U.S. DEPARTMENT OF
ENERGY

PNNL-21268

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

STOMP
Subsurface Transport Over Multiple Phases

STOMP-CO2 and -CO2e Guide

Version 1.0

M.D. White D.J. Watson
D.H. Bacon S.K. White
B.P McGrail Z.F. Zhang

April 2012

Pacific Northwest
NATIONAL LABORATORY

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.

(9/2003)

PNNL-21268

STOMP

Subsurface Transport Over Multiple Phases

STOMP-CO2 and -CO2e Guide

Version 1.0

M.D. White D.J. Watson
D.H. Bacon S.K. White
B.P. McGrail Z.F. Zhang

April 2012

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

Pacific Northwest National Laboratory
Richland, Washington 99352

Preface

This STOMP (Subsurface Transport Over Multiple Phases) guide document describes the theory, use, and application of the STOMP-CO2 and STOMP-CO2e operational modes. These operational modes of the STOMP simulator are configured to solve problems involving the geologic sequestration of CO_2 in saline reservoirs. STOMP-CO2 is the isothermal version and STOMP-CO2e is the nonisothermal version. These core operational modes solve the governing conservation equations for component flow and transport through geologic media; where, the STOMP-CO2 components are water, CO_2 and salt and the STOMP-CO2e operational mode also includes an energy conservation equation. Geochemistry can be included in the problem solution via the ECKEChem (Equilibrium-Conservation-Kinetic-Equation Chemistry) module, and geomechanics via the EPRMech (Elastic-Plastic-Rock Mechanics) module. This addendum is designed to provide the new user with a full guide for the core capabilities of the STOMP-CO2 and -CO2e simulators, and to provide the experienced user with a quick reference on implementing features. Several benchmark problems are provided in this addendum, which serve as starting points for developing inputs for more complex problems and as demonstrations of the simulator's capabilities.

STOMP-CO2 and -CO2e are written in Fortran 90 with dynamic memory allocation. The codes can be configured for either a banded or conjugate gradient linear system solver. The simulators are provided as source code to encourage the open exchange of scientific and mathematical ideas, but this requires that the user compile and link the code into an executable. In writing this addendum the authors have assumed that the reader is familiar with numerical simulation of multifluid subsurface flow and reactive transport and with the computing environment on which they plan to compile and execute the STOMP-CO2 and -CO2e simulators. The simulator is maintained following a configuration management plan as a collection of source code files. Assembly of the library files into a single source code or executable occurs through a software maintenance utility. Version numbers are assigned to individual files in the STOMP library of files and those version numbers are reported to standard output and the "output" file for the active files in the executable at the conclusion of the execution. The memory requirements for executing the STOMP-CO2 and -CO2e simulators are dependent on the complexity of the 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.

Summary

Geologic sequestration is currently being practiced and scientifically evaluated as a critical component in a broad strategy, comprising new practices and technologies, for mitigating global climate change due to anthropogenic emissions of CO_2. Demonstrating that geologic sequestration of CO_2 is safe and effective, and gaining public acceptance of sequestration technologies is critically important in meeting these global climate change challenges. Monitored field-scale demonstrations of geologic sequestration of carbon dioxide will contribute greatly toward growing trust and confidence in the technology; however, pilot demonstrations ultimately will not be the norm for new geological sequestration deployments. Instead, scientists, engineers, regulators, and ultimately the public will rely on numerical simulations to predict the performance of geologic repositories for carbon dioxide sequestration.

The U.S. Department of Energy (DOE), through the National Environmental Technology Laboratory (NETL) has requested the development of numerical simulation capabilities for quantifying the permanent storage capacity, leakage rates, and public risks associated with geologic sequestration of CO_2. In conjunction with this request. the Zero Emissions Research and Technology (ZERT) Center has been created with the mission of conducting basic and applied research that supports the development of new technologies for minimizing emissions of anthropogenic carbon dioxide and other greenhouse gases that impact global climate change. As a member of the ZERT Center, the Pacific Northwest National Laboratory[1] (PNNL) is conducting research associated with geologic sequestration of CO_2 that includes the thermochemistry of supercritical CO_2-brine mixtures, mineralization kinetics, leakage and microseepage of CO_2, and new materials for CO_2 capture. In addition to these research activities, PNNL is developing new scalable CO_2 reservoir simulation capabilities for its multifluid subsurface flow and transport simulator, STOMP (Subsurface Transport Over Multiple Phases).

Potential geologic reservoirs for sequestering CO_2 include deep saline aquifers, depleted or partially depleted natural gas and petroleum reservoirs, coal beds, and hydrate-bearing formations. The STOMP-CO2 and STOMP-CO2e simulators were designed to investigate the geologic sequestration of CO_2 in saline reservoirs, whose upper depth is greater than 800-m below ground surface. To sequester CO_2 in deep saline reservoirs, CO_2 is injected into the reservoir through "screened" intervals. The injected CO_2 is only slightly miscible with the reservoir brine and forms a CO_2 dominate phase, which can be a liquid or gas. A liquid phase forms for pressures greater than the saturation pressure for temperatures below the critical point for CO_2 (31.06°C and 7.3825 MPa), and a gas phase forms for pressures less than the saturation pressure, up to the critical pressure. For temperature and pressure conditions above the critical point, a supercritical gas phase forms. The solubility of water in the CO_2 dominate phase is low, but contributes greatly to the near-well saturation profiles when dry CO_2 is injected into the reservoir.

The principal objective for geologic sequestration is trapping CO_2 deep in the subsurface in a stable, immobile form, such that under ambient reservoir conditions the sequestered CO_2 does not leak into the atmosphere or Underground Sources of Drinking Water (USDW). For deep saline geologic reservoirs there are four primary trapping mechanisms: 1) structural, 2) dissolution, 3) hydraulic and 4) mineralization. Structural trapping occurs when the predominately CO_2 phase (gas) becomes immobile by impermeable barriers (e.g., overlying caprock, fault). Dissolution trapping occurs when the injected CO_2 dissolves into an immobile aqueous phase (i.e., brine). Hydraulic trapping occurs when the mobile and connected gas phase forms immobile ganglia of trapped gas with the imbibition of aqueous phase (i.e., the replacement of a nonwetting fluid with a wetting fluid). Both gaseous and liquid CO_2 are considered to be nonwetting fluids compared with the aqueous phase. Mineralization is the conversion of dissolved or gas phase CO_2 into carbonate minerals. Mineralization occurs through two mechanisms, depending on the aqueous saturation

[1] Pacific Northwest National Laboratory is operated for the U.S. Department of Energy by Battelle Memorial Institute

states. Under moderate to high aqueous saturation conditions, carbonate formation occurs via dissolution of the primary minerals with a reduction in brine pH and subsequent precipitation of carbonate minerals. Under low aqueous saturation conditions the thin water film on surfaces of the formation minerals catalyzes a more rapid carbonation reaction. None of the trapping mechanisms are irreversible. Changes in the subsurface environment (e.g., pressure reduction, temperature increases, or geomechanical alterations) could result in the liberation of trapped CO_2. The core modeling capabilities in the STOMP-CO2 and -CO2e include the three first trapping mechanisms (i.e., structural, dissolution, and hydraulic). The ECKEChem module allows for mineralization trapping.

The STOMP-CO2 and -CO2e simulators solve conservation equations for component mass (i.e., water, CO_2, and salt) and energy on a structured orthogonal grid. In their native form the conservation equations are expressed as partial differential equations. Except for certain geometries, boundary conditions, and simplified property functions there are no analytic solutions to these equations, leaving numerical solutions as the only option for general problems. To solve these conservation equations numerically they are first converted to algebraic form using the integral finite difference approach applied to structured orthogonal grids and Euler-backward time differences. The resulting algebraic equations are closed through a series of constitutive equations. Although in algebraic form, the system of conservation equations and constitutive equations are highly nonlinear. Newton-Raphson iteration is used to resolve these nonlinearities.

Each conservation equation is solved for a primary variable. The primary variables must be chosen to be independent and completely define the state of the system. All secondary variables must be determined from the primary variable set. The STOMP-CO2 simulator has three primary variables and the STOMP-CO2e simulator has four primary variables, with the extra primary variable being from the energy conservation equation. Because of phase appearances and disappearances, three sets of primary variables are used by STOMP-CO2 and -CO2e. A primary variable switching scheme is used to transition between primary variable sets with phase condition transitions. To ensure smooth transitions across phases, the primary variable switching occurs between Newton-Raphson iterations within a single time step.

The constitutive equations relate the primary variables with all of the secondary variables. Secondary variables include thermodynamic and transport properties and phase compositions. The collection of constitutive equations that describe the phase conditions, phase compositions, and phase densities, assuming phase equilibria, will be referred to as the equation of state. The primary variables and the equation-of-state secondary variables are the foundation for computing the remaining thermodynamic and transport secondary variables. This document provides a brief description of the calculation procedure used in STOMP-CO2 and -CO2e for each secondary variable. Components migrate across the computational domain from grid cell to grid cell across surface interfaces by advective, dispersive and diffusive fluxes. Energy additionally is transported by conduction. The transport equations that describe the flow of phases, components and energy through the subsurface are additionally described in this document for the STOMP-CO2 and -CO2e simulators.

STOMP-CO2 and -CO2e numerical simulators are computer codes that read input files, execute calculations and write output files. The input files are used to control the simulation time stepping and algorithm options, assign property data, define material dependent constitutive equations, set initial conditions, prescribe drivers (e.g., boundary conditions, sources, wells), and request output. The input files read by these simulators can vary from a single short text file to an array of input files. Developing meaningful simulation results for a geologic sequestration problem requires both conceptualizing the problem correctly and correctly converting that conceptualization into input files. This document provides input formatting instructions for the core capabilities in STOMP-CO2 and -CO2e. The formatting instructions for the ECKEChem geochemistry module and EPRMech geomechanics module are described in separate documents (White and McGrail 2005, White and Fang *in prep*).

Benchmark problems are useful for demonstrating the simulation capabilities of a particular computer code especially if alternative solutions are available from other codes or analytic solutions. Whereas, comparisons against field demonstrations, pilot-scale tests, and industrial-scale injections are additionally valuable, benchmarking problems can be used to isolate particular features of the simulator. Input files for benchmark problems are an invaluable source for users to understand modeling approaches and formatting requirements. As there are often alternatives to modeling a particular benchmark problem, input files provide excellent examples of successful modeling solutions. This document includes a series of benchmark problems for the STOMP-CO2 and -CO2e simulators, including those with geochemistry and geomechanics.

KEYWORDS: carbon dioxide, geologic sequestration, multifluid flow and transport, heat transfer, numerical simulation, modeling, greenhouse gases, reactive transport, mineralization, geomechanics, geochemistry

Acknowledgments

Development, verification and documentation of the STOMP-CO2 and STOMP-CO2e simulators by the Pacific Northwest National Laboratory (PNNL) was funded by the U.S. Department of Energy (DOE) Office of Fossil Energy as part of the Zero Emissions Research and Technology (ZERT II) program.

ZERT: Dr. Lee Spangler, Associate Vice President for Research
Office of the Vice President for
Research, Creativity, and Technology Transfer
Montana State University

Nomenclature

Roman Symbols

A_m	specific reactive surface area, m^2/kg
b	molality, mol solute/kg solvent
D	diffusion coefficient, m^2/s
E_a	activation energy, J/mol
g	acceleration of gravity, m/s^2
h	head, m
$\mathbf{J}$	diffusive flux rate, kg/m^2 s
k	thermal conductivity, W/m K
k	kinetic-reaction rate constant, mol/s m^2
$\mathbf{k}$	intrinsic permeability, m^2
$\mathbf{k}_e$	effective thermal conductivity vector, W/m K
k_r	relative permeability
k_{25}	kinetic-reaction rate constant at 25°C, mol/s m^2
K	equilibrium constant
K_{eq}	chemical equilibrium constant
L	well-segment length, m
m	mass rate density, kg/s m^3
M	molecular weight, kg/kmol
n_D	diffusive porosity
P	pressure, Pa
q	volumetric flow rate, m^3/s
q	power density, W/m^3
Q	ion activity product
r	radius, m
R	Land's parameter
R	ideal gas constant, J/kg K
R_k	reaction rate, mol/s
s	saturation
s	well skin factor
$\bar{s}$	effective saturation
$\bar{\bar{s}}$	apparent saturation

t	time, s
T	temperature, C or K
$T(C)$	temperature, C
$T(K)$	temperature, K
u	internal energy, J/kg
V	volume, m^3
$\mathbf{V}$	Darcy velocity vector, m/s
$\bar{V}$	average partial molar volume, m^3/kmol
WI	well index, m^3
x	x-direction distance, m
y	y-direction distance, m
z	z-direction distance, m
$\mathbf{z}$	unit gravitation direction vector
Z	compressibility factor

Greek Symbols

β	scaling factor
Δ	differential
μ	viscosity, Pa s
ρ	density, kg/m^3
σ	interfacial tension, N/m
τ	tortuousity factor
ϕ	porosity
ϕ	fugacity coefficient
χ	mole fraction
ω	mass fraction

Subscripts

b	brine
c	critical
$CO2$	carbon dioxide
e	effective
ex	excess property

f	fracture
f	free
g	gas
gf	free gas
gl	gas-aqueous
gr	gas residual
gt	trapped gas
H2O	water
l	aqueous
lf	free aqueous
lr	aqueous residual
m	matrix
o	grid-cell equivalent
od	oven dried
p	precipitated salt
r	reduced
s	salt
sat	saturated conditions
unsat	unsaturated conditions
w	well
x	x-coordinate direction
y	y-coordinate direction
z	z-coordinate direction
zd	zero density

Superscripts

0	reference conditions
CO2	carbon dioxide
H2O	water
min	minimum
mp	Webb matching point
potential	maximum possible
salt	salt
w	well

Contents

Figures

Tables

1.0 Introduction

Numerical simulation comprises two elements: a computer code that implements the mathematics of our scientific understanding, and a user that implements the computer code as an analytical tool. Successful numerical simulation requires both a proper implementation of the mathematics by the code developer and a proper implementation of the computer code by the code user. Successful numerical simulations provide a quantitative understanding of complex processes. Unsuccessful numerical simulation can lead us to a false understanding of a system. This guide is intended to provide the reader with an understanding of how the mathematics of geologic sequestration has been implemented in the STOMP-CO2 and -CO2e simulators. Equally importantly this guide is intended to provide the STOMP-CO2 and -CO2e users with an understanding of how to apply the code to geologic sequestration problems. Scientific software, such as the STOMP-CO2 and -CO2e simulators, are generally works in progress. Scientific discoveries often lead to new conceptual models, which become implemented in the simulator as new alogrithms. Advances in numerical schemes and computer architectures also spur changes in scientific software. As such this document is really a snapshot in time of the state of the STOMP-CO2 and -CO2e simulators.

1.1.1.1 Background

The Subsurface Transport Over Multiple Phases (STOMP) simulator (White and Freedman, 2007; Nichols and White, 2007a, 2007b; Freedman and White, 2007; White and Oostrom, 2006; White and McGrail, 2005; White and Oostrom, 2000), is PNNL's analytical tool for investigating coupled processes involving multifluid flow, heat transport, geochemistry, and geomechanics in the subsurface. The simulator was initially developed to assess nuclear waste repository performance, but greatly expanded its application domains over its nearly two-decade development life. The simulator is now being applied to support laboratory and field investigations: geologic nuclear waste repositories; radionuclide transport; unsaturated zone hydrology; reactive barriers; nuclear waste tank thermal histories; surface barriers; freeze walls; soil desiccation; soil vapor extraction; volatile organic fate and migration; dense nonaqueous phase migration and natural attenuation; geologic sequestration of greenhouse gases; coupled reactive transport; oil shale production; and natural gas hydrate production. STOMP's capabilities and numerical schemes continue to be developed with the current emphasis being gas hydrate production via CO_2 injection, oil shale production, and coupled geomechanical modeling. The STOMP development team at PNNL currently has concurrent development activities for co-sequestration of greenhouse gases, ternary hydrate systems, coupled geomechanical modeling, and the implementation of vertical equilibrium capabilities for geologic sequestration modeling. An associated concurrent project is investigating the scalability of the parallel implementation of the simulator (eSTOMP) on the latest generation of supercomputers. Whereas STOMP is routinely used for environmental assessments at U.S. DOE sites, its code structure and active development team allows it to be extended to new research and application domains.

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 the 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 that 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.

1.2 Algorithm Design

The overall flow path for the STOMP-CO2 and -CO2e simulators comprises three principal components: 1) initialization, 2) time stepping, and 3) closure. The flow chart for these components is shown in Figure 1.1. During initialization the input file is read twice. The first input file read is used to allocate and initialize memory for all the global variables (i.e., those variables passed among routines via modules). The second input file read is used to define the problem. The input file readers are not foolproof but do report error messages when input errors are noted. Once the input file has been successfully read twice, the initial states are checked for thermodynamic consistency, phase conditions are established, and the primary variables are selected for every grid cell. The next stage of initialization involves computing all the secondary variables at grid cells and boundary surfaces. If geomechanics calculations are included in the simulation, then the initial stress state of the domain is established. If coupled wells are included in the simulation, then the well trajectories are determined and the wells are equilibrated with the formation. Once the well trajectories are defined, the coupled conservation equations and coupled well equations are sequenced such that the band width of the Jacobian matrix is minimized. If geochemistry calculations are included in the simulation, then the initial chemical state of the system is established. With the initial primary and secondary variables computed, the next stage of initialization is to compute the initial surface fluxes at both internal surfaces and boundary surfaces (e.g., volumetric phase flux, component advective/diffusive flux, heat flux). The final initialization stage is to record user requested output to the various output files (i.e., output, plot.xxx, and surface). If a zero-time-step simulation is requested, the simulator stops at the end of the intialization, after recording output. This option is very useful for computing property values or examining initial states.

The time stepping component is where the simulator loops over time, with each time-step loop solving for the conditions at a new point in time. Geomechanics, reactive transport (geochemistry) and solute transport are solved sequentially with the coupled multifluid flow and heat transport. Each new time step begins with the assignment of a time step, which is determined from the previous time step parameters (e.g., time step and convergence) and user input (e.g., maximum time step, minimum time step, time-step acceleration factor, output times, boundary condition times, source times, coupled well times). The next step is to store the old time step values of the primary and secondary variables. Old-time-step values of fluxes are not stored. Next, boundary conditions and sources at the new time step are assigned. Fluxes across the interior and boundary surfaces are then computed. The Jacobian matrix, solution vector, and problem vector are all set to zero. The Jacobian matrix and problem vector are then loaded for all active nodes and interior surfaces (i.e., non-boundary surfaces). The Jacobian matrix and problem vector is then corrected for boundary surface fluxes. At this point the Jacobian matrix and problem vector are complete and the linear system is either solved directly with a banded solver or iteratively. The resulting solution vector from the linear system solve is corrections (updates) to the primary variables. The primary variables are then updated and residuals (errors) in the conservation equations are determined. The updated primary variables are then used to establish new phase conditions and a new primary variable set for each grid cell. Increments to the primary variables for the numerical derivatives are determined, and the secondary variables are computed using native and incremented primary variables. Convergence is then established using the conservation equation residuals; where the highest relative residual across the computational domain determines convergence.

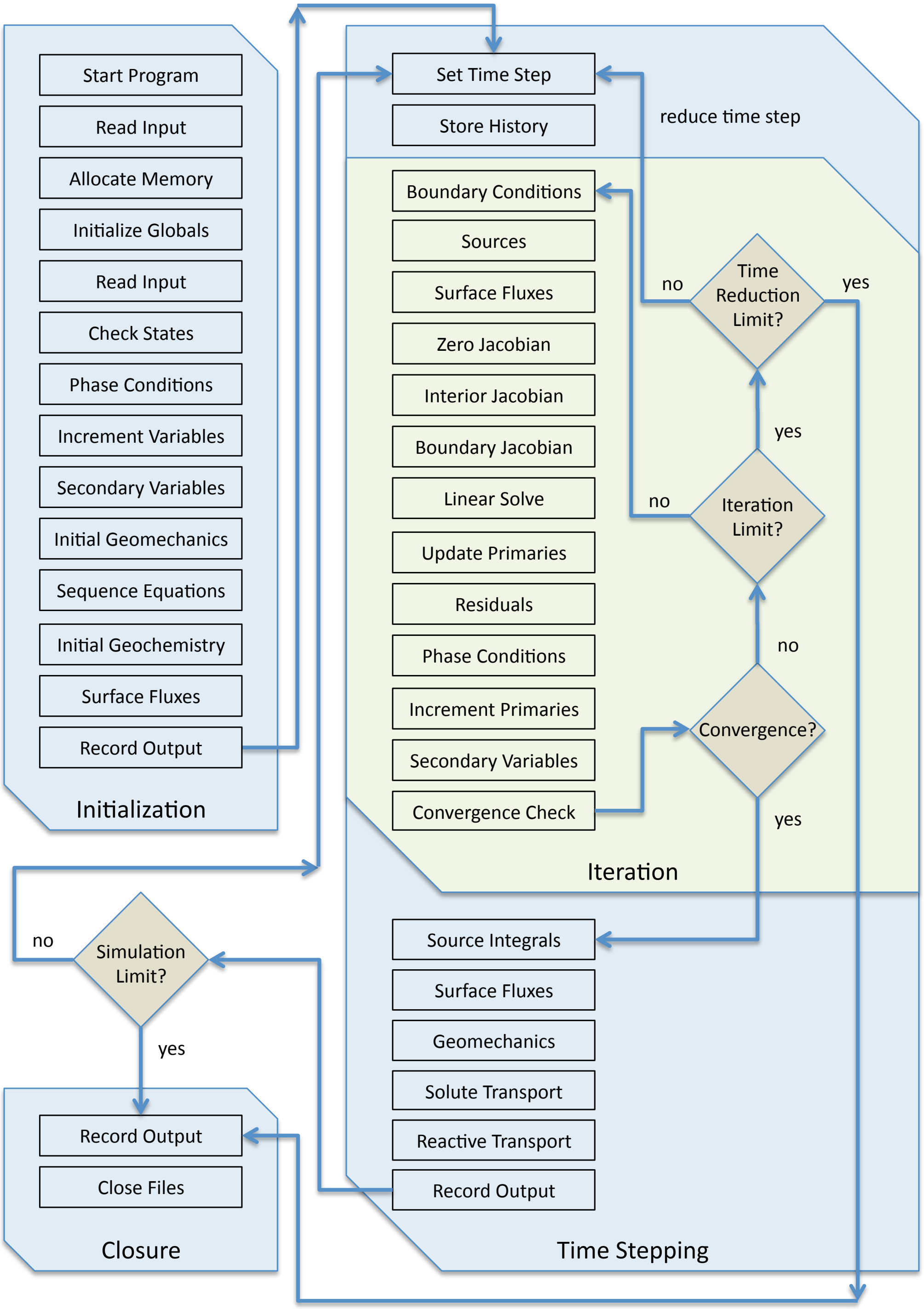

Figure 1.1. STOMP-CO2 and -CO2e Algorithmic Flow Chart

If convergence has occurred for all conservation equations and for all grid cells, then the simulation proceeds to the geomechanics, solute transport and reactive transport routines. Prior to calling these routines integrals for sources are computed and surface fluxes are updated using the converged values of the primary and secondary variables. Before starting a new time step, output is recorded and the simulation limits are checked. If the simulation time or number of time steps has reached a user defined limit then the simulation moves to closure at which point output is recorded, all simulation files are closed, and the program stops. If convergence has not occurred, then the iteration limit is checked. If the number of Newton-Raphson iterations exceeds the user-defined limit, then the time step is reduced and the time step is repeated at a reduced time step amount. If the number of Newton-Raphson iterations is less than the user-defined limit then another iteration loop occurs, starting with a calculation of the boundary conditions. If the iteration limit is exceeded, then the new reduced time step is checked against the minimum time step or the number of allowed consecutive time-step reductions is checked. If either of these checks fail, then the simulation transfers to closure, resulting in recording of the output and cessation of the simulation.

1.3 Sequential and Scalable Implementations

There are two implementations of STOMP-CO2 and -CO2e; the serial version and the scalable version. The scalable versions are referred to as eSTOMP-CO2 and eSTOMP-CO2e; where the proceeding "e" indicates an extensible code. Extensible or scalable refers to the ability of the simulator to execute on multiple processor computers. The serial implementation of the simulators is written in Fortran 90. The direct banded linear system solver is fully integrated into the simulator, but the iterative linear system solver requires linking the simulator with the SPLIB package (Bramley and Wang, 1995), developed at Indiana University. The scalable implementations are also written in Fortran 90, but require linkage to three external packages: 1) Global Arrays (GA) (Nieplocha et al, 2006), 2) MPI (Gropp et al., 1999), and 3) PETSc (Balay et al., 2008). A central concept for the scalable implementation is that the computational domain is partitioned over multiple computer processors, resulting in reduced computational time. For simulation problems with large domains (e.g., >100k grid cells), reductions in computational time can be realized using the scalable implementations on massively parallel computers. The sequential implementations of the simulator were used as the basis for developing their scalable counterparts, using the Global Array (GA) toolkit. The GA toolkit supports a one-sided, shared memory style programming model on both shared and distributed memory platforms. One-sided communication is an alternative to the paired sends and receives in MPI (Message Passing Interface). This approach can have advantages in efficiency and flexibility in addressing load balancing and multiple component, multiple data applications. Another benefit of using GA is that it provides a global address space abstraction for distributed arrays that can dramatically simplify programming for problems involving spatial arrays of data, such as those typically used in computational fluid dynamics applications. As the banded and SPLIB linear system solvers used in the sequential implementation of the simulators are not scalable, the parallel extensible toolkit for scientific computation (PETSc) is used for the linear system solve. The guiding principals for the development of the scalable implementations of the simulators have been 1) to keep the source code of the scalable implementation accessible to the domain scientists and engineers that are familiar with the sequential implementation and 2) to maintain uniformity between the input and output files between the sequential and scalable implementations. The first guiding principal has been achieved in large part because of the GA toolkit.

The GA programming toolkit, developed at the Pacific Northwest National Laboratory, makes parallel program coding easier and provides high efficiency and portability (Nieplocha et al, 2006). The one-sided asynchronous remote communication (no processor interactions are required to access remote data) implemented in the GA model is conceptually similar to shared memory but is not limited to shared memory architectures. The GA model provides the ease of coding of shared-memory models and the portability of message-passing models, such as MPI. GA also enables programmers to take advantage of existing MPI software because of the compatibility of GA with MPI. For the case of eSTOMP-CO2 and -CO2e, this has made it possible to use GA in the grid component without interfering with the MPI calls in the PETSc solver libraries. The shared memory model of the GA toolkit provides a portable interface that maintains the global index space of distributed arrays. This allows the distributed array to be used as if the data were stored in shared memory even though the data may be physically distributed across different processors. Programmers can access data in a similar fashion to serial programming using explicit library calls, without having to know the low-level details of the communication and data distribution. Rather than writing parallel applications from scratch or significantly restructuring serial application code, GA allows developers, in many cases, to make simple modifications to serial code to develop scalable parallel code.

2.0 Mathematical Formulation

2.1 Conservation Equations

The STOMP-CO2 simulator assumes isothermal conditions and solves three coupled mass conservation equations: water mass, CO_2 mass and salt mass. The STOMP-CO2e simulator is the nonisothermal counterpart to STOMP-CO2 and solves an additional coupled conservation equation for thermal energy. The conservation equations equate the change in the conserved quantity within a volume over time with the net flux of the conserved quantity into the volume plus any net source of the conserved quantity within the volume. For the energy equation, the conserved quantity within a volume is formulated in terms of phase internal energy; the fluxes of energy are by mobile phase advection and thermal diffusion; energy flux associated with component diffusive flux is ignored; and energy sources are either associated with mass sources or heat sources:

$$\frac{\partial}{\partial t}\left[\sum_{\gamma=l,g,p}\left(\phi\,\rho_\gamma\,s_\gamma\,u_\gamma\right)\right]=-\sum_{\gamma=l,g}\nabla\left(\rho_\gamma\,h_\gamma\,\mathbf{V}_\gamma\right)+\nabla\left(\mathbf{k}_e\,\nabla T\right)+\sum_{\gamma=l,g}\left(h_\gamma\,m_\gamma\right)+q \tag{2.1}$$

Advective fluxes of the mobile phases are computed according to Darcy's law

$$\mathbf{V}_\gamma=-\frac{k_{r\gamma}\,\mathbf{k}}{\mu_\gamma}\left(\nabla P_\gamma+\rho_\gamma\,g\,\mathbf{z}\right);\;\; for\;\gamma=l,g \tag{2.2}$$

Water (H2O) is assumed to exist in the aqueous (l) and gas (g) phases under equilibrium conditions. The conservation equation for water mass considers water flux via advection and molecular diffusion through the mobile phases:

$$\frac{\partial}{\partial t}\left[\sum_{\gamma=l,g}\left(\phi\,\rho_\gamma\,s_\gamma\,\omega_\gamma^{H2O}\right)\right]=-\sum_{\gamma=l,g}\nabla\left(\rho_\gamma\,\omega_\gamma^{H2O}\,\mathbf{V}_\gamma\right)-\sum_{\gamma=l,g}\nabla\left(\mathbf{J}_\gamma^{H2O}\right)+\sum_{\gamma=l,g}\left(\omega_\gamma^{H2O}\,m_\gamma\right) \tag{2.3}$$

Diffusive fluxes of components through the mobile fractions of the aqueous and gas phases are computed from gradients in molar concentration, considering molecular diffusion, but ignoring hydraulic dispersion:

$$\mathbf{J}_\gamma^i=-\phi\,\rho_\gamma\,s_\gamma\,\frac{M^i}{M_\gamma}\left(\tau_\gamma\,D_\gamma^i\right)\nabla\,\chi_\gamma^i;\;\; for\;\gamma=l,g \tag{2.4}$$

CO_2 (CO2) is assumed to exist in the aqueous (l) and gas (g) phases under equilibrium conditions. The conservation equation for CO_2 mass considers CO_2 flux via advection and molecular diffusion through the mobile phases:

$$\frac{\partial}{\partial t}\left[\sum_{\gamma=l,g}\left(\phi\,\rho_\gamma\,s_\gamma\,\omega_\gamma^{CO2}\right)\right]=-\sum_{\gamma=l,g}\nabla\left(\rho_\gamma\,\omega_\gamma^{CO2}\,\mathbf{V}_\gamma\right)-\sum_{\gamma=l,g}\nabla\left(\mathbf{J}_\gamma^{CO2}\right)+\sum_{\gamma=l,g}\left(\omega_\gamma^{CO2}\,m_\gamma\right) \tag{2.5}$$

Salt (s) is assumed to only exist dissolved in the aqueous phase (*l*) and precipitated (*p*), as reflected in its conservation of mass equation:

$$\frac{\partial}{\partial t}\left[\phi\,\rho_l\,s_l\,\omega_l^s + \phi\,\rho_p\,s_p\right] = -\nabla\left(\rho_l\,\omega_l^s\,\mathbf{V}_l\right) - \nabla\left(\mathbf{J}_l^s\right) + \omega_l^s\,m_l \tag{2.6}$$

Each conservation equation is solved for a single unknown, referred to as the primary variable. The constitutive equations relate the primary variables to the secondary variables, as described in Sections 2.2 through 2.7. Primary variables for the conservation equations are selected such that they specify the state condition and vary depending on the phase condition. STOMP-CO2 and STOMP-CO2e use a primary variable switching scheme, which is described in Section 3.1.1.

2.2 Equations of State

The intended applications for the STOMP-CO2 and -CO2e simulators are geologic storage of CO_2 in deep saline reservoirs. For the range of temperature and pressure conditions anticipated for geologic sequestration projects, five phases are possible: 1) water-rich liquid (aqueous), 2) CO_2-rich vapor (gas), 3) CO_2-rich liquid (liquid-CO_2), 4) clathrated CO_2 (hydrate), and 5) crystalline salt (precipitated salt). The STOMP-CO2 and -CO2e simulators do not consider hydrate formation, which are generally limited to high-pressure conditions and temperatures below 15°C. If these conditions are anticipated, the STOMP-HYD and STOMP-HYD-KE simulators have capabilities for modeling hydrate formation/dissociation. The equations of state express: 1) the existence of phases given the temperature, pressure, and water, CO_2, and salt concentration, 2) the partitioning of components among existing phases, and 3) the density of the existing phases. Data tables containing constants used by STOMP-CO2 and STOMP-CO2e are found in Appendix A.

