved in the aqueous phase (per unit mass aqueous phase). The longitudinal and transverse hydraulic dispersivities are properties only of the rock/soil type. *Longitudinal Dispersivity* is defined with respect to dispersion along the flow path and is assumed to be independent of the flow direction with respect to the porous media structure. Likewise, *Transverse Dispersivity* is defined with respect to dispersion transverse to the flow path, independent of the flow direction. Refer to the STOMP Theory Guide (White and Oostrom 2000) for a more complete description of the solid-aqueous interface partitioning and hydraulic dispersion of transported solutes.

4.4.33 Solution Control Card

This card controls many general operational aspects of a simulation. Three *Execution Modes* are recognized: *Normal, Restart,* or *Initial Conditions*. In the *Normal* and *Initial Conditions* modes, initial state conditions are declared through the *Initial Conditions Card*. In the *Restart* mode, initial state conditions are assigned via a restart file from a previous execution or declared through the *Initial Conditions Card*, using the special *overwrite* option on selected parameters. Unless specified through the *Output Control Card*, restart files (i.e., *restart.n*) are generated at each *plot.n* write event, and have name extensions that correspond to the generating time step (e.g., the file restart.28 would have been generated at the conclusion of time step 28). Restart files are text files that contain simulation time and control information, and a collection of field variables needed to redefine the simulation state for the operational mode. In the *Normal* mode, the simulator executes from a declared start time, using an initial state declared through the *Initial Conditions Card*, until the declared stop time, the declared number of time steps, an execution error, or a sequence of convergence failures. In the *Initial Conditions* mode, the simulator reads the input file checking for formatting errors and stops; no time steps are executed. In the *Restart* mode, the simulator executes from either a declared start time or the start time specified in the restart file, using an initial state defined by a previous execution, until the declared stop time, the declared number of time steps, an execution error, or a sequence of convergence failures. In the *Restart* mode, the initial state defined by the restart file can be modified using the special '*overwrite*' option on selected parameters, declared through the *Initial Conditions Card*.

By default, the *Normal* and *Restart* modes use Euler time differencing (i.e., first order backward). The *Second Order* option switches the time-differencing scheme to second order backward. The *Restart* mode offers an option to read named restart files by including the keyword '*file*' in the *Execution Mode Option* character string. This option triggers the code to read an additional character string, which is the name of the restart file. For example, the input line, *restart using second order time differencing from file,restart.3456,* would start the simulator using second-order backward time differencing with the initial state declared through the field parameters in the file *restart.3456.* In addition to this option, the *Restart* mode offers an option to read restart files, generated by other operational modes by including the key word '*mode*.' For example, the input line, *restart mode file,restart.3456,1*, would start the simulator with the initial state declared through the field parameters in the file *restart.3456*, which was created from a previous *STOMP-W (Water)* operational mode (i.e., *Mode 1*). This is considered an advanced option, as the user is responsible for defining the initial system state, for the current operational mode, from a combination of parameters from the restart file and *Initial Conditions Card*.

The *STOMP-W (Water)* operational modes have two additional options associated with the *Execution Mode Option*: *Dynamic Domain* and *No Flow*. The *Dynamic Domain* option conserves computations by temporarily removing nodes from the computational domain where changes in the flow field are insignificant. The active computational domain is updated every time step. The *No Flow* option (*STOMP-W w/ Transport* operational mode only) is used for transporting solutes through a steady-state flow field. For these simulations, the flow field is computed once during the initial time step and then remains unaltered.

The governing flow and transport equations to be solved are declared through the *Operational Mode Option*, which has two components: coupled flow equations and solute transport equations. The solved coupled flow equations are declared by choosing an operational mode from those listed in Table 2.1 (e.g., *STOMP-W* (*Water)*, *STOMP-WO (Water-Oil*)). Operational modes are referred to by name or number; see Table 2.1 for a correspondence between operational mode name and numbers (e.g., *STOMP-W* (*Water)* or *Mode 1*, *STOMP-WAE* (*Water-Air-Energy)* or *Mode 3*, *STOMP-WCS* (*Water-CO$_2$-NaCl)* or *Mode 32*). Because execution performance is indirectly proportional to the number of solved governing equations, the user should select an operational mode that is most appropriate for the problem of interest. For example, if the physical system to be modeled does not contain thermal or gas pressure gradients and is void of oil, then the most appropriate operational mode would be the *Water* operational mode, where only the water mass conservation equation is solved. Executing the described physical system under the *STOMP-WA* (*Water-Air*) operational mode would yield nearly identical results. However, the execution time would be significantly increased. Considerable attention should be given to the selection of an appropriate operational mode for the physical system of interest. In terms of efficient and representative simulations, it is equally important not to eliminate critical physical phenomena through erroneous assumptions, as it is not to solve superfluous governing equations. Each operational mode allows the solution of solute transport equations, which are indicated by including the keyword *Transport* in the operational mode (e.g., *Water-Salt w/ Transport* or *Water-Air and Solute Transport*). Solute transport is solved using the Patankar method, unless the keyword *TVD* appears, which indicates the TVD method. The TVD method for coupled salt, dissolved oil, or surfactant transport is indicated with the keyword *LFL*.

Execution periods refer to a period of simulation time. The simulator allows the user to specify a single or multiple execution periods. For each execution period, the user can control the initial time step, maximum time step, time step acceleration factor, maximum number of Newton-Raphson iterations, and convergence criterion. Recommended values for the Time Step Acceleration Factor, Maximum Number of Newton-Raphson Iterations, and Convergence Criterion are 1.25, 8, and 1.e-06, respectively. Simulations involving complex phase transitions often require more Newton-Raphson iterations to reach convergence, because of the design of the phase transition algorithms. For these types of simulations, a value of 16 is recommended for the Maximum Number of Newton-Raphson Iterations. Except under special circumstances, it is not recommended to change the value for the Convergence Criterion from its recommended value. This value has proven through numerous applications to achieve a good balance between accuracy and execution speed.

Field variables, which include physical, thermodynamic, and hydrologic properties, are defined in the finite-difference formulation at the node centers. Conversely, flux variables are defined at node interfaces. Computation of flux variables requires knowledge of field variables at node interfaces. Values of flux variables at node interfaces are evaluated by averaging the field values for the two nodes adjoining an interfacial surface. Interfacial averaging schemes may be declared individually for each field variable through the *Interfacial Averaging Variables* input.

The default interfacial averaging schemes for the simulator are shown in Table 4.4. For simulations of physical systems involving heat transfer, it should be noted that convergence problems might arise if the density properties are not averaged with upwind weighting. Likewise, infiltration problems typically demonstrate strong dependencies on the relative permeability of the infiltrating fluid.

Table 4.4. Default Interfacial Averaging Options

Field Variable	Interfacial Averaging
Aqueous Density	Upwind
Aqueous Relative Permeability	Upwind
Aqueous Viscosity	Harmonic
Dissolved Air Diffusion	Harmonic
Dissolved Oil Diffusion	Harmonic
Dissolved Salt Diffusion	Harmonic
Gas Density	Upwind
Gas Relative Permeability	Upwind
Gas Viscosity	Harmonic
Hydraulic Dispersion	Harmonic
Intrinsic Permeability	Harmonic

NAPL Density	Upwind
NAPL Relative Permeability	Upwind
NAPL Viscosity	Harmonic
Oil Vapor Diffusion	Harmonic
Solute Diffusion	Harmonic
Thermal Conductivity	Harmonic
Water Vapor Diffusion	Harmonic

4.4.34 Source Card

This card allows the user to control sources and/or sinks of mass, energy, and solutes by defining time-varying sources. By definition, sinks are negative sources, and sources refer to an influx of mass, energy, or solute into a node. Sources can be specified for interior or boundary nodes and are functionally analogous to *Neumann* type boundary conditions. Sources applied to inactive nodes are not recognized. Sources are time varying; however, unlike boundary conditions, multiple sources may be applied to a node during the same time period. The simulator controls time steps to agree with time transitions in sources. Source inputs depend on the operational mode, with the required inputs becoming more complex with increasing number of solved equations. Sinks withdraw mass, energy, or solutes from a node. The physical properties of the fluids withdrawn through sinks equal the properties of the fluids of the node. Sources inject mass, energy, or solutes into a node. The physical properties for fluids injected through sources are computed from the specified input parameters.

Sources of type *Aqueous Volumetric* and *Aqueous Mass* inject aqueous fluid. If specified, dissolved air and oil will be injected with the aqueous fluid. For non-isothermal problems, the injected fluid would also transport an amount of heat into the node based on the enthalpy of the entering aqueous fluid. Sources of type *Gas Volumetric w/ Mass Fraction*, *Gas Volumetric w/ Relative Humidity*, *Gas Mass w/ Mass Fraction*, and *Gas Mass w/ Relative Humidity* are similar to the aqueous sources, in that mass and heat injected into the node would include contributions from water vapor, air, and oil vapors. The source type suffix *w/ Mass Fraction* indicates that water vapor, air, and vapor concentrations in the gas phase will be specified through inputs of mass fractions for these quantities. Similarly, the source type suffix *w/ Relative Humidity* indicates that water vapor, air, and oil vapor concentrations in the gas phase will be specified through inputs of relative humidity for the water and oil. Sources of type *NAPL Volumetric* and *NAPL Mass* inject NAPL. Because dissolution of air and water in the NAPL phase is neglected by the simulator, these sources represent pure NAPL. Energy sources are specified as type *Power* or *Power Density*, where *Power* type sources inject energy per unit time and *Power Density* type sources inject energy per unit time per node volume. *Solute* sources inject solute mass in the assumed units for solute (e.g., Ci, pCi, gm, kg, mol, kgmol).