2.2.1 Water-Vapor Pressure

Water-vapor pressure in geologic media is a function of temperature and capillary pressure. For saturated to nearly saturated conditions (i.e., low capillary pressure conditions), the water-vapor pressure equals the saturated water-vapor pressure, which is computed according to the k-function (Meyer et al., 1993):

$$P_{sat}^{H2O} = P_c^{H2O}\,\exp\left[\begin{array}{l}\left(T_r^{H2O}\right)^{-1}\left(\dfrac{\sum\limits_{i-1}^{5} k_i\left(1-T_r^{H2O}\right)^i}{1+k_6\left(1-T_r^{H2O}\right)+k_7\left(1-T_r^{H2O}\right)^2}\right)\\ -\left(\dfrac{\left(1-T_r^{H2O}\right)}{k_8\left(1-T_r^{H2O}\right)^2+k_9}\right)\end{array}\right] \tag{2.7}$$

where, the *k* parameters are given in Table A.1. As the wetting fluid (i.e., aqueous) saturation decreases in a geologic media, the radius of curvature across the aqueous-gas interface decreases, resulting in a corresponding decrease in the water-vapor pressure. The Kelvin equation (Nitao 1988) relates the water-vapor partial pressure to the saturated water-vapor pressure:

$$P_g^{H2O} = P_{sat}^{H2O} \exp\left[\frac{-P_{gl}}{\rho_l\left(\frac{R\,T(K)}{M_l}\right)}\right] \tag{2.8}$$

A plot of the saturated water-vapor pressure function versus temperature is shown in Figure 2.1.

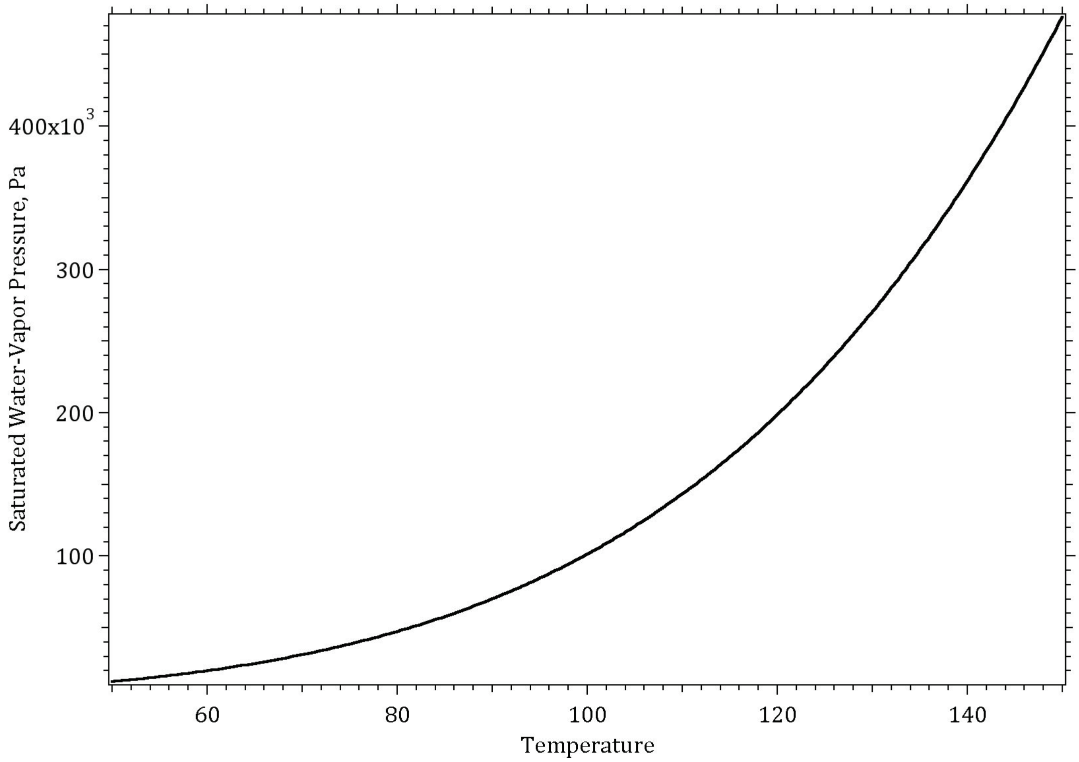

Figure 2.1. Saturated Water-Vapor Pressure versus Temperature

2.2.2 CO$_2$ Thermodynamic Property Table

Thermodynamic properties for CO$_2$ are computed via interpolation from a property data table, which is read and structured during the initialization stage of the simulation. Externally the CO$_2$ property data table is stored in a file named "co2_prop.dat." The property table was developed from the equation of state for CO$_2$ published by Span and Wagner (1996). The table includes the subcritical gas, subcritical liquid, and supercritical gas regions up to temperatures of 1100 K and pressures of 800 MPa. Internally, the table comprises 72 pressure columns with a range of 36 to 70 temperature rows per column, for a total of 4131 temperature-pressure points. The pressure and temperature spacings are nonuniform, with a higher concentration of temperature and pressures being used around the liquid-gas saturation line and the critical point. The liquid-gas saturation line is embedded in the property data table using double temperature points along the pressure column. At each temperature-pressure point, the property data table contains density, enthalpy, internal energy, fugacity, and entropy. Bilinear interpolation is used to calculate property values from the table for the supercritical gas and subcritical gas regions, with the saturation-line data points serving as upper limits in the subcritical region. The two pressure columns are located with a bisection search algorithm, and the two temperature rows for each pressure column are also located with the bisection search algorithm, limited by the liquid-gas saturation line. For the subcritical liquid region a two-staged interpolation approach is used. First, temperature interpolation at the upper pressure occurs between the two bounding temperatures, then temperature interpolation along the saturation line occurs between the two bounding temperatures, and finally, pressure interpolation occurs between the saturation pressure and upper pressure.

2.2.3 Phase Equilibria

Phase equilibria calculations are required whenever more than one phase exists in a grid cell. These calculations determine the distribution of components in each phase, assuming phase equilibria. Three phase conditions are possible in STOMP-CO2 and -CO2e: 1) aqueous saturated without entrapped gas; 2) aqueous and mobile gas; and 3) aqueous saturated with entrapped gas. In the aqueous saturated without entrapped gas phase condition, only the aqueous phase is present and the amount of dissolved CO_2 in the aqueous phase is a primary unknown. For this phase condition, phase equilibria calculations are used to determine the solubility limit for CO_2 in the aqueous phase. If the concentration of dissolved CO_2 in the aqueous phase exceeds the solubility limit, then the phase condition switches to aqueous and mobile gas. In the aqueous and mobile gas phase condition, two phases exist; the aqueous phase and the CO_2 predominate phase (gas), which can be scCO_2, subcritical liquid, or subcritical gas. Whereas a fraction of the gas phase must be mobile for this phase condition, immobile gas is additionally possible. In the aqueous saturated with entrapped gas phase two phase conditions exists; the aqueous and gas phases; where the gas phase can be scCO_2, subcritical liquid, or subcritical gas. When aqueous and gas phase both exist in a grid cell, then phase equilibria calculations are used to determine the concentrations of water and CO_2 in the two phases.

Phase equilibria calculations in STOMP-CO2 and -CO2e use the formulations of Spycher et al. (2003) for temperatures below 100°C and Spycher and Pruess (2010) for temperatures above 100°C, with corrections for dissolved salt provided in Spycher and Pruess (2010). The Spycher formulations are based on the Redlich-Kwong equation of state with parameters fitted from published experimental data for CO_2-H_2O systems. At equilibrium conditions the mole fraction of water in the gas phase and mole fraction of CO_2 in the aqueous phase are computed per Equations (2.9-2.16):

$$\chi_g^{H2O} = \frac{\left(1-B\right)}{\left(\frac{1}{A}-B\right)};\ \chi_l^{CO2} = B\left(1-\chi_g^{H2O}\right) \tag{2.9}$$

$$A = \frac{K_{H2O}^{0}}{\phi_{H2O}\,P}\exp\left[\frac{\left(P-P^{0}\right)\bar{V}_{H2O}}{R\,T(K)}\right];\ B = \frac{\phi_{CO2}\,P}{\left(10^{3}/M^{H2O}\right)K_{CO2}^{0}}\exp\left[-\frac{\left(P-P^{0}\right)\bar{V}_{CO2}}{R\,T(K)}\right] \tag{2.10}$$

$$\begin{aligned}\log\left(K_{H2O}^{0}\right) &= -2.209 + 3.097x10^{-2}\,T(C) - 1.098x10^{-4}\,T(C)^{2} + 2.048x10^{-7}\,T(C)^{3}\\ \log\left(K_{CO2(g)}^{0}\right) &= 1.189 + 1.304x10^{-2}\,T(C) - 5.446x10^{-5}\,T(C)^{2}\\ \log\left(K_{CO2(l)}^{0}\right) &= 1.169 + 1.368x10^{-2}\,T(C) - 5.380x10^{-4}\,T(C)^{2}\end{aligned} \tag{2.11}$$

$$\ln\left(\phi_{H2O}\right)=\ln\left(\frac{V}{V-b_{mix}}\right)+\left(\frac{b_{H2O}}{V-b_{mix}}\right)-\left(\frac{2\,a_{H2O-CO2}}{R\,T^{1.5}\,b_{mix}}\right)\ln\left(\frac{V+b_{mix}}{V}\right)$$
$$+\left(\frac{a_{mix}\,b_{H2O}}{R\,T^{1.5}\,b_{mix}}\right)\left[\ln\left(\frac{V+b_{mix}}{V}\right)-\left(\frac{b_{mix}}{V+b_{mix}}\right)\right]-\ln\left(\frac{P\,V}{R\,T(K)}\right)$$
$$\ln\left(\phi_{CO2}\right)=\ln\left(\frac{V}{V-b_{mix}}\right)+\left(\frac{b_{CO2}}{V-b_{mix}}\right)-\left(\frac{2\,a_{CO2}}{R\,T^{1.5}\,b_{mix}}\right)\ln\left(\frac{V+b_{mix}}{V}\right)$$
$$+\left(\frac{a_{mix}\,b_{CO2}}{R\,T^{1.5}\,b_{mix}}\right)\left[\ln\left(\frac{V+b_{mix}}{V}\right)-\left(\frac{b_{mix}}{V+b_{mix}}\right)\right]-\ln\left(\frac{P\,V}{R\,T(K)}\right) \tag{2.12}$$

$$P=\left(\frac{R\,T(K)}{V-b_{mix}}\right)-\left(\frac{a_{mix}}{T(K)^{0.5}\,V\left(V+b_{mix}\right)}\right) \tag{2.13}$$

$$\begin{aligned}
&a_{mix}=a_{CO2};\ b_{mix}=b_{CO2}\\
&a_{CO2}=7.54x10^{7}-4.13x10^{4}\ T(K),\ bar\ cm^{6}\ K^{0.5}\ mol^{-2}\\
&b_{CO2}=27.80,\ cm^{3}/mol;\ b_{H2O}=18.18,\ cm^{3}/mol\\
&a_{H2O-CO2}=7.89x10^{7},\ bar\ cm^{6}\ K^{0.5}\ mol^{-2}\\
&P^{0}=1\,bar;\ \bar{V}_{H2O}=18.1\,cm^{3}/mol;\ \bar{V}_{CO2}=32.6\,cm^{3}/mol
\end{aligned} \tag{2.14}$$

The phase molar volume of the gas phase is computed by recasting Equation (2.13) into cubic form and solving using the Nickalls scheme (Nickalls 1993):

$$V^{3}-V^{2}\left(\frac{R\,T(K)}{P}\right)-V\left(\frac{R\,T(K)\,b_{mix}}{P}-\frac{a_{mix}}{P\,T(K)^{0.5}}+b_{mix}^{2}\right)-\left(\frac{a_{mix}\,b_{mix}}{P\,T(K)^{0.5}}\right) \tag{2.15}$$

The solution to the cubic in Equation (2.15) has three roots. The molar volume of the gas form is always the largest root, and the molar volume of the liquid form is always the smallest root. The selected form depends on an evaluation of the work done in the transition from gas to liquid form along two paths and the stability criteria:

$$w_1 = P\left(V_{gas} - V_{liquid}\right)$$

$$w_2 = R\,T(K)\ln\left(\frac{V_{gas}-b_{mix}}{V_{liquid}-b_{mix}}\right) + \frac{a_{mix}}{b_{mix}\;T(K)^{0.5}}\ln\left[\frac{\left(V_{gas}+b_{mix}\right)V_{liquid}}{\left(V_{liquid}+b_{mix}\right)V_{gas}}\right] \tag{2.16}$$

$$if\ \left(w_2 - w_1\right) > 0\ then\ gas\ \ form;\ if\ \left(w_2 - w_1\right) < 0\ then\ liquid\ \ form$$

$$if\ \left(w_2 - w_1\right) = 0\ then\ both\ gas\ and\ liquid\ \ forms$$

Plots of the H_2O mole fraction in the gas phase and CO_2 mole fraction in the aqueous phase are shown in Figure 2.2 as a function of pressure at 50°C, 90°C, and 130°C.

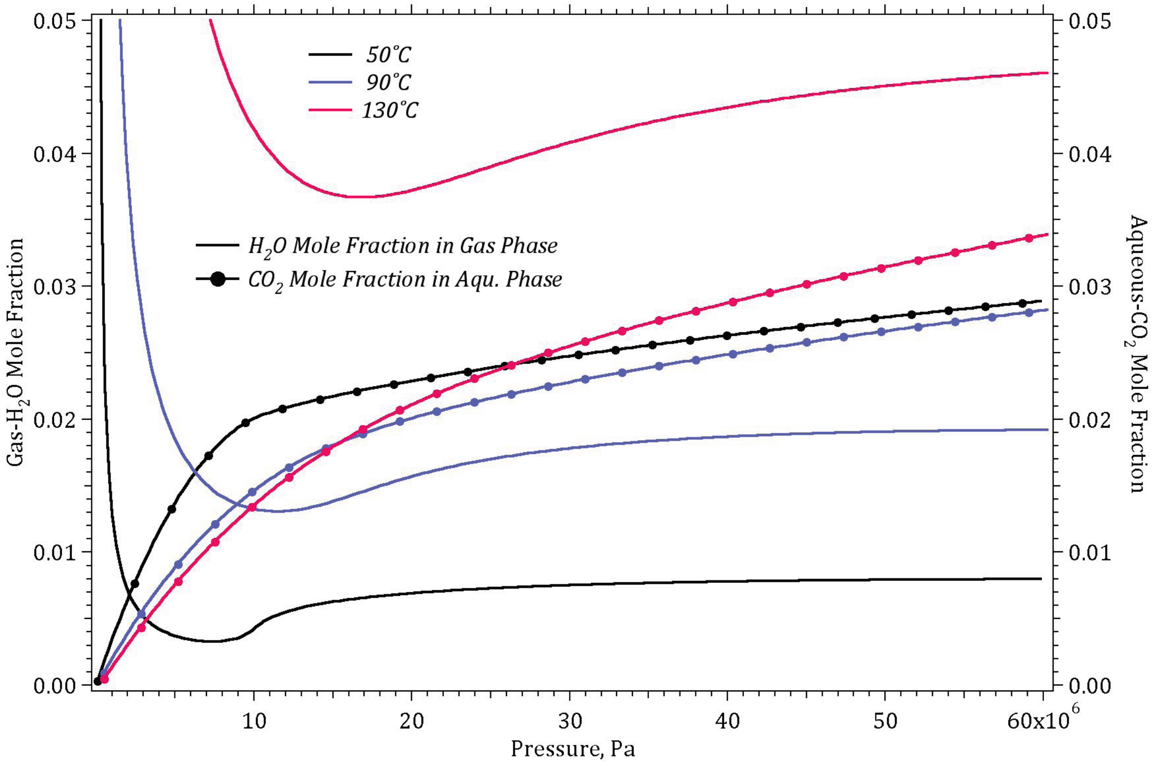

Figure 2.2. H_2O Mole Fraction in the Gas Phase and CO_2 Mole Fraction in the Aqueous Phase as a Function of Pressure at 50°C, 90°C, and 130°C

To account for a saline aqueous phase 1) the salt mole fraction in the aqueous phase is included in the equilibrium equation; and 2) the CO_2 activity coefficient is altered by the salt concentration in the aqueous phase following the model of Spycher and Pruess (2010). The result is an alteration to 1) the equilibrium expression in Equation (2.9), and 2) the *B* parameter in Equation (2.10). The model accounts for aqueous concentrations of Na, K, Ca, and Mg cations and Cl and SO_4 anions. The parameters in the model are temperature dependent:

$$
\begin{gathered}
\chi_g^{H2O} = \frac{\left(1 - B' - \chi_l^{salt}\right)}{\left(\frac{1}{A} - B'\right)}; \;\; \chi_l^{CO2} = B'\left(1 - \chi_g^{H2O}\right) \\
\chi_l^{salt} = \frac{\upsilon\, m_{salt}}{\left(10^3 / M^{H2O}\right) + \upsilon\, m_{salt} + m_{CO2}} \\
m_{CO2} = \frac{\chi_l^{CO2}\left(10^3 / M^{H2O}\right)}{\chi_l^{H2O}}; \;\; \chi_l^{H2O} = 1 - \chi_l^{CO2} - \chi_l^{salt} \\
\chi_g^{H2O} = \frac{\left(1 - B'\right)\left(10^3 / M^{H2O}\right)}{\left[\left(\frac{1}{A} - B'\right)\left(\upsilon\, m_{salt} + \left(10^3 / M^{H2O}\right)\right)\right] + \upsilon\, m_{salt}\, B'} \\
\upsilon \textit{ is the stoichiometric number of ions in the dissolved salt} \\
\left(\textit{i.e.}, 2 \textit{ for } NaCl, 3 \textit{ for } CaCl_2\right) \\
B' = \frac{\phi_{CO2}\, P}{\left(10^3 / M^{H2O}\right)\gamma'_{CO2}\, K_{CO2}^0} \exp\left[-\frac{\left(P - P^0\right)\bar{V}_{CO2}}{R\,T(K)}\right] \\
\gamma'_{CO2} = \left[1 + \frac{\sum m_{i \neq CO2}}{\left(10^3 / M^{H2O}\right)}\right] \exp\left[\begin{array}{l} 2\,\lambda\left(m_{Na} + m_K + 2\,m_{Ca} + 2\,m_{Mg}\right) + \\ \xi\, m_{Cl}\left(m_{Na} + m_K + m_{Ca} + m_{Mg}\right) - 0.07\, m_{SO4} \end{array}\right] \\
\lambda = 2.217x10^{-4}\, T(K) + \frac{1.074}{T(K)} + \frac{2648}{\left(T(K)\right)^2} \\
\xi = 1.30x10^{-5}\, T(K) - \frac{20.12}{T(K)} + \frac{5259}{\left(T(K)\right)^2}
\end{gathered}
\tag{2.17}
$$

Plots of the H_2O mole fraction in the gas phase and CO_2 mole fraction in the aqueous phase are shown in Figure 2.3 for an aqueous salt mass fraction of 0.1 as a function of pressure at 50°C, 90°C, and 130°C. Adding salt to the mixture results in lower amounts of CO_2 in the aqueous phase (i.e., salting out effect), as can be seen by comparing Figures 2.2 and 2.3.

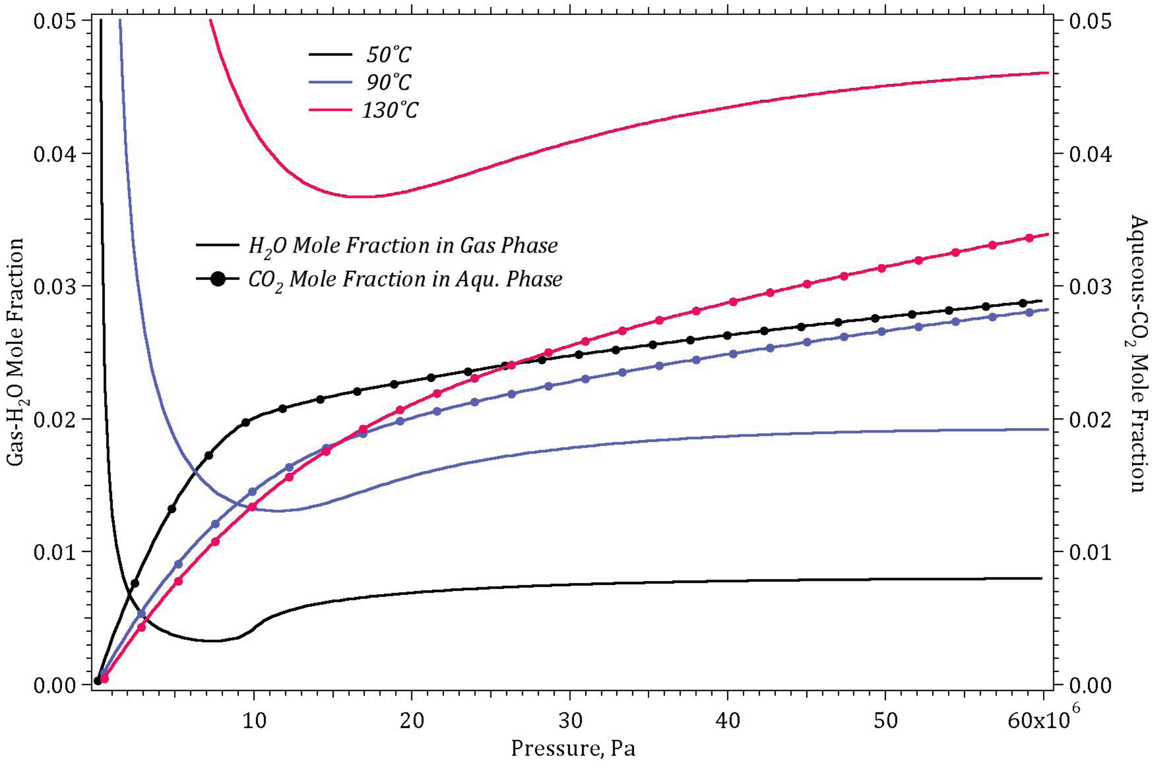

Figure 2.3. H_2O Mole Fraction in the Gas Phase and CO_2 Mole Fraction in the Aqueous Phase as a Function of Pressure at 50°C, 90°C, and 130°C for an Aqueous Salt Mass Fraction of 0.1

2.2.4 Aqueous Density

Aqueous density is computed from a series of calculations starting with the pure-water density, computed as a function of temperature and pressure using the ASME steam table formulations (Meyer et al. 1993):

$$
\begin{aligned}
\rho_l^{H2O} = M^{H2O} \Bigg[& V_c^{H2O} \Big\{ A_{12}\, a_5\, Z^{-5/17} + A_{13} + A_{14}\, T_r^{H2O} + A_{15} \left(T_r^{H2O}\right)^2 \\
& + A_{16} \left(a_6 - T_r^{H2O}\right)^{10} + A_{17} \left(a_7 + \left(T_r^{H2O}\right)^{19} \right)^{-1} \\
& - \left(A_{18} + 2\, A_{19}\, P_r^{H2O} + 3\, A_{20} \left(P_r^{H2O}\right)^2 \right) \left(a_8 + \left(T_r^{H2O}\right)^{11} \right)^{-1} \\
& - A_{21} \left(T_r^{H2O}\right)^{18} \left(a_9 + \left(T_r^{H2O}\right)^2 \right) \left(-3 \left(a_{10} + P_r^{H2O} \right)^{-4} + a_{11} \right) \\
& + 3\, A_{22} \left(a_{12} - T_r^{H2O} \right) \left(P_r^{H2O}\right)^2 + 4\, A_{23} \left(P_r^{H2O}\right)^3 \left(T_r^{H2O}\right)^{-20} \Big\} \Bigg]^{-1} \\
Z = {} & Y + \left(a_3\, Y^2 - 2\, a_4 T_r^{H2O} + 2\, a_5\, P_r^{H2O} \right)^{1/2} \\
Y = {} & 1 - a_1 \left(T_r^{H2O}\right)^2 - a_2 \left(T_r^{H2O}\right)^{-6} \\
T_r^{H2O} = {} & \frac{T(K)}{T_c^{H2O}(K)} ;\; P_r^{H2O} = \frac{P}{P_c^{H2O}}
\end{aligned}
\tag{2.18}
$$

Plots of the pure-water liquid density versus pressure at 50, 100, and 150°C are shown in Figure 2.4.

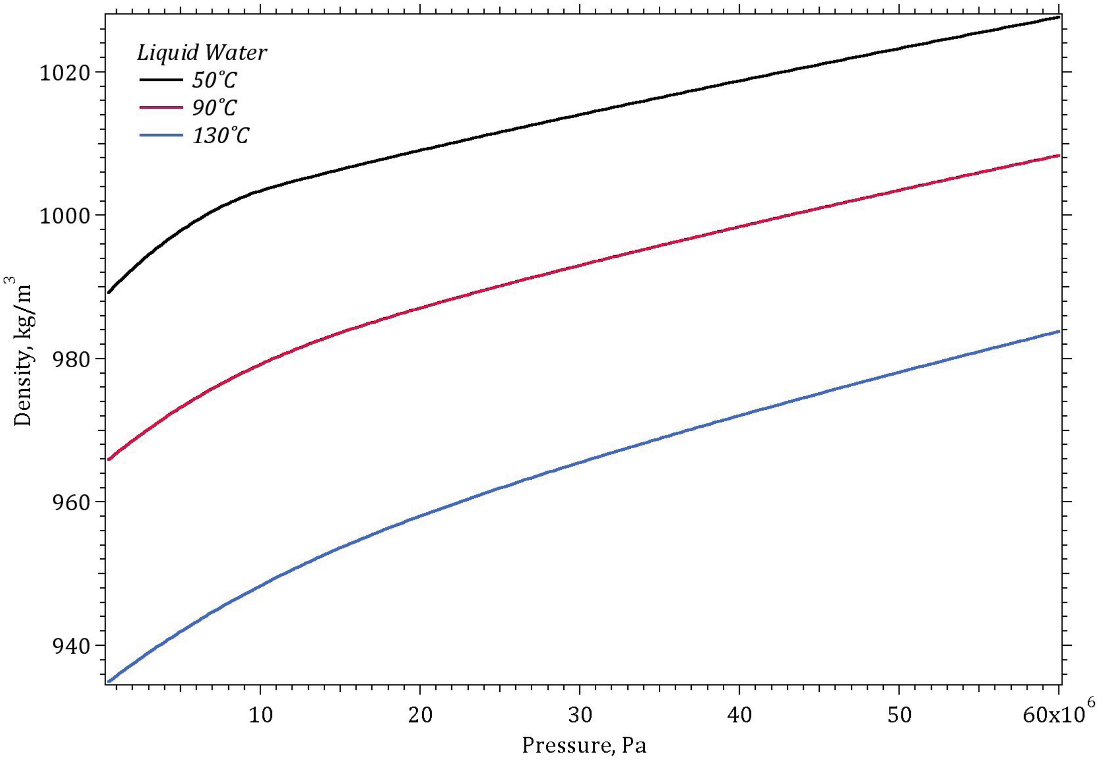

Figure 2.4. Pure-Water Liquid Density as a Function of Pressure at 50, 100, and 150°C

The pure-water density is then corrected for salt concentration using the Haas [1976] formulation to compute a brine density:

$$b_l^{salt} = \frac{10^3\,\omega_l^{salt}}{M^{salt}\left(1-\omega_l^{salt}\right)};\; V_l^{H2O} = \left(10^{-3}\,\rho_l^{H2O}\right)^{-1}$$

$$\phi = c_1 + c_2\,V_l^{H2O} + c_3\left(V_l^{H2O}\right)^2 + \left(c_4 + c_5\,V_l^{H2O}\right)\left(\frac{V_l^{H2O}}{V_c^{H2O} - V_l^{H2O}}\right)^2\left(b_l^{salt}\right)^{0.5} \tag{2.19}$$

$$\rho_b = \left(10^3 + b_l^{salt}\,M^{salt}\right)\left(10^3\,V_l^{H2O} + \phi\,b_l^{salt}\right)$$

Brine density versus salt mass fraction in the aqueous phase is shown in Figure 2.5 for temperatures of 50, 100, and 150°C at 10 MPa.

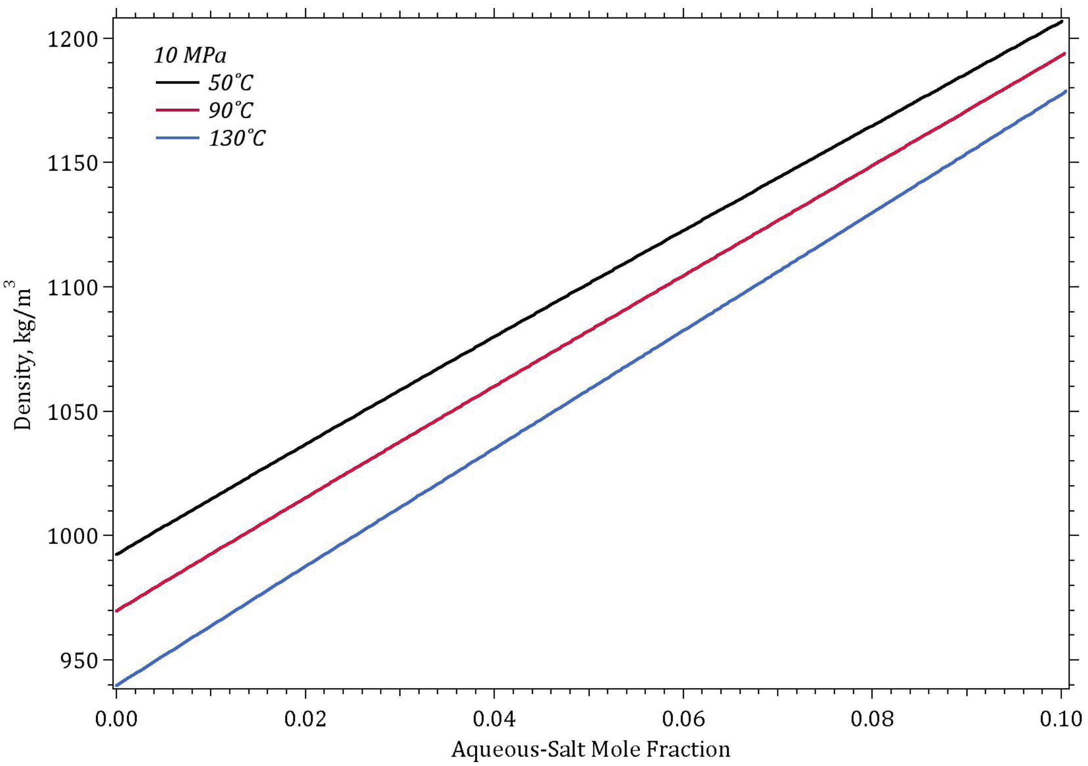

Figure 2.5. Brine Density versus Salt Mass Fraction in the Aqueous Phase for Temperatures of 50, 100, and 150°C at 10 MPa

The aqueous density is computed from the brine density, dissolved CO_2 mass fraction, and temperature using the formulation of Anderson (1992) and the Poynting correction factor (Prausnitz et al., 1986):

$$\begin{aligned}
\rho_l &= \frac{\rho_b}{\left(1 + c_l^{CO2} - \omega_l^{CO2}\right)} \\
c_l^{CO2} &= \frac{\upsilon_l^{CO2}\, \rho_b\, \omega_l^{CO2}}{M^{CO2}} \\
\upsilon_l^{CO2} &= d_1 + d_2\, T(C) + d_3 \left(T(C)\right)^2 + d_4 \left(T(C)\right)^3 + d_5 \left(T(C)\right)^4
\end{aligned} \qquad (2.20)$$

Plots of aqueous density under equilibrium conditions with CO_2 as a function of pressure, per Figure 2.2, at temperatures of 50°, 90°, and 130°C are shown in Figure 2.6. Adding salt to the mixture increases the aqueous density, as shown in Figure 2.6, for equilibrium conditions with CO_2 as a function of pressure, per Figure 2.3, at temperatures of 50°, 90°, and 130°C, with an aqueous-salt mass fraction of 0.1.