Well sources model production or injection wells using a non-linear well model. Positive flow rates are considered as injection wells and negative flow rates are considered as production wells. The well model essentially computes the depth of liquid (water) in the well through a Newton-Raphson iteration scheme that balances flux across the well casing with the source pumping rate. For injection wells, the depth of water in the well is limited by a well bottom pressure constraint, expressed in pressure units. For pumping wells the contribution from seepage flow is ignored. Wells are vertical features in the STOMP simulator and must be defined over a contiguous array of vertical nodes. By default, well bore storage is considered in the well model and the well volume, as defined by the *Borehole Radius,* is removed from the node volume. Both of these features can be ignored by including the key words *No Storage* and *No Volume* in the *Source Type Option*.

Time variations of sources are controlled through declaring multiple source times. All *Source Time* inputs are referenced against the *Initial Time* specified in the *Solution Control* card or obtained from a *restart* file. A source declared with a single *Source Time* implies that the source is time invariant and the specified *Source Time* represents the start time for the source. Prior to the start time the source will be zero, and from the start time to execution completion the source will be as specified. If a source is declared with multiple *Source Times*, then the first time listed equals the start time, the last time listed equals the stop time, and the intermediate times are transition points. For simulation times outside of the start and stop time limits, zero source conditions apply. For simulation times between two *Source Times*, linear interpolation of the sources is applied. Step source changes can be simulated by defining duplicate *Source Times* the first time would indicate the completion of the previous source, and the second time would indicate the start of the new source. At the completion of the step source, another set of duplicate *Source Time* declarations would be used. Step sources are convenient methods to introduce slugs of fluids, heat, or solute into an interior node. For the Reactive Transport modules, this card has been modified to allow for sources to be specified for reactive species. Reactive species sources are specified in the same manner as solute sources, using a species rate (i.e., mol/s) or species density rate (i.e., $mol/m^3_{\text{grid volume}}$ s). Required input includes the source type, species name, source domain, the number of time points, and for each time point: the time, and species source rate and units.

4.4.35 Species Link Card

This card, used for the STOMP modes with reactive transport, associates reactive species with components in the coupled flow and transport equations and defines which species name defines the system pH. Currently, the following coupled flow and transport components can be associated: aqueous water, gas water, aqueous air, gas air, aqueous CO_2, gas CO_2, aqueous CH_4, gas CH_4, aqueous salt, and solid salt. Required inputs include the number of reactive species links, species names, and linked components (i.e., Aqueous pH, Aqueous Water, Gas Water, Aqueous Air, Gas Air, Aqueous CO2, Gas CO2, Aqueous CH4, and Gas CH4). The reactive transport module ECKEChem is described by White and McGrail (2005).

4.4.36 Surface Flux Card

This card allows the user to define surfaces to track fluxes of fluid mass, fluid volume, heat, and/or solutes. A surface defined with this card can be composed of rectangular areas of coplanar surfaces on exterior boundaries or between interior nodes. Output from the surface flux integration routines is written to the *surface* file and contains flux rate and integral data for each defined surface at every time step. The types of fluxes that can be tracked depend on the operational mode. Declaration of surfaces is similar to defining boundary condition surfaces. Surfaces are defined by referencing a group of coplanar nodes and a surface direction with respect to the nodes. For the Cartesian coordinate system, the terms west, south, and bottom refer to the negative *x*-, *y*-, and *z*-directions, respectively, and the terms east, north, and top refer to the positive *x*-, *y*-, and *z*-directions, respectively. For the cylindrical coordinate system, the terms west, south, and bottom refer to the negative *r*-, θ-, and *z*-directions, respectively, and the terms east, north, and top refer to the positive *r*-, θ-, and *z*-directions, respectively. For example, a surface to track the flux rate and integral of a particular solute entering the water table could be defined for a simulation with saturated conditions along the bottom boundary surface by referencing the node group along the bottom of the computational grid and defining the *Surface Flux Orientation* as *Bottom*. One surface flux rate and integral value is computed for each defined surface and represents the summation of surface flux contributions from the individual surfaces in the coplanar group of surfaces.

For Reactive Transport modules, this card has been modified to allow for reactive species surface flux output to the surface file. Reactive species surface flux output is specified in the same manner as solute surface flux output, with the exception that the keyword *Species* is used instead of *Solute*. Required input includes the surface flux type, species name, and surface flux rate and integral units, surface direction, and surface domain.

4.4.37 Thermal Properties Card

This card allows the user to assign values to the thermal conductivity, specific heat, and short-wave albedo for each defined rock/soil type. Every rock/soil type defined on the *Rock/Soil Zonation Card* must be referenced. With the *IJK Indexing* option, node dependent parameters are entered via external files and node independent parameters are entered directly on the card. This card is required only for simulations involving the solution of the energy conservation equation. Declaration of the thermal conductivity depends on the operational mode and function option. Refer to the STOMP Theory Guide (White and Oostrom 2000) for a description of thermal conductivity functions.

The *Constant* option fixes the thermal conductivity to a constant value, independent of temperature or saturation. The *Parallel* option requires the thermal conductivity of the soil grains, and models thermal conductivity with an equivalent parallel path model dependent on porosity, phase saturations, and temperature. The *Linear* and *Somerton* options scale the thermal conductivity between the unsaturated and saturated values depending on phase saturation. Both the *Campbell* and *Cass* options describe the thermal conductivity as a function of saturation using a polynomial function. However, the *Cass* option allows the additional specification of non-isothermal flow enhancement factors. A primary assumption with the simulator is that principal components of the thermal conductivity tensor are aligned with the principal coordinate directions. For cylindrical coordinate systems the radial, azimuthal, and vertical permeabilities correspond with the *x-*, *y-*, and *z*-direction values, respectively.

Short-wave albedo input is required for calculating the amount of incoming radiation that is reflected by the bare soil. There are four options for specifying how the ground surface albedo is calculated: (1) *Plum and Xiu Albedo*, (2) *Wang Albedo*, (3) *Briegleb Albedo*, and (4) *Constant Albedo*. Details on the various albedo options can be found in Ward et al. (2005).

4.4.38 UCODE Control Card

This card allows users to control UCODE executions through the STOMP input file. Parameters and options read from this card are written to external files that are then read and used by UCODE. This card is offered as a convenience to users executing UCODE in conjunction with STOMP, but is not required if the user prefers to externally create the controlling files for UCODE. The control parameters read on this card are written to the external file *out_uc.sto*. As with other STOMP input, this card uses comma delimited fields. Output written to *out_uc.sto*, however, is properly formatted for UCODE input. More detailed information on this card can be found in Ward et al. (2005).

4.4.39 Well Card

This card allows users to specify monitoring and pumping wells for the *STOMP-WO (Water-Oil)* mode. Input variables are well type, well radius, well aqueous depth, well NAPL depth, well head pressure, dissolved oil saturation, number of screen intervals, and well times.

5.0 Compilation and Execution

In its native form, the STOMP simulator is a collection of files, which contain either global routines or those associated with a particular operational mode. For users outside of the PNNL, the STOMP simulator is distributed as assembled source coding for a particular operational mode, with the associated files including modules, example input files, and required external packages (e.g., SPLIB). Assembly of the source coding occurs through the *make* utility (Talbott 1988) before the code is distributed. For users within the PNNL, the *make* utility additionally can be used directly to create an executable. Except for external packages, the STOMP simulator is coded in FORTRAN. Distributing an open source allows users to read and modify the simulator, which will hopefully result in an open exchange of scientific ideas. A drawback of an open source is that the user is responsible for compiling and linking the source code to generate an executable. This inherently assumes that the user has a FORTRAN compiler and is familiar with its use for generating code. Advanced users, interested in modifying the code, should additionally be familiar with using a symbolic debugger (often provided with the FORTRAN compiler). The unassembled STOMP source is coded in a combination of FORTRAN 77 and FORTRAN 90. With respect to memory allocation, the assembled source code can be configured in two forms: static and dynamic memory.

For those users with only access to a FORTRAN 77 compiler, (e.g., g77, f77, pgf77) the code must be configured in static memory form. In this form, the source code will include a *parameters* and a *commons* file. As FORTRAN 77 is unable to dynamically allocate memory, the user is responsible for editing the *parameters* file to define array dimensions to statically allocate memory during compilation. For users with access to a FORTRAN 90 compiler (e.g., f90, pgf90, ifc), the source code can be configured for dynamic memory allocation. In the dynamic memory form, both the *parameters* and *commons* files are replaced with a series of FORTRAN 90 modules, in the file *allo.f.* As described below, this module file must be compiled prior to compiling the source code. When configured under the dynamic memory option a utility named *step* is included in the source code. When incorporated, the *step* utility becomes the first subroutine called and reads a STOMP *input* file to determine dimensioned array requirements. These values (i.e., parameters) are then used to allocate memory for the dimensioned arrays via a call to the subroutine *alloc*. Subroutine *alloc* makes a series of memory allocations and memory checks. If STOMP attempts to allocate more memory than available on the computer, the simulation stops and an error message is printed. Memory allocation under the dynamic memory option only occurs during the execution startup (i.e., memory is never deallocated until the execution stops). The dynamic memory option is generally preferred as it allows the user to execute problems without having to create a *parameters* file and recompile the code with changes in the *input* file. However, if the dynamic option is not available, the *step* routines (*step.bd* or *step.sp*) can be executed separately to create a *parameters* file. Table 5.1 summarizes the memory allocation options for the sequential modes.