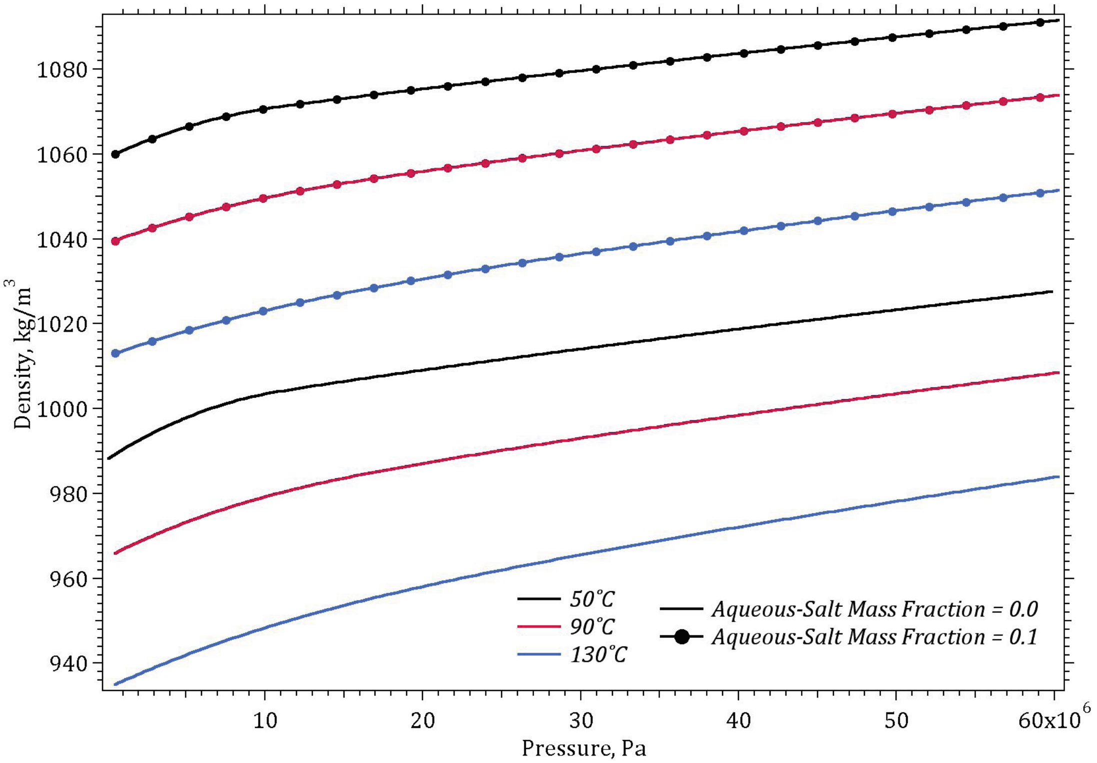

Figure 2.6. Aqueous Density as a Function of Pressure at 50, 90, and 130˚C for Aqueous Salt Mass Fractions of 0.0 and 0.1

2.2.5 Gas Density

Gas phase density is computed from the pure CO2 and pure water vapor densities using mass fraction weighting, where the gas mass fractions are computed from the phase equilibria (Section 2.2.3):

$$\rho_g = \omega_g^{CO2}\,\rho_g^{CO2} + \omega_g^{H2O}\,\rho_g^{H2O} \tag{2.21}$$

The pure CO2 density is computed as a function of temperature and CO2 partial pressure from a tabular interpolation of the Span and Wagner (1996) equation of state for CO2 (Section 2.2.2). Plots of pure CO2 density as a function of temperature and pressure over the temperature range from 0˚ to 130˚C and pressure range from 10 to 100 bar are shown in Figure 2.7; where the possible phase regions are indicated for each pressure.

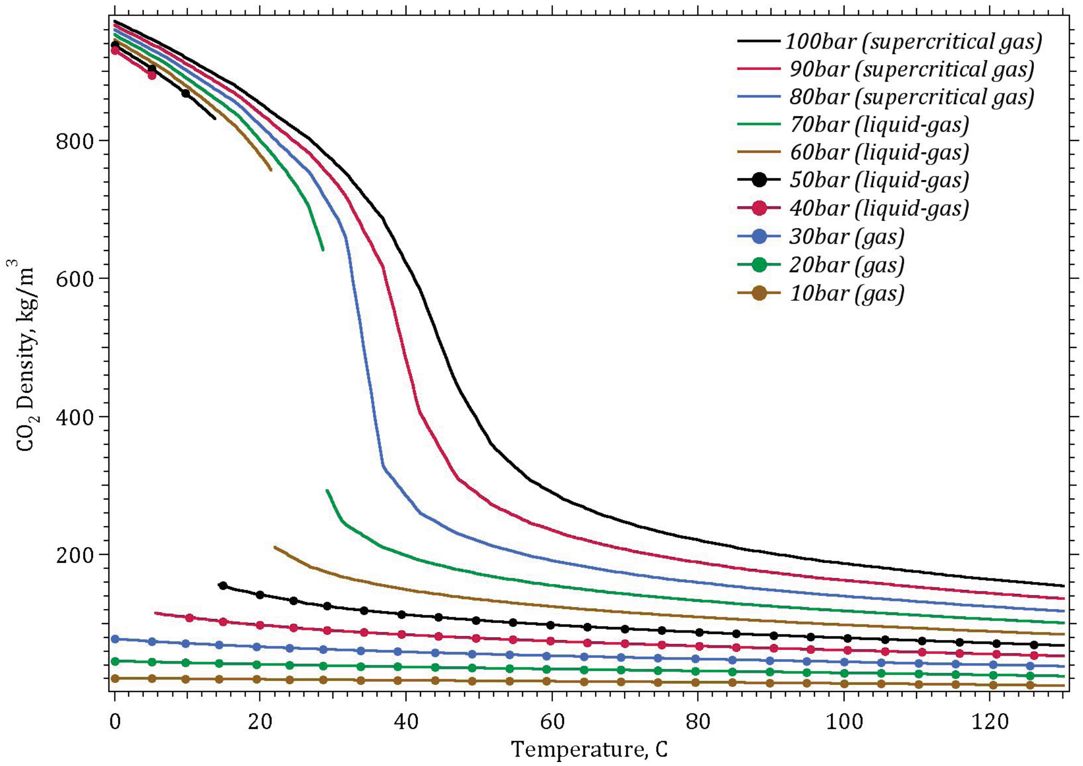

Figure 2.7. CO_2 Density as a Function of Temperature and Pressure

The pure water-vapor density is computed as a function of temperature and water-vapor partial pressure using the ASME steam table formulations (Meyer et al. 1993):

$$\rho_g^{H2O} = M^{H2O}\left[V_c^{H2O}\left\{\left(\frac{l_1\, T_r^{H2O}}{P_r^{H2O}}\right) - \sum_{i=1}^{5}\left(i\left(P_r^{H2O}\right)^{(i-1)} \sum_{j=1}^{n_i} B_{i,j}\, X^{z_{i,j}} \right) \right.\right.$$

$$\left.\left. - \sum_{i=6}^{8}\left\{ \frac{(i-2)\left(P_r^{H2O}\right)^{(1-i)} \sum_{j=1}^{n_i} B_{i,j}\, X^{z_{i,j}}}{\left(P_r^{H2O}\right)^{(2-i)} \sum_{k=1}^{l_i} b_{i,k}\, X^{x_{i,j}}} \right\} + 11\left(\frac{P_r^{H2O}}{\beta_L}\right)^{10} \sum_{i=0}^{6} B_{9,j}\, X^{i} \right\}\right]^{-1}$$

$$n_j = [2,3,2,2,3,2,2,2];\ l_j = [0,0,0,0,0,1,1,2]$$

$$z_{j,1} = [13,18,18,25,32,12,24,24];\ z_{j,2} = [3,2,10,14,28,11,18,14] \tag{2.22}$$

$$z_{j,3} = [0,1,0,0,24,0,0,0]$$

$$x_{j,1} = [0,0,0,0,0,14,19,54];\ x_{j,2} = [0,0,0,0,0,0,0,27]$$

$$X = \exp\left[b\left(1 - T_r^{H2O}\right)\right]$$

$$\beta_L = L_0 + L_1\, T_r^{H2O} + L_2\left(T_r^{H2O}\right)^2$$

$$T_r^{H2O} = \frac{T(K)}{T_c^{H2O}(K)};\ P_r^{H2O} = \frac{P}{P_c^{H2O}}$$

A plot of the water-vapor density as a function of temperature for saturated conditions is shown in Figure 2.8. The increase in water-vapor density with temperature is due to corresponding increase in water-vapor pressure with temperature. For a gas-aqueous system at constant pressure, the overall gas density drops with increasing temperature due to the relatively low solubility of water in the gas phase and the decrease in gas phase density with temperature, as also shown in Figure 2.8.

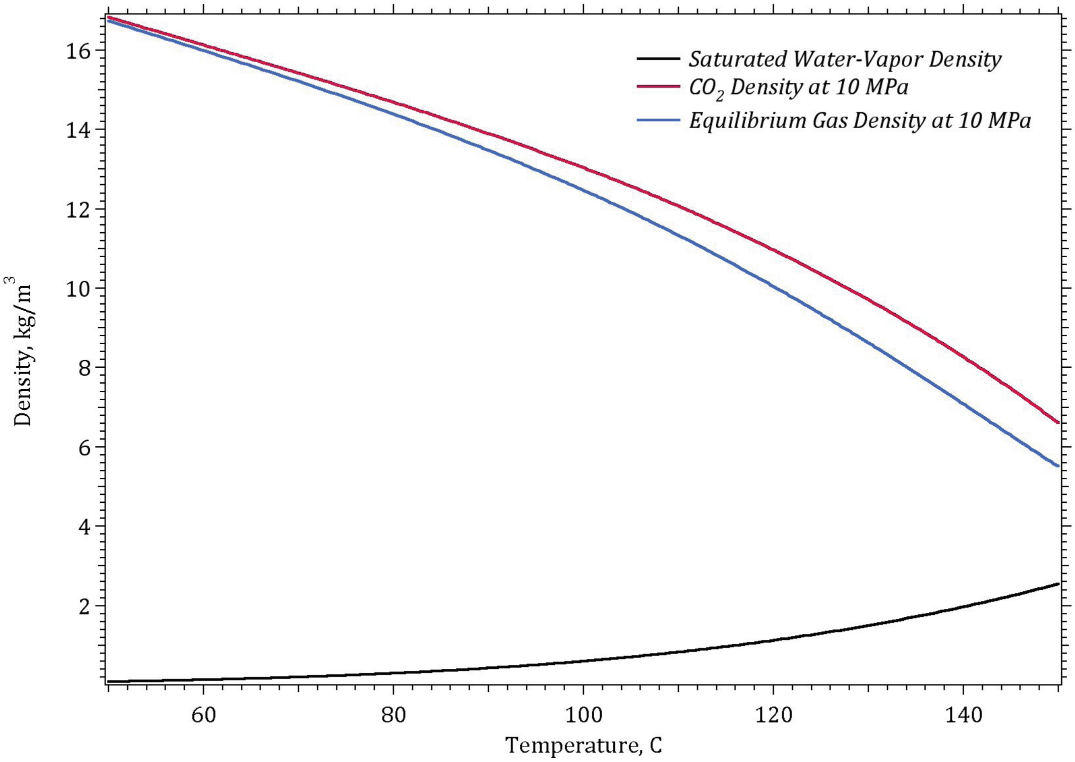

Figure 2.8. Water-Vapor and Gas Density as a Function of Temperature at a Constant Pressure of 10 MPa

2.2.6 Precipitated Salt Density

Precipitated salt density is computed as a function of temperature and pressure from the formulation of Battistelli et al. (1997):

$$\rho_s = 2165.0 \exp\left(-1.2e-4\,T\left(C\right) + 4.e-11\,P\left(Pa\right)\right) \tag{2.23}$$

2.3 Thermodynamic Properties

The thermodynamic properties considered here are the state variables of enthalpy and internal energy. These variables are only computed in the STOMP-CO2e simulator, which additionally solves the conservation of energy equation.

2.3.1 Aqueous Enthalpy and Internal Energy

The aqueous enthalpy is computed from the temperature, brine enthalpy, pure CO_2 enthalpy, the heat of CO_2 solution, the aqueous component concentrations using the formulation of Battistelli et al. (1997), where the CO_2 heat of solution is computed with the Himmelblau formulation and the Henry's coefficient for CO_2

solubility in brine is computed using the formulation of Battistelli et al. (1997) as a function of temperature and salt molality:

$$
\begin{aligned}
&h_l = \omega_l^{H2O}\, h_b + \omega_l^{CO2}\left(h_g^{CO2} + h_{sol}^{CO2}\right)\\
&h_{sol}^{CO2} = \frac{-R\left(T\left(K\right)\right)^2 \dfrac{\partial\, H_b^{CO2}}{\partial\, T}}{M^{CO2}}\\
&H_b^{CO2} = Y\left(10^{\left(b_b^s\, Z\right)}\right);\ Y = \sum_{i=0}^{5} b_i \left(T\left(C\right)\right)^i;\ Z = \sum_{i=0}^{4} c_i \left(T\left(C\right)\right)^i
\end{aligned}
\tag{2.24}
$$

where the *b* and *c* parameters are given in Table A.2. The aqueous phase is nearly incompressible; therefore, the aqueous internal energy is assumed to equal the aqueous enthalpy. The pure CO_2 enthalpy is computed as a function of temperature and CO_2 partial pressure from a tabular interpolation of the Span and Wagner (1996) equation of state for CO_2 (Section 2.2.2). Brine enthalpy and internal energy are computed from the pure-water liquid enthalpy, salt enthalpy, and enthalpy of salt solution using the formulation of Gudmundsson and Thrainsson (1989) as a function of temperature and salt concentration.

$$
\begin{aligned}
&h_b = \omega_b^{H2O}\, h_l^{H2O} + \omega_b^s\, h_p^s + b_b^s\, h_{sol}^s\\
&h_{sol}^s = \left(\frac{4184.0}{1000.0 + M^s\, b_b^s}\right)\sum_{i=0}^{3}\sum_{j=0}^{2} a_{i,j}\left(T\left(C\right)\right)^i \left(b_b^s\right)^j\\
&h_p^s = \left(\frac{4184.0}{M^s}\right)\sum_{i=1}^{3} b_i\,\left(T\left(C\right)\right)^i
\end{aligned}
\tag{2.25}
$$

where the *a* and *b* parameters are given in Table A.3. The pure-water liquid enthalpy is computed from the ASME steam table formulations (Meyer et al. 1993), where the reference state is defined as the pure-water liquid internal energy at 0.01˚C:

$$
\begin{aligned}
h_l^{H2O} &= \left(\frac{P_c^{H2O}}{\rho_c^{H2O}} \right) A_0\, T_r^{H2O} - \sum_{i=1}^{10} \left(i-2 \right) A_i \left(T_r^{H2O} \right)^{\left(i-1\right)} \\
&+ A_{11} \left[Z \left\{ 17 \left(\frac{Z}{29} - \frac{Y}{12} \right) + 5\, T_r^{H2O} \left(\frac{Y'}{12} \right) \right\} + a_4\, T_r^{H2O} - \left(a_3 - 1 \right) T_r^{H2O}\, Y\, Y' \right] Z^{\left(-5/17\right)} \\
&+ \left\{ \begin{array}{l} A_{12} - A_{14} \left(T_r^{H2O} \right)^2 + A_{15} \left(9\, T_r^{H2O} + a_6 \right) \left(a_6 - T_r^{H2O} \right)^9 \\ + A_{16} \left(20 \left(T_r^{H2O} \right)^{19} + a_7 \right) \left(\left(T_r^{H2O} \right)^{19} + a_7 \right)^{-2} \end{array} \right\} P_r^{H2O} \\
&- \left(12 \left(T_r^{H2O} \right)^{11} + a_8 \right) \left(\left(T_r^{H2O} \right)^{11} + a_8 \right)^{-2} \left(A_{17}\, P_r^{H2O} + A_{18} \left(P_r^{H2O} \right)^2 + A_{19} \left(P_r^{H2O} \right)^3 \right) \\
&+ A_{20} \left(T_r^{H2O} \right)^{18} \left(17\, a_9 + 19 \left(T_r^{H2O} \right)^2 \right) \left\{ \left(a_{10} + P_r^{H2O} \right)^3 + a_{11}\, P_r^{H2O} \right\} \\
&+ A_{21}\, a_{12} \left(P_r^{H2O} \right)^3 + 21\, A_{22} \left(T_r^{H2O} \right)^{-20} \left(P_r^{H2O} \right)^4 \\
Z &= Y + \left(a_3\, Y^2 - 2\, a_4\, T_r^{H2O} + 2\, a_5\, P_r^{H2O} \right)^{\left(1/2\right)} \\
Y &= 1 - a_1 \left(T_r^{H2O} \right)^2 - a_2 \left(T_r^{H2O} \right)^{-6} \\
Y' &= -2\, a_1\, T_r^{H2O} + 6\, a_2 \left(T_r^{H2O} \right)^{-7} \\
T_r^{H2O} &= \frac{T(K)}{T_c^{H2O}(K)};\; P_r^{H2O} = \frac{P(Pa)}{P_c^{H2O}(Pa)}
\end{aligned}
\qquad (2.26)
$$

where the *a* and *A* parameters are given in Table A.4.

2.3.2 Gas Enthalpy and Internal Energy

Gas enthalpy is computed from the pure CO_2 and pure water vapor enthalpies using mass fraction weighting, where the gas mass fractions are computed from the phase equilibria (Section 2.2.3):

$$
h_g = \omega_g^{CO2}\, h_g^{CO2} + \omega_g^{H2O}\, h_g^{H2O} \qquad (2.27)
$$

The pure CO_2 density is computed as a function of temperature and CO_2 partial pressure from a tabular interpolation of the Span and Wagner (1996) equation of state for CO_2 (Section 2.2.2). The reference point for pure CO_2 enthalpy is the ideal gas state at 298.15 K (25°C) and 0.101325 MPa (1 atm). Plots of pure CO_2 enthalpy as a function of temperature and pressure over the temperature range from 0° to 130°C and pressure range from 10 to 100 bar are shown in Figure 2.9, where the possible phase regions are indicated for each pressure.

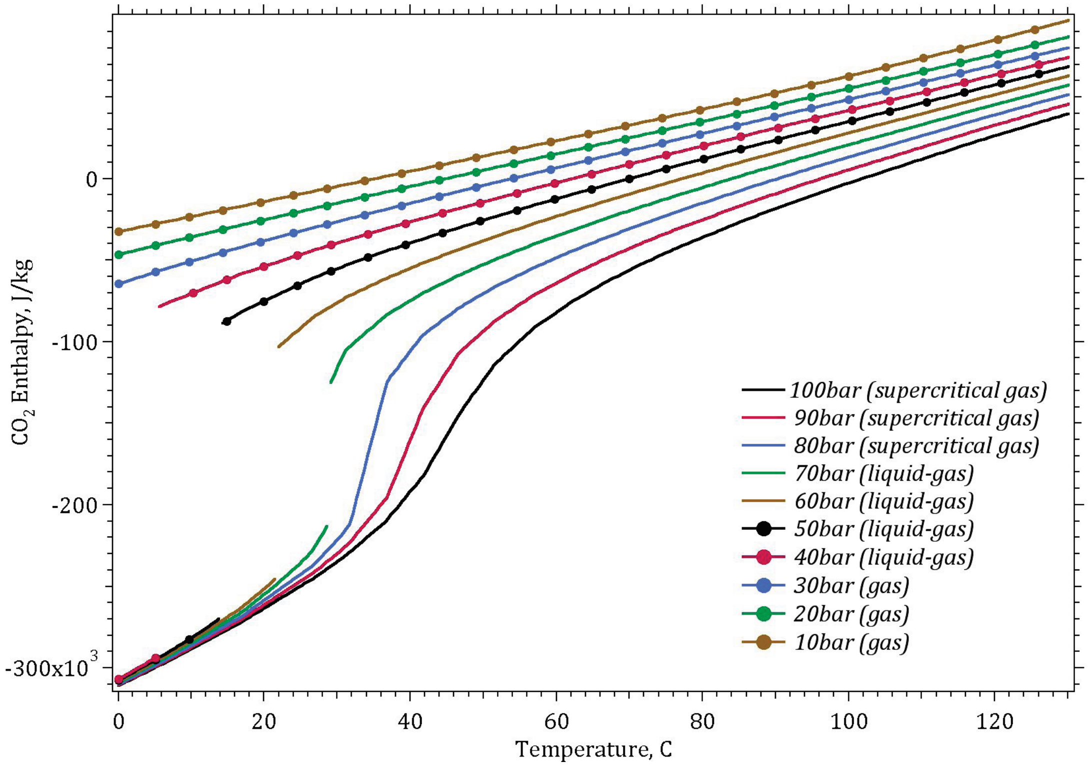

Figure 2.9. CO_2 Enthalpy as a Function of Temperature and Pressure

Pure water vapor enthalpy is computed from the ASME steam table formulations (Meyer et al. 1993), where the reference state is defined the pure-water liquid internal energy at 0.01˚C:

$$h_g^{H2O} = \left(\frac{P_C^{H2O}}{\rho_C^{H2O}} \right) \left\{ \begin{aligned} & B_O\, T_R^{H2O} - \sum_{i=1}^{5} B_{0,i} \left(i-2 \right) \left(T_R^{H2O} \right)^{\left(i-1 \right)} \\ & - \sum_{j=1}^{5} \left(P_R^{H2O} \right)^j \sum_{i=1}^{n_j} B_{j,i} \left(1 + z_{j,i}\, b\, T_R^{H2O} \right) X^{\left(z_{j,i} \right)} \\ & - \sum_{j=6}^{8} \left[\frac{ \displaystyle\sum_{m=1}^{n_j} B_{j,i}\, X^{\left(z_{j,i} \right)} \left(\begin{aligned} & \left[1 + z_{j,i}\, b\, T_R^{H2O} \right] \\ & - \frac{ b\, T_R^{H2O} \displaystyle\sum_{k=1}^{l_j} x_{j,k}\, b_{j,k}\, X^{\left(x_{j,k} \right)} }{ \left(T_R^{H2O} \right)^{\left(2-j \right)} + \displaystyle\sum_{k=1}^{l_j} b_{j,k}\, X^{\left(x_{j,k} \right)} } \end{aligned} \right) }{ \left(P_R^{H2O} \right)^{\left(2-j \right)} + \displaystyle\sum_{k=1}^{l_j} b_{j,k}\, X^{\left(x_{j,k} \right)} } \right] \\ & + P_R^{H2O} \left(\frac{P_R^{H2O}}{\beta_L} \right)^{10} \sum_{j=0}^{6} \left[\left(1 + T_R^{H2O} \left(\frac{10\, \beta_L'}{\beta_L} \right) + j\, b \right) B_{9,j}\, X^j \right] \end{aligned} \right\} \tag{2.28}$$

$$X = \exp\left(b \left(1 - T_R^{H2O} \right) \right);\ \beta_L = L_0 + L_1\, T_R^{H2O} + L_2 \left(T_R^{H2O} \right)^2;\ \beta_L' = L_1 + 2\, L_2\, \theta$$

$$T_R^{H2O} = \frac{T(K)}{T_c^{H2O}(K)};\ P_R^{H2O} = \frac{P(Pa)}{P_c^{H2O}(Pa)}$$

where the *b*, *B*, *l*, *L*, *n*, *x*, and *z* parameters are given in Table A.5. The gas internal energy is computed from the gas enthalpy, gas pressure, and gas density:

$$u_g = h_g - \frac{P_g}{\rho_g} \tag{2.29}$$

2.3.3 Precipitated Salt Enthalpy and Internal Energy

Precipitated salt enthalpy is computed using the empirical formulation of Lide and Kehiaian (1994), where the reference point is 273.15 K (0˚C):

$$h_s = A_0 + \sum_{i=1}^{5} \frac{A_i \left(T(K)\right)^i}{i} \tag{2.30}$$

where the *A* parameters are given in Table A.6. The precipitated salt internal energy is assumed to be equal to the precipitated salt enthalpy.

2.4 Transport Properties

Transport properties considered here are fluid state properties of viscosity, thermal conductivity, and diffusion coefficients, used to calculate the transport of components or energy through the geologic media.

2.4.1 Aqueous Viscosity

The aqueous viscosity is computed from the pure-water liquid viscosity, then adjusted for salt and dissolved CO_2. The pure-water liquid viscosity is computed as a function of temperature, pressure, and density using ASME steam table formulations (Meyer et al. 1993):

$$\begin{gathered}
Y = \sqrt{T_r^{H2O}} \left[\sum_{i=0}^{3} \frac{h_i}{\left(T_r^{H2O}\right)^i} \right]^{-1} \\
Z = \sum_{j=0}^{6} \sum_{i=0}^{5} h_{j,i} \left[\left(\frac{1}{T_r^{H2O}} - 1 \right)^i \right] \left(\rho_r^{H2O} - 1 \right)^j \\
\mu_l^{H2O} = \mu_c^{H2O} \left[Y \exp\left(\rho_r^{H2O}\, Z \right) \right] \\
T_r^{H2O} = \frac{T(K)}{T_c^{H2O}};\ \rho_r^{H2O} = \frac{\rho}{\rho_c^{H2O}} \\
T_c^{H2O} = 647.27\, K;\ \rho_c^{H2O} = 317.763\, kg/m^3;\ \mu_c^{H2O} = 5.5071\, x\, 10^{-5}\ Pa\, s
\end{gathered} \tag{2.31}$$

where the *h* parameters are given in Table A.7. A plot of the pure-water liquid viscosity as a function of temperature is shown in Figure 2.10.

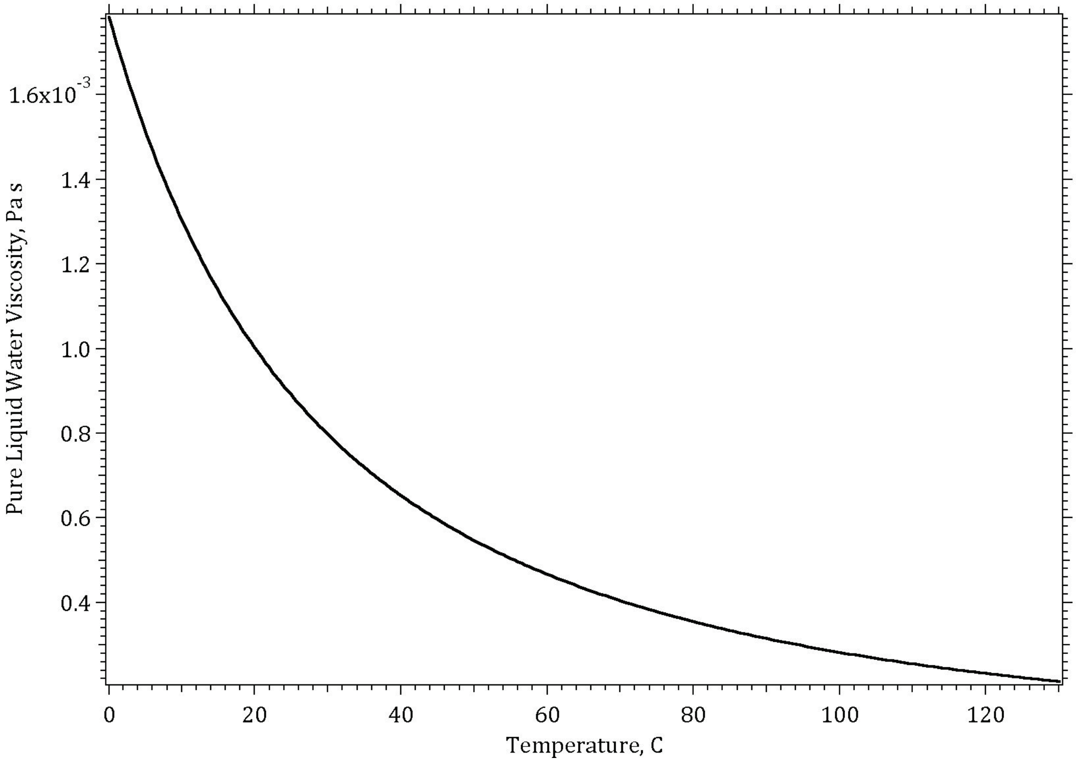

Figure 2.10. Pure-Water Liquid Viscosity

Brine viscosity is computed from the pure-water liquid viscosity as a function of salt molality and temperature using the formulation of Phillips et al (1981):

$$\mu_b = \mu_l^{H2O}\left(1 + a_1\, b_l^s + a_2\left(b_l^s\right)^2 + a_3\left(b_l^s\right)^3 + a_4\, T(C)\left(1-\exp\left(a_5\, b_l^s\right)\right)\right) \tag{2.32}$$

where the *a* parameters are given in Table A.8. Aqueous viscosity is computed from the brine viscosity, pure CO_2 viscosity, and aqueous-CO_2 mole fraction, using the formulation of Grunberg and Nissan (Reid et al. 1987):

$$\mu_l = \exp\left[\left(1-\chi_l^{CO2}\right)\ln\left(\mu_b\right) + \chi_l^{CO2}\ln\left(\mu^{CO2}\right)\right] \tag{2.33}$$

2.4.2 Gas Viscosity

Gas viscosity is computed from the pure component viscosities of CO_2 and water vapor using an extension proposed by Wilke of the Chapman-Enskog theory for multi-component gas mixtures at low densities (Reid et al. 1987). The semi-empirical expression of Wilke relates the gas mixture viscosity to pure component vapor viscosities and component gas mole fractions:

$$\mu_g = \sum_{i=H2O,CO2} \frac{\chi_g^i \, \mu_g^i}{\displaystyle\sum_{j=H2O,CO2} \chi_g^j \, \Phi_g^{i,j}}$$

$$\Phi_g^{i,j} = \left(\frac{1}{\sqrt{8}}\right)\left(1+\frac{M^i}{M^j}\right)^{(-1/2)}\left[1+\left(\frac{\mu_g^i}{\mu_g^j}\right)^{1/2}+\left(\frac{M^j}{M^i}\right)^{1/4}\right]^2 \qquad (2.34)$$

Pure CO2 viscosity is computed by combining the zero-density-limit-viscosity with the excess-viscosity using the formulation of Fenghour et al. (1998):

$$\mu_g^{CO2} = 10^{-6}\left(\mu_{zd}^{CO2}+\mu_{ex}^{CO2}\right)$$

$$\mu_{zd}^{CO2} = \frac{1.00697\sqrt{T_r^{CO2}}}{\exp\left\{\displaystyle\sum_{i=0}^{4} a_i\left[\log\left(T_r^{CO2}\right)\right]^i\right\}} ; T_r^{CO2} = \frac{T(K)}{251.196} \qquad (2.35)$$

$$\mu_{ex}^{CO2} = b_1\,\rho_g^{CO2} + b_2\left(\rho_g^{CO2}\right)^2 + b_3\,\frac{\left(\rho_g^{CO2}\right)^6}{\left(T_r^{CO2}\right)^3} + b_4\left(\rho_g^{CO2}\right)^8 + b_5\,\frac{\left(\rho_g^{CO2}\right)^8}{T_r^{CO2}}$$

where, the *a* and *b* parameters are given in Table A.9. Pure water-vapor viscosity is computed as a function of temperature, pressure and density using the ASME steam table formulations (Meyer et al. 1993), per Equation (2.31), where, the *b* parameters are given in Table A.7.

2.4.3 Aqueous Thermal Conductivity

Aqueous thermal conductivity is computed as a function of the pure-water liquid thermal conductivity as a function of temperature and salt aqueous mass fraction using the empirical Yusufova correlation (Ozbek and Phillips, 1980):

$$k_l = k_b = k_l^{H2O}\left[\begin{array}{l} 1-\left(c_1+c_2\,T(C)+c_3\left(T(C)\right)^2\right)\omega_l^s(\%) \\ +\left(c_4+c_5\,T(C)+c_6\left(T(C)\right)^2\right)\left(\omega_l^s(\%)\right)^2 \end{array}\right] \qquad (2.36)$$

where the *c* parameters are given in Table A.10. The pure-water liquid thermal conductivity is computed as a function of temperature and density using the ASME steam table formulations (Meyer et al. 1993):

$$k_l^{H2O} = X_1 + \left\{ \begin{array}{l} \left(\dfrac{d_1}{\left(T_R^{H2O}\right)^{10}} + d_2 \right) \left(\rho_R^{H2O}\right)^{9/5} \exp\left(C_1 \left(1 - \left(\rho_R^{H2O}\right)^{14/5} \right) \right) \\ + d_3\, S \left(\left(\rho_R^{H2O}\right)^{Q} \right) \exp\left(\left(\dfrac{Q}{1+Q} \right) \left(1 - \left(\rho_R^{H2O}\right)^{(1+Q)} \right) \right) \\ + d_4 \exp\left(C_2 \left(T_R^{H2O}\right)^{3/2} + \dfrac{C_3}{\left(\rho_R^{H2O}\right)^5} \right) \end{array} \right\} \tag{2.37}$$

$$\theta' = \left|1 - T_R^{H2O}\right| + C_4;\ Q = 2 + \frac{C_5}{\left(\theta'\right)^{3/5}}$$

$$for\ \theta' \geq 1,\ S = \frac{1}{\theta'};\ for\ \theta' < 1,\ S = \frac{C_6}{\left(\theta'\right)^{3/5}}$$

$$X_1 = X_0 + b_1 + b_2\, \rho_R^{H2O} + b_3 \exp\left(B_1 \left(\rho_R^{H2O} + B_2 \right)^2 \right)$$

$$X_0 = \left(T_R^{H2O}\right)^{1/2} \sum_{i=0}^{3} a_i \left(T_R^{H2O}\right)^i$$

where the *a*, *b*, *B*, *C,* and *d* parameters are given in Table A.11.

2.4.4 Gas Thermal Conductivity

Gas thermal conductivity is computed using the Mason and Saxena modification to the Waasiljewa equation (Reid et al. 1987) from the pure CO_2 and pure-water vapor thermal conductivities:

$$
\begin{aligned}
k_g &= \frac{\chi_g^{H2O}\, k_g^{H2O}}{\mathrm{X}_g^{H2O}} + \frac{\chi_g^{CO2}\, k_g^{CO2}}{\mathrm{X}_g^{CO2}} \\
\mathrm{X}_g^{H2O} &= \chi_g^{H2O} + \chi_g^{CO2}\, \phi^{H2O-CO2} \\
\mathrm{X}_g^{CO2} &= \chi_g^{H2O} \phi^{CO2-H2O} + \chi_g^{CO2} \\
\phi^{H2O-CO2} &= \frac{\left[1 + \left(\dfrac{\phi^{H2O}}{\phi^{CO2}} \right)^{1/2} \left(\dfrac{M^{CO2}}{M^{H2O}} \right)^{1/4} \right]^2}{\left[8 \left(1 + \dfrac{M^{H2O}}{M^{CO2}} \right) \right]^{1/2}} \\
\phi^{CO2-H2O} &= \frac{\left[1 + \left(\dfrac{\phi^{CO2}}{\phi^{H2O}} \right)^{1/2} \left(\dfrac{M^{H2O}}{M^{CO2}} \right)^{1/4} \right]^2}{\left[8 \left(1 + \dfrac{M^{CO2}}{M^{H2O}} \right) \right]^{1/2}} \\
\phi^{H2O} &= \Gamma^{H2O} \left[\exp\left(0.0464\, T_R^{H2O} \right) - \exp\left(-0.2412\, T_R^{H2O} \right) \right] \\
\phi^{CO2} &= \Gamma^{CO2} \left[\exp\left(0.0464\, T_R^{CO2} \right) - \exp\left(-0.2412\, T_R^{CO2} \right) \right] \\
\Gamma^{H2O} &= 210.0 \left[\frac{T_c^{H2O}\,(K) \left(M^{H2O} \right)^3}{\left(P_c^{H2O} (bar) \right)^4} \right] \\
\Gamma^{CO2} &= 210.0 \left[\frac{T_c^{CO2}\,(K) \left(M^{CO2} \right)^3}{\left(P_c^{CO2} (bar) \right)^4} \right]
\end{aligned}
\tag{2.38}
$$

2.4.5 Aqueous Molecular Diffusion Coefficients

Dissolved CO_2 and salt diffuse through the aqueous phase in response to their concentration gradients. A molar balance of water diffusion is maintained. The molecular aqueous diffusion coefficient for CO_2 can be specified as a constant or computed as a function of aqueous viscosity and CO_2 gas viscosity using the formulation of Renner (1998), with an upper limit restriction based on literature reference data:

$$D_l^{CO2} = \min\left[6.391e+3 \left(\mu_l \left(cP \right)^{-0.1584} \right) \left(\mu_g \left(cP \right)^{6.911} \right), 10^{-9} \right] \left(\frac{m^2}{s} \right) \tag{2.39}$$

The molecular aqueous diffusion coefficient for salt can be specified as a constant or computed as a function of temperature and salt concentration using the method of Nernst-Haskell for dilute solutions and the method of Gordon for concentrated solutions (Reid et al. 1987); where the mean ionic activity is computed according to the method of Bromley (1973):

$$\begin{gathered} D_l^s = {}^{25^\circ C}D_w^s \left[\left(\frac{{}^{25^\circ C}\mu_l^w}{\mu_b} \right) \right] \left(1 + b_l^s \, \frac{\partial \ln\left(\gamma\right)}{\partial \, b_l^s} \right) \left(\frac{T\left(K\right)}{298.15\,K} \right) \left(\frac{m^2}{s} \right) \\ \frac{\partial \ln\left(\gamma\right)}{\partial \, b_l^s} = 2.30258 \left\{ \begin{array}{l} \left(\dfrac{-0.2555}{\sqrt{b_l^s}\left(1+\sqrt{b_l^s}\right)} \right) + \left(\dfrac{0.2555}{\left(1+\sqrt{b_l^s}\right)^2} \right) \\ + \left(\dfrac{0.06+0.6\,B}{\left(1+1.5\,b_l^s\right)^2} \right) - \left(\dfrac{3\left(0.06+0.6\,B\right)b_l^s}{\left(1+1.5\,b_l^s\right)^3} \right) + B \end{array} \right\} \\ B = 0.0547; \;\; {}^{25^\circ C}D_w^s = 2.254\,x\,10^{-9}\,\frac{m^2}{s} \end{gathered} \tag{2.40}$$

The aqueous molecular diffusion coefficients are converted to effective aqueous diffusion coefficients for the geologic media via the aqueous tortuosity factor, aqueous saturation, and porosity.

2.4.6 Gas Molecular Diffusion Coefficients

Water vapor diffuses through the CO_2-rich phase (i.e., gas) in response to its concentration gradient. A molar balance of CO_2 diffusion is maintained. The molecular diffusion coefficient for water vapor in the gas phase is computed via the method of Wilke and Lee (Reid et al., 1987):

$$D_g^{H2O} = \frac{\left[3.03 - \left(\frac{0.98}{\sqrt{\bar{M}}}\right)\right]\left(10^{-7}\right)\left(T(K)\right)^{3/2}}{P(bar)\sqrt{\bar{M}}\left(\bar{\sigma}\right)^2 \Omega_D}\left(\frac{m^2}{s}\right)$$

$$\Omega_D = \frac{A_1}{\left(T^*\right)^{A_2}} + \frac{A_3}{\exp\left(A_4\, T^*\right)} + \frac{A_5}{\exp\left(A_6\, T^*\right)} + \frac{A_7}{\exp\left(A_8\, T^*\right)}$$

$$T^* = \frac{T(K)}{\sqrt{\left(\frac{\varepsilon}{k}\right)^{H2O}\left(\frac{\varepsilon}{k}\right)^{CO2}}} \tag{2.41}$$

$$\bar{M} = \frac{2}{\left(\frac{1}{M^{H2O}} + \frac{1}{M^{CO2}}\right)};\ \bar{\sigma} = \frac{\left(\sigma^{H2O} + \sigma^{CO2}\right)}{2}$$

$$\sigma^{H2O} = 3.190008977\,\mathring{A};\ \sigma^{CO2} = 3.795165630\,\mathring{A}$$

$$\left(\frac{\varepsilon}{k}\right)^{H2O} = 429.18\,\text{K};\ \left(\frac{\varepsilon}{k}\right)^{CO2} = 95.85245\,\text{K}$$

where the *A* parameters are given in Table A.12.

2.5 Phase Saturation

Aqueous saturation is the fraction of pore space filled with aqueous phase and is computed from the scaled gas-aqueous capillary pressure using functional forms (e.g., van Genuchten or Brooks and Corey), or via user provided data. Interpolation of user provided data can be linear or cubic spline. Aqueous saturations below the residual saturation are computed via extensions to the functional forms of the aqueous saturation versus capillary pressure relations. Gas saturation is the fraction of pore space filled with gas phase. Gas saturation can be free (mobile) or trapped (immobile). A single grid cell can contain free gas only, free and trapped gas, or trapped gas only.

2.5.1 Interfacial Tension Scaling

Capillary pressures used to calculate aqueous saturations can be scaled using interfacial tension scaling:

$$\beta_{gl} = \frac{\sigma^*}{\sigma_{gl}} \tag{2.42}$$

where the reference surface tension is the interfacial surface tension of the fluid pairs used to determine the saturation versus capillary pressure function.

2.5.2 Aqueous Saturation

The van Genuchten (1980) correlation relates the scaled capillary pressure to the actual aqueous saturation via four parameters:

$$\bar{s}_l = \frac{s_l - s_{lr}}{1 - s_{lr}} = \left(1 + \left(\alpha \, \beta_{gl} \, h_{gl} \right)^n \right)^{-m} \; ; \; h_{gl} = \left(\frac{P_g - P_l}{\rho_l^* \, g} \right) \tag{2.43}$$

The Brooks and Corey (1966) correlation is a three-parameter function that makes use of an entry pressure to inhibit desaturation for scaled capillary pressures below the entry pressure:

$$\begin{aligned} &for \;\; \beta_{gl} \, h_{gl} \leq \varphi : \; \bar{s}_l = \frac{s_l - s_{lr}}{1 - s_{lr}} = 1 \\ &for \;\; \beta_{gl} \, h_{gl} > \varphi : \; \bar{s}_l = \frac{s_l - s_{lr}}{1 - s_{lr}} = \left(\frac{\varphi}{\beta_{gl} \, h_{gl}} \right)^{\lambda} \; ; \; h_{gl} = \left(\frac{P_g - P_l}{\rho_l^* \, g} \right) \end{aligned} \tag{2.44}$$

The Haverkamp et al. (1977) correlation is similar to that of Brooks and Corey (1966) in that it makes use of an entry pressure, but is a four-parameter function:

$$\begin{aligned} &for \;\; \beta_{gl} \, h_{gl} \leq \varphi : \; \bar{s}_l = \frac{s_l - s_{lr}}{1 - s_{lr}} = 1 \\ &for \;\; \beta_{gl} \, h_{gl} > \varphi : \; \bar{s}_l = \frac{s_l - s_{lr}}{1 - s_{lr}} = \left(\frac{\alpha}{\alpha + \left(h_{gl} - \varphi \right)^{\beta}} \right) ; \; h_{gl} = \left(\frac{P_g - P_l}{\rho_l^* \, g} \right) \end{aligned} \tag{2.45}$$

Dual-porosity or equivalent-continuum models for aqueous-gas systems relate the gas-aqueous capillary pressure to the bulk aqueous saturation for fractured geologic media through two functions [Klavetter and Peters 1986; Nitao 1988]. One function relates the gas- aqueous capillary pressure to the matrix aqueous saturation and the other relates the gas-aqueous capillary pressure to the fracture aqueous saturation. The pivotal assumption associated with the dual porosity function is that the fracture and matrix pressures are in equilibrium. This assumption neglects transient fracture-matrix interactions. The dual-porosity function is implemented in STOMP-CO2 and -CO2e simulators for the van Genuchten (1980), Eqn. (2.43), or Brooks and Corey (1966), Eqn. (2.44) aqueous saturation versus capillary pressure functions. The effective actual aqueous saturation for the dual-porosity model is computed from a porosity-weighted combination of the actual fracture and matrix aqueous saturations:

$$s_l = \frac{s_{lf}\, n_{Df} + s_{lm}\left(1-n_{Df}\right) n_{Dm}}{n_{Df} + \left(1-n_{Df}\right) n_{Dm}}$$

$$\bar{s}_{lf} = \frac{s_{lf} - s_{lrf}}{1 - s_{lrf}} = \left(1 + \left(\alpha_f\, \beta_{gl}\, h_{gl}\right)^{n_f}\right)^{-m_f} \;;\; h_{gl} = \left(\frac{P_g - P_l}{\rho_l^*\, g}\right) \tag{2.46}$$

$$\bar{s}_{lm} = \frac{s_{lm} - s_{lrm}}{1 - s_{lrm}} = \left(1 + \left(\alpha_m\, \beta_{gl}\, h_{gl}\right)^{n_m}\right)^{-m_m} \;;\; h_{gl} = \left(\frac{P_g - P_l}{\rho_l^*\, g}\right)$$

where, the van Genuchten (1980) function is used to demonstrate the split calculation of the effective fracture and matrix aqueous saturations.

2.5.3 Trapped Gas Saturation

A fundamental assumption for the gas entrapment model in STOMP-CO2 and -CO2e is that the aqueous phase is the wetting phase relative to the gas phase. Gas entrapment is assumed to occur only when the aqueous phase is on an imbibition path (i.e., increasing aqueous saturation). Gas saturation can be free or trapped:

$$s_g = 1 - s_l = s_{gf} + s_{gt} \tag{2.47}$$

where the trapped gas is assumed to be in the form of aqueous occluded ganglia and immobile. A complete theoretical model for scanning path hysteresis and nonwetting fluid entrapment was developed by Parker and Lenhard (1987). The entrapment model used in the STOMP-CO2 and -CO2e simulators is formulated after the simplifications of the Parker and Lenhard model published by Kaluarachchi and Parker (1992). The potential effective trapped gas saturation varies between zero and the effective maximum trapped gas saturation as a function of the historical minimum value of the apparent aqueous saturation:

$$\bar{s}_{gt}^{potential} = \frac{s_{gt}^{potential}}{1 - s_{lr}} = \bar{s}_{gt}^{\max}\left(\frac{1 - \bar{\bar{s}}_l^{\min}}{1 - \bar{\bar{s}}_l^{\min}\left(1 + \bar{s}_{gt}^{\max}\right)}\right) = \left(\frac{1 - \bar{\bar{s}}_l^{\min}}{1 - R\left(1 - \bar{\bar{s}}_l^{\min}\right)}\right)$$

$$\bar{\bar{s}}_l^{\min} = \bar{s}_l^{\min} = \frac{s_l^{\min} - s_{lr}}{1 - s_{lr}};\; R = \frac{1}{\bar{s}_{gt}^{\max}} - 1 \tag{2.48}$$

$$\bar{\bar{s}}_l = \bar{s}_l + \bar{s}_{gt} = \frac{s_l + s_{gt} - s_{lr}}{1 - s_{lr}};\; \bar{s}_l = \frac{s_l - s_{lr}}{1 - s_{lr}};\; \bar{s}_{gt} = \frac{s_{gt}}{1 - s_{lr}}$$

where *R* is the Land's parameter (1968). The effective trapped gas saturation varies between zero and the effective potential trapped gas saturation as a function of the apparent aqueous saturation:

$$\bar{s}_{gt} = \frac{s_{gt}}{1-s_{lr}} = \left(\frac{1-\bar{\bar{s}}_l^{\min}}{1-R\left(1-\bar{\bar{s}}_l^{\min}\right)} \right) - \left(\frac{1-\bar{\bar{s}}_l}{1-R\left(1-\bar{\bar{s}}_l\right)} \right) \tag{2.49}$$

The modeling of gas entrapment is optional in STOMP-CO2 and -CO2e. When the gas entrapment model is active, the effective aqueous saturations in Eqns. (2.43-2.45) are replaced with the apparent aqueous saturations, which makes the apparent aqueous saturations a function of the capillary pressure.

2.5.4 Extension Below Residual Aqueous Saturation

When CO_2 is injected into a deep saline geologic reservoir, initially the aqueous phase is displaced. As injection continues, the aqueous relative permeability drops, especially near the injection point, and the formation rock begins to desaturate through desiccation by the injected CO_2. Conventional capillary pressure and saturation functions (e.g., van Genuchten, Eqn. 2.43, Brooks and Corey, Eqn. 2.44, and Haverkamp, Eqn. 2.45) are limited to saturations above the residual saturation. For these functions, the capillary pressure tends toward infinity as the aqueous saturation approaches the residual saturation. This limitation can be overcome by specifying an extension option for the either the van Genuchten (1980) or Brooks and Corey (1966) characteristic functions. The Fayer and Simmons (1995) extension modifies the aqueous residual saturation as a function of capillary pressure:

$$s_{lr} = \left(1 - \frac{\ln\left(h_{gl}\right)}{\ln\left(h_{od}\right)} \right) s'_{lr} \tag{2.50}$$

where the primed value is the specified residual saturation and the oven-dried head is equivalent to a capillary pressure of 10^9 Pa (~10^5 m).

The Webb (2000) extension divides the capillary pressure-saturation function into two regimes. For low aqueous saturation the function follows a logarithmic form, and for moderate to high saturations the function follows the specified form (e.g., van Genuchten, Brooks and Corey). Transition between the two forms occurs at the matching point, which occurs at the point where the two functions have matching slopes. The low-saturation function is a linear function on a semilog plot:

$$s_l = \frac{\log\left(\dfrac{h_{gl}}{h_{gl}^{mp}} \right)}{\gamma} + s_l^{mp}; \ \gamma = -\frac{\log\left(\dfrac{h_{od}}{h_{gl}^{mp}} \right)}{s_l^{mp}} \tag{2.51}$$

The matching point saturation and capillary pressure head is determined by setting the partial derivative of the aqueous saturation with respect to the capillary pressure head in Eqn. (2.51) equal to the partial derivative of the aqueous saturation with respect to the capillary pressure head in the specified function form (e.g., van Genuchten, Eqn. 2.43, Brooks and Corey, Eqn. 2.44, and Haverkamp, Eqn. 2.45), setting the capillary pressure head to the matching point capillary pressure head, setting the aqueous saturation to the matching point aqueous saturation and solving for the matching point aqueous saturation. A nonlinear solve is required to determine the matching point saturation and capillary pressure head for the van Genuchten, Eqn. 2.43, Brooks and Corey, Eqn. 2.44 functions, but this solution is only required once, during simulation initialization. An oven-dried head of 10^9 Pa (~10^5 m) is used for the Webb extension model.

2.6 Phase Relative Permeability

Phase relative permeability is computed from the aqueous, gas, and trapped gas saturations using empirical functions. Aqueous saturations below the aqueous residual saturation are assumed to be immobile and trapped gas is assumed to be immobile.

2.6.1 Aqueous Relative Permeability

A number of functional forms are available for computing the aqueous relative permeability, including tabular input. The Burdine (1953) and Mualem (1976) forms are associated with the van Genuchten (1980) or Brooks and Corey (1966) capillary pressure-saturation functions. When combined with the van Genuchten function (Eqn. 2.43), the Burdine (1953) and Mualem (1976) functions, respectively, for aqueous relative permeability depend on the effective aqueous saturation and porosity distribution factor:

$$\begin{aligned} k_{rl} &= \left(\bar{s}_l\right)^2 \left[1 - \left(1 - \left(\bar{s}_l\right)^{1/m} \right)^m \right] \\ k_{rl} &= \left(\bar{s}_l\right)^{1/2} \left[1 - \left(1 - \left(\bar{s}_l\right)^{1/m} \right)^m \right]^2 \end{aligned} \tag{2.52}$$

When combined with the Brooks and Corey function (Eqn. 2.44), the Burdine (1953) and Mualem (1976) functions, respectively, for aqueous relative permeability depend on the effective aqueous saturation and porosity distribution factor:

$$\begin{aligned} k_{rl} &= \left(\bar{s}_l\right)^{\left(3+\frac{2}{\lambda}\right)} \\ k_{rl} &= \left(\bar{s}_l\right)^{\left(\frac{5}{2}+\frac{2}{\lambda}\right)} \end{aligned} \tag{2.53}$$

The Corey (1977) and Fatt and Klikoff (1959) functions, respectively, are simple exponential functions of the effective aqueous saturation:

$$\begin{aligned} k_{rl} &= \left(\bar{s}_l\right)^4 \\ k_{rl} &= \left(\bar{s}_l\right)^3 \end{aligned} \tag{2.54}$$

The Haverkamp (1977) aqueous relative permeability model is formulated in terms of capillary pressure instead of effective aqueous saturation:

$$\begin{aligned} &for \;\; \beta_{gl}\, h_{gl} \le \varphi:\; k_{rl} = 1 \\ &for \;\; \beta_{gl}\, h_{gl} > \varphi:\; k_{rl} = \left[\frac{A}{A + \left(h_{gl} - \varphi \right)^{\gamma}} \right] \end{aligned} \tag{2.55}$$

The Touma and Vauclin (1986) and the Modified Corey (1977) functions, respectively, are both expanded exponential forms:

$$\begin{aligned} k_{rl} &= a \left(\bar{s}_l \right)^{b} \\ k_{rl} &= a \left(\frac{s_l - s_{lr}}{1 - s_{lr} - s_{gr}} \right)^{b} \end{aligned} \tag{2.56}$$

Dual-porosity or equivalent-continuum models (Klavetter and Peters 1986; Nitao 1988) for aqueous relative permeability combine the matrix and fracture aqueous relative permeabilities into an effective aqueous relative permeability:

$$k_{rl} = \frac{k_m\, k_{rlm} \left(1 - n_{Df} \right) - k_f\, k_{rlf}\, n_{Df}}{k_m \left(1 - n_{Df} \right) - k_f\, n_{Df}} \tag{2.57}$$

where, the matrix and fracture intrinsic permeabilities can have components in the three principal grid directions, making the aqueous relative permeability anisotropic. The theory of Lenhard and Parker (1987) accounts for the pore-space distribution of trapped gas; where the aqueous relative permeability of a given effective aqueous saturation is greater with trapped gas than without. This results from the assumption that trapped gas is totally occluded by the aqueous phase, which displaces the aqueous phase into larger pore spaces than without trapped gas. This effect is not considered in STOMP-CO2 and -CO2e, in that the aqueous relative permeability for trapped gas conditions is only a function of the effective aqueous saturation.

2.6.2 Gas Relative Permeability

A number of functional forms are available for computing the gas relative permeability, including tabular input. The Burdine (1953) and Mualem (1976) forms are associated with the van Genuchten (1980) or Brooks and Corey (1966) capillary pressure-saturation functions. When combined with the van Genuchten function, (Eqn. 2.43), the Burdine (1953) and Mualem (1976) functions, respectively, for gas relative permeability depend on the effective free gas saturation, apparent aqueous saturation, and porosity distribution factor:

$$k_{rg} = \left(\bar{s}_{gf} \right)^{2} \left[\left(1 - \left(\bar{\bar{s}}_l \right)^{1/m} \right)^{m} \right]; \;\; \bar{s}_{gf} = 1 - \bar{\bar{s}}_l \tag{2.58a}$$

$$k_{rg} = \left(\bar{s}_{gf} \right)^{1/2} \left[\left(1 - \left(\bar{\bar{s}}_l \right)^{1/m} \right)^{m} \right]^{2} \;; \;\; \bar{s}_{gf} = 1 - \bar{\bar{s}}_l \tag{2.58b}$$

When combined with the Brooks and Corey function (Eqn. 2.44), the Burdine (1953) and Mualem (1976) functions, respectively, for gas relative permeability depend on the apparent aqueous saturation and porosity distribution factor:

$$k_{rg} = \left(\bar{s}_{gf}\right)^2 \left[\left(1 - \left(\bar{\bar{s}}_l\right)^{\left(1+\frac{2}{\lambda}\right)}\right)\right]; \; \bar{s}_{gf} = 1 - \bar{\bar{s}}_l$$

$$k_{rg} = \left(\bar{s}_{gf}\right)^{1/2} \left[\left(1 - \left(\bar{\bar{s}}_l\right)^{\left(1+\frac{1}{\lambda}\right)}\right)\right]^2 \; ; \; \bar{s}_{gf} = 1 - \bar{\bar{s}}_l \tag{2.59}$$

The Fatt and Klikoff (1959) function is a simple exponential function of the effective free gas saturation or apparent aqueous saturation:

$$k_{rg} = \left(\bar{s}_{gf}\right)^3; \; \bar{s}_{gf} = 1 - \bar{\bar{s}}_l \tag{2.60}$$

The Corey (1977) and Free Corey functions are exponential forms, which include a residual gas saturation. As such, these functions, respectively, use the actual aqueous saturation:

$$k_{rg} = a\left(1 - \left(\frac{s_l - s_{lr}}{1 - s_{lr} - s_{gr}}\right)\right)^2 \left(1 - \left(\frac{s_l - s_{lr}}{1 - s_{lr} - s_{gr}}\right)^2\right); \; 1 - s_l = s_g = s_{gf} + s_{gt}$$

$$k_{rg} = a\left(1 - \left(\frac{s_l - s_{lr}}{1 - s_{lr} - s_{gr}}\right)\right)^b \; ; \; 1 - s_l = s_g = s_{gf} + s_{gt} \tag{2.61}$$

The van Genuchten (1980) gas relative permeability model is similar to the combined Mualem and van Genuchten function (Eqn. 2.58b), but additionally includes a residual gas saturation:

$$k_{rg} = \left(\bar{s}_{gf}\right)^{1/2} \left[\left(1 - \left(1 - \bar{s}_{gf}\right)^{1/m}\right)^m\right]^2 ; s_{gf} = \left(1 - \bar{\bar{s}}_l\right)\left(1 - s_{lr}\right); \bar{s}_{gf} = \frac{s_{gf} - s_{gr}}{1 - s_{gr}} \tag{2.62}$$

Dual-porosity or equivalent-continuum models (Klavetter and Peters 1986; Nitao 1988) for aqueous relative permeability combine the matrix and fracture aqueous relative permeabilities into an effective aqueous relative permeability:

$$k_{rg} = \frac{k_m \, k_{rgm} \left(1 - n_{Df}\right) - k_f \, k_{rgf} \, n_{Df}}{k_m \left(1 - n_{Df}\right) - k_f \, n_{Df}} \tag{2.63}$$

where, the matrix and fracture intrinsic permeabilities can have components in the three principal grid directions. The gas relative permeability is not considered to be a directional tensor, so the maximum value computed from the three principal grid directions is used as the scalar value.

2.7 Transport Parameters

Transport parameters described in this document are the effective properties for geologic permeability, phase tortuosity, and thermal conductivity.

2.7.1 Permeability

Permeability is computed as the product of the rock intrinsic permeability and the permeability reduction factor. The permeability tensor in STOMP-CO2 and -CO2e has zero off-diagonal elements, requiring the specification of intrinsic permeability in the three principal axis directions, but intrinsic permeabilities for inactive principal directions are not required. For example, in a one-dimensional radial problem, only the intrinsic permeability in the radial direction needs to be specified. The intrinsic permeability is specified via the *Hydraulic Properties Card* (Sections 4.2.9 and B.9) and can be specified with units of intrinsic permeability (e.g., m^2, Darcy, mD) or with units of hydraulic conductivity (e.g., m/day, cm/hr). If units of hydraulic conductivity are specified, the intrinsic permeability is computed assuming that the hydraulic conductivity was specified for the density and viscosity of water at 25°C and 1 atm. Permeability reduction occurs via salt precipitation following the formulation of Pruess and Garcia (2002) and Verma and Pruess (1988). The permeability reduction model uses a tube-in-series model to represent permeability changes:

$$\frac{k}{k_0} = \theta^2 \left[\frac{1 - f + \left(\dfrac{f}{\omega^2}\right)}{1 - f + f \left(\dfrac{\theta}{\theta + \omega - 1}\right)^2} \right]$$
$$\theta = \frac{1 - s_s - \phi_r}{1 - \phi_r}; \;\; \omega = 1 + \frac{\left(\dfrac{1}{f}\right)}{\left(\dfrac{1}{\phi_r} - 1\right)} \tag{2.64}$$

where, ϕ_r denotes the fraction of original porosity at which permeability goes to zero and f is the fractional length of the pore body. A plot of the reduction in permeability ratio versus salt saturation for ϕ_r of 0.8, 0.7, and 0.6, and f of 0.8 and 0.5 is shown in Figure 2.11.

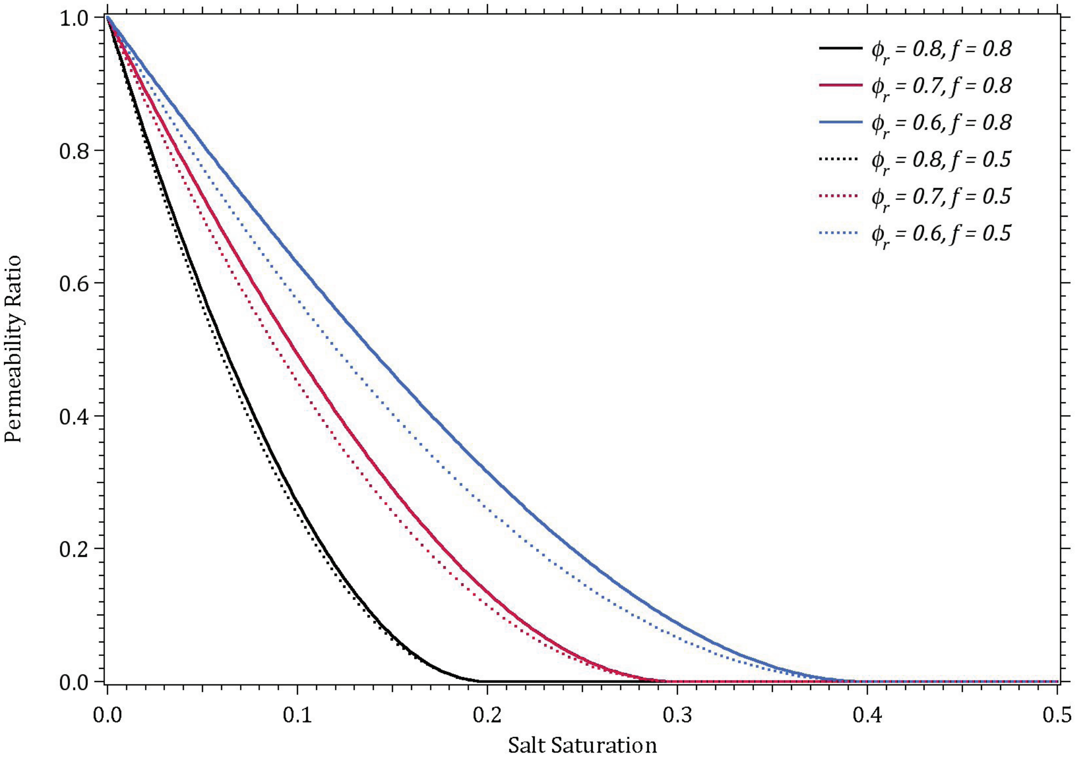

Figure 2.11. Permeability Reduction with Salt Precipitation

2.7.2 Tortuosity

Component diffusion through geologic media depends on the phase molecular diffusion coefficient (Sections 2.4.5 and 2.4.6), porosity, phase saturation, and the tortuosity factor; where the tortuosity factor effectively reduces the molecular diffusion coefficient for increased diffusion path length. In STOMP-CO2 and -CO2e the tortuosity factor can be specified as a constant or as a function of porosity and phase saturation using the formulation of Millington and Quirk (1959):

$$\tau_l = \left(n_D\right)^{1/3}\left(s_l\right)^{7/3};\ \tau_g = \left(n_D\right)^{1/3}\left(s_g\right)^{7/3} \tag{2.65}$$

2.7.3 Thermal Conductivity

Three functional forms are available in STOMP-CO2 and -CO2e for computing the effective thermal conductivity. As with permeability, the thermal conductivity tensor has zero off-diagonal elements, requiring the specification of rock thermal conductivity in the three principal axis directions, but rock thermal conductivities for inactive principal directions are not required. The DeVries model (1966) computes effective thermal conductivity using a phase volumetric weighting scheme from the rock thermal conductivity, phase thermal conductivities (Sections 2.4.3 and 2.4.4), and phase saturations:

$$\mathbf{k}_e = \mathbf{k}_s\,(1 - n_D) + k_l\, n_D\, s_l + k_g\, n_D\, s_g \tag{2.66}$$

The linear model computes the effective thermal conductivity as the aqueous-saturation scaled average between the unsaturated and saturated effective thermal conductivities:

$$\mathbf{k}_e = \mathbf{k}_{unsat} + s_l \left(\mathbf{k}_{sat} - \mathbf{k}_{unsat} \right) \tag{2.67}$$

The Somerton et al. (1973, 1974) model computes the effective thermal conductivity as the square root of the aqueous-saturation scaled average between the unsaturated and saturated effective thermal conductivities:

$$\mathbf{k}_e = \mathbf{k}_{unsat} + \sqrt{s_l} \left(\mathbf{k}_{sat} - \mathbf{k}_{unsat} \right) \tag{2.68}$$

2.8 Coupled Well Model

The coupled well model fully integrates the well equations into the coupled conservation equations (Eqns. 1.2 through 1.6). The current formulation of the coupled well model ignores pressure drops in the well due to friction losses, which results in a well under hydrostatic conditions for the well fluid. The coupled well model has two modes: 1) flow controlled and 2) pressure controlled. In the flow controlled mode, the unknown for the coupled well model is the pressure at the starting point of the screened intervals, and in the pressure controlled mode the unknown is the well flow rate given a specified pressure at the starting point of the screened intervals. A coupled well is specified by declaring linear sections of screened intervals and a time varying well flow rate and pressure limit. If the well flow rate can be achieved within the pressure limit, then the well is flow controlled and well pressure becomes the principal unknown. If the well flow cannot be achieved within the pressure limit, then the well pressure is fixed and the well flow rate becomes the principal unknown. Transitions between the coupled well being flow and pressure controlled is handled automatically. For the flow controlled mode, the well pressure becomes the primary variable for the coupled field conservation equations and well model equations in the Jacobian matrix. For the pressure controlled mode, the well pressure is fixed. The diagonal term for the well primary variable in the Jacobian matrix is set to 1.0 and the problem vector is set to 0.0.