Selected operational modes of the STOMP simulator are available for scalable execution on multiple processor (i.e., parallel) computers. These versions of the simulator are written in pure FORTRAN 90 with imbedded directives that are interpreted by a FORTRAN preprocessor. Without the preprocessor, the scalable version of the simulator can be executed sequentially on a single processor computer. The scalable versions of the STOMP simulator carry the '-*Sc*' designator.

Table 5.1. Memory Options for Sequential Modes

Memory Option	FORTRAN Compiler	Include Files	Module File	Notes
Static	f77	*commons* *parameters*		• User generated *parameters* • Recompilation w/ input change
Dynamic	f90		*allo.f*	• No recompilation w/ input change • Greater 1 GB memory on Linux

5.1 Compilation

Compiling the source code into an executable differs between operating systems, compilers, and memory options. This section will describe the differences between memory options, using the UNIX[1] as an example operating system. For the static memory option, the following files will be provided with the assembled source coding: *stomp#_[sp,bd].f*, *commons*, and *parameters*; where the # in the filename *stomp#_[sp,bd].f* refers to the operational mode number and the solver options *sp* and *bd* refer to the conjugate gradient or banded solvers, respectively (e.g., *stomp4_bd.f* is the source code for the *STOMP-WO (Water-Oil)* operation mode with the banded solver). To create an executable on a UNIX system, assuming the *commons* and *parameters* files were in the same directory as the source code, the user would issue the following command:

```
f77 -I . -o stomp4_bd.e stomp4_bd.f
```

For the conjugate gradient solver, the compiler must link to the *splib* library, and the compilation command would be:

```
f77 -I . -o stomp4_sp.e stomp4_sp.f $SPLIB_PATH/splib.a
```

where `$SPLIB_PATH` is the path to the *splib* library. For the dynamic memory option, the following files will be provided with the assembled source coding: *stomp#_[sp,bd].f*, and *allo.f*. To create an executable on a UNIX system, the user would issue the following command:

```
f90 -c allo.f
f90 -c stomp4_bd.f
f90 -o stomp4_bd.e allo.o stomp4_bd.o
```

For the conjugate gradient solver, the compiler must link to the *splib* library, and the compilation command would be:

```
f90 -c allo.f
f90 -c stomp4_sp.f
f90 -o stomp4_sp.e allo.o stomp4_sp.o $SPLIB_PATH/splib.a
```

where, `$SPLIB_PATH` is the path to the *splib* library.

[1] UNIX is a registered trademark of AT&T Information Systems.

5.2 Parameters

For the static memory option, the user is responsible for creating a *parameters* file. Parameters are used by the FORTRAN programming language and compilers to statically allocate memory for storage of variables. The FORTRAN 77 language is unable to allocate memory dynamically, therefore, all memory storage requirements must be defined at compilation time. No execution errors will occur if the memory allocated is greater than required by the simulation, unless the memory requirements exceed the computer's capabilities. Unless necessary, the user should avoid executing simulations that require the use of virtual memory. The time required to swap data between the virtual memory storage device and the active memory typically yields poor execution speeds. The STOMP simulator requires two types of parameters (declared and computed) to be defined, prior to compilation. The user is responsible for properly assigning all of the declared parameters. Declared parameters are assigned by modifying the *parameters* file supplied with the STOMP simulator using a text editor (word processor) or by creating a new *parameters* file. The equations for the computed parameters must be included in each *parameters* file after the declared parameters. All declared parameters, except for switch type parameters, must have minimum values of 1. Undersized parameters will generally yield execution errors, which may or may not be detected by the system. Oversized parameters are permissible, but can result in excessive memory allocation.

5.3 Execution

Executing the simulator is straight forward and only requires that the executable version of the code and an input file named *input* reside in the current directory. For restart simulations, a restart file named *restart* must also reside in the current directory. Because restart files are created with an extension that corresponds with the generating time step, the user must rename the appropriate restart file to *restart*. For a UNIX operating system, execution is started by typing in the name of the executable file. Execution will be indicated by the printing of a STOMP title banner and program disclaimer to the standard input/output device (e.g., screen). Two types of error messages may be generated during a STOMP execution. The first type is a system generated message that typically indicates a memory, FORTRAN, or other system error identified by the system. The second type of error messages refer to those generated by the STOMP code, which typically refer to input, parameter, or convergence failure type messages. STOMP generated messages are divided into three categories according to severity. The most severe are *ERROR* messages, which abort the program execution. Undersized parameters are typical of errors which yield *ERROR* messages, because execution of the simulator with undersized parameters may yield gross errors, or even worse subtle errors which may pass undetected in the results. Next on the severity level are the *WARNING* messages, which generally warrant notice by the user that a problem with the input file probably exists. The least severe are *NOTE* messages, which are used to record events like the absence of an optional input card.

When an execution is terminated with a STOMP generated *ERROR* message two types of information appear on the standard input/output device. The first piece of information is an error message with or without related data that indicates the nature of the error. Input formatting errors are trapped to a limited degree by the simulator. Unfortunately, the input error message that results may not always agree with the actual location of the identified input error. This is because the original error may have skipped detection. For example, a missing input item on an input line would not be identified until STOMP identified an incorrect data type or the end of the line was reached. The user should expect to find input errors slightly before the location indicated by the generating error message. The second type of information, which is generated with an *ERROR* message, is a calling sequence. The calling sequence is a slash delineated list of routine names which were called prior to the identified error. The calling sequence for an input error on the *Aqueous Relative Permeability Card* for the *Water* operational mode would appear as

```
/STOMP1/RDINPT1/RDRPL/WRMSG
```

where the called routine names are delineated with a forward slash. For this example, the main routine *STOMP1* called the subroutine *RDINPT1* to read the input file for the *Water* operational mode, which called *RDRPL* the global routine to read the *Aqueous Relative Permeability Card*, which identified an error and called *WRMSG* to write an error message to the standard input/output device (e.g., screen).

6.0 Output Files and Post-Processing

The STOMP simulator can generate, depending on the requested output and saturation function type, data files for restarting a simulation and three types of simulation result files. Every execution produces an output file named *output*. If the simulation concludes without a fatal error, then a *restart.n* and a *plot.n* file are also generated. Restart and plot file names include extensions (i.e., *n*) which correspond to the time step for which the file was written. For example, a restart file named *restart.39* would have been written at the conclusion of the 39th time step. A surface file named *surface* is generated whenever a Surface Flux Card is included in the *input* file. All output files generated by the simulator (*output, restart.n, plot.n,* and *surface* files) are ASCII text files.

6.1 Output File

An output file named *output* is generated with every execution of the simulator. If a file named *output* already exists in the current directory then that file will be erased and overwritten. The user is responsible for renaming previous *output* files prior to executing the simulator. An *output* file begins with a welcome statement, disclaimer, and banner, which should appear similar to

```
 Welcome to ...


                           STOMP
        Subsurface Transport Over Multiple Phases


 This file was produced by STOMP, a numerical simulator
 developed by the Pacific Northwest National Laboratory, with
 support from the VOC-Arid Integrated Demonstration Project,
 Office of Technology Development, U.S. Department of Energy.
 Results from this version of STOMP should not be used for
 license related applications.

 For inquiries or assistance:  Call (509) 372-6070

                      ---  OUTPUT  ---
```

Following this banner is an input record, which documents the simulation input. An input record is included on the *output* file to document the simulation in the event that the *input* file is destroyed or becomes separated from the *output* file. The input record is formatted differently than an *input* file, but contains all of the information listed on an *input* file. Optional cards which are not included in the *input* file are noted in the input record. If the simulator identifies an input error, an error message will appear in the input record at the point the error was noted. Input cards are read by the simulator in a specific order and will appear in the input record in that order. Because input cards can be organized randomly on the *input* file the card order on the input record may not match that on the *input* file. The reference node output record follows the input record.

The reference node output record is a table of simulation data and selected reference node variables, which are printed according to the frequency requested on the Output Control Card with the input item *Reference Node Output File Frequency*. A table column header is printed to delineate the columns every

10 print records. Definitions of the header abbreviations for the reference node output are given in Table 6.1. Each print record will show simulation data and reference node variable data for each reference node requested. Simulation data comprises the time step, the reference node number, the simulation time, the simulation time step, and the number of Newton-Raphson iterations required to reach convergence. Reference node variable data comprise values of the variables specified on the Output Control Card through the Reference Node Variable inputs. Reference nodes are indexed by node number. The corresponding *x*-, *y*-, and *z*-direction indices are shown for each reference node on the line preceding the column headers. For example, a simulation with reference node 272 at *x*-, *y*-, and *z*-direction indices of 2, 3, and 6, respectively would have a reference node header line that appeared as

```
Reference Node(s) (  2, 3,  6:     272)
```

Column headers for reference node variables are delineated with an abbreviated title and associated units. Units are enclosed in brackets below the variable abbreviation. A portion of reference node output record for a horizontal heat pipe problem involving 50 nodes for time steps 70 through 73 appeared as

```
Reference Node(s) (  1,  1,  1:      1) ( 50,  1,  1:     50)
 Step  Node    Time      Timestep  Itr      T           SL           PL
             [day   ]   [day   ]        [c       ]                 [pa     ]
   70     1  2.837E+01  1.793E+00  8  7.1893E+01  9.9893E-01  1.0133E+05
   70    50  2.837E+01  1.793E+00  8  1.3564E+02  2.0992E-02 -1.9896E+08
   71     1  3.061E+01  2.242E+00  9  7.1895E+01  9.9903E-01  1.0133E+05
   71    50  3.061E+01  2.242E+00  9  1.4039E+02  1.9265E-02 -2.2722E+08
   72     1  3.341E+01  2.802E+00  9  7.1878E+01  9.9933E-01  1.0133E+05
   72    50  3.341E+01  2.802E+00  9  1.4598E+02  1.7489E-02 -2.6050E+08
   73     1  3.692E+01  3.503E+00  8  7.1869E+01  9.9951E-01  1.0133E+05
   73    50  3.692E+01  3.503E+00  8  1.5225E+02  1.5759E-02 -2.9758E+08
```

where the reference nodes are nodes 1 and 50, and the reference node variable data includes the temperature, reported in degrees Celsius, the aqueous saturation, and the aqueous pressure, reported in Pascal (absolute).