The well trajectory for the coupled well model is specified by declaring linear segments of screened intervals. The screened intervals do not need to be contiguous and can be declared outside of the computational domain. Only those sections of the screened intervals that reside within the computational domain will be considered in the coupled well calculations. The starting point for the screened intervals is the first point at which a declared screened interval crosses the computational domain. This point becomes the point at which the well pressure is defined. The well trajectory can transition more than once within a grid cell and grid cells can contain multiple well segments from the same or different wells. Connections between the well intervals and computational nodes are calculated within the STOMP-CO2 and -CO2e simulators. The flexible well trajectory was implemented using a modification to the Peaceman well index formulation for describing the coupling between the flow rate between the well and reservoir as a function of well pressure, reservoir pressure, well properties, and reservoir properties. The modification, known as the projection well index, involves the projection of the linear well interval segments onto the principal orthogonal axes of the computational grid. A Peaceman well index is then computed for each of the three orthogonal directions and the overall well index for the node is computed as the square root of the sum of the squares of the directional Peaceman well indices.

A well index is defined as the ratio of the well flow rate, times the viscosity of the well fluid, divided by the difference in wellbore and grid-cell pressures:

$$WI = \frac{q\,\mu_\gamma}{\left(P_\gamma^w - P_\gamma\right)} \tag{2.69}$$

The classical approach to the well problem is the Peaceman model for the well index:

$$WI = \left[\frac{2\,\pi\,\sqrt{k_x\,k_y}\;\Delta z}{\ln\left(\frac{r_o}{r_w}\right) + s} \right] \tag{2.70}$$

which is based on single-phase steady-state radial flow from a vertical well section into a grid cell. The radius of the grid cell is defined as the radial position at which the grid-cell pressure is equal to the pressure obtained from the analytical radial solution to the flow problem:

$$q = \left(\frac{2\,\pi\,\sqrt{k_x\,k_y}\;\Delta z}{\mu_\gamma} \right) \left(\frac{P_\gamma^w - P_\gamma}{\ln\left(\frac{r_o}{r_w}\right)} \right) \tag{2.71}$$

where the grid cell radius is a function of the grid cell geometry and intrinsic permeability:

$$r_o = 0.28 \left[\frac{\left(\left\{ \frac{k_y}{k_x} \right\}^{1/2} \Delta x^2 + \left\{ \frac{k_x}{k_y} \right\}^{1/2} \Delta y^2 \right)^{1/2}}{\left(\left\{ \frac{k_y}{k_x} \right\}^{1/4} + \left\{ \frac{k_x}{k_y} \right\}^{1/4} \right)} \right] \tag{2.72}$$

Well segments in STOMP-CO2 and -CO2e are not restricted to a vertical orientation, which requires the use of a projection well index (Shu, 2005); where the well trajectory is projected on to the coordinate axes, as shown in Figure 2.12.

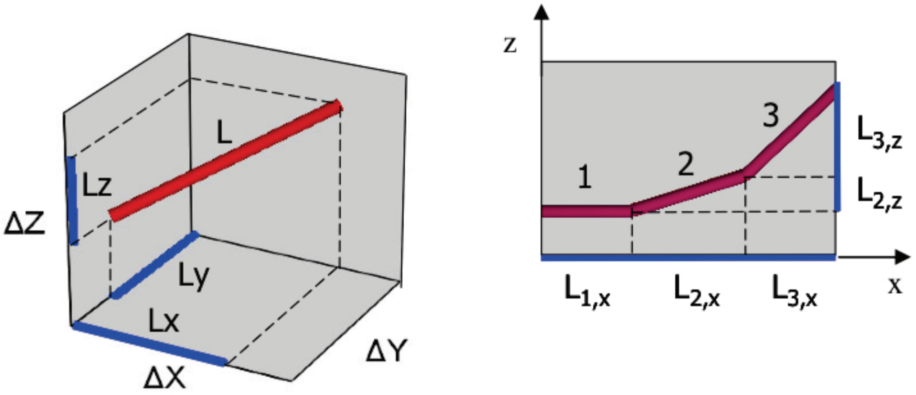

Figure 2.12. Well Trajectory Projected onto Coordinate Axes and Well Segments Projected onto Coordinate Axes (image from Shu, 2005)

Multiple well segments within a single grid cell, as shown in the right-hand schematic in Figure 2.12, are combined by summing projections onto the coordinate axes:

$$L_x = \sum_i L_x^i ;\; L_y = \sum_j L_y^j ;\; L_z = \sum_k L_z^k \tag{2.73}$$

Well segment projections are combined into a single well index, using the projection well model (Shu, 2005):

$$WI_x = \left[\frac{2\pi \sqrt{k_y\, k_z}\, L_x}{\ln\left(\dfrac{r_{o,x}}{r_w}\right) + s} \right];\; WI_y = \left[\frac{2\pi \sqrt{k_x\, k_z}\, L_y}{\ln\left(\dfrac{r_{o,y}}{r_w}\right) + s} \right];\; WI_z = \left[\frac{2\pi \sqrt{k_y\, k_x}\, L_z}{\ln\left(\dfrac{r_{o,z}}{r_w}\right) + s} \right] \tag{2.74}$$

where the directional equivalent radii are defined in terms of directional intrinsic permeability and grid-cell dimensions:

$$r_{o,x} = 0.28\left[\frac{\left(\left\{\frac{k_y}{k_z}\right\}^{1/2}\Delta z^2 + \left\{\frac{k_z}{k_y}\right\}^{1/2}\Delta y^2\right)^{1/2}}{\left(\left\{\frac{k_y}{k_z}\right\}^{1/4} + \left\{\frac{k_z}{k_y}\right\}^{1/4}\right)}\right]$$

$$r_{o,y} = 0.28\left[\frac{\left(\left\{\frac{k_x}{k_z}\right\}^{1/2}\Delta z^2 + \left\{\frac{k_z}{k_x}\right\}^{1/2}\Delta x^2\right)^{1/2}}{\left(\left\{\frac{k_x}{k_z}\right\}^{1/4} + \left\{\frac{k_z}{k_x}\right\}^{1/4}\right)}\right];\ WI = \sqrt{\left(WI_x\right)^2 + \left(WI_y\right)^2 + \left(WI_z\right)^2} \qquad (2.75)$$

$$r_{o,z} = 0.28\left[\frac{\left(\left\{\frac{k_x}{k_y}\right\}^{1/2}\Delta y^2 + \left\{\frac{k_y}{k_x}\right\}^{1/2}\Delta x^2\right)^{1/2}}{\left(\left\{\frac{k_x}{k_y}\right\}^{1/4} + \left\{\frac{k_y}{k_x}\right\}^{1/4}\right)}\right]$$

3.0 Numerical Solution

The STOMP-CO2 simulator solves three coupled mass conservation equations, expressed in partial differential form in Eqns. (2.3), (2.5) and (2.6). The STOMP-CO2e simulator solves an additional conservation of thermal energy equation, expressed in differential form in Eqn. (2.1). A general analytical solution to the coupled conservation equations and constitutive equations is not available. These equations are solved numerically, subject to initial conditions and boundary conditions. The conservation equations equate the change in the conserved quantity within a control volume over time with the net flow of the conserved quantity into the control volume across the control volume surfaces. The numerical solution is founded on the concept of discretizing both time and space. Temporal discretization occurs through time stepping; where the simulator solves the system of conservation and constitutive equations at some new point in time, starting with a known solution at an earlier point in time. For the first time step, initial conditions define the known solution for the starting time. Spatial discretization occurs through gridding; where the physical domain is divided into discrete grid volumes. STOMP-CO2 and STOMP-CO2e both use the integral finite difference spatial discretization scheme, which assumes that state properties for the entire grid volume can be represented by those values at the centroid of the grid volume.

3.1 Discretization and Linearization

To solve these conservation equations numerically, the equations are converted to algebraic form through spatial and temporal discretization using the integral finite difference approach on structured orthogonal grids and Euler-backward time differences. STOMP-CO2 and -CO2e use structured grids with 7-point stencils for three-dimensional domains, 5-point stencils for two-dimensional domains, and 3-point stencils for one-dimensional domains. Cartesian and cylindrical type structured grids can be specified with minimal input. The more complex boundary-fitted type structure grids are typically specified through hexahedron vertices. The algebraic equations that result from spatial and temporal discretization are closed through a series of constitutive equations, a partial list of which are shown in Eqns. (2.7) through (2.67). The closed system of equations is highly nonlinear, which makes direct solution impossible. Newton-Raphson iteration was used to resolve these nonlinearities, where a set of primary variables is selected for each grid cell. Primary unknowns for the conservation equations are chosen to be independent and able to completely define the system state, which implies the ability to compute the suite of secondary variables. With the possibility for phase appearances and disappearances, there does not exist a single set of primary variables that can be used to define the system state for every phase condition possibility.

3.1.1 Primary Variable Switching

To overcome this numerical difficulty, a primary variable switching scheme was developed that changes the primary variable set with phase conditions. To ensure smooth transitions across phases, the primary variable switching occurs between Newton-Raphson iterations within a single time step. Both STOMP-CO2 and -CO2e use three primary variable sets; where only the primary unknown for the CO_2 mass conservation equation switches between phase conditions, as shown in Table 3.1.

Table 3.1. Primary Variable Sets

Energy	H_2O	CO_2	Salt
Phase Condition #1		$\bar{\bar{s}}_l = 1;\ s_{gt} = 0$	
T	P_l	ω_l^{CO2}	ψ_b^s
Phase Condition #2		$\bar{\bar{s}}_l < 1;\ s_{gt} \geq 0$	
T	P_l	P_g	ψ_b^s
Phase Condition #3		$\bar{\bar{s}}_l = 1;\ s_{gt} > 0$	
T	P_l	s_{gt}	ψ_b^s

In all phase conditions, the aqueous dissolved salt can be under saturated, saturated, or over saturated. In Phase Condition #1, CO_2 is totally dissolved in the aqueous phase and the primary unknown for the CO_2 conservation equation is the aqueous CO_2 mass fraction. For the over-saturated salt condition, the aqueous salt concentration is fixed at the saturation level and the remaining salt is in precipitated form. In Phase Condition #2 free gas is present and a portion of the total gas phase can be trapped. The primary unknown for the CO_2 conservation equation is the gas pressure, and the aqueous CO_2 mass fraction is at the solubility limit. In Phase Condition #2, the aqueous phase can disappear, but only through extension to the capillary pressure - saturation function with the gas-aqueous capillary pressure reaching oven-dried conditions. In Phase Condition #3, all of the gas is trapped and the primary unknown is the trapped gas saturation. For Phase Conditions #1 and #3, the gas pressure is set to the sum of the aqueous pressure and the scaled gas entry pressure. In Phase Condition #3, the historical minimum effective aqueous saturation is computed as a function of the trapped gas saturation, according to Eqn. (2.49), using an effective aqueous saturation of 1.0.

Transitions between phase conditions occur between Newton-Raphson iterations. There is no limit to the number of phase transitions during a single time step. Transitions between Phase Condition #1 and #2 occur when the aqueous CO_2 concentration exceeds its aqueous solubility limit. Transitions between Phase Condition #1 and #3 are not permitted, as a fundamental assumption concerning gas entrapment is that it occurs during aqueous imbibition and a Phase Condition #1 to #3 transition indicates an aqueous drainage path. Transitions between Phase Condition #2 and #1 occur when the gas pressure becomes lower than the aqueous pressure plus the scaled gas entry pressure and the trapped gas saturation is zero. Transitions between Phase Condition #2 and #3 occur when the gas pressure becomes lower than the sum of the aqueous pressure and the scaled gas entry pressure, and the trapped gas saturation is greater than zero. Transitions between Phase Condition #3 and #1 occur when the trapped gas saturation goes to zero. Transitions between Phase Condition #3 and #2 occur when the trapped gas saturation exceeds the maximum trapped gas saturation. This scheme is equivalent to requiring the gas saturation to form a continuous phase across pore throats, before allowing the gas to become mobile.

3.1.2 Primary Variable Updating and Convergence

The Newton-Raphson iteration solution scheme yields a linear system of equations that relate the Jacobian matrix, the solution vector and the problem vector. The Jacobian matrix is comprised of partial derivatives of the residual (error) of a conservation equation at a particular grid cell (node) with respect to a primary variable for the node within the grid stencil. For STOMP-CO2 there are three primary variables per node and for STOMP-CO2e there are four primary variables per node. Therefore the Jacobian matrix is order 3 x nodes or 4 x nodes for STOMP-CO2 or STOMP-CO2e, respectively. The solution vector comprises corrections to the primary variables at each node, and the problem vector contains the negative of the conservation equation residuals for the current iterate values of the primary variables. The STOMP simulator can be configured to call a direct banded or indirect conjugate gradient linear system solver. The direct banded solver is numerically efficient for smaller problems; generally those involving less than 10,000 unknowns. The conjugate gradient solver is required for larger problems because of its computational efficiency and memory requirements. The linear system solver returns the solution vector corrections to the primary variables for every active node. The computed corrections are based on the state of the residual and the gradient of the residual with respect to the primary variables. The residual functional forms are complex with respect to the primary variables and for conditions remote from the roots (i.e., primary variable solution that yield zero residuals) the corrections returned from the linear system solver can be excessive. Whereas relaxation is a convenient scheme for dampening corrections to primary variables for nonlinear solution systems, the preferred approach in the STOMP-CO2 and -CO2e simulators is to limit primary variable corrections.

Each time step involves a number of Newton-Raphson iterations, where the initial guess to the primary variables are the converged primary variables at the conclusion of the previous time step. At the start of each iteration, the primary variables are used to determine the phase condition for every active node. The partial derivatives in the Jacobian matrix are computed numerically in STOMP-CO2 and -CO2e, which requires setting an increment for each primary variable. Once the phase conditions are established and the primary variable increments are set, the constitutive equations are solved to determine the secondary variables. The secondary variables are then used to determine conservation equation residuals and residual partial derivatives, which comprise the Jacobian matrix. The linear system solver then returns corrections to the primary variables. Before continuing on to the next iteration, the primary variables are updated with the corrections and a convergence check is conducted. STOMP-CO2 and -CO2e use a global convergence requirement, which means that convergence requirements must be met for all primary variables in every active node. The metric for convergence is that either the residual falls below a specified fraction of the total conserved quantity in the node, or the correction to the primary variable falls below a specified fraction of a reference value. Converged solutions proceed to a new time step, after reporting any requested results. Unconverged solutions proceed to a new iteration or result in a convergence failure. Convergence failures occur when the number of Newton-Raphson iterations exceed a specified value. When this occurs, the time step is cut by a specified fraction, and the time step is restarted. If the time step falls below a specified value from a sequence of time-step cuts, then the simulation stops.

3.2 Algorithm Structure and Flow Path

The algorithmic structure for the core of the STOMP-CO2 and -CO2e simulators is based on four components: 1) initialization, 2) time-stepping loop, 3) Newton-Raphson iteration loop, and 4) closure. During initialization, the input file is read twice. The first input file read is used to determine memory requirement and to allocate memory to the global arrays. The second input read is used to define the problem. After reading the input file the second time, the initial conditions are checked for errors and initial phase conditions are set. With the initial phase conditions and primary variables set, all secondary variables are computed. Next the Jacobian matrix structure is defined, including consideration of bandwidth, coupled wells, and vertical equilibrium domains. Before starting a new time step, all surface fluxes are computed.

At the start of a new time step, the time-step quantity is determined, user-requested output is recorded, and old time-step information is stored. The next series of calculations are preparatory for building the Jacobian matrix. Boundary condition properties are computed. Source/sink contributions are determined. Internal surface and boundary surface fluxes are computed. The Jacobian matrix and problem vector are then loaded in conservation equation sequence: 1) thermal energy (STOMP-CO2e only), 2) water mass, 3) CO_2 mass, and 4) salt mass, assuming no-flow adiabatic conditions for all boundary surfaces. Next, the Jacobian matrix and problem vector are modified for boundary conditions applied to boundary surfaces. The Jacobian matrix is then further modified for vertical equilibrium domains and coupled wells. Once constructed, the Jacobian matrix and problem vector are submitted to the linear system solver, which returns the solution vector of primary variable corrections. The primary variable corrections are then used to update the primary variables and check for convergence.

If convergence is not achieved and the iteration count is less than or equal to the specified limit, then the updated primary variables are used to set the phase condition, assign primary variable increments, and compute the secondary variables. The number of Newton-Raphson iterations is incremented and new Newton-Raphson iteration is started. If convergence is not achieved and the iteration count is greater than the specified limit, then the primary variables and phase conditions are reset to their old-time step values. Secondary variables are computed, the time-step is reduced by a user specified amount, and a new time step is started. If the reduced time-step quantity is below a user specified value, then the simulation halts, creating a restart file and recording user-specified output. If convergence is achieved, the updated primary variables are used to set phase conditions, assign primary variable increments, compute secondary variables, and compute interior and boundary surface fluxes for use in nonreactive and reactive transport. A new time step quantity is determined, user-requested output is recorded, and a new time-step loop is started. The simulation continues in this manner until the user-specified maximum number of time steps or user-specified simulation time limit is reached.

4.0 Input File

4.1 Introduction

STOMP-CO2 and -CO2e inputs can vary from a single file to a series of files. The principal input file is simply named “input.” All other input files, whose names are user defined, are called from the input file. The input file is a text file which is divided into cards, lines, and fields. Being a text file it can be created or modified with any text editor. The card designation is in honor of punched cards that have been programming machines since their use in the 1720s to control textile looms in France and computers until they became obsolete in the mid-1980s. Cards are named and identified with a “~” prior to the card name. Card names without the preceding “~” are not interpreted by the simulator. Input lines for each card are located immediately below the card title line. Blank lines within a card structure are not permitted, but any line with a “#” or “!” as the first character is considered to be a comment line. Lines are divided into fields by commas. A closing comma is always required to identify the end of the field. All leading and trailing spaces in a field input are ignored. Field inputs can be one of three types: 1) character strings, 2) real numbers, or 3) integers. Character strings are used to name entities, select options, or specify units. For example, named entities are rock/soils, solutes, or reactive species. Options are user selected choices, such as model formulations, solution schemes, initial condition variables, boundary condition types, or output requests. Units represent the units associated with dimensional variables. Internally, STOMP uses the International System of Units (SI); where, the base units are length in meters (m), time in seconds (s), mass in kilograms (kg), temperature in degrees Celsius (˚C), and molar mass in kilomole (kmol). Recognized units are shown in Table 4.1. Unit inputs can be any combination of the recognized units; where spaces between units imply multiplication of units and only a single division symbol can be used within the character string. The “^” is used to represent exponential notation in the input file (e.g., m^2 = m^2). Real-number inputs are non-integer number inputs. Real numbers can include a decimal point or scientific notation. Integer-number inputs cannot contain a decimal point nor be expressed in scientific notation.

Table 4.1. Input Units

Unit	Description	SI Conversion	Base Units
1	One	1.e+0	N/A
a	Angstrom	1.e-10	m
ang	Angstrom	1.e-10	m
angstrom	Angstrom	1.e-10	m
ao	Angstrom	1.e-10	m
aqu	Aqueous	1.e+0	N/A
aqueous	Aqueous	1.e+0	N/A
atm	Atmosphere	1.01325e+5	kg/m s^2
bar	Bar	1.e+5	kg/m s^2
btu	BTU	1.0544e+3	kg m^2/s
c	˚Celsius	1.e+0	˚C
cal	Calories	4.184e+0	kg m^2/s
ci	Curies	1.e+0	N/A
cm	Cm	1.e-2	m
cp	Centipoise	1.e-3	kg/m s
cst	centiStokes	1.e-6	m^2/s
d	Day	8.64e+4	s
darcy	Darcy	0.9869e-12	m^2

day	Day	8.64e+4	s
debyes	Debyes	1.e+0	(kg/m s^{2})$^{(1/2)}$
deg	degrees	1.745329252e-2	rad
degree	degrees	1.745329252e-2	rad
dyn	dynes	1.e-5	kg m/s^{2}
dynes	dynes	1.e-5	kg m/s^{2}
f	˚Fahrenheit	5.555556e-1	˚C
ffa	football field area	5351.215104e+0	m^{2}
ffl	football field length	109.728e+0	m
ffvh_cm	volume football field area x cm	53.51215104e+0	m^{3}
ffvh_ft	volume football field area x foot	1631.05037e+0	m^{3}
ffvh_in	volume football field area x inch	135.920864e+0	m^{3}
ffvh_m	volume football field area x meter	5351.215104e+0	m^{3}
ffvh_yd	volume football field area x yard	4893.1511e+0	m^{3}
ft	feet	3.048e-1	m
furlong	furlong	2.01168e+2	m
g	gram	1.e-3	kg
gal	gallon	3.7854e-3	m^{3}
gas	gas	1.e+0	N/A
gm	gram	1.e-3	kg
gpa	gigaPascal	1.e+9	kg/m s^{2}
gram	gram	1.e-3	kg
hc	hydraulic conductivity	1.039102952D-07	N/A
hour	hour	3.6e+3	s
hp	horse power	7.457e+2	kg m^{2}/s^{3}
hr	hour	3.6e+3	s
in	inch	2.54e-2	l
j	Joule	1.e+0	kg m^{2}/s^{2}
joule	Joule	1.e+0	kg m^{2}/s^{2}
k	˚Kelvin	1.e+0	˚C
kg	kilogram	1.e+0	kg
kgmol	kilomole	1.D+0	kmol
kj	kiloJoule	1.D+3	kg m^{2}/s^{2}
kjoule	kiloJoule	1.D+3	kg m^{2}/s^{2}
kmol	kilomole	1.D+0	kmol
kmole	kilomole	1.D+0	kmol
kpa	kiloPascal	1.D+3	kg/m s^{2}
l	liter	1.D-3	m^{3}
langley	Langley	4.186D+4	kg/s^{2}
lb	pound	4.5359D-1	kg
lbf	poundforce	4.4482D+0	kg m/s^{2}
lbm	poundmass	4.5359e-1	kg

lbmol	poundmole	4.5359e-1	kmol
liq	liquid	1.e+0	N/A
liter	liter	1.e-3	m^3
m	meter	1.e+0	m
md	milliDarcy	0.9869e-15	m^2
mdarcy	milliDarcy	0.9869e-15	m^2
mg	milligram	1.e-6	kg
mi	mile	1.609344e+3	m
mile	mile	1.609344e+3	m
min	minute	6.e+1	s
ml	milliliter	1.e-6	m^3
mm	millimeter	1.e-3	m
mmt	million metric tonne	1.e+9	kg
mol	mole	1.e-3	kmol
mole	mole	1.e-3	kmol
mpa	megaPascal	1.e+6	kg/m s^2
mt	metric tonne	1.e+3	kg
n	Newton	1.e+0	kg m/s^2
napl	NAPL	1.e+0	N/A
newton	Newton	1.e+0	kg m/s^2
oil	oil	1.e+0	N/A
p	Poise	1.e-1	m/kg s
pa	Pascal	1.e+0	kg/m s^2
pci	picoCuries	1.e+0	N/A
plant	plant	1.e+0	N/A
psf	pounds per square foot	4.7880556e+1	kg/m s^2
psi	pounds per square inch	6.8948e+3	kg/m s^2
r	˚Rankine	5.555556e-1	˚C
rad	radian	1.e+0	rad
rad	radian	1.e+0	rad
radian	radian	1.e+0	rad
rod	rod	5.0292e+0	m
s	second	1.e+0	s
sec	second	1.e+0	s
slug	slug	1.4594e+1	kg
sol	solute	1.e+0	N/A
solid	solid	1.e+0	N/A
st	Stokes	1.e-4	m^2/s
ton	ton	9.0718494e+2	kg
tonne	metric tonne	1.e+3	kg
voc	VOC	1.e+0	N/A
w	Watt	1.e+0	kg^2 m/s^3
water	water	1.e+0	N/A
week	week	6.048e+5	s
wh	water head	9.7935332e+03	m/kg^2 s^2
wk	week	6.048e+5	s
yd	yard	9.144e-1	m

year	year	3.15576e+7	s
yr	year	3.15576e+7	s

4.2 Card Descriptions

The following sections provide a brief synopsis of the input cards recognized by STOMP-CO2 and STOMP-CO2e. Detailed formatting instructions for the input cards is described in Appendix B.

4.2.1 Simulation Title Card

This card primarily provides a means to document a simulation. Information recorded in this card is rewritten in the *output* file, which then serves as a permanent record of the simulation. The user is encouraged to use descriptive titles and to briefly describe the 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.

4.2.2 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 for 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 for selected parameters, declared through the *Initial Conditions Card*.

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,stomp-w*, 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 simulation. 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*.

Solute transport is specified by including the keyword *Transport* in the *Execution Mode* field. By default, solute transport is solved using the Patankar method. If the keyword *TVD* appears in the *Execution Mode* field, then the TVD scheme is used for transport. If the keywords *Roe Superbee* appear in the *Execution Mode* field, then the Roe Superbee scheme is used for transport. Reactive transport is specified by including the keyword *ECKEChem* in the *Execution Mode* field. The reactive transport algorithms use the same transport schemes as the solute transport model, and therefore are controlled through the keyword options *TVD* and *Roe Superbee*. The *No Flow* option, used in conjunction with the *Normal* or *Restart* execution modes, results in the coupled flow and transport equations only being computed once. This option can be used to eliminate the flow calculations each time step for a reactive transport problem with a steady flow field.

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.

Two additional solution controls can be specified for each *Execution Period* input line: 1) the minimum time step, and 2) the time-step cut factor. These optional solution controls must be specified together. The minimum-time-step parameter sets the minimum time step for an execution period. Without this control STOMP-CO2 and -CO2e quit after four time-step reductions. Cyclic injection well schedules can yield simulations that execute with large time steps during steady injection periods, but require small time steps during the injection startup. The minimum-time-step parameter allows the code to cut the time step to the minimum value before quitting. The time-step cut factor is set to 0.2 by default, which means that time steps are cut to 20% of their value when a convergence failure occurs. Execution performance can be improved for some simulations by altering this factor.

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 4.2. 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 4.2. Default Interfacial Averaging Options

Field Variable	Interfacial Averaging Scheme
Aqueous Diffusion	Harmonic
Gas Diffusion	Harmonic
Aqueous Density	Upwind
Gas Density	Upwind
Aqueous Viscosity	Harmonic
Gas Viscosity	Harmonic
Aqueous Relative Permeability	Upwind
Gas Relative Permeability	Upwind
Aqueous Enthalpy	Upwind
Gas Enthalpy	Upwind
Effective Thermal Conductivity	Harmonic
Intrinsic Permeability	Harmonic
Porosity	Harmonic

4.2.3 Grid Card

The STOMP simulator's integral 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 an external grid file. Four different types of external grid files are recognized: 1) vertices, 2) node vertices, 3) Eclipse, and 4) EarthVision. The vertices file assumes that adjacent nodes have co-located vertices, requiring only ((*nx*+1)*(*ny*+1)*(*nz*+1)) vertex inputs. The node vertices file requires the specification of eight vertices for every node, (i.e., (nx*ny*nz)*8).

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 toward increasing node numbers. The grid dimensions that are specified on the *Grid Card* refer to node surfaces; therefore, for grids with non-uniform spacing, for each grid direction the number of entries required is one plus the number of nodes. Node volumes are defined by their bounding surfaces. Cylindrical coordinate 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.

4.2.4 Internal Boundary Surfaces Card

This card is used to define the location of internal boundary surfaces. Internal boundary surfaces sever the internal connections between adjacent nodes, making the fluxes between these nodes controlled by the applied boundary conditions and not the gradients in temperature, pressure, or concentration between the nodes. By default, internal boundary surfaces have a *Zero Flux* type boundary. Therefore, an internal boundary surface without an associated boundary condition is an impermeable interior surface. Unique boundary conditions can be applied to either side of an internal boundary surface. Internal boundary surfaces are specified 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. Internal boundary surfaces can be specified directly via this card or by using an external file. If a split, or discontinuity in the interior of the grid occurs, then internal boundary surfaces are implicitly defined and do not need to be explicitly defined in the *Internal Boundary Surfaces Card*.

4.2.5 Inactive Nodes Card

This card is used to declare inactive nodes within the computational domain. 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).

4.2.6 Rock/Soil Zonation Card

In the STOMP simulator, hydrologic flow and transport properties are associated with rock/soil types; where rock/soil types are user defined names. 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, are case insensitive, and contain no more than 64 characters. The following key words have special meanings when used within a rock/soil name: '*dp*, '*dual porosity*,' '*fractured*;' where, '*dp*,' '*dual porosity*,' and '*fractured*' are 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 cell 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., via 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*, *Gas Relative Permeability Card, Solute/Porous Media Card*, *Salt Transport Card*, *Geomechanical Link* Card, *Geomechanical Properties Card*, *Lithology Card*, and *Solid Species Card.*

4.2.7 Vertical Equilibrium Card

The vertical equilibrium assumption essentially converts three-dimensional domains into two-dimensional domains, or two-dimensional domains into one-dimensional domains by assuming static equilibrium of the gas and aqueous phases in the vertical direction. For STOMP-CO2e, the vertical equilibrium assumption further assumes a single temperature in the vertical direction across the vertical equilibrium domain. The vertical equilibrium assumption can be applied to the entire computational domain or across portions of the domain. The *Vertical Equilibrium Card* is used define regions of the computational domain to be under the vertical equilibrium assumption. Computational regions outside of the vertical equilibrium domains use the conventional STOMP-CO2 and -CO2e solution approaches.

4.2.8 Mechanical Properties Card

This card allows the user to assign values to the particle density, porosity, 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. Compressibility is specified via three different approaches: 1) *Specific Storativity*, 2) *Bulk Compressibility*, or 3) *Pore Compressibility*. If the *Specific Storativity* is specified it is converted to bulk compressibility, using the specified *Diffusive Porosity* and reference values for water density and compressibility. A default *Specific Storativity* of 10^{-7} Pa^{-1} is assumed for *null* entries. Bulk and pore compressibilities can be specified with or without a *Reference Pressure*. Without a *Reference Pressure* specification, the initial pressure is used as the *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). 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 (Section 2.7.2). 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*).

4.2.9 Hydraulic Properties Card

This card is used 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 requires two parameters, which are entered via this card: *Pore-Body Fractional Length* and *Fractional Critical Porosity* (Section 2.7.1).

4.2.10 Thermal Properties Card

This card allows the user to assign values to the thermal conductivity and specific heat 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 (i.e., STOMP-CO2e). A collection of thermal conductivity models are available, as defined through the *Thermal Conductivity Function Option* (Section 2.7.3). The *Constant* option fixes the thermal conductivity to a constant value, independent of temperature or saturation. The Parallel option invokes the DeVries (1966) model, which computes effective thermal conductivity using a phase volumetric weighting scheme from the rock thermal conductivity, phase thermal conductivities (Sections 2.4.3 and 2.4.4), and phase saturations. The Linear model computes the effective thermal conductivity as the aqueous-saturation scaled average between the unsaturated and saturated effective thermal conductivities. The Somerton et al. (1973, 1974) model computes the effective thermal conductivity as the square root of the aqueous-saturation scaled average between the unsaturated and saturated effective thermal conductivities.

4.2.11 Saturation Function Card

Saturation functions relate the gas-aqueous capillary pressure to aqueous, gas, and entrapped gas saturations (Section 2.5). Model options and parameters for these functions are specified through the *Saturation Function 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. Functional forms for the saturation-capillary pressure functions are preferred; however, 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. Functional models in STOMP-CO2 and -CO2e include the van Genuchten model (1980), the Brooks and Corey model (1966), and the Haverkamp model (1977). These van Genuchten and Brooks and Corey models can be used in association with the dual-porosity model (Klavetter and Peters 1986; Nitao 1988). The van Genuchten and Brooks and Corey models can also be implemented with gas entrapment by using the keyword, *Entrapment*. The entrapment model requires one additional input parameter *Actual Maximum Trapped Gas Saturation* (Section 2.5.2). To extend the saturation function below the aqueous residual saturation, functional extensions are required. STOMP-CO2 and -CO2e recognizes two extension functions: Fayer and Simmons (1995) and Webb (2000) (Section 2.5.3). Both of these extension functions work with both the van Genuchten and Brooks and Corey characteristics functions, without requiring additional input as the oven-dried head is assumed to be equal to a capillary pressure of 10^9 Pa (~10^5 m).

4.2.12 Aqueous Relative Permeability Card

The aqueous relative permeability functions relate phase saturations with aqueous relative permeability (Section 2.6.1). The *Aqueous Relative Permeability Card* is used to specify model options and parameters. 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 Mualem and Burdine aqueous relative permeability functions are also dependent on the saturation function type and are strictly applicable to the van Genuchten (1980) and Brooks and Corey (1966) 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 aqueous relative permeability-saturation functions are preferred, but 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.

4.2.13 Gas Relative Permeability Card

The gas relative permeability functions relate phase saturations with gas relative permeability (Section 2.6.2). The *Gas Relative Permeability Card* is used to specify model options and parameters. 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 Mualem and Burdine gas relative permeability functions are also dependent on the saturation function type and are strictly applicable to the van Genuchten (1980) and Brooks and Corey (1966) 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 gas relative permeability-saturation functions are preferred, but 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.

4.2.14 Solute/Fluid Interaction Card

Solutes are nonreactive passive tracers that are transported via advection and diffusion through the aqueous and gas phases, with equilibrium partitioning between the gas, aqueous and solid phases. Being passive tracers the phase properties are assumed to be independent of solute concentration. Solute concentrations in STOMP-CO2 and STOMP-CO2e are unspecified in terms of the solute mass, which means solute concentrations are specified only with respect to the volumetric component (e.g., 1/m^3, 1/ml). The solute mass in the numerator is implied (e.g., Curies, gm, kmol). In developing an input file, it is critical that the user keep the mass component of the solute units consistent. The *Solute/Fluid Interactions Card* is used to define solute names, phase diffusion coefficients, gas-aqueous partitioning and solute radioactive decay. 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 cannot contain 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.

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 partition coefficients define the equilibrium distribution of solute between the aqueous and gas phases. Partition coefficients that define the equilibrium adsorption of solute onto the solid phase are defined on the *Solute/Porous Media Interaction Card*. 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. 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.

4.2.15 Solute/Porous Media Interaction Card

A principal assumption concerning solute transport in STOMP-CO2 and STOMP-CO2e is that the aqueous phase is the wetting fluid. This implies a gas-aqueous interface and an aqueous-solid interface. Aqueous-gas partitioning is specified via the *Solute/Fluid Interaction Card*. This card is used 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.

4.2.16 Salt Transport Card

Aqueous diffusion coefficients for salt in STOMP-CO2 and -CO2e are computed internally (Section 2.4.5). This input card is not required for STOMP-CO2 and -CO2e simulations. It is only used to define geologic media dependent dispersivities for salt. 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. *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.

4.2.17 Coupled Well Card

Wells specified with the *Coupled Well Card* inject fluids into the formation under a specified mass injection rate, subject to a pressure limit. When the mass injection rate can be met without exceeding the specified pressure limit (e.g., fracture pressure limit), then the well is considered to be flow controlled and the wellhead pressure becomes the unknown. When the mass injection rate cannot be met without exceeding the specified pressure limit, then the well is considered to be pressure controlled and the mass injection rate becomes the unknown. Numerical convergence of the nonlinear conservation and constitutive equations is greatly enhanced by fully integrating the well equations into the reservoir field equations. Hence, this card specifies a well model whose solution is fully integrated into the reservoir field equations in STOMP-CO2 and STOMP-CO2e. A projection modification to the Peaceman well index formulation allows for an arbitrary well trajectory and a grid-free well trajectory specification simplifies grid convergence studies (see Section 2.8). Wells specified via the *Coupled Well Card* are specified using straight-line well intervals; where only the screened intervals are specified. Injection wells should be specified starting nearest the wellhead. The well interval points are specified according to their *x*, *y*, and *z* coordinate points in space with respect to the specified grid system. Points defining a well interval can be located outside or inside the computational domain, but only the portion of the well within the computational domain is included in the well model.

Because a series of well interval point pairs are used to specify the well trajectory and the points are specified according to their *x*, *y*, and *z* coordinate points in space, the same well trajectory specification can be used for a variety of computational grids. The coupled well model is connected to the computational domain by coupling well nodes with field nodes. These connections are determined during the initialization stages of a simulation. For dynamic memory simulations, the well connection procedure is executed twice; once to determine memory requirements for the well and Jacobian matrix; and a second time to define well nodes and their connections to field nodes. Each well interval can have a unique radius and skin factor. The injection schedule can vary over time and can include double time entries to handle sharp transitions in the well schedule. Linear interpolation is used between the injection schedule times, as with the schedule specifications for boundary conditions and source/sinks.

4.2.18 Initial Conditions Card

The *Initial Conditions Card* is used to assign starting values to primary variables. The first input line of this card requires that the *Initial Saturation Option* be specified where the choices are 1) *aqueous pressure* and *gas pressure*, 2) *aqueous pressure* and *aqueous saturation*, or 3) *gas pressure* and *aqueous saturation*. The specified option applies to the entire computational domain. If the *aqueous pressure* and *gas pressure* option is chosen, then aqueous pressure and gas pressure must be specified everywhere in the domain, and the initial phase saturations will be determined from the characteristic functions. If *aqueous pressure* and *aqueous saturation* is chosen, then the aqueous pressure and aqueous saturation must be specified everywhere in the domain and the initial gas pressure will be determined by solving the characteristic functions inversely. Otherwise, if *gas pressure* and *aqueous saturation* is chosen, then the gas pressure and aqueous saturation must be specified everywhere in the domain and the initial aqueous pressure will be determined by solving the characteristic functions inversely.

Variables that can be initialized through the *Initial Conditions Card* are shown in the list below:

- *aqueous pressure*
- *gas pressure*
- *aqueous saturation*
- *temperature*
- *actual trapped gas saturation*

- *relative trapped gas saturation*
- *CO_2 relative saturation*
- *CO_2 mass fraction*
- *CO_2 aqueous concentration*
- *salt relative saturation*
- *salt mass fraction*
- *salt aqueous concentration*
- *solute concentration*
- *species gas molar concentration*
- *species aqueous molality*
- *species aqueous molar concentration*
- *species volumetric molar concentration*

Variables that are not specified are assigned default values, as shown in Table 4.3. Initial condition inputs are converted to primary variable values through flash calculations. The flash calculation routines additionally have checks for inconsistent initial conditions specifications. For example if saturated aqueous conditions were specified through the initial *gas pressure* and *aqueous pressure* and the initial *CO_2 aqueous concentration* and *salt aqueous concentration* were specified with values that exceeded the solubility limit for CO_2 in the aqueous phase, then the code would halt with an input error message.

Table 4.3. Initial Condition Defaults

Variable	Symbol	Default
Temperature	T	20.0 ˚C
Pressure of phase γ	P_γ	101325.0 Pa
Saturation of phase γ	s_γ	0.0
Mass Fraction in phase γ	$\omega_\gamma^{H2O}, \omega_\gamma^{CO2}, \omega_\gamma^{s}$	0.0
Mole Fraction in phase γ	$\chi_\gamma^{H2O}, \chi_\gamma^{CO2}, \chi_\gamma^{s}$	0.0
Solute concentration in phase γ	C_γ	0.0 1/m^3
Species concentration in phase γ	C_γ	0.0 1/m^3
Aqueous salt concentration	ρ_l^s	0.0 kg/m^3

The gradient utilities of the initial condition card are invaluable and should be used when possible. Gradients to the initial conditions allow the user to specify that the initial value of field variables will vary 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 that 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.

Nonreactive species (i.e., solutes) are initialized in terms of solute per unit volume, where the volume can refer to the total node volume, the aqueous phase volume or the gas phase volume. 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. Reactive species (i.e., species) are initialized in terms of molar concentration or aqueous molality. 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/$m^3_{aqueous}$). For more complete descriptions on the initialization of reactive species, refer to the ECKEChem addendum (White and McGrail, 2005).

4.2.19 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. To apply a boundary condition to an internal boundary surface, defined via the *Internal Boundary Surfaces Card*, 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 that the user has not created an ill-posed problem or that the simulation 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.

For STOMP-CO2 three separate boundary types must be declared for each boundary surface: 1) aqueous, 2) gas, and 3) salt. For STOMP-CO2e four separate boundary types must be declared for each boundary surface: 1) energy, 2) aqueous, 3) gas, and 4) salt. The boundary types are shown in Table 4.4. Solutes and reactive species require additional boundary type inputs. The boundary types for solutes and reactive species are shown in Table 4.5.

Table 4.4. Boundary Types

Energy	Aqueous	Gas	Salt
Dirichlet	Dirichlet	Dirichlet	Inflow Aqu. Molality
Neumann	Neumann	Neumann	Inflow Aqu. Conc.
Zero Flux	Zero Flux	Zero Flux	Inflow Aqu. Rel. Sat.
Outflow	Hydraulic Gradient	Hydraulic Gradient	Inflow Aqu. Mass Frac.
Initial Condition	Initial Condition	Initial Condition	Aqu. Molality
			Aqu. Conc.
			Aqu. Mass Frac.
			Zero Flux
			Outflow
			Initial Condition

Table 4.5. Solute and Reactive Species Boundary Types

Solute	Aqueous Species	Gas Species
Inflow-Outflow Aqu.	Inflow-Outflow	Inflow-Outflow
Inflow-Outflow Gas	Inflow	Inflow
Inflow-Outflow	Outflow	Outflow
Inflow Aqu.	Concentration	Concentration
Inflow Gas	Zero Flux	Zero Flux
Inflow		
Outflow		
Volumetric Conc.		
Aqu. Conc.		
Gas Conc.		
Zero Flux		
Initial Condition		

The *Dirichlet* boundary type is used to specify a field value (e.g., temperature or pressure) at the boundary surface. The *Neumann* boundary type allows the user to specify a flux (e.g., heat flux or volumetric phase flux) at the boundary surface. The *Zero Flux* boundary type is used to impose no flow and/or transport conditions across the boundary. 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 *Initial Conditions* boundary type fixes the boundary field variables (e.g., temperature, pressure, aqueous salt mass fraction) 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 salt boundary types *Aqu. Molality*, *Aqu. Conc.*, and *Aqu. Mass Frac.* are *Dirichlet* type boundaries where the user has options for specifying the aqueous salt concentration. The salt boundary types *Inflow Aqu. Molality*, *Inflow Aqu. Conc.*, and *Inflow Aqu. Mass Frac.* are *Inflow* type boundaries where the user has options for specifying the aqueous salt 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.

For nonreactive species (i.e., solutes) a unique boundary type must be declared for each solute. For reactive species (i.e., species) only two boundary types are declared for all species; one for aqueous species and one for gas species. Chemical reaction systems generally involve large numbers of reactive species, which has necessitated some formatting changes for this card. There are two departures from the conventional formatting for the reactive species: 1) not all species concentrations need to be specified (typically only primary species concentrations are specified), and 2) line returns are permitted within a boundary condition input line. 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. For more complete descriptions on the boundary condition types and input for reactive species, refer to the ECKEChem addendum (White and McGrail, 2005).

4.2.20 Source Card

This card allows the user to control sources and/or sinks of mass, energy, or solutes by defining time-varying sources. By definition, sinks are negative sources, and sources refer to an influx of mass, energy, or solutes 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. 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. Source type options for STOMP-CO2 and –CO2e are shown in Table 4.6.

Table 4.6. Source Types

Energy	Aqueous		Gas	Salt	Solute
Power	Volume Rate		Volume Rate	Density Rate	Density Rate
Power Density	Salt	CO_2	Water	Mass Rate	Rate
	Conc.	Conc.	Rel. Humid.		
	Rel. Sat.	Rel. Sat.	Mass Frac.		
	Mass Frac.	Mass Frac.	No Water		
	Molality	No CO_2	Mass Rate		
	Mass Rate		Water		
	Salt	CO_2	Rel. Humid.		
	Conc.	Conc.	Mass Frac.		
	Rel. Sat.	Rel. Sat.	No Water		
	Mass Frac.	Mass Frac.			
	Molality	No CO_2			

Energy sources can be declared in terms of power or power per unit node volume (i.e., *Power Density*). Aqueous sources can be declared in terms of volumetric or mass rates. Aqueous sources additionally require the specification of the dissolved concentrations of CO_2 and salt. Specified concentrations are verified via the flash routines. Gas sources can be declared in terms of volumetric or mass rates. Gas sources additionally require the specification of the water-vapor concentration. Salt sources can be associated with an aqueous source or specified directly in terms of a mass rate or mass rate per unit node volume (i.e., *Density Rate*). Solute sources are only specified directly in terms of a mass rate or mass rate per unit node volume. Aqueous or gas sinks, however, remove solutes according to the nodal concentration. Sources can vary over time between sources and sinks. Aqueous and gas sinks removed dissolved components (e.g., CO_2, salt, water vapor) according to nodal concentrations.

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., $mol/m^3{}_{grid\ volume}$ 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. For more complete descriptions on the specification of reactive species sources, refer to the ECKEChem addendum (White and McGrail, 2005).

4.2.21 Output Control Card

This card allows the user to control output written to the *output* file, *plot.n* 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 may be 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.

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 that 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 to 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 that 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. 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. Variables that can be specified for reference node output are shown in Table 4.7.

Table 4.7 Reference Node Output Options

apparent aqueous saturation	integrated precip. salt mass	vert. int . CO_2 mass area
aqueous CO_2 concentration	integrated aqueous CO_2 mass	vert. int. gas CO_2 mass
aqueous CO_2 diffusion coeff.	integrated aqueous water	vert. int. gas CO_2 mass area
aqueous CO_2 fraction mass	integrated CO_2 mass	water mass source rate
aqueous CO_2 fraction mole	integrated CO_2 mass source	water source int mass
aqueous Courant number	integrated gas CO_2 mass	water total mass
aqueous density	integrated gas water mass	water vapor part press
aqueous enthalpy	integrated trapped CO_2 mass	well mass CO_2 integral
aqueous fracture saturation	integrated water mass	well mass CO_2 rate
aqueous gauge pressure	integrated water mass src.	well mass nodal CO_2 rate
aqueous hydraulic head	mean effective stress	well mass nodal water rate
aqueous internal energy	node number	well mass water integral
aqueous matrix saturation	phase condition	well mass water rate

aqueous moisture content	rock/soil type	well pressure
aqueous pressure	salt aqueous fraction mole	well-node pressure
aqueous relative perm.	salt mass source integral	x aqueous volumetric flux
aqueous salt concentration	salt mass source rate	x displacement
aqueous salt mass fraction	salt saturation	x gas vol volumetric flux
aqueous saturation	salt volumetric conc.	x intrinsic perm.
aqueous thermal conduc.	similarity variable	x normal strain
aqueous viscosity	solute aqueous conc.	x salt flux
aqueous water conc.	solute aqueous mole fraction	x solute flux
aqueous water fraction mass	solute gas conc.	xnc aqueous volumetric flux
CO_2 mass source rate	solute gas mole fraction	xnc gas volumetric flux
CO_2 source integrated mass	solute integrated mass	xnc salt flux
CO_2 total mass	solute source	y aqueous volumetric flux
diffusive porosity	solute volumetric conc.	y displacement
effective trap gas saturation	species aqueous conc.	y gas volumetric flux
gas CO_2 concentration	species gas conc.	y intrinsic perm.
gas CO_2 mass fraction	species integrated mass.	y normal strain
gas Courant number	species source.	y salt flux
gas density	species volumetric conc.	y solute flux
gas enthalpy	temperature	ync aqueous volumetric flux
gas fracture saturation	total salt mass	ync gas volumetric flux
gas gauge pressure	trapped gas saturation	ync salt flux
gas hydraulic head	vert. equil. aqu. press.	z aqueous volumetric flux
gas internal energy	vert. equil. aqu. rel. perm.	z displacement
gas matrix saturation	vert. equil. aqu. sat.	z gas volumetric flux
gas pressure	vert. equil. gas press.	z intrinsic perm.
gas relative perm.	vert. equil. gas rel. perm.	z normal strain
gas saturation	vert. equil. gas sat.	z salt flux
gas thermal conduc.	vert. equil. inter eleva	z solute flux
gas viscosity	vert. equil. trap gas sat	znc aqueous volumetric flux
gas water concentration	vert. int. aqu CO_2 mass	znc gas volumetric flux
gas water diffusion coeff.	vert. int. aqu. CO_2 mass area	znc salt flux
gas-aqueous scaling	vert. int. CO_2 mass	

Plot files are written at the conclusion of an execution, by default, and at each requested *Plot File Output Time*. 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 both operational modes 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.

4.2.22 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. Variables that can be specified for surface flux output are shown in Table 4.8.

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.

Table 4.8 Surface Flux Output Options

aqueous CO_2 mass flux	gas volumetric flux
aqueous mass flux	heat flux
aqueous volumetric flux	salt mass flux
CO_2 mass flux	solute flux
gas CO_2 mass flux	conservation comp. species flux
gas mass flux	kinetic comp. species flux

4.2.23 Geomechanics Properties Card

This card is only required for simulations that include geomechanical calculations using the EPMech (Elastic-Plastic Geomechanics) module. The *Geomechanics Properties Card* allows the user to assign values to the *Young's Modulus*, *Poisson's Ratio*, *Biot Coefficient*, and *Thermal Coefficient of Expansion* 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.

4.2.24 Geomechanics Link Card

This card is only required for simulations that include geomechanical calculations using the EPMech (Elastic-Plastic Geomechanics) module. The *Geomechanics Link Card* allows the user to assign function forms and functional parameters that define the linkage between the geomechanical simulation and the coupled flow and transport simulation. These linkages are specified through three relationships: 1) porosity and mean effective stress, 2) intrinsic permeability and porosity, and 3) capillary pressure and intrinsic permeability or porosity, which must be declared 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. The recognized *Porosity-Mean Stress Function* types are *Davis* or *none*. The *Davis* function requires the specification of two parameters: 1) *Davis-Davis Porosity-Mean Stress Function Exponent*, and 2) *Residual Porosity at High Stress*. The recognized *Intrinsic Permeability-Porosity Function* types are *Davis* or *none*. The *Davis* function requires the specification of one parameter: 1) *Davis-Davis Intrinsic Permeability-Porosity*. The recognized *Capillary Pressure-Permeability/Porosity Function* types are *Leverett* or *none*.

4.2.25 Geomechanics Boundary Conditions Card

This card is only required for simulations that include geomechanical calculations using the EPMech (Elastic-Plastic Geomechanics) module. The *Geomechanics Boundary Condition Card* defines the boundary conditions for the geomechanical simulation. Geomechanical boundary conditions may be applied to any boundary surface or surface dividing active and inactive nodes. Geomechanical boundary conditions may be applied only to surfaces of active nodes. To apply a geomechanical 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. To apply a geomechanical boundary condition to an internal boundary surface, defined via the *Internal Boundary Surfaces Card*, the surface is referenced by the active node. Geomechanical boundary conditions are time varying. The user is not allowed to assign multiple geomechanical boundary conditions to a boundary surface during the same time period, but multiple geomechanical boundary conditions can be applied to a boundary surface over different time periods. The simulator controls time steps to agree with time transitions in geomechanical boundary conditions.

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.

Every *Geomechanical Boundary Condition* requires the specification of five *Geomechanical Boundary Types*: 1) *X-Direction Traction Boundary Type*, 2) *Y-Direction Traction Boundary Type*, 3) *Z-Direction Traction Boundary Type, 4) Normal Displacement Boundary Type*, and 5) *Shear Displacement Boundary Type*. *X-Direction Traction Boundary Types* defined on *West* or *East* surfaces require *Normal Displacements*; and *X-Direction Traction Boundary Types* defined on *South*, *North*, *Top*, or *Bottom* surfaces require *Shear Displacements*. *Y-Direction Traction Boundary Types* defined on *North* or *South* surfaces require *Normal Displacements*; and *Y-Direction Traction Boundary Types* defined on *West*, *East*, *Top*, or *Bottom* surfaces require *Shear Displacements*. *Z-Direction Traction Boundary Types* defined on *Bottom* or *Top* surfaces require *Normal Displacements*; and *Z-Direction Traction Boundary Types* defined on *South*, *North*, *West*, or *East* surfaces require *Shear Displacements*.

5.0 Benchmark Problems

5.1 Introduction

The U.S. Department of Energy National Laboratories, including the Pacific Northwest National Laboratory, have developed and are continually improving a suite of numerical simulators for modeling geologic sequestration of greenhouse gases that collectively represent the state of science and coupled-process modeling. A critical component associated with developing these analytic tools is the verification, benchmarking, and validation of the computer code -- the implementation of the mathematical models and numerical solution schemes. Historically, simulators have been verified against analytical solutions, benchmarked against alternative implementations, or validated against laboratory or field experiments. Generally all three checks are used during the simulator development process, with benchmarking and validation dominating as the code becomes mature through extensive application. This section demonstrates the application of the STOMP-CO2 and -CO2e simulators to a series of internationally recognized benchmarking problems.

Numerical simulation capabilities for modeling sequestration processes associated with the fate of greenhouse gasses injected into geologic reservoirs have evolved rapidly during the past decade. Ten years ago the accepted state of the science was numerical simulators that could address dissolution, structural and mineralization trapping, isothermal conditions, aqueous-based geochemistry, and decoupled geomechanics for idealized conceptualizations of the subsurface. The standard for geologic sequestration modeling is continually advancing, but now includes capabilities for predicting hydraulic trapping, nonisothermal conditions, transitions to subcritical conditions, ground-surface interactions, injection wells, co-sequestration, supercritical-CO_2-based geochemistry, coupled hydrology-geochemistry-geomechanics, heterogeneous basin-scale domains, and wettability transitions. Numerical simulators have become common-practice analytical tools for scientists and engineers. When properly applied, these tools provide valuable understanding of complex geologic sequestration processes, and have become invaluable in determining the suitability of sequestration projects. Numerical simulators for geologic sequestration are founded on mathematical descriptions of physical processes and solution algorithms.

5.2 Radial Flow of Supercritical CO_2 from an Injection Well

Radial flow of injected supercritical CO_2 into simplified fresh-water and saline aquifers is compared. This problem is identical to Problem 3 of the code intercomparison problems developed under the GeoSeq Project (Pruess et al. 2002) and addresses two-fluid flow of CO_2 and aqueous for a simplified flow geometry and aquifer properties. A constant mass injection rate of CO_2 is applied from a line source at the center of the infinite radial domain into an aquifer with homogeneous and isotropic hydrologic properties. Gravity and inertial effects are ignored by using a one-dimensional radial computational domain. The problem has a similarity solution, where dependence on the radial distance (*r*) and time (*t*), is replaced by the similarity variable ($\xi = r^2/\,t$), (O'Sullivan 1981; Doughty and Pruess 1992).

5.2.1 Problem Description

Geologic sequestration of anthropologic CO_2 into subsurface reservoirs, including brine aquifers, partially or fully depleted oil and gas reservoirs, and coal beds, is currently being implemented or evaluated globally. Numerical simulation has been used and will continue to be useful in determining the feasibility of sequestering CO_2 into particular reservoirs, developing injection protocols, and monitoring sequestration. The credibility of numerical simulation to accurately model the multifluid subsurface flow, transport, and reactive processes needs to be established before it will become an accepted engineering tool. The primary objective of the code intercomparison exercises of the GeoSeq Project (Pruess et al. 2002) was to evaluate the ability of numerical simulators to model critical processes associated with CO_2 sequestration in geologic reservoirs.

This problem involves the injection of supercritical CO_2 into an infinite-acting one-dimensional radial domain with an aquifer thickness of 100 m. The porous medium is assumed homogenous and isotropic and gravity effects are ignored. Injection occurs at a constant rate of 100 kg/s. The multifluid processes of interest for this problem are two-phase flow of CO_2 and brine, subject to relative permeability and capillarity effects, the effects of pressure and salinity on phase density, phase viscosity and CO_2 solubility and precipitation of salt with dry-out of the formation. Whereas, this problems contains nonlinearities in the thermodynamic and hydrologic transport properties, the problem solution for time and radial distance can be reduced through the similarity variable ($\xi = r^2/\,t$). This allows results to be reported using radial profiles at a fixed time or a time series at a fixed radial distance. The original GeoSeq problem requested that results be reported over the similarity variable range 10^{-8} m²/s $\leq \xi \leq 10^{-1}$ m²/s.

The capillary pressure-saturation relation is described using the van Genuchten formulation (van Genuchten 1980):

$$\bar{s}_l = \frac{s_l - s_{lr}}{1 - s_{lr}} = \left[1 + \left(\alpha\,\beta_{gl}\,h_{gl}\right)^n\right]^{-m} \;;\; m = 1 - \frac{1}{n} \qquad (5.1)$$

The aqueous relative permeability relation is described using the van Genuchten capillary pressure function with the Mualem porosity distribution function (van Genuchten 1980):

$$k_{rl} = \left(\bar{s}_l\right)^{1/2}\left[1 - \left(1 - \left(\bar{s}_l\right)^{1/m}\right)^m\right]^2 \qquad (5.2)$$

The gas relative permeability relation is described using the Modified Corey formulation, which includes an irreducible gas saturation:

$$k_{rg} = \left(1 - \left(\frac{s_l - s_{lr}}{1 - s_{lr} - s_{gr}} \right) \right)^2 \left(1 - \left(\frac{s_l - s_{lr}}{1 - s_{lr} - s_{gr}} \right)^2 \right) \tag{5.3}$$

Simulation parameters are shown in Table 5.1.

Table 5.1. Simulation Parameter Values

Parameter Description		Parameter Value
Intrinsic Permeability		10^{-13} m^2
Porosity		0.12
Pore Compressibility		4.5 x 10^{-10} Pa^{-1}
Aquifer Thickness		100 m
Saturation Function	s_{lr}	0.0
Saturation Function	n	1.84162
Saturation Function	α	0.5 m^{-1}
Aqu. Rel. Perm.	s_{lr}	0.30
Aqu. Rel. Perm.	m	0.457
Gas. Rel. Perm.	s_{gr}	0.05
Gas Rel. Perm.	s_{lr}	0.30
Initial Aquifer Pressure		120 bar
Initial Aquifer Temperature		45˚C
Initial Aquifer Salinity		15 wt.% NaCl
CO_2 Injection Rate		100 kg/s

5.2.2 Zero-Salinity Input

Because this problem involves isothermal conditions, STOMP-CO2 was selected for the simulations. Time stepping and grid spacing were not specified as part of the original GeoSeq problem description, but left to the discretion of the modeler. For this problem a domain ranging from 0.3 to 100,000.0 m was specified using 100 grid cells, with the grid spacing increasing exponentially. An initial time step of 0.001 s was specified with an ending time of 10,000 days. The STOMP-CO2 input file is shown in Section 5.2.2.1; where indented lines indicate a continuation of a previous line. Narratives for selected input cards follow.

5.2.2.1 Zero-Salinity Input File

```
~Simulation Title Card
1,
Problem 3,
M.D. White,
Pacific Northwest Laboratory,
21 May 2002,
09:45 AM PST,
10,
Intercomparison of simulation models for CO2 disposal in
underground storage reservoirs.
Test Problem 3: Radial Flow from a CO2 Injection Well
This problem addresses two-phase flow of CO2 and water
for simplified flow geometry and medium properties.  The
aquifer into which injection is made is assumed infinite-acting,
homogenoeus, and isotropic. Gravity and inertial effects are
neglected, injection is made at a constant mass rate, and flow
is assumed 1-D radial (line source).  Under the conditions
stated the problem has a similarity solution where dependence on
radial distance R and time t occurs only through the similarity
variable x = R2/t (O’Sullivan 1981; Doughty and Pruess 1992).

~Solution Control Card
Normal,
STOMP-CO2,
1,
0,day,1.e+5,day,1.e-3,s,1.e+4,day,1.15,16,1.e-06,
10000,
Variable Aqueous Diffusion,
Variable Gas Diffusion,
0,

~Grid Card
Cylindrical,
100,1,1,
0.3,m,0.34068267,m,0.386882272,m,0.439346951,m,0.498926308,m,0.566585156,m,
0.643419145,m,0.730672508,m,0.829758203,m,0.9422808,m,1.070062462,m,
1.215172455,m,1.379960655,m,1.567095601,m,1.779607712,m,2.020938356,m,
2.294995583,m,2.606217409,m,2.959643684,m,3.360997708,m,3.81677891,m,
4.334368098,m,4.922146988,m,5.589633925,m,6.347638032,m,7.208434242,m,
8.185962079,m,9.296051391,m,10.55667869,m,11.98825828,m,13.61397279,m,
15.46014866,m,17.55668242,m,19.9375248,m,22.64123061,m,25.71158298,m,
29.19830246,m,33.15785213,m,37.65435198,m,42.76061723,m,48.55933748,m,
55.14441582,m,62.62248937,m,71.11465626,m,80.75843656,m,91.70999929,m,
104.1466914,m,118.2699096,m,134.308362,m,152.5217712,m,173.2050808,m,
196.6932312,m,223.3665839,m,253.6570806,m,288.0552382,m,327.1180922,m,
371.4782167,m,421.853969,m,479.0611216,m,544.0260733,m,617.8008506,m,
701.5801442,m,796.7206557,m,904.7630673,m,1027.456991,m,1166.789304,m,
1325.016317,m,1504.700323,m,1708.751078,m,1940.472932,m,2203.618331,m,
2502.448588,m,2841.802888,m,3227.176651,m,3664.810527,m,4161.79145,m,
4726.16741,m,5367.077773,m,6094.901285,m,6921.424143,m,7860.030856,m,
8925.920993,m,10136.35532,m,11510.93531,m,13071.92059,m,14844.58935,m,
16857.64778,m,19143.69485,m,21739.75025,m,24687.85387,m,28035.74657,m,
31837.64332,m,36155.1111,m,41058.06594,m,46625.90509,m,52948.79278,m,
60129.12031,m,68283.16417,m,77542.96894,m,88058.48564,m,100000,m,
0.0,deg,45.0,deg,
0.0,m,100.0,m,
~Rock/Soil Zonation Card
1,
Aquifer,1,100,1,1,1,1,

~Mechanical Properties Card
Aquifer,2650,kg/m^3,0.12,0.12,Compressibility,4.5e-10,1/Pa,100.0,bar,Millington and
    Quirk,

~Hydraulic Properties Card
Aquifer,1.e-13,m^2,,,,,0.8,0.8,

~Saturation Function Card
Aquifer,van Genuchten,0.5,1/m,1.84162,0.0,0.457,0.0,

~Aqueous Relative Permeability Card
Aquifer,Mualem Irreducible,0.457,0.30,

~Gas Relative Permeability Card
Aquifer,Corey,0.05,0.30,

~Salt Transport Card
Aquifer,0.0,m,0.0,m,

~Initial Conditions Card
Gas Pressure,Aqueous Pressure,
3,
Gas Pressure,120.0,Bar,,,,,,,1,100,1,1,1,1,
Aqueous Pressure,120.0,Bar,,,,,,,1,100,1,1,1,1,
Temperature,45.0,C,,,,,,,1,100,1,1,1,1,

~Source Card
1,
Gas Mass Rate,Water-Vapor Mass Fraction,1,1,1,1,1,1,1,
0,s,120.0,bar,12.5,kg/s,0.0,

~Boundary Conditions Card
1,
East,Aqu. Dirichlet,Gas Dirichlet,Aqu. Mass Frac.,
100,100,1,1,1,1,1,
0,s,120.0,bar,0.0,120.0,bar,1.0,0.0,,

~Output Options Card
2,
33,1,1,
63,1,1,
1,1,s,m,deg,6,6,6,
12,
Aqueous Pressure,Pa,
Gas Pressure,Pa,
Aqueous Saturation,,
Gas Saturation,,
Aqueous Density,kg/m^3,
Gas Density,kg/m^3,
Aqueous Viscosity,Pa s,
Gas Viscosity,Pa s,
Aqueous Relative Permeability,,
Gas Relative Permeability,,
CO2 Aqueous Mass Fraction,,
Water Gas Mass Fraction,,
4,
30,day,
100,day,
1000,day,
10000,day,
13,
Aqueous Pressure,Pa,
Gas Pressure,Pa,
Aqueous Saturation,,
Gas Saturation,,
Aqueous Density,kg/m^3,
Gas Density,kg/m^3,
Aqueous Viscosity,Pa s,
Gas Viscosity,Pa s,
Aqueous Relative Permeability,,
Gas Relative Permeability,,
CO2 Aqueous Mass Fraction,,
Water Gas Mass Fraction,,
X Node Centroid,m,
```

5.2.2.2 Solution Control Card

The simulation started at 0 days and ended at 100,000 days. An initial time step of 0.001 s was used to capture early time events. A moderate time-step acceleration factor of 1.15 was used to capture more time points, but not inhibit time-step growth.