Convergence failures and subsequent time step reductions are also noted within the reference node output record. Refer to Chapter 5 for a discussion on convergence errors. Three types of convergence error messages may appear within the output. The first type indicates that the update to a primary variable exceeded the maximum allowable change. An example of this type of convergence error for a horizontal heat pipe problem appeared as

```
         ---  Excessive Primary Variable Change  ---
    Temperature =  3.80318E+02 Node =     50
    Water Pressure = -1.33730E+09 Node =     50
    Gas Pressure =  3.07342E+05 Node =     50
    Time Step Reduced From  2.9383E+00 day to  5.8766E-01 day
```

where an excessive change to at least one primary variable was noted at node 50 and the time step was reduced from 2.938 to 0.5877 days and repeated. The second type indicates that a converged solution was not obtained within the maximum number of Newton-Raphson iterations. An example of this type of convergence error for a three-phase volatile infiltration problem appeared as

```
          ---  Convergence Failure  ---
    Water Equation Maximum Residual =  1.3313E-02 Node =    152
    Air Equation Maximum Residual =  8.2280E-07 Node =     76
    VOC Equation Maximum Residual =  1.0472E-02 Node =    154
    Time Step Reduced From  7.0121E-03 hr to  1.4024E-03 hr
```

where the maximum normalized residuals for the water and oil mass conservation equations were noted, which exceeded the convergence criterion. After this convergence failure, the time step was reduced from 0.00701 to 0.00140 hours and repeated. The third type indicates that the linear system solver has failed to reach a solution. An example of this type of convergence error for a three-phase volatile infiltration problem appeared as

```
          ---  Singular Matrix  ---
    Jacobian Matrix Index =    339
    Node Number           =    113
    Time Step Reduced From  1.4849E-02 hr to  2.9697E-03 hr
```

where a zero or nearly zero diagonal was noted for the oil equation at node 113. After this convergence failure, the time step was reduced from 0.0148 to 0.00297 hours and repeated. All three types of error messages include additional information that pertains to the specific convergence problem. Convergence errors result in a reduction in the time step with a repeated attempt to solve the system of governing equations. Four successive convergence failures without an intermediate successfully converged time step results in termination of the execution. Output files are concluded with the following closing message

```
          ---  End of STOMP Simulation  ---
```

Post-processing of the *output* file can be done with the Perl script *outputTo.pl*. The Perl program transforms the output file into a formatted input for Gnuplot, Grapher, Igor, Matlab, and Tecplot. The minimum command line is

outputTo.pl

after which the user is prompted for the plotting package name, plotting package input file name, STOMP output file name, reference nodes, and reference node variables. However, the command line can be expanded to include these options. A few examples of more expanded command lines are

outputTo.pl -an -t Gnuplot output.dat output
-- prints all nodes (variables and title will be prompted) to file output.dat

outputTo.pl -av -an Tecplot ref_nodes.dat output
-- prints all nodes and variables to file ref_nodes.dat

outputTo.pl Grapher ref_nodes.dat output
-- prints (variables and nodes will be prompted) to file ref_nodes.dat

outputTo.pl -av -t Tecplot tecplot.dat output
--prints all variables (nodes and title will be prompted) to file tecplot.dat

Table 6.1. Reference Node Output Record Abbreviations

Abbreviation	Definition
ANFL	Free NAPL-aqueous interfacial area
ANTL	Trapped NAPL-aqueous interfacial area
ASL	Apparent aqueous saturation
ASLM	Minimum effective aqueous saturation
AST	Apparent total liquid saturation
BGL	Gas-aqueous surface tension scaling factor
Cn[a]	Solute volumetric concentration
CGn[a]	Solute gas volumetric concentration
CGA	Air mass concentration in the gas phase
CGO	Oil mass concentration in the gas phase
CGW	Water mass concentration in the gas phase
CLn[a]	Solute aqueous volumetric concentration
CLA	Air mass concentration in the aqueous phase
CLO	Oil mass concentration in the aqueous phase
CLW	Water mass concentration in the aqueous phase
CNn[a]	Solute NAPL volumetric concentration
CNO	Oil mass concentration in the NAPL phase
CNW	Water mass concentration in the NAPL phase
CRNG	Maximum local gas phase courant number
CRNL	Maximum local aqueous phase courant number
CRNN	Maximum local NAPL phase courant number
CS	Salt volumetric concentration
CSL	Salt aqueous volumetric concentration
CSO	Sorbed oil volumetric concentration
DAPS	Dissolved air percent saturation
DSGF	Dual porosity gas saturation (fracture)
DSGM	Dual porosity gas saturation (matrix)
DSLF	Dual porosity aqueous saturation (fracture)
DSLM	Dual porosity aqueous saturation (matrix)
E_SA	Actual evaporation rate
ESGT	Effective trapped gas saturation
ESML	Minimum effective aqueous saturation
ESNT	Effective trapped NAPL saturation
FAI	Fracture adjacent index
FW_S	Surface water mass flux
GPG	Gas pressure (gauge)
GPL	Aqueous pressure (gauge)
GPN	NAPL pressure (gauge)
HHG	Gas hydraulic head (water equivalent)
HHL	Aqueous hydraulic head (water equivalent)
HHN	NAPL hydraulic head (water equivalent)
HKL	Aqueous solute coefficient

Table 6.1. (contd)

Abbreviation	Definition
HKNF	Free-NAPL solute coefficient
HKNT	Trapped-NAPL solute coefficient
IMA	Integrated air or CO_2 mass
IMGA	Integrated air or CO_2 mass in gas
IMGO	Integrated oil mass in gas, or, Integrated CH_4 mass in gas
IMGT	Integrated air in trapped gas phase
IMGW	Integrated water mass in gas phase
IMHA	Integrated hydrate CO_2 mass
IMHO	Integrated hydrate CH_4 mass
IMHW	Integrated hydrate water mass
IMLA	Integrated air in aqueous phase
IMLO	Integrated oil in aqueous phase, or, Integrated CH_4 mass in aqueous phases
IMLW	Integrated water in aqueous phase
IMO	Integrated oil mass
IMSA	Integrated air or CO_2 mass source
IMSO	Integrated oil or CH_4 mass source
IMSW	Integrated water mass source
IMW	Integrated water mass
IPLW	Integrated well aqueous pumping
IPNW	Integrated mineral CO_2 or well NAPL pumping
MCL	Aqueous moisture content
MCN	NAPL moisture content
MCO2	Mineralized CO_2
MCT	Total liquid moisture content
MPH	Matric Potential Head
OEC	Osmotic efficiency coefficient
PA	Atmospheric pressure
PATH	Scanning path (-1=drying, 1=wetting)
PE_SA	Potential evaporation rate
PG_S	Surface gas pressure
PG	Gas pressure (absolute)
PH	Hydrate pressure (absolute)
PHCN	Phase condition index
PL_S	Surface aqueous pressure
PL	Aqueous pressure (absolute)
PLWB	Aqueous well depth
PM_A	Atmospheric precipitation rate
PM_S	Surface mass precipitation rate
PN	NAPL pressure (absolute)
PORD	Diffusive porosity

Table 6.1. (contd)

Abbreviation	Definition
POSM	Osmotic pressure
PT_SA	Potential transpiration rate
PV_S	Surface vapor pressure, or, Surface volumetric precipitation rate
PVA	Air or CO_2 partial pressure
PVW	Water vapor partial pressure
PW	Monitoring well pressure
QH_S	Surface sensible heat flux
QL_S	Surface latent heat flux
QLW	Aqueous well flow rate
QLWI	Integrated aqueous well flow volume
QNW	NAPL well flow rate
QNWI	Integrated NAPL well flow volume
QTW	Total well flow rate
QTWI	Integrated total liquid well flow volume
RABS	Bare-soil aerodynamic resistance
RFIM	Rainfall interception mass
RH	Atmospheric relative humidity
RHOG	Gas density
RHOH	Hydrate density
RHOL	Aqueous density
RHON	NAPL density
RKG	Gas relative permeability
RKL	Aqueous relative permeability
RKLX	Aqueous relative permeability (x-direction)
RKLY	Aqueous relative permeability (y-direction)
RKLZ	Aqueous relative permeability (z-direction)
RKN	NAPL relative permeability
RL_S	Surface net long-wave radiation
RN	Atmospheric solar radiation
RS_P	Plant stomatal resistance
RSPX	Plant stomatal resistance
RS_S	Surface net short-wave radiation
RSZN	Rock/Solute zone number
RT_S	Surface net total radiation
SG	Gas saturation
SGT	Trapped gas saturation
SGTL	Aqueous trapped gas saturation
SGTN	NAPL trapped gas saturation
SH	Hydrate saturation
SL_S	Surface aqueous saturation
SL	Aqueous saturation