5.2.2.3 Grid Card

A one-dimensional radial grid was used, with a geometrically increasing grid spacing. An outer radial distance of 100,000 m was used to simulate an infinite domain. A 45° wedge was used, which means the CO_2 injection rate needed to be 1/8th of the specified 100 kg/s.

5.2.2.4 Initial Conditions Card

The initial gas and aqueous pressures were set to 120 bar throughout the domain, yielding aqueous saturated conditions.

5.2.2.5 Source Card

CO_2 was introduced into the domain using a *Gas Mass Rate* source option, with a gas water mass fraction of 0.0. The specified gas pressure is not used in the simulation. For this type of source, CO_2 mass and water mass are injected directly into the inner radial node at the specified rate.

5.2.2.6 Boundary Condition Card

An outer radial boundary was set for the aqueous and gas phases that maintained the initial pressure condition on the boundary surface.

5.2.2.7 Output Control Card

The reference node output at nodes 33 and 63 had node centroids of 18.7471 and 850.742 m, respectively. The plot file outputs were requested at 30, 100, 1,000 and 10,000 days. Both the reference node output and plot file outputs were used to generate the similarity variable plots, demonstrating the generated similarity solution.

5.2.3 Zero-Salinity Results

The evolution of the gas front is shown in Figure 5.1 for four points in time. The coarse grid spacing for the outer radial nodes tends to smear the leading edge of the front profile. Near the injection point, at the center of the domain, the injected dry CO_2 eventually desiccates the rock. The plots in Figure 5.1 were created using the plotTo.pl perl script utility on the plot files that were generated at the respective days. Accuracy of the numerical simulation can be examined by plotting the results in terms of the similarity variable. Figure 5.2 shows the results for pressure as a function of the similarity variable at four times (30, 100, 1000 and 10000 days) and two radial distances (18.7471 and 850.742 m), which correspond to the centroids of nodes 33 and 63. The agreement of the results between the different time points and radial distance is good, verifying the similarity property of the numerical solution. The plots at different points in time were generated from the plot files and the plotTo.pl perl script. The plots at different radial distance points were generated from the reference-node section of the output file using the outputTo.pl perl script. Figures 5.3 and 5.4 show gas saturation and aqueous dissolved CO_2 mass fraction.

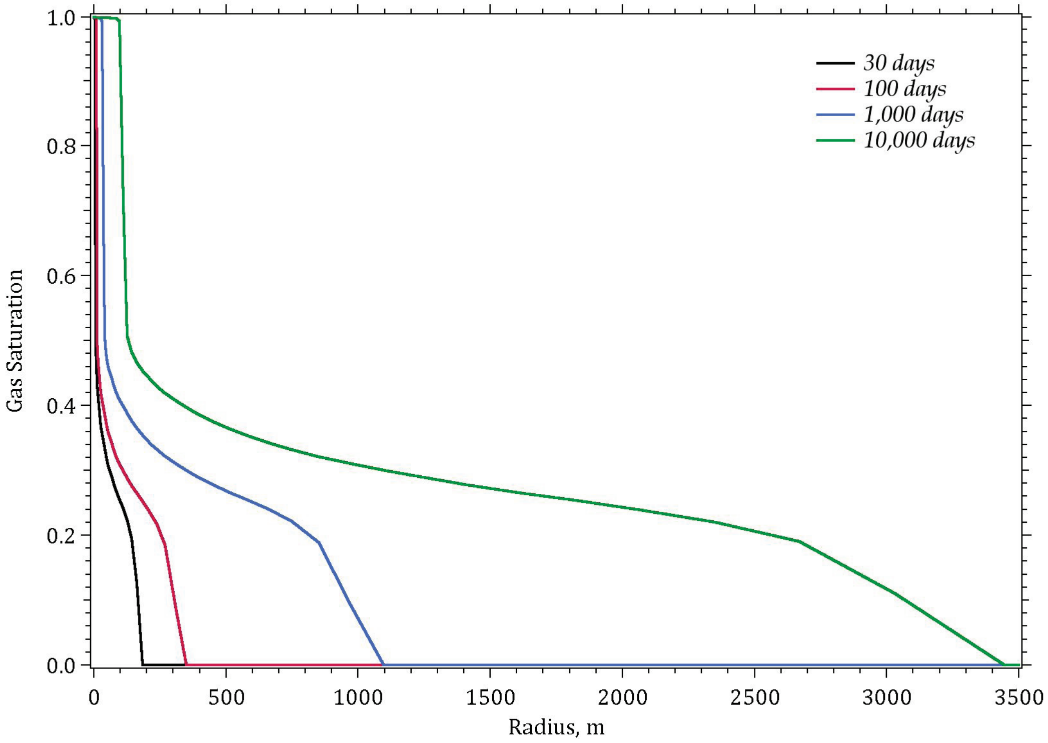

Figure 5.1. Gas Saturation Profile 30, 100, 1,000, and 10,000 Days (Zero Salinity)

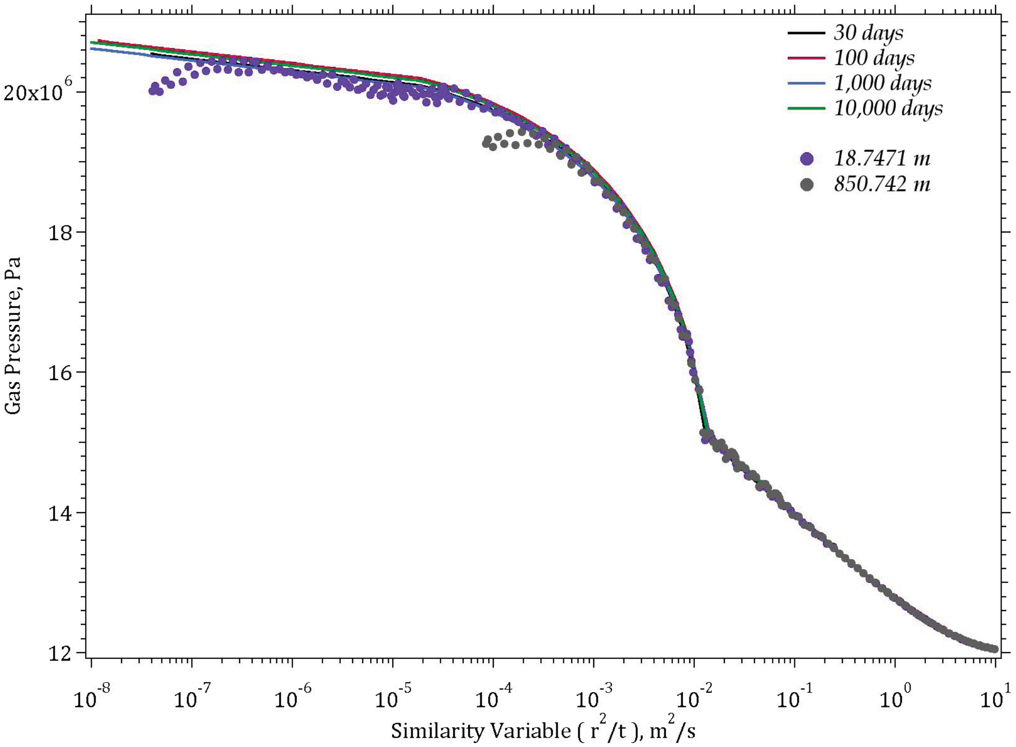

Figure 5.2. Gas Pressure versus the Similarity Variable, Using Both Reference-Node and Plot-File Data

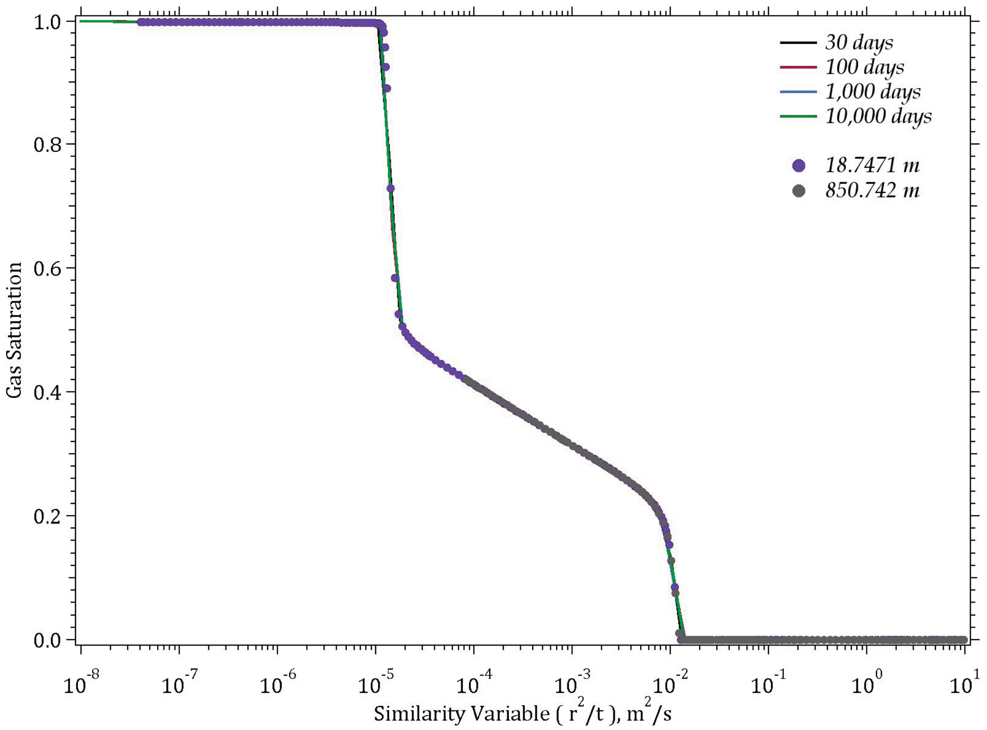

Figure 5.3. Gas Saturation versus the Similarity Variable, Using Both Reference-Node and Plot-File Data

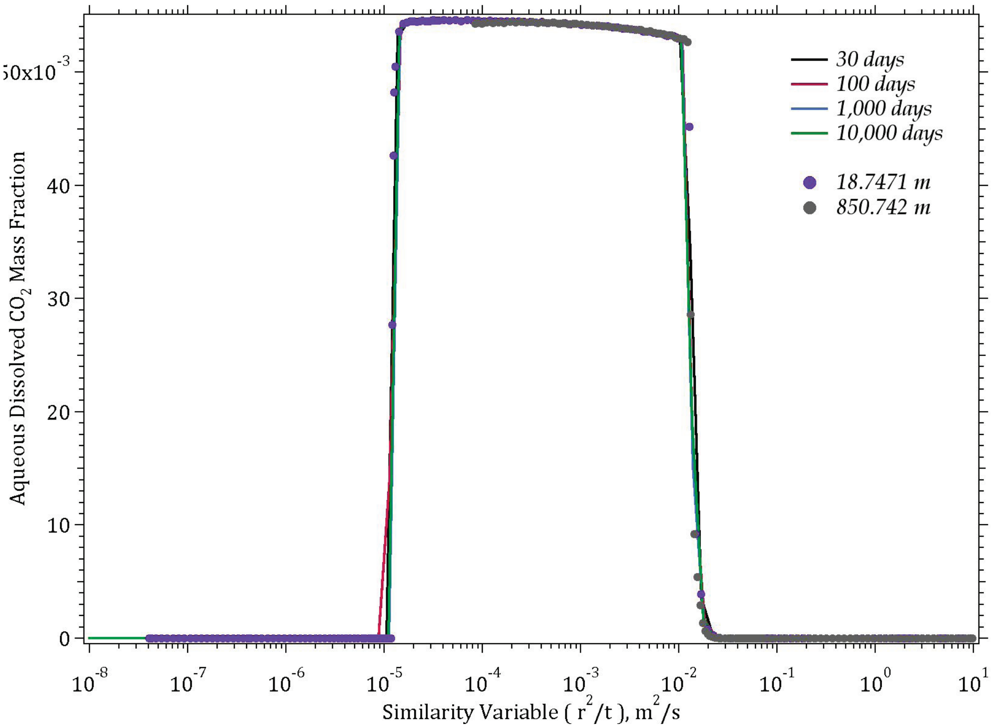

Figure 5.4. Aqueous CO_2 Mass Fraction versus the Similarity Variable, Using Both Reference-Node and Plot-File Data

5.2.4 Salinity Input

The salinity input only differs slightly from the zero-salinity input. The differences between the two input files are discussed in the input card narratives. The STOMP-CO2 input file is shown in Section 5.2.4.1; where indented lines indicate a continuation of a previous line. Narratives for selected input cards follow.

5.2.4.1 Salinity Input File

```
~Simulation Title Card
1,
Problem 3,
M.D. White,
Pacific Northwest Laboratory,
21 May 2002,
09:45 AM PST,
10,
Intercomparison of simulation models for CO2 disposal in
underground storage reservoirs.
Test Problem 3: Radial Flow from a CO2 Injection Well
This problem addresses two-phase flow of CO2 and water
for simplified flow geometry and medium properties.  The
aquifer into which injection is made is assumed infinite-acting,
homogenoeus, and isotropic. Gravity and inertial effects are
neglected, injection is made at a constant mass rate, and flow
is assumed 1-D radial (line source).  Under the conditions
stated the problem has a similarity solution where dependence on
radial distance R and time t occurs only through the similarity
variable x = R2/t (O'Sullivan 1981; Doughty and Pruess 1992).
```

```
~Solution Control Card
Normal,
STOMP-CO2,
1,
0,day,1.e+5,day,1.e-3,s,1.e+4,day,1.15,16,1.e-06,
10000,
Variable Aqueous Diffusion,
Variable Gas Diffusion,
0,

~Grid Card
Cylindrical,
100,1,1,
0.3,m,0.34068267,m,0.386882272,m,0.439346951,m,0.498926308,m,0.566585156,m,
0.643419145,m,0.730672508,m,0.829758203,m,0.9422808,m,1.070062462,m,
1.215172455,m,1.379960655,m,1.567095601,m,1.779607712,m,2.020938356,m,
2.294995583,m,2.606217409,m,2.959643684,m,3.360997708,m,3.81677891,m,
4.334368098,m,4.922146988,m,5.589633925,m,6.347638032,m,7.208434242,m,
8.185962079,m,9.296051391,m,10.55667869,m,11.98825828,m,13.61397279,m,
15.46014866,m,17.55668242,m,19.9375248,m,22.64123061,m,25.71158298,m,
29.19830246,m,33.15785213,m,37.65435198,m,42.76061723,m,48.55933748,m,
55.14441582,m,62.62248937,m,71.11465626,m,80.75843656,m,91.70999929,m,
104.1466914,m,118.2699096,m,134.308362,m,152.5217712,m,173.2050808,m,
196.6932312,m,223.3665839,m,253.6570806,m,288.0552382,m,327.1180922,m,
371.4782167,m,421.853969,m,479.0611216,m,544.0260733,m,617.8008506,m,
701.5801442,m,796.7206557,m,904.7630673,m,1027.456991,m,1166.789304,m,
1325.016317,m,1504.700323,m,1708.751078,m,1940.472932,m,2203.618331,m,
2502.448588,m,2841.802888,m,3227.176651,m,3664.810527,m,4161.79145,m,
4726.16741,m,5367.077773,m,6094.901285,m,6921.424143,m,7860.030856,m,
8925.920993,m,10136.35532,m,11510.93531,m,13071.92059,m,14844.58935,m,
16857.64778,m,19143.69485,m,21739.75025,m,24687.85387,m,28035.74657,m,
31837.64332,m,36155.1111,m,41058.06594,m,46625.90509,m,52948.79278,m,
60129.12031,m,68283.16417,m,77542.96894,m,88058.48564,m,100000,m,
0.0,deg,45.0,deg,
0.0,m,100.0,m,

~Rock/Soil Zonation Card
1,
Aquifer,1,100,1,1,1,1,

~Mechanical Properties Card
Aquifer,2650,kg/m^3,0.12,0.12,Pore Compressibility,4.5e-10,1/Pa,100.0,bar,Millington
    and Quirk,

~Hydraulic Properties Card
Aquifer,1.e-13,m^2,,,,,0.8,0.8,

~Saturation Function Card
Aquifer,van Genuchten,0.5,1/m,1.84162,0.0,0.457,0.0,

~Aqueous Relative Permeability Card
Aquifer,Mualem Irreducible,0.457,0.30,

~Gas Relative Permeability Card
Aquifer,Corey,0.05,0.30,

~Salt Transport Card
Aquifer,0.0,m,0.0,m,
~Initial Conditions Card
Gas Pressure,Aqueous Pressure,
4,
Gas Pressure,120.0,Bar,,,,,,,1,100,1,1,1,1,
Aqueous Pressure,120.0,Bar,,,,,,,1,100,1,1,1,1,
Temperature,45.0,C,,,,,,,1,100,1,1,1,1,
Salt Mass Fraction,0.15,,,,,,,,1,100,1,1,1,1,

~Source Card
1,
Gas Mass Rate,Water-Vapor Mass Fraction,1,1,1,1,1,1,1,
0,s,120.0,bar,12.5,kg/s,0.0,

~Boundary Conditions Card
1,
East,Aqu. Dirichlet,Gas Dirichlet,Aqu. Mass Frac.,
100,100,1,1,1,1,1,
0,s,120.0,bar,0.0,120.0,bar,1.0,0.15,,

~Output Options Card
2,
33,1,1,
63,1,1,
1,1,s,m,deg,6,6,6,
7,
Gas Saturation,,
Salt Saturation,,
Salt Aqueous Mass Fraction,,
CO2 Aqueous Mass Fraction,,
Gas Pressure,Pa,
Diffusive Porosity,,
Similitude Variable,,
4,
30,day,
100,day,
1000,day,
10000,day,
7,
Gas Saturation,,
Salt Saturation,,
Salt Aqueous Mass Fraction,,
CO2 Aqueous Mass Fraction,,
Gas Pressure,Pa,
Diffusive Porosity,,
Similitude Variable,,
```

5.2.4.2 Initial Conditions Card

An initial salinity of 0.15 wt.% was specified with an additional input line in the *Initial Conditions Card*, using the *Salt Mass Fraction*. This initial condition option sets the aqueous salt mass fraction to the specified value.

5.2.4.3 Boundary Condition Card

To maintain salinity on the outer radial boundary the salt boundary condition was specified, using an aqueous salt mass fraction of 0.15.

5.2.4.4 Output Control Card

The number of requested variables was reduced for both the reference-node sand plot-file outputs, but salt saturation and aqueous salt mass fraction were added to the list. Salt saturation refers to the saturation of precipitated salt. As the salt precipitates it reduces the effective porosity and permeability of the geologic media.

5.2.5 Salinity Results

The initial aqueous salt concentration of 15 wt% was below the solubility limit. As the injected CO_2 desiccates the geologic media the salt concentration increases beyond the solubility limit resulting in salt precipitation, as shown in Figure 5.5. Comparing Figures 5.2 and 5.6, shows that the gas pressures late in time or near the injection point are higher than those for the simulation without salt due to the permeability reduction that occurs with salt precipitation. A value of 0.8 was used for the fraction of original porosity at which permeability goes to zero and the fractional length of the pore body. Because the gas injection is specified as a constant mass rate and the amount of precipitate salt is small, the gas saturation profiles are similar between the zero-salinity and salinity simulations (Figures 5.3 and 5.7). The saturated salt concentrations resulted in lower aqueous CO_2 concentrations (i.e., the salting-out effect), as seen by comparing Figures 5.4 and 5.8.

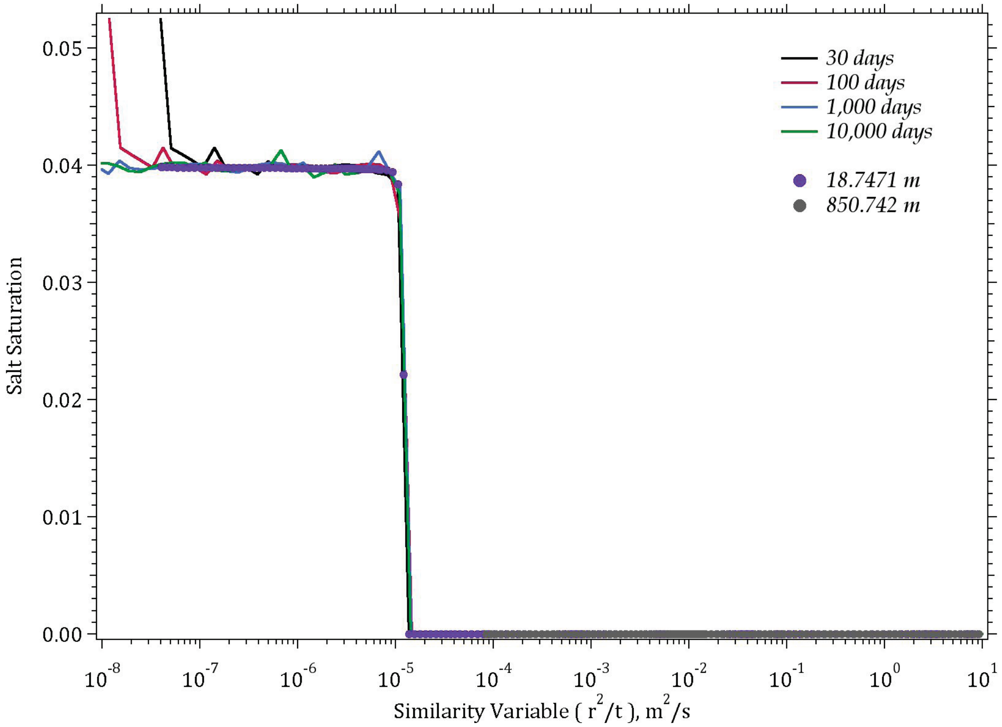

Figure 5.5. Salt Saturation versus the Similarity Variable, Using Both Reference-Node and Plot-File Data

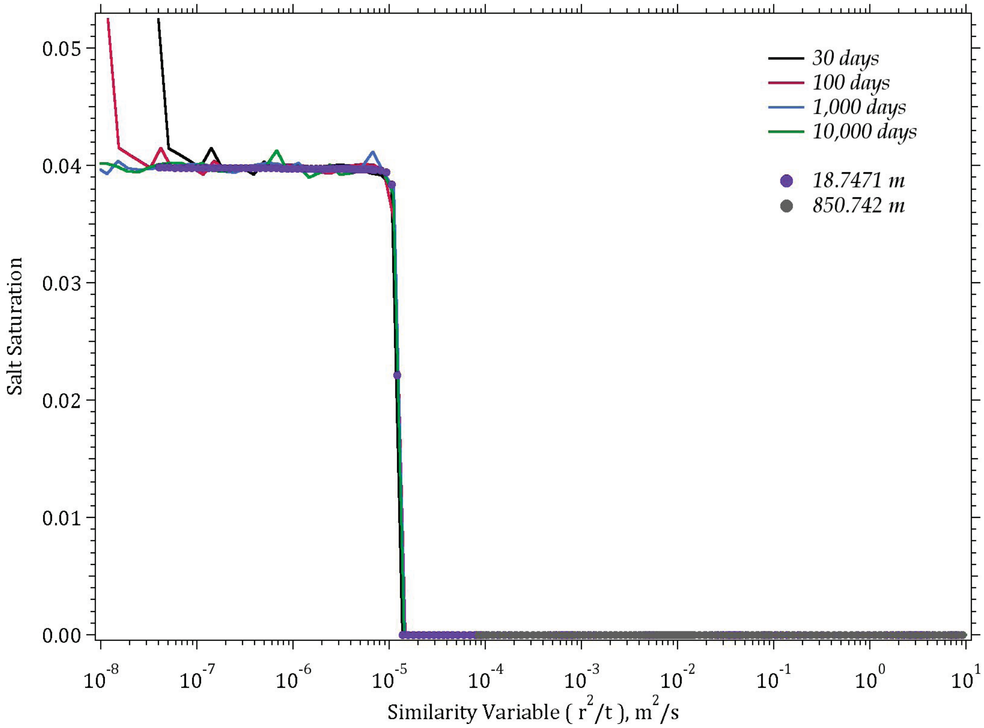

Figure 5.6. Gas Pressure versus the Similarity Variable, Using Both Reference-Node and Plot-File Data

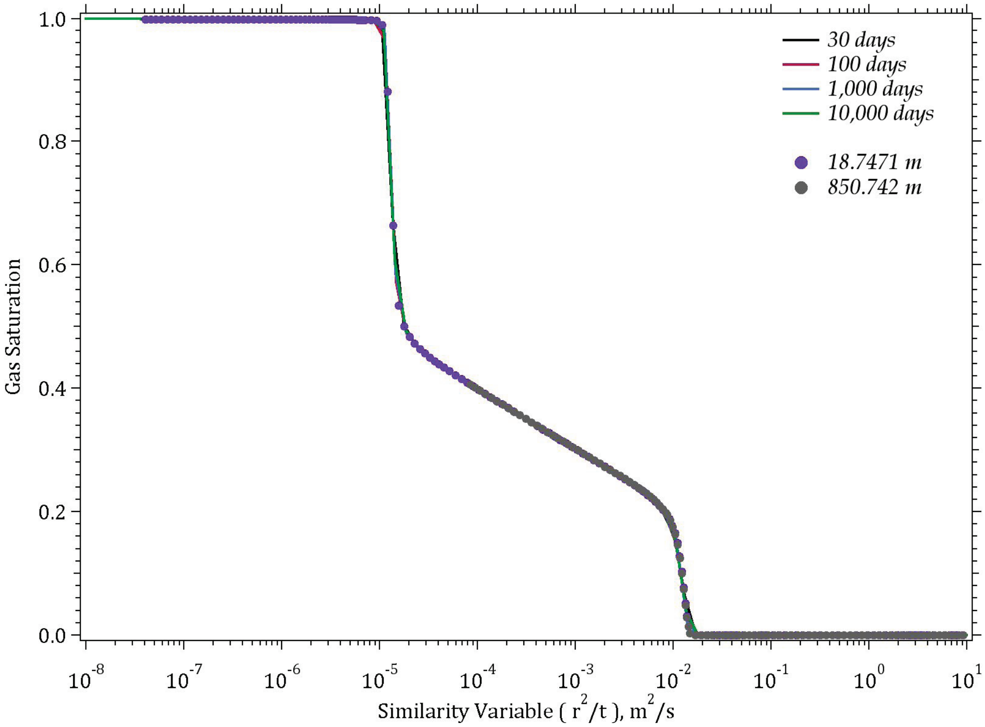

Figure 5.7. Gas Saturation versus the Similarity Variable, Using Both Reference-Node and Plot-File Data

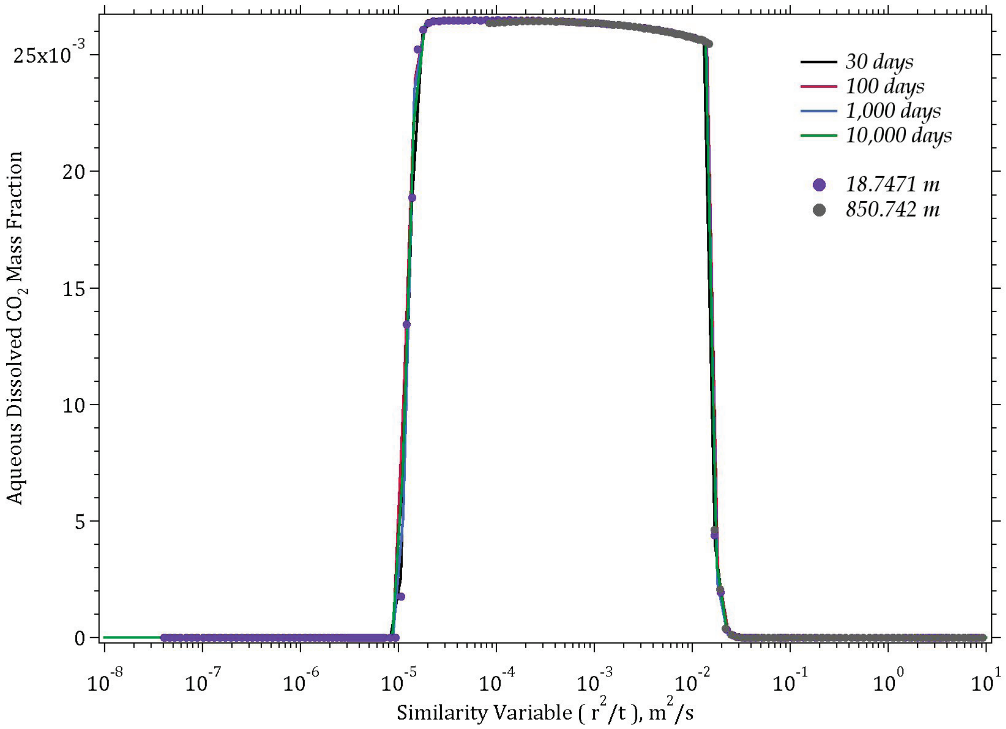

Figure 5.8. Aqueous CO_2 Mass Fraction versus the Similarity Variable, Using Both Reference-Node and Plot-File Data

5.3 Discharge of Sequestered CO_2 Along a Fault Zone

Loss of CO_2 from a deep fresh-water aquifer through a leaky fault is investigated. This problem is identical to Problem 4 of the code intercomparison problems developed under the GeoSeq Project (Pruess et al. 2002) and addresses two-fluid flow of CO_2 and aqueous for a simplified, one-dimensional vertical flow geometry. The problem is designed to investigate the transport of CO_2 from the disposal aquifer to another aquifer 500 m above, through an intersecting vertical fault. The vertical fault is idealized using a one-dimensional geometry and constant pressure boundary conditions (Pruess and Garcia 2002).

5.3.1 Problem Description

Geologic sequestration of anthropologic CO_2 into subsurface reservoirs, including brine aquifers, partially or fully depleted oil and gas reservoirs, and coal beds, is currently being implemented or evaluated globally. Numerical simulation has been used and will continue to be useful in determining the feasibility of sequestering CO_2 into particular reservoirs, developing injection protocols, and monitoring sequestration. The credibility of numerical simulation to accurately model the multifluid subsurface flow, transport, and reactive processes needs to be established before it will become an accepted engineering tool. The primary objective of the code intercomparison exercises of the GeoSeq Project (Pruess et al. 2002), was to evaluate the ability of numerical simulators to model critical processes associated with CO_2 sequestration in geologic reservoirs.

This problem involves the leakage of CO_2 from the injection aquifer to another aquifer situated 500 m above, through an idealized 25-m leaky fault as shown in Figure 5.9.

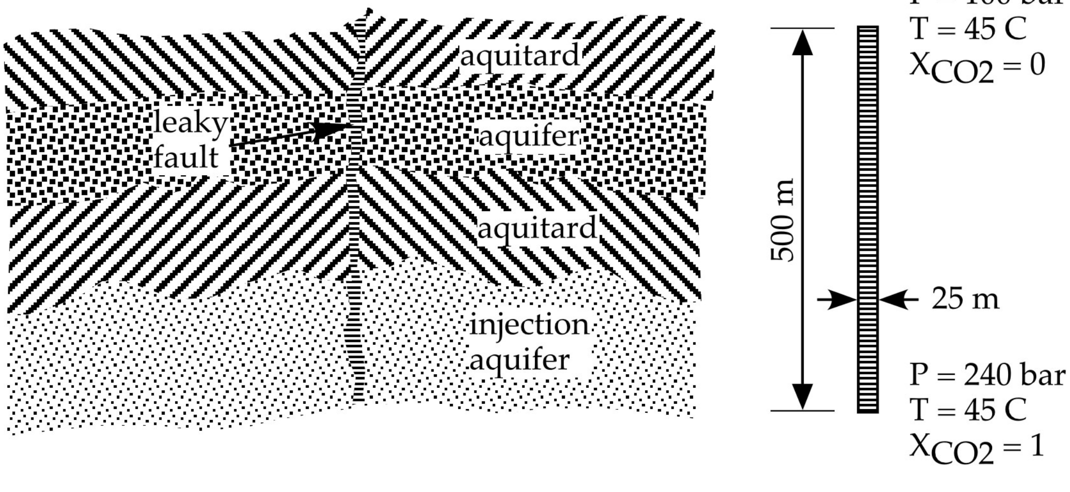

Figure 5.9. Schematic of Fault-Connected Aquifers and Idealized Fault with Boundary Conditions, from Pruess et al. (2002)

Initially, the system is under saturated hydrostatic conditions (pure water) relative to the 100–bar pressure in the upper aquifer. Temperature is held constant throughout the simulation at 45 C. At time zero the gas pressure in the lower aquifer is increased to 240 bar causing an immiscible displacement of water by upward migrating CO_2 with concurrent dissolution of CO_2 into the aqueous phase.