Table 6.1. (contd)

Abbreviation	Definition
SLW	Well water saturation
SN	NAPL saturation
SNF	Free NAPL saturation
SNM	Mobile NAPL saturation
SNR	Residual NAPL saturation
SNT	Trapped NAPL saturation
SPCM	Space monitor
SS	Precipitated salt saturation
ST	Total-liquid saturation
SRCn[(a)]	Solute source integral
SRCA	Air or CO_2 mass source rate
SRCO	Oil mass source rate
SRCQ	Energy source rate
SRCS	Salt mass source rate
SRCT	Thermal energy source rate
SRCW	Water mass source rate
SRIA	Air or CO_2 mass source integral
SRIO	Oil mass source integral
SRIS	Salt or surfactant mass source integral
SRIT	Thermal energy mass source integral
SRIW	Water mass source integral
ST	Total liquid saturation
SXLA	Saturated CO_2 aqueous mass fraction
T_P	Plant temperature
T_S	Surface temperature
T_SA	Actual transpiration rate
T	Temperature
TA	Atmospheric temperature
THKX	Equivalent thermal conductivity (x-direction)
THKY	Equivalent thermal conductivity (y-direction)
THKZ	Equivalent thermal conductivity (z-direction)
TMA	Total air mass
TMO	Total oil mass
TMW	Total water mass
TMS	Total salt mass
TNO	NAPL oil mass
TPNL	NAPL-aqueous trapping number
TSO	Sorbed oil mass, kg oil
UCn[(a)]	Solute flux (x-direction)
UG_W	Gas well cross-screen flux
UG	Gas Darcy velocity (x-direction)
UGNC	Gas Darcy velocity (x-direction, node-centered)

Table 6.1. (contd)

Abbreviation	Definition
UL_W	Water well cross-screen flux
UL	Aqueous Darcy velocity (x-direction)
ULNC	Aqueous Darcy velocity (x-direction, node-centered)
ULO	Dissolved Oil Flux (x-direction)
ULOC	Dissolved oil flux (x-direction, node centered)
UK	Intrinsic permeability (x-direction)
UN	NAPL Darcy velocity (x-direction)
UNNC	NAPL Darcy velocity (x-direction, node-centered)
UQ	Energy flux (x-direction)
UQNC	Energy flux (x-direction, node-centered)
US	Salt flux (x-direction)
USNC	Salt flux (x-direction, node-centered)
VCn[(a)]	Solute flux (y-direction)
VG	Gas Darcy velocity (y-direction)
VGNC	Gas Darcy velocity (y-direction, node-centered)
VISG	Gas phase viscosity
VISL	Aqueous phase viscosity
VISN	NAPL viscosity
VL	Aqueous Darcy velocity (y-direction)
VLNC	Aqueous Darcy velocity (y-direction, node-centered)
VLO	Dissolved Oil Flux (y-direction)
VLOC	Dissolved oil flux (y-direction, node centered)
VK	Intrinsic permeability (y-direction)
VN	NAPL Darcy velocity (y-direction)
VNNC	NAPL Darcy velocity (y-direction, node-centered)
VQ	Energy flux (y-direction)
VQNC	Energy flux (y-direction, node-centered)
VS	Salt flux (y-direction)
VSNC	Salt flux (y-direction, node-centered)
WCn[(a)]	Solute flux (z-direction)
WDL	Aqueous well depth
WDN	NAPL well depth
WDT	Total liquid well depth
WG_W	Gas vertical well flux
WG	Gas Darcy velocity (z-direction)
WGNC	Gas Darcy velocity (z-direction, node-centered)
WK	Intrinsic permeability (z-direction)
WL_W	Aqueous vertical well flux
WL	Aqueous Darcy velocity (z-direction)
WLNC	Aqueous Darcy velocity (z-direction, node-centered)
WLO	Dissolved Oil Flux (z-direction)
WLOC	Dissolved oil flux (z-direction, node centered)

Table 6.1. (contd)

Abbreviation	Definition
WN	NAPL Darcy velocity (z-direction)
WNNC	NAPL Darcy velocity (z-direction, node-centered)
WQ	Energy flux (z-direction)
WQNC	Energy flux (z-direction, node-centered)
WS	Salt flux (z-direction); Atmospheric Wind Speed
WSNC	Salt flux (z-direction, node-centered)
XGA	Air or CO_2 mass fraction in the gas phase
XGO	Oil mass fraction in the gas phase
XGW	Water mass fraction in the gas phase
XGWW	Monitoring well water mass fraction in the gas phase
XHA	Hydrate CO_2 mass fraction
XHO	Hydrate CH_4 mass fraction
XHW	Hydrate water mass fraction
XLA	Air or CO_2 mass fraction in the aqueous phase
XLAW	Monitoring well dissolved air or CO_2 mass fraction
XLO	Oil mass fraction in the aqueous phase
XLS	Salt mass fraction in the aqueous phase
XLW	Water mass fraction in the aqueous phase
XMGA	Air or CO_2 mole fraction in the gas phase
XMGO	Oil mole fraction in the gas phase
XMGW	Water mole fraction in the gas phase
XMHA	Hydrate CO_2 mole fraction
XMHO	Hydrate CH_4 mole fraction
XMHW	Hydrate water mole fraction
XMLA	Aqueous CO_2 mole fraction
XMLS	Aqueous salt mole fraction
XMNO	Oil mole fraction in the NAPL phase
XMNW	Water mole fraction in the gas phase
XNW	Water mass fraction in the NAPL phase
XNO	Oil mass fraction in the NAPL phase
XSO	Sorbed oil mass fraction
YGn[(a)]	Solute fraction in the gas phase
YLn[(a)]	Solute fraction in the aqueous phase
YNn[(a)]	Solute fraction in NAPL
(a) n refers to a solute number.	

6.2 Plot File

A plot file is generated by default at the conclusion of every execution and otherwise when requested on the Output Control Card with the Plot File Output Times input items. If a file named *plot.n* (where *n* refers to the time step extension) already exists in the current directory then the file will be erased and overwritten. The user is responsible for renaming previous *plot.n* files prior to executing the simulator. A plot file begins with a welcome statement, disclaimer, and banner, which should appear similar to

```
 Welcome to ...


                           STOMP
        Subsurface Transport Over Multiple Phases


 This file was produced by STOMP, a numerical simulator
 developed by the Pacific Northwest National Laboratory, with
 support from the VOC-Arid Integrated Demonstration Project,
 Office of Technology Development, U.S. Department of Energy.
 Results from this version of STOMP should not be used for
 license related applications.

 For inquiries or assistance:  Call (509) 372-6070

                      ---  PLOT  ---
```

Following this header are data on the current time step, simulation time, and number of nodes in three coordinate directions. The remaining portion of a *plot.n* file comprises arrays of geometric and variable data. These data are arranged in groups, each with a title line that indicates the type of data. Data groups are arranged in rows with 10 data items per row. Each group of data contains a field or surface variable listed sequentially for every node or surface in the computational domain. The first four groups of data appear in every *plot.n* file and contain the x- or radial-direction node positions, y- or azimuthal-direction node positions, z-direction node positions, and node volumes for every node in the computational domain. The remaining groups of data correspond to the requested *Plot File Variables.* Field variable data groups list the value of the field variable at the node centroid. Surface variables and/or fluxes are written to the plot file twice. The first flux variable data group lists the value of the flux variable at the node centroid, whereas the second group lists the value at the node surfaces. Node centroid values for flux variables are simple arithmetic averages of the two node surface values.

Post-processing of the *plot* files can be done with the Perl script *plotTo.pl*. The Perl program transforms the output file into a formatted input for Gnuplot, Grapher, Igor, Matlab, and Tecplot. The minimum command line is

plotTo.pl

after which the user is prompted for the plotting package name, plotting package input file name, STOMP plot file name(s), and variable names. However, the command line can be expanded to include these options. A few examples of more expanded command lines are

plotTo.pl Surfer plot.dat plot.123
--prints to file plot.dat

plotTo.pl -t -ts Tecplot plots.dat plot.123 plot.456 plot.789
-- prints to file plots.dat

plotTo.pl -t Gnuplot plots.dat plot.[0-9][0-9]
-- all available plot files ranging from plot.0 to plot.99 are printed to plots.day

plotTo.pl Tecplot plots.dat plot.*
-- all plot files will be printed to plots.dat.

plotTo.pl -t Tecplot plots.dat `ls -1rt plot.*`
-- plot files in the listing made by `ls -1rt plot.*` are printed to plots.dat

6.3 Surface File

A surface file named *surface* is generated during an execution if surfaces were defined on the *Surface Flux Card*. If a file named *surface* already exists in the current directory then that file will be erased and overwritten. The user is responsible for renaming previous *surface* files prior to executing the simulator. A *surface* file begins with a welcome statement, disclaimer, and banner, which should appear similar to

```
 Welcome to ...


                           STOMP
        Subsurface Transport Over Multiple Phases


 This file was produced by STOMP, a numerical simulator
 developed by the Pacific Northwest National Laboratory, with
 support from the VOC-Arid Integrated Demonstration Project,
 Office of Technology Development, U.S. Department of Energy.
 Results from this version of STOMP should not be used for
 license related applications.

 For inquiries or assistance:  Call (509) 372-6070

                      ---  SURFACE  ---
```

Following the header are columns of surface flux rate and integral data. Surface flux information is written to the *surface* file every time step, at the conclusion of every time step. The first column of data in the *surface* file is the time step in units, which are specified with the *Output Time Units* variable on the Output Control Card. Each surface flux output is written in two columns, the first column being the surface flux rate at the current time step and the second being the integral of the surface flux rate from the beginning of the current simulation. Surface flux integral data are not carried over on restart simulations. Each column in the *surface* file is titled with an abbreviated header and associated units shown in brackets below the column title. Definitions of the header abbreviations for the *surface* file are given in Table 6.2. Each header title is followed with an integer enclosed in parentheses. The integer refers to the surface flux number, as defined on the Surface Flux Card (i.e., the surface flux definition order).

Post-processing of the *surface* files can be done with the Perl script *surfaceTo.pl*. The Perl program transforms the output file into a formatted input for Gnuplot, Grapher, Igor, Matlab, and Tecplot. The minimum command line is

surfaceTo.pl

after which the user is prompted for the plotting package name, plotting package input file name, STOMP surface file name, surfaces, and surface-flux variable names. However, the command line can be expanded to include these options. A few examples of more expanded command lines are

surfaceTo.pl -as -t Gnuplot surface.dat surface
-- prints all surfaces (variables and title will be prompted) to file output.dat

surfaceTo.pl -av -as Tecplot surface.dat surface
-- prints all surfaces and variables to file surface.dat

surfaceTo.pl Grapher surface.dat surface
-- prints (variables and surfaces will be prompted) to file surface.dat

surfaceTo.pl -av Tecplot surface.dat surface
-- prints all variables (surfaces will be prompted) to file surface.dat

6.4 Restart File

A restart file is generated by default at the conclusion of every execution and otherwise when requested on the Output Control Card with the Plot File Output Times input items. If a file named *restart.n* (where *n* refers to the time step extension) already exists in the current directory then the file will be erased and overwritten. The user is responsible for renaming previous *restart.n* files prior to executing the simulator. The user has the option to specify a “no restart” or “final restart” as a plot variable. A restart file begins with a welcome statement, disclaimer, and banner, which should appear similar to

```
 Welcome to ...


                           STOMP
        Subsurface Transport Over Multiple Phases


 This file was produced by STOMP, a numerical simulator
 developed by the Pacific Northwest National Laboratory, with
 support from the VOC-Arid Integrated Demonstration Project,
 Office of Technology Development, U.S. Department of Energy.
 Results from this version of STOMP should not be used for
 license related applications.

 For inquiries or assistance:  Call (509) 372-6070

                      ---  RESTART  ---
```

All data values in restart files are in SI units with pressures expressed in gauge. The first data line of a restart file contains the following timing and simulation information: time, time step, maximum time step, time step acceleration factor, convergence criterion, maximum number of Newton-Raphson iterations, number of time steps, and number of nodes. The remaining lines are node data lines, which contain the following field variable data for the current time step for every node (one line per node): temperature, aqueous pressure, gas pressure, NAPL pressure, oil vapor pressure, water vapor pressure, aqueous saturation, gas saturation, and NAPL saturation. If the simulation includes solute transport then the node data lines will also contain solute volumetric concentrations for each solute. All node data lines are terminated with an integer value for the phase condition.

Table 6.2. Surface Flux Column Title Abbreviations

Abbreviation	Definition
AE	Surface actual evaporation
AT	Surface potential evaporation
PE	Surface actual transpiration
PT	Surface actual evaporation
NLWR	Surface net long-wave radiation
NSWR	Surface net short-wave radiation
NTR	Surface net total radiation
UAGOR	Gas-advective oil-mass flux rate (x-direction)
UAGOI	Gas-advective oil-mass flux rate integral (x-direction)
UAMR	Total CO_2 mass flux rate (x-direction)
UAMI	Total CO_2 mass flux rate integral (x-direction)
UCMR	Solute flux (x-direction)
UCMI	Solute flux integral (x-direction)
UDGOR	Gas-diffusive oil-mass flux rate (x-direction)
UDGOI	Gas-diffusive oil-mass flux rate integral (x-direction)
UGAAR	Gas advective air mass flux (x-direction)
UGAAI	Gas advective air mass flux integral (x-direction)
UGAMR	Gas CO_2 mass flux rate (x-direction)
UGAMI	Gas CO_2 mass flux rate integral (x-direction)
UGAQR	Gas advective heat flux (x-direction)
UGAQI	Gas advective heat flux integral (x-direction)
UGAWR	Gas advective water mass flux (x-direction)
UGAWI	Gas advective water mass flux integral (x-direction)
UGDAR	Gas diffusive air mass flux (x-direction)
UGDAI	Gas diffusive air mass flux integral (x-direction)
UGDQR	Gas diffusive heat flux (x-direction)
UGDQI	Gas diffusive heat flux integral (x-direction)
UGDWR	Gas diffusive water mass flux (x-direction)
UGDWI	Gas diffusive water mass flux integral (x-direction)
UGMR	Gas mass flux rate (x-direction)
UGMI	Gas mass flux rate integral (x-direction)
UGOR	Gas-total oil-mass flux rate (x-direction)
UGOI	Gas-total oil-mass flux rate integral (x-direction)
UGOMR	Gas CH_4 mass flux rate (x-direction)
UGOMI	Gas CH_4 mass flux rate integral (x-direction)
UGVR	Gas volumetric flux rate (x-direction)
UGVI	Gas volumetric flux rate integral (x-direction)
ULAMR	Aqueous CO_2 mass flux rate (x-direction)
ULAMI	Aqueous CO_2 mass flux rate integral (x-direction)
ULMR	Aqueous mass flux rate (x-direction)
ULMI	Aqueous mass flux rate integral (x-direction)
ULOMR	Aqueous CH_4 mass flux rate (x-direction)

Table 6.2. (contd)

Abbreviation	Definition
ULOMI	Aqueous CH_4 mass flux rate integral (x-direction)
ULVR	Aqueous volumetric flux rate (x-direction)
ULVI	Aqueous volumetric flux rate integral (x-direction)
UNMR	NAPL mass flux rate (x-direction)
UNMI	NAPL mass flux integral (x-direction)
UNVR	NAPL volumetric flux rate (x-direction)
UNVI	NAPL volumetric flux integral (x-direction)
UOMR	Total CH_4 mass flux rate (x-direction)
UOMI	Total CH_4 mass flux rate integral (x-direction)
UQVR	Energy flux rate (x-direction)
UQVI	Energy flux integral (x-direction)
USMR	Salt mass flux rate (x-direction)
USMI	Salt mass flux integral (x-direction)
UWMR	Water mass flux (x-direction)
UWMI	Water mass flux integral (x-direction)
VAGOR	Gas-advective oil-mass flux rate (y-direction)
VAGOI	Gas-advective oil-mass flux rate integral (y-direction)
VAMR	Total CO_2 mass flux rate (y-direction)
VAMI	Total CO_2 mass flux rate integral (y-direction)
VCMR	Solute flux (y-direction)
VCMI	Solute flux integral (y-direction)
VDGOR	Gas-diffusive oil-mass flux rate (y-direction)
VDGOI	Gas-diffusive oil-mass flux rate integral (y-direction)
VGAAR	Gas advective air mass flux (y-direction)
VGAOR	Gas advective oil mass flux (y-direction)
VGAMR	Gas CO_2 mass flux rate (y-direction)
VGAMI	Gas CO_2 mass flux rate integral (y-direction)
VGAOI	Gas advective oil mass flux integral (y-direction)
VGAWR	Gas advective water mass flux (y-direction)
VGAWI	Gas advective water mass flux integral (y-direction)
VGDAR	Gas diffusive air mass flux (y-direction)
VGDAI	Gas diffusive air mass flux integral (y-direction)
VGDQR	Gas diffusive heat flux (y-direction)
VGDQI	Gas diffusive heat flux integral (y-direction)
VGDWR	Gas diffusive water mass flux (y-direction)
VGDWI	Gas diffusive water mass flux integral (y-direction)
VGMR	Gas mass flux rate (y-direction)
VGMI	Gas mass flux rate integral (y-direction)
VGOR	Gas-total oil-mass flux rate (x-direction)
VGOI	Gas-total oil-mass flux rate integral (x-direction)
VGOMR	Gas CH_4 mass flux rate (y-direction)
VGOMI	Gas CH_4 mass flux rate integral (y-direction)

Table 6.2. (contd)