Results to be calculated are CO_2 mass fluxes (kg/m^2 s) over both gas and aqueous phases at the fault inlet (bottom) and outlet (top). Aqueous phase flux (kg/m^2 s) is to be calculated at the fault outlet. Fluxes are to be reported for a range of times from 10^3 to 10^{11} s. Profiles of gas saturation and dissolved CO_2 mass fraction at times of 1 x 10^7 and 2 x 10^7 s are to reported, along with the CO_2 inventory in the aqueous and gas phases at those times.

The capillary pressure-saturation relation is described using the van Genuchten formulation (van Genuchten 1980):

$$\bar{s}_l = \frac{s_l - s_{lr}}{1 - s_{lr}} = \left[1 + \left(\alpha \, \beta_{gl} \, h_{gl} \right)^n \right]^{-m} \; ; \; m = 1 - \frac{1}{n} \tag{5.4}$$

The aqueous relative permeability relation is described using the van Genuchten capillary pressure function with the Mualem porosity distribution function (van Genuchten 1980):

$$k_{rl} = \left(\bar{s}_l \right)^{1/2} \left[1 - \left(1 - \left(\bar{s}_l \right)^{1/m} \right)^m \right]^2 \tag{5.5}$$

The gas relative permeability relation is described using the Modified Corey formulation, which includes an irreducible gas saturation:

$$k_{rg} = \left(1 - \left(\frac{s_l - s_{lr}}{1 - s_{lr} - s_{gr}} \right) \right)^2 \left(1 - \left(\frac{s_l - s_{lr}}{1 - s_{lr} - s_{gr}} \right)^2 \right) \tag{5.6}$$

Simulation parameters are shown in Table 5.2.

Table 5.2. Simulation Parameter Values

Parameter Description		Parameter Value
Intrinsic Permeability		10^{-13} m^2
Porosity		0.35
Pore Compressibility		4.5 x 10^{-10} Pa^{-1}
Fault Height		500 m
Fault Width		25 m
Saturation Function	s_{lr}	0.0
Saturation Function	n	1.84162
Saturation Function	α	0.5 m^{-1}
Aqu. Rel. Perm.	s_{lr}	0.30
Aqu. Rel. Perm.	m	0.457

Gas. Rel. Perm.	s_{gr}	0.05
Gas Rel. Perm.	s_{lr}	0.30
Initial Aquifer Pressure		hydrostatic w/ 100 bar at top
Initial Aquifer Temperature		45˚C
Initial Aquifer Salinity		0 wt.% NaCl
CO_2 Injection Pressure		240 bar at bottom

Time stepping and grid spacing were not specified as part of the original GeoSeq problem description but were in fact left to the discretion of the modeler. For this problem, the 500-m fault was modeled using vertical 100 grid cells with a uniform height of 5 m. The width of the domain matched the width of the fault (25 m) and a 1-m depth was used. To achieve hydrostatic conditions, an initial simulation was executed for a period of 10^{11} s imposing 100-bar pressure conditions at the fault top and zero flux boundary conditions at the fault bottom. The results from this hydrostatic simulation were then used as initial conditions for the transient simulation, which used an initial time step of 1 s, with a time-step acceleration factor of 1.25 for a total time of 10^{11} s.

5.3.2 Hydrostatic Input

Because this problem involves isothermal conditions, STOMP-CO2 was selected for the simulations. Time stepping and grid spacing were not specified as part of the original GeoSeq problem description, but left to the discretion of the modeler. For this problem a one-dimensional vertical column of 100 uniformly spaced nodes was used to model the fault. This problem was modeled using two simulations: 1) hydrostatic and 2) transient. The hydrostatic simulation was conducted to create hydrostatic conditions in the fault using closed bottom boundary and a fixed aqueous pressure of 100 bar at the top surface. The transient simulation was then initialized with the restart file that was generated from the hydrostatic simulation. The STOMP-CO2 input file is shown in Section 5.3.2.1; where indented lines indicate a continuation of a previous line. Narratives for selected input cards follow.

5.3.2.1 Hydrostatic Input File

```
~Simulation Title Card
1,
Problem 4,
M.D. White,
Pacific Northwest Laboratory,
26 August 2002,
14:45 AM PST,
10,
Intercomparison of simulation models for CO2 disposal in
underground storage reservoirs.
Test Problem 4: CO2 Discharge Along a Fault Zone
This problem explores CO2 loss from storage through a leaky fault,
using a highly simplified 1-D linear flow geometry. It is envisioned
that an aquifer into which CO2 disposal is made is intersected by a
vertical fault, which establishes a connection through an otherwise
impermeable caprock to another aquifer 500 m above the storage aquifer.
This situation is idealized by assuming 1-D flow geometry and constant
pressure boundary conditions (Pruess and Garcia, 2000).
```

```
~Solution Control Card
Normal,
STOMP-CO2,
1,
0,s,1.e+11,s,1.e+3,s,1.e+11,s,1.25,16,1.e-06,
10000,
Variable Aqueous Diffusion,
Variable Gas Diffusion,
0,

~Grid Card
Uniform Cartesian,
1,1,100,
25.0,m,
1.0,m,
5.0,m,

~Rock/Soil Zonation Card
1,
Fault,1,1,1,1,1,100,

~Mechanical Properties Card
Fault,2650,kg/m^3,0.35,0.35,Compressibility,4.5e-10,1/Pa,100.0,bar,Millington and
    Quirk,

~Hydraulic Properties Card
Fault,1.e-13,m^2,1.e-13,m^2,1.e-13,m^2,0.8,0.8,

~Saturation Function Card
Fault,van Genuchten,0.5,1/m,1.84162,0.0,0.457,0.0,

~Aqueous Relative Permeability Card
Fault,Mualem Irreducible,0.457,0.30,

~Gas Relative Permeability Card
Fault,Corey,0.05,0.30,

~Salt Transport Card
Fault,0.0,m,0.0,m,

~Initial Conditions Card
Gas Pressure,Aqueous Pressure,
3,
Gas Pressure,148.80475,Bar,,,,,-0.0981,1/m,1,1,1,1,1,100,
Aqueous Pressure,148.80475,Bar,,,,,-0.0981,1/m,1,1,1,1,1,100,
Temperature,45.0,C,,,,,,,1,1,1,1,1,100,

~Boundary Conditions Card
1,
Top,Aqueous Dirichlet,Gas Dirichlet,Aqueous Mass Fraction,
1,1,1,1,100,100,1,
0,s,100.0,bar,0.0,100.0,bar,1.0,0.0,,
~Output Options Card
4,
1,1,1,
1,1,10,
1,1,90,
1,1,100,
1,1,s,m,6,6,6,
5,
Gas Saturation,,
CO2 Gas Mass Fraction,,
CO2 Aqueous Mass Fraction,,
Gas Pressure,Pa,
Diffusive Porosity,,
0,
5,
Gas Saturation,,
CO2 Gas Mass Fraction,,
CO2 Aqueous Mass Fraction,,
Gas Pressure,Pa,
Diffusive Porosity,,
```

5.3.2.2 Solution Control Card

The hydrostatic conditions simulation was initiated with a constant density hydrostatic gradient in pressure, which should be close to the computed hydrostatic conditions that consider compressibility. As the initial conditions are close to the hydrostatic conditions a large initial time step was specified for the hydrostatic conditions simulation.

5.3.2.3 Grid Card

A one-dimensional vertical column of 100 nodes was used to model the fault with a uniform z-direction spacing of 5 m, x-direction spacing of 25 m, and y-direction spacing of 1m, yielding a fault height of 500 m and width of 25 m.

5.3.2.4 Initial Conditions Card

The initial gas and aqueous pressures were set to 148.80475 bar at the lowest node in the domain, with a vertical gradient of -0.0981 bar/m, which is hydrostatic for an incompressible aqueous density of 1000 kg/m^3. The temperature was set to the problem conditions of 45˚C.

5.3.2.5 Boundary Condition Card

Boundary conditions for all boundary surfaces except for the fault top were unspecified, which implies no flow across the unspecified boundary surfaces. The fault top boundary used Dirichlet conditions for the aqueous and gas boundary types, with equal pressures of 100 bar. The aqueous CO$_2$ relative saturation was set to 0.0, and the gas water-vapor relative saturation was set to 1.0.

5.3.3 Transient Input

The hydrostatic simulation required 77 time steps to complete, generating a restart file named *restart.77*. This restart file was then used to initialize the transient simulation. The pressure at the lowest node in the domain changed during the simulation from 14.8805 MPa to 14.8590 MPa, indicating a slight shift in the pressure profile from that specified in the *Initial Conditions Card* of the hydrostatic simulation. The STOMP-CO2 input file is shown in Section 5.3.3.1; where indented lines indicate a continuation of a previous line. Narratives for selected input cards follow.

5.3.3.1 Transient Input File

```
~Simulation Title Card
1,
Problem 4,
M.D. White,
Pacific Northwest Laboratory,
26 August 2002,
14:45 AM PST,
10,
Intercomparison of simulation models for CO2 disposal in
underground storage reservoirs.
Test Problem 4: CO2 Discharge Along a Fault Zone
This problem explores CO2 loss from storage through a leaky fault,
using a highly simplified 1-D linear flow geometry. It is envisioned
that an aquifer into which CO2 disposal is made is intersected by a
vertical fault, which establishes a connection through an otherwise
impermeable caprock to another aquifer 500 m above the storage aquifer.
This situation is idealized by assuming 1-D flow geometry and constant
pressure boundary conditions (Pruess and Garcia, 2002).

~Solution Control Card
Restart File,restart.77,
H2O-NaCl-CO2,
1,
0,s,1.e+11,s,1.e+0,s,1.e+11,s,1.25,16,1.e-06,
10000,
Variable Aqueous Diffusion,
Variable Gas Diffusion,
0,

~Grid Card
Uniform Cartesian,
1,1,100,
25.0,m,
1.0,m,
5.0,m,

~Rock/Soil Zonation Card
1,
Fault,1,1,1,1,1,100,

~Mechanical Properties Card
Fault,2650,kg/m^3,0.35,0.35,Compressibility,4.5e-10,1/Pa,100.0,bar,Millington and
     Quirk,

~Hydraulic Properties Card
Fault,1.e-13,m^2,1.e-13,m^2,1.e-13,m^2,0.8,0.8,

~Saturation Function Card
Fault,van Genuchten,0.5,1/m,1.84162,0.0,0.457,0.0,

~Aqueous Relative Permeability Card
Fault,Mualem Irreducible,0.457,0.30,

~Gas Relative Permeability Card
Fault,Corey,0.05,0.30,

~Salt Transport Card
Fault,0.0,m,0.0,m,
~Boundary Conditions Card
2,
Top,Aqueous Dirichlet,Gas Dirichlet,Aqueous Mass Fraction,
1,1,1,1,100,100,1,
0,s,100.0,bar,0.0,100.0,bar,1.0,0.0,,
Bottom,Aqueous Zero Flux,Gas Dirichlet,Aqueous Mass Fraction,
1,1,1,1,1,1,1,
0,s,,,0.0,240.0,bar,0.0,0.0,,

~Output Options Card
4,
1,1,1,
1,1,10,
1,1,90,
1,1,100,
1,1,s,m,6,6,6,
8,
Gas Saturation,,
CO2 Gas Mass Fraction,,
CO2 Aqueous Mass Fraction,,
Gas Pressure,Pa,
Diffusive Porosity,,
Integrated CO2 Mass,kg,
Integrated Aqueous CO2 Mass,kg,
Integrated Gas CO2 Mass,kg,
2,
1.e+07,s,
2.e+07,s,
7,
Gas Saturation,,
CO2 Gas Mass Fraction,,
CO2 Aqueous Mass Fraction,,
Gas Pressure,Pa,
Diffusive Porosity,,
Gas Density,kg/m^3,
Aqueous Density,kg/m^3,

~Surface Flux Card
3,
Total CO2 Flux,kg/s,kg,Bottom,1,1,1,1,1,1,
Total CO2 Flux,kg/s,kg,Top,1,1,1,1,100,100,
Aqueous Mass Flux,kg/s,kg,Top,1,1,1,1,100,100,
```

5.3.3.2 Solution Control Card

The restart file generated at the end of the hydrostatic simulation contains primary variables for every node and timing data, including the simulation time of 10^{11} s. The restart file created by the hydrostatic simulation was read by specifying *Restart File* as the *Execution Option*, and then specifying *restart.77* as the *Restart File Name*. The timing data in the restart file can be used or overwritten. To avoid having the transient simulation start from the time of 10^{11} s, the timing data was overwritten by specifying timing information in the *Solution Control Card*. The transient simulation started at 0 s and ended at 10^{11} s. An initial time step of 1.0 s was used to capture early time events. A conventional time-step acceleration factor of 1.25 was specified and 16 Newton-Raphson iterations were permitted.

5.3.3.3 Initial Conditions Card

Because the initial conditions for the transient simulation are contained in the restart file, the *Initial Condition Card* was not included in the input file.

5.3.3.4 Boundary Condition Card

For the transient simulation the *Gas Boundary Type* was set to *Dirichlet* on the *bottom* surface of the lowest node in the fault and the gas pressure was set to a constant value of 240.0 bar. The relative saturation of water in the gas phase was set to zero and the *Aqueous Boundary Type* was set to *Zero Flux*.

5.3.3.5 Output Control Card

Reference node output was requested at two nodes near the bottom and top of the fault. To track the amounts of CO_2 in the fault over time three integrated CO_2 mass were specified in the *Reference Node Variable* list: 1) *Integrated C02 Mass*, 2) *Integrated Aqueous C02 Mass*, and 3) *Integrated Gas C02 Mass*, which tracks the total, aqueous dissolved, gas phase CO_2 masses, respectively. To create profiles of gas saturation and the aqueous dissolved mass fraction of CO_2 at times of 1 x 10^7 and 2 x 10^7 s, plot files were requested the *Plot File Times* of 1.e+7 and 2.e+7 s, and *Plot File Variables* of *Gas Saturation* and *Aqueous CO2 Mass Fraction* were requested.

5.3.3.6 Surface Flux Card

To track the CO_2 mass flux crossing the fault bottom and top the *Surface Flux Type* of *Total CO2 Flux* was requested for the *bottom* surface of the lowest node and for the *top* surface of the upper most node in the domain. The surface normal for a *bottom* surface is downward (i.e., negative z direction), therefore, flux into the fault, in the positive z direction is negative. The surface normal for a *top* surface is upward (i.e., positive z direction), therefore, flux exiting the fault, in the positive z direction is positive.

5.3.4 Transient Results

In response to a step change in pressure at the lower fault boundary condition, CO_2 migrates up the fault, displacing the aqueous phase and concurrently dissolving into the aqueous phase. Gas saturation profiles at 1 x 10^7 and 2 x 10^7 s are shown in Figure 5.10. Aqueous dissolved CO_2 mass fraction profiles at 1 x 10^7 and 2 x 10^7 s are shown in Figure 5.11. Dissolution of CO_2 in the aqueous phase for the thermodynamic conditions of this problem is subject to strong non-idealities. The STOMP simulator contains two solubility formulations, with and without the Poynting correction factor. Without the Poynting correction factor the CO_2 solubility increases with pressure which yields the slope in dissolved CO_2 mass fraction with depth shown in Figure 5.11. The Poynting correction factor reduces this solubility at higher pressures. The time dependence of CO_2 and water mass fluxes are shown in Figures 5.12 through 5.14. Because of the step change in boundary pressure, initially the CO_2 flux entering the fault is large, but then decreases until CO_2 breaks through the fault top, at approximately 2.75 x 10^7 s. As with the CO_2 flux, water flux at the fault top increases rapidly, transitions to a quasi-steady flux and then decreases rapidly after CO_2 breakthrough at the fault top. Aqueous flux then slowly declines as water evaporates into the dry CO_2 stream. Total CO_2 inventories in the aqueous and gas phases at 1 x 10^7 s are 83.7 and 396.1 tonnes, respectively; and at 2 x 10^7 s are 143.5 and 681.9 tonnes, respectively.

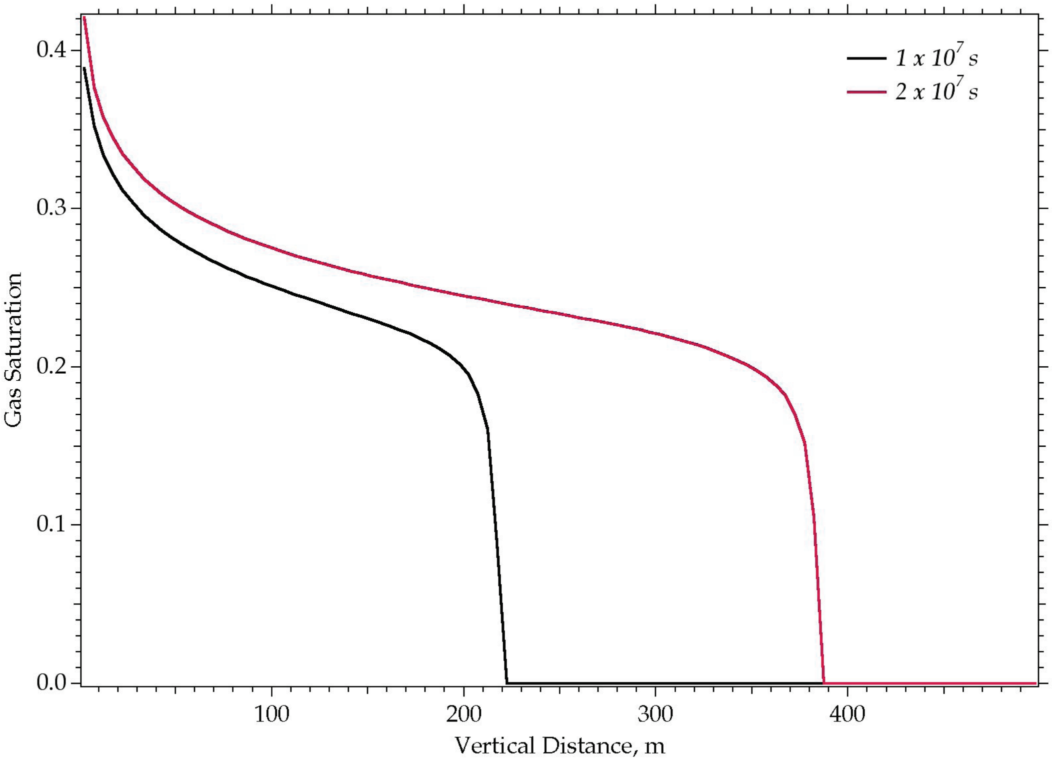

Figure 5.10. Gas Saturation Profiles at 1 x 10^7 and 2 x 10^7 s

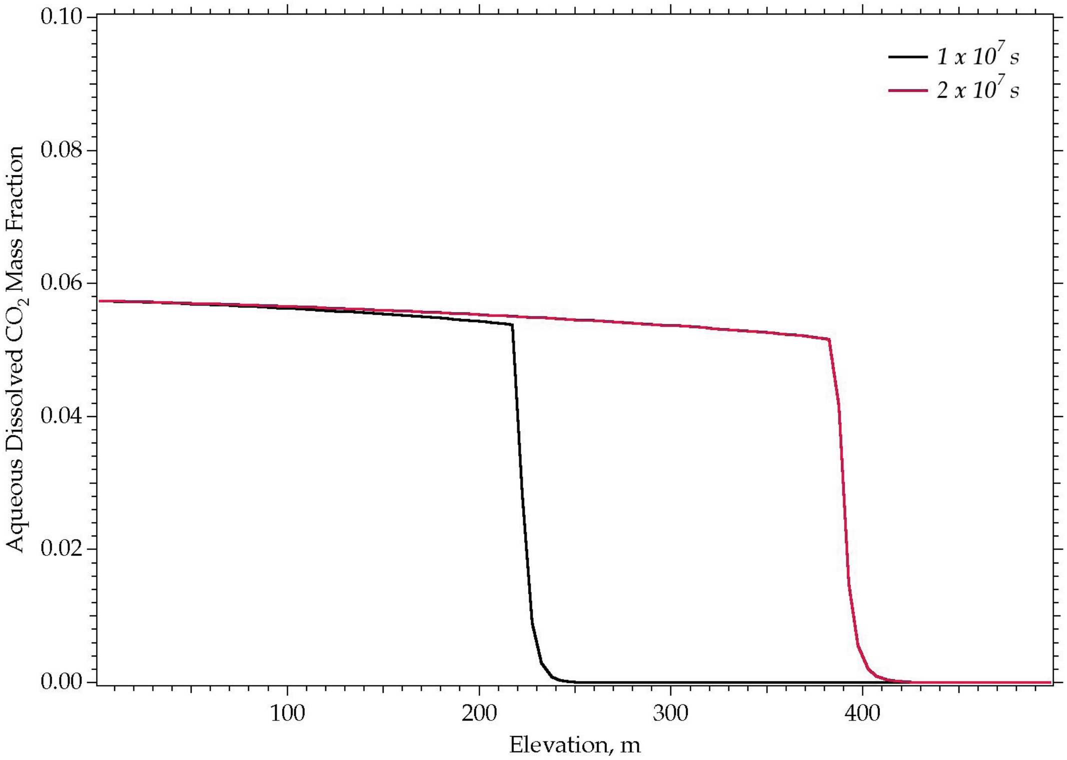

Figure 5.11. Dissolved CO_2 Mass Fraction Profiles at 1 x 10^7 and 2 x 10^7 s

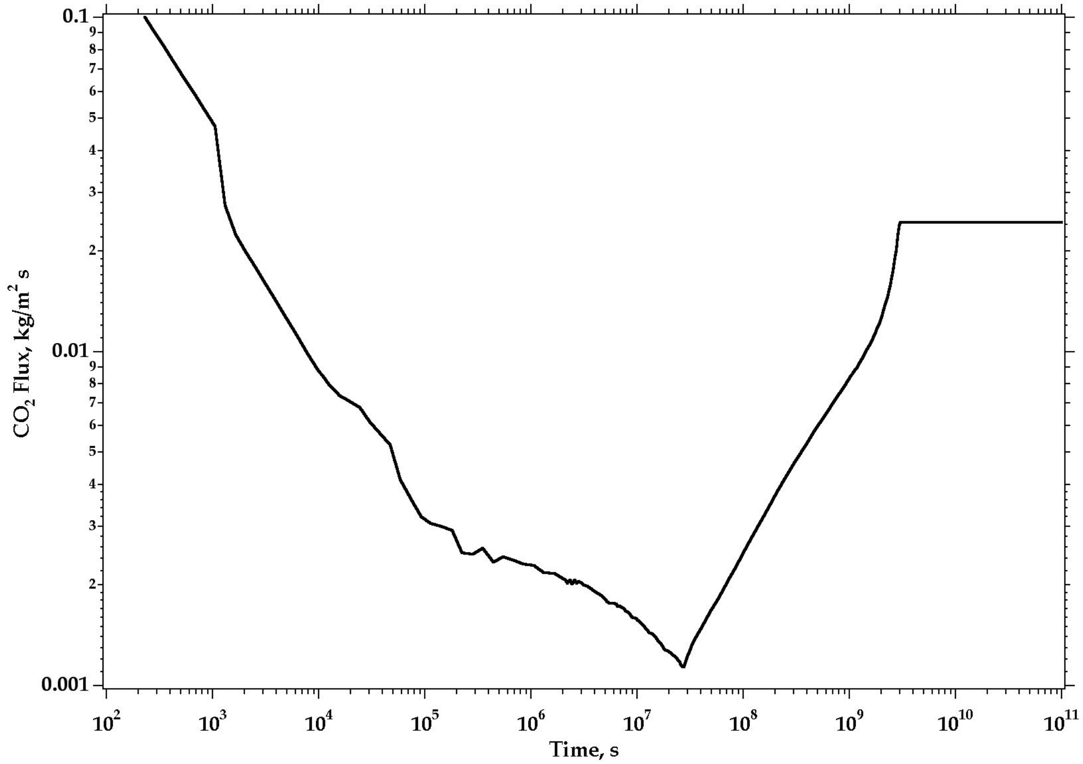

Figure 5.12. CO_2 Flux at the Fault Bottom, kg/m^2 s

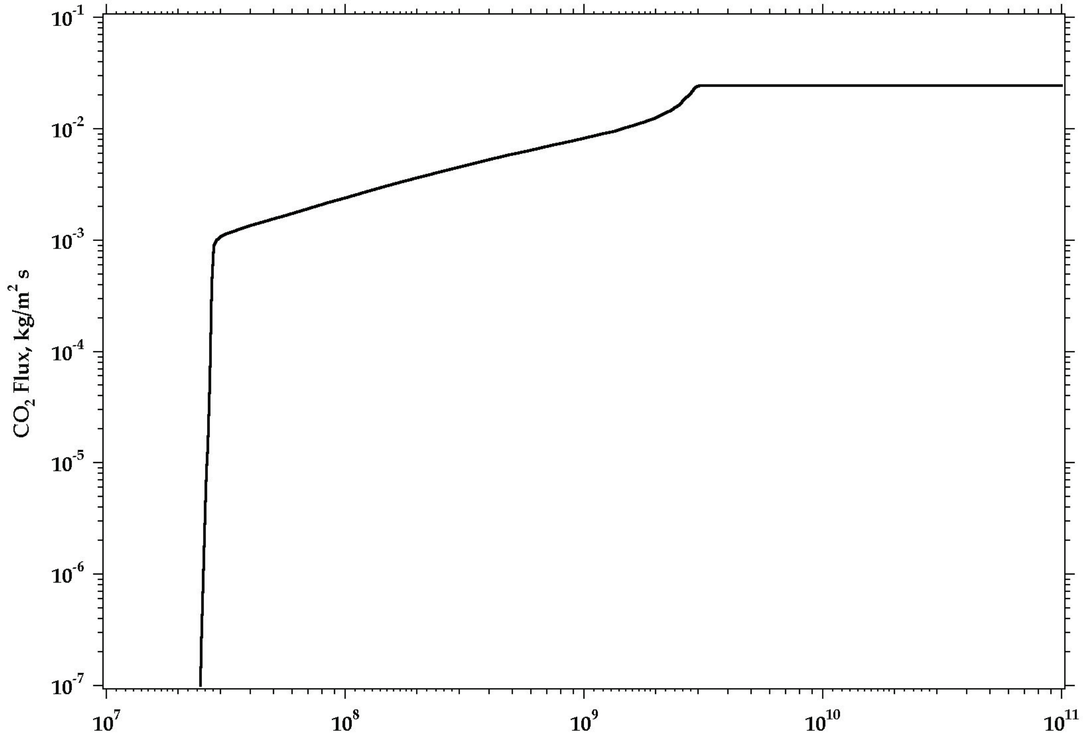

Figure 5.13. CO_2 Flux at the Fault Top, kg/m^2 s

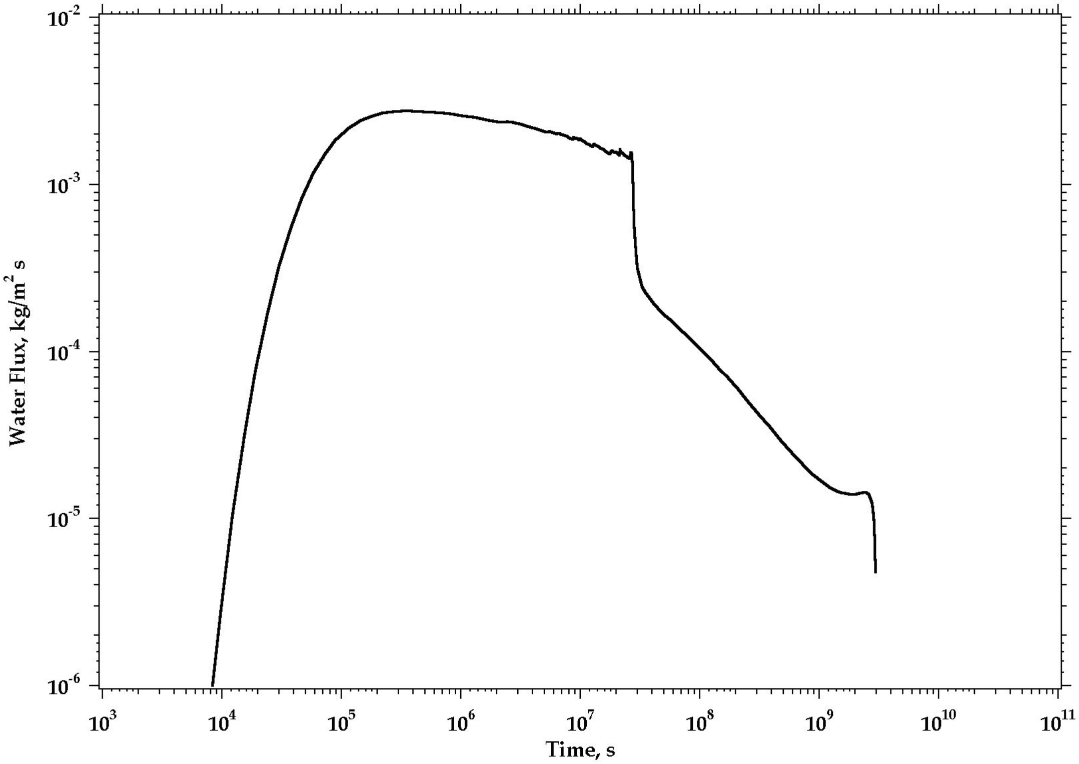

Figure 5.14. Water Flux at th Fault Top, kg/m^2 s

5.4 Mineral Trapping in a Glauconitic Sandstone Aquifer

This problem addresses geochemical effects of CO_2 injection into a glauconitic sandstone aquifer, and analyzes the impact of CO_2 immobilization through carbonate precipitation. This problem is based on Problem 5 of the code intercomparison problems developed under the GeoSeq Project (Pruess et al. 2002). Batch reaction modeling of the geochemical evolution of this aquifer is performed in the presence of CO_2 at high pressure. The problem is based on (Gunter et al. 1997), who modeled water-rock reactions when CO_2 is injected into a glauconitic sandstone aquifer in the Alberta Sedimentary Basin, Canada.

5.4.1 Problem Description

This problem addresses geochemical effects of CO_2 injection into a glauconitic sandstone aquifer, and analyzes the impact of CO_2 immobilization through carbonate precipitation. Batch reaction modeling of the geochemical evolution of this aquifer is performed in the presence of CO_2 at high pressure. The chemical reactions caused by CO_2 injection begin with the dissolution of CO_2 in water to form weak carbonic acid:

$$CO_2\left(g\right)+H_2O=CO_2\left(aq\right) \tag{5.7}$$

This is followed by dissociation of carbonic acid to form the bicarbonate ion:

$$CO_2\left(aq\right)=H^++HCO_3^- \tag{5.8}$$

The increased acidity causes dissolution of primary host rock minerals, which in turn causes complexing of dissolved cations with the bicarbonate ion such as

$$Ca^{2+}+HCO_3^-=CaHCO_3^+ \tag{5.9}$$

The dissolved bicarbonate species react with divalent cations to precipitate carbonates. Formation of calcium, magnesium, and ferrous carbonates are expected to be the primary mechanism by which CO_2 is immobilized (Gunter et al. 1997):

$$\begin{aligned} HCO_3^-+Ca^{2+}&=CaCO_3(s)+H^+\\ HCO_3^-+Mg^{2+}&=MgCO_3(s)+H^+\\ HCO_3^-+Fe^{2+}&=FeCO_3(s)+H^+ \end{aligned} \tag{5.10}$$

The glauconitic sandstone aquifer (Alberta Sedimentary Basin, Canada) is a medium- to fine-grained litharenite. The average mineral composition is shown in Table 5.3. The average porosity is 12%. A representative glauconite chemical composition and thermodynamic properties were estimated from descriptions of the mineralogical compositions of glauconite and its paragenesis as reported in the published literature (Xu et al. 2001). Oligoclase was incorporated as a solid solution of plagioclase, and the thermodynamic properties of oligoclase were calculated from calorimetric studies of plagioclase solid solutions reported in the literature. Furthermore, organic matter was assumed to be present in the glauconitic sandstone, and was represented by the generic