Abbreviation	Definition
VGVR	Gas volumetric flux rate (y-direction)
VGVI	Gas volumetric flux rate integral (y-direction)
VLAMR	Aqueous CO_2 mass flux rate (y-direction)
VLAMI	Aqueous CO_2 mass flux rate integral (y-direction)
VLMR	Aqueous mass flux rate (y-direction)
VLMI	Aqueous mass flux rate integral (y-direction)
VLOMR	Aqueous CH_4 mass flux rate (y-direction)
VLOMI	Aqueous CH_4 mass flux rate integral (y-direction)
VLVR	Aqueous volumetric flux rate (y-direction)
VLVI	Aqueous volumetric flux rate integral (y-direction)
VNMR	NAPL mass flux rate (y-direction)
VNMI	NAPL mass flux integral (y-direction)
VNVR	NAPL volumetric flux rate (y-direction)
VNVI	NAPL volumetric flux integral (y-direction)
VOMR	Total CH_4 mass flux rate (y-direction)
VOMI	Total CH_4 mass flux rate integral (y-direction)
VQVR	Energy flux rate (y-direction)
VQVI	Energy flux integral (y-direction)
VSMR	Salt mass flux rate y-direction)
VSMI	Salt mass flux integral (y-direction)
VWMR	Water mass flux (y-direction)
VWMI	Water mass flux integral (y-direction)
WAGOR	Gas-advective oil-mass flux rate (z-direction)
WAGOI	Gas-advective oil-mass flux rate integral (z-direction)
WAMR	Total CO_2 mass flux rate (z-direction)
WAMI	Total CO_2 mass flux rate integral (z-direction)
WCMR	Solute flux (z-direction)
WCMI	Solute flux integral (z-direction)
WGAAR	Gas advective air mass flux (z-direction)
WGAAI	Gas advective air mass flux integral (z-direction)
WGAMR	Gas CO_2 mass flux rate (z-direction)
WGAMI	Gas CO_2 mass flux rate integral (z-direction)
WGAQR	Gas advective heat flux (z-direction)
WGAQI	Gas advective heat flux integral (z-direction)
WGAWR	Gas advective water mass flux (z-direction)
WGAWI	Gas advective water mass flux integral (z-direction)
WGDAR	Gas diffusive air mass flux (z-direction)
WGDAI	Gas diffusive air mass flux integral (z-direction)
WGDQR	Gas diffusive heat flux (z-direction)
WGDQI	Gas diffusive heat flux integral (z-direction)
WGDWR	Gas diffusive water mass flux (z-direction)
WGDWI	Gas diffusive water mass flux integral (z-direction)

Table 6.2. (contd)

Abbreviation	Definition
WDGOR	Gas-diffusive oil-mass flux rate (z-direction)
WDGOI	Gas-diffusive oil-mass flux rate integral (z-direction)
WGMR	Gas mass flux rate (z-direction)
WGMI	Gas mass flux rate integral (z-direction)
WGOR	Gas-total oil-mass flux rate (z-direction)
WGOI	Gas-total oil-mass flux rate integral (z-direction)
WGOMR	Gas CH_4 mass flux rate (z-direction)
WGOMI	Gas CH_4 mass flux rate integral (z-direction)
WGVR	Gas volumetric flux rate (z-direction)
WGVI	Gas volumetric flux rate integral (z-direction)
WLAMR	Aqueous CO_2 mass flux rate (z-direction)
WLAMI	Aqueous CO_2 mass flux rate integral (z-direction)
WLMR	Aqueous mass flux rate (z-direction)
WLMI	Aqueous mass flux rate integral (z-direction)
WLOMR	Aqueous CH_4 mass flux rate (z-direction)
WLOMI	Aqueous CH_4 mass flux rate integral z-direction)
WLVR	Aqueous volumetric flux rate (z-direction)
WLVI	Aqueous volumetric flux rate integral (z-direction)
WMB	Surface water mass balance
WNMR	NAPL mass flux rate (z-direction)
WNMI	NAPL mass flux integral (z-direction)
WNVR	NAPL volumetric flux rate (z-direction)
WNVI	NAPL volumetric flux integral (z-direction)
WOMR	Total CH_4 mass flux rate (z-direction)
WOMI	Total CH_4 mass flux rate integral (z-direction)
WQVR	Energy flux rate (z-direction)
WQVI	Energy flux integral (z-direction)
WSMR	Salt mass flux rate (z-direction)
WSMI	Salt mass flux integral (z-direction)
WWMR	Water mass flux (z-direction)
WWMI	Water mass flux integral (z-direction)

7.0 References

ANSI. 1978. *American National Standard Programming Language FORTRAN.* X3.9-1978, American National Standards Institute, Broadway, New York.

Brooks, RH, and AT Corey. 1964. *Hydraulic properties of porous media.* Hydrology Papers 3, Colorado State University, Fort Collins, Colorado.

DOE Order 413.2A. 2001. Laboratory Directed Research and Development. U. S. Department of Energy, Washington, D.C.

Fang Y, GT Yeh, and WD Burgos. 2003. "A General Paradigm to Model Reaction-Based Biogeochemical Process in Batch Systems." *Water Resources Research* 39(4):1083.

Poeter EP and MC Hill. 1998. *Documentation of UCODE, a Computer Code for Universal Inverse Modeling*. U.S. Geological Survey Water-Resources Investigations Report 98-4080, 116 pp.

Reid RC, JM Prausnitz, and BE Poling. 1987. *The Properties of Gases and Liquids*. McGraw-Hill, New York.

Shuttleworth, JW and J.S Wallace. 1985. "Evaporation from sparse crops-an energy combination theory." *Quart. J. R. Met. Soc.*, 111:839-855.

Talbott S. 1988. *Managing Projects with Make.* O'Reilly and Associates, Inc., Newton, Massachusetts.

van Genuchten, M. 1980. "A Closed-Form Equation for Predicting the Hydraulic Conductivity of Unsaturated Soils." *Soil Sci. Am. J.* 44:892-898.

Ward AL, MD White, EJ Freeman, and ZF Zhang. 2005. *STOMP: Subsurface Transport Over Multiple Phases. Version 1.0. Addendum: Sparse Vegetation Evapotranspiration Model for the Water-Air-Energy Operational Mode.* PNNL-15465, Pacific Northwest National Laboratory, Richland, Washington.

White MD and BP McGrail. 2005. *STOMP: Subsurface Transport Over Multiple Phases. Version 1.0. Addendum: ECKEChem Equilibrium-Conservation-Kinetic Equation Chemistry and Reactive Transport.* PNNL-15482, Pacific Northwest National Laboratory, Richland, Washington.

White MD and M Oostrom. 2000. *STOMP Subsurface Transport Over Multiple Phases Theory Guide Version 2.0.* PNNL-12030, Pacific Northwest National Laboratory, Richland, Washington.

8.0 STOMP Related Publications

This section includes a partial list of peer-reviewed and conference proceedings papers in which the STOMP simulator has been used for numerical flow and transport applications.

Ataie-Ashtiani B, SM Hassanizadeh, M Oostrom, MA Celia, and MD White. 2001. "Numerical Simulation and Homogenization of Two-Phase Flow in Heterogeneous Porous Media." *J. of Contaminant Hydrology* 49: 87-110.

Ataie-Ashtiani B, SM Hassanizadeh, MA Celia, M Oostrom, and MD White. 2000. "Effective Two-Phase Flow Parameters for Heterogeneous Porous Media." In *Proceedings of Computational Methods in Water Resources XIII*, LR Bentley et al. (eds.), pp. 755-762, Kluwer Ac. Publ. London.

Ataie-Ashtiani B, SM Hassanizadeh, M Oostrom, and MD White. 2000. "Numerical Simulation and Homogenization of Two-Phase Flow in Heterogeneous Porous Media." In *Groundwater Updates, Proceedings of the IAHR International Symposium on Groundwater*, K Sato and Y Iwasa (eds.), pp. 333-338, Springer-Verlag, Tokyo.

Davis BM, JD Istok, and L Semprini. 2002. "Push-Pull Partitioning Tracer Tests Using Radon-222 to Quantify Nonaqueous Phase Liquid Contamination." *J. Contam. Hydrol*. 58:129-146.

Gee GW, AL Ward, ZF Zhang, GS Campbell, and J Mathison. 2002. "The Influence of Hydraulic Nonequilibrium on Pressure Plate Data." *Vadose Zone Journal* 1:172-178.

Gee GW, ZF Zhang, SW Tyler, WH Albright, and MJ Singleton. 2005. "Chloride Mass Balance: Cautions in Predicting Increased Recharge Rates." *Vadose Zone Journal* 4(1):72-78.

Hofstee C and M Oostrom. 2002. "Formation of Residual NAPL in Three-Phase Systems. Experiments and Numerical Simulations." In *Proceedings of Computational Methods in Water Resources XIII*, SM Hassanizadeh et al. (eds.), pp. 193-202.

Hofstee C, M Oostrom, JH Dane, and RW Walker. 1998. "Infiltration and Redistribution of Perchloroethylene in Partially Saturated, Stratified Porous Media." *J. of Contaminant Hydrol*. 34:293-313.

Lenhard RJ, M Oostrom, CS Simmons, and MD White. 1995. "Investigation of Density-Dependent Gas Advection of Trichloroethylene: Experiment and a Model Validation Exercise. *J. of Contaminant Hydrol*. 19:47-67.

Lenhard RJ, M Oostrom, and MD White. 1995. "Modeling Fluid Flow and Transport in Variably Saturated Porous Media with the STOMP Simulator. 2. Verification and Validation Exercises." *Adv. in Water Resour*. 18:365-374.

Oostrom M, JH Dane, and TW Wietsma. 2005. "Removal of Carbon Tetrachloride from a Layered Porous Medium by Means of Soil Vapor Extraction Enhanced by Desiccation and Water Table Reduction." *Vadose Zone J*. 4:1170-1182.

Oostrom M, JH Dane, and TW Wietsma. 2005. "Carbon Tetrachloride Removal from a Heterogeneous Porous Medium by Two Soil Vapor Extraction Techniques. In *Proceedings of the American Geophysical Union Hydrology Days*, JA Ramirez (ed.), pp. 187-197. Colorado State University, Fort Collins, Colorado.

Oostrom M, C Hofstee, and JH Dane. 1997. "Light Nonaqueous-Phase Liquid Movement in a Variably Saturated Sand." *Soil Science Society of America J*. 61:1547-1554.

Oostrom M, C Hofstee, JH Dane, and MD White. 1996. "Simulation of a Quantitative Two-Dimensional Multi-Fluid Flow Experiment." In *Proceedings of the XI International Conference on Computational Methods in Water Resources*, AA Aldema et al. (eds.), pp. 449-556, Kluwer Ac. Publ. London.

Oostrom M, C Hofstee, RJ Lenhard, and TW Wietsma. 2003. Flow Behavior and Residual Saturation Formation of Injected Carbon Tetrachloride in Unsaturated Heterogeneous Porous Media." *J. of Contaminant Hydrology* 64:93-112.

Oostrom M, C Hofstee, RC Walker, and JH Dane. 1999. "Movement and Remediation of Trichloroethylene in a Saturated, Heterogeneous Porous Medium. 1. Spill Behavior and Initial Dissolution." *J. of Contaminant Hydrol*. 37:159-178.

Oostrom M, C Hofstee, RC Walker, and JH Dane. 1999. Movement and Remediation of Trichloroethylene in a Saturated, Heterogeneous Porous Medium. 2. Pump-and-Treat and Surfactant Flushing." *J. of Contaminant Hydrol*. 37:179-197.

Oostrom M, C Hofstee, and TW Wietsma. 2006. Behavior of a Viscous LNAPL under Variable Water Table Conditions: Experiment and Numerical Simulation." *J. of Soil and Sediment Contam*. In Press.

Oostrom M, C Hofstee, and TW Wietsma. 2006. LNAPLs do not Always Float: An Example Case of a Viscous LNAPL Under Variable Water Table Conditions." In *Proceedings of the American Geophysical Union Hydrology Days*, JA Ramirez (ed.), pp. 119-130, Colorado State University, Fort Collins, Colorado.

Oostrom M and RJ Lenhard. 1996. "Radon Transport into Dwellings: Considering Groundwater as a Potential Source." *Geoph. Res. Lett*. 23:1577-1580.

Oostrom M and RJ Lenhard. 1998. "Comparison of Relative Permeability-Saturation-Pressure Parametric Models for Infiltration and Redistribution of a Light Nonaqueous Phase Liquid in Sandy Porous Media." *Adv. in Water Resour*. 21:145-157.

Oostrom M and RJ Lenhard. 2003. "Carbon Tetrachloride Flow Behavior in Unsaturated Hanford Caliche Material: An Investigation of Residual NAPL." *Vadose Zone J*. 2:25-33.

Oostrom M, RJ Lenhard, P van Geel, MD White, and TW Wietsma. 2005. Comparison of Models Describing Residual NAPL Formation in the Vadose Zone." *Vadose Zone J*. 4:163-174.

Oostrom M, MD White, and ML Brusseau. 2001. "Theoretical Estimation of Interfacial Areas of Entrapped and Free Nonwetting Fluids in Porous Media." *Advances in Water Resources* 24:887-898.

Oostrom M, RJ Lenhard, and MD White. 1995. Infiltration of Dense and Light Nonaqueous Phase Liquids in Partly Saturated Sand Columns." In *Proceedings of the American Geophysical Union Hydrology Days*, H Morel-Seytoux (ed.), pp. 215-226, Hydrology Days Publications, Atherton, California.

Oostrom M, RJ Lenhard, and MD White. 1996. "Numerical Evaluation of Groundwater as a Supply for Radon in Dwellings." In *Proceedings of the XI International Conference on Computational Methods in Water Resources*, AA Aldema et al. (eds.), pp. 193-200, Kluwer Ac. Publ. London.

Oostrom M, PD Thorne, MD White, MJ Truex, and TW Wietsma. 2003. "Numerical Modeling to Assess DNAPL Movement and Removal at the Scenic Site Operable Unit near Baton Rouge, Louisiana: A Case Study." *J. of Soil and Sediment Contamination*, 12:1-26.

Oostrom M, MJ Truex, PD Thorne, and TW Wietsma. 2006. "Three-Dimensional Multifluid Flow and Transport at the Brooklawn Site near Baton Rouge, LA. A Case Study." *J. of Soil and Sediment Contam*. (In Press).

Rockhold ML, CJ Murray, and MJ Fayer. 1999. "Conditional Simulation and Upscaling of Soil Hydraulic Properties." In *Proceedings of the International Workshop on Characterization and Measurement of the Hydraulic Properties of Unsaturated Porous Media*, M Th Van Genuchten, FJ Leij, and L Wu (eds.), University of California, Riverside, pp. 1391-1401.

Rockhold ML, RE Rossi, and RG Hills. 1996. "Application of Similar Media Scaling and Conditional Simulation for Modeling Water Flow and Tritium Transport at the Las Cruces Trench Site." *Water Resour. Res*. 32(3):595-609.

Rockhold ML. 1999. "Parameterizing Flow and Transport Models for Field-Scale Applications in Heterogeneous, Unsaturated Soils." In *Assessment on Non-Point Source Pollution in the Vadose Zone*, DL Corwin, K Loague, and TR Ellsworth (eds.), Geophysical Monograph 108, American Geophysical Union, Washington, D.C., pp. 243-260.

Schroth MH, JD Istok, JS Selker, M Oostrom, and MD White. 1998. "Multifluid Flow in Bedded Porous Media: Laboratory Experiments and Numerical Simulations." *Adv. in Water Resour*. 22:169-183.

Thornton EC, L Zhong, and M Oostrom. 2006. "Development of a Field Design for In Situ Gaseous Treatment of Sediment Based on Laboratory Column Test Data and Numerical Analysis." *J. or Environmental Eng*. (In Press).

Ward AL, ZF Zhang, GW Gee. 2006. "Upscaling Unsaturated Hydraulic Parameters for Flow Through Heterogeneous Anisotropic Sediments." *Adv. in Water Resour*. 29:268-280.

White MD and RJ Lenhard. 1993. "Numerical Analysis of a Three-Phase System with a Fluctuating Water Table." In *Proceedings of Thirteenth Annual American Geophysical Union Hydrology Days, Fort Collins, Colorado*. Hydrology Days Publication, Atherton, California.

White MD and BP McGrail. 2002. "Numerical Investigations of Multifluid Hydrodynamics During Injections of Supercritical CO_2 into Porous Media." In *Proceedings of GHGT-6, Sixth International Conference on Greenhouse Gas Control Technologies*, Kyoto, Japan. Elsevier Science, Amsterdam, The Netherlands.

White MD and M Oostrom. 1997. "Mathematical Modeling of Surfactant-Enhanced Mobilization and Kinetic Dissolution of DNAPLs in Soil Columns." In *Proceedings of the American Geophysical Union Hydrology Days*, H Morel-Seytoux (ed.), pp. 349-363, Hydrology Days Publications, Atherton, California.

White MD and M Oostrom. 1998. Modeling Surfactant-Enhanced Nonaqueous Phase Liquid Remediation of Porous Media." *Soil Science* 163:931-940.

White MD and M Oostrom. 2001. "NAPL Migration in Response to Hydraulic Controls at the Brooklawn Site near Baton Rouge." In *Proceedings of the 21th American Geophysical Union Hydrology Days*, HJ Morel-Seytoux (ed.), pp. 241-254, Colorado State University, Fort Collins, Colorado.

White MD, M Oostrom, and RJ Lenhard. 1995. "Modeling Fluid Flow and Transport in Variably Saturated Porous Media with the STOMP Simulator. 1. Nonvolatile Three-Phase Model Description." *Adv. in Water Resour*. 18:353-364.

White MD, M Oostrom and RJ Lenhard. 2004. A Practical Model for Mobile, Residual, and Entrapped NAPL in Water-Wet porous Media." *Groundwater* 42:734-746.

White MD, MD Williams, and M Oostrom. 1999. Simulating Air Entrapment, Mobilization, and Dissolution for Fluctuating Water Table Conditions." In *Proceedings of the American Geophysical Union Hydrology Days*, HJ Morel-Seytoux (ed.), pp. 467-482, Hydrology Days Pub., Atherton, California.

White MD and AL Ward. 2005. "Numerical Simulations of Surface Barriers for Shrub-Steppe Ecoregions." In *Proceedings of the American Geophysical Union Hydrology Days*, JA Ramirez (ed.), pp. 224-236, Colorado State University, Fort Collins, Colorado.

Williams MD and M Oostrom. 2000. "Re-Oxygenation of Anoxic Water in a Laboratory Fluctuating Water Table System." *J. of Hydrology* 230:70-85.

Zhang ZF, AL Ward, and GW Gee. 2003. "Describing the Unsaturated Hydraulic Properties of Anisotropic Soils Using a Tensorial Connectivity-Tortuosity (TCT) Concept." *Vadose Zone Journal* 2(3):313-321.

Zhang ZF, AL Ward, and GW Gee. 2003. "Inversely Estimating Soil Hydraulic Parameters through Field Drainage Experiment." *Vadose Zone Journal* 2:201-211.

Zhang ZF, AL Ward, and GW Gee. 2004. "A Combined Parameter Scaling and Inverse Technique to Upscale the Unsaturated Hydraulic Parameters for Heterogeneous Soils." *Water Resour. Res*. 40(8), W08306, doi:10.1029/2003WR002925.

Zhang ZF, AL Ward, and GW Gee. 2004. "A Parameter Scaling Concept for Estimating Field-Scale Hydraulic Functions of Layered Soils." *Journal of Hydraulic Research* 42sup:93-103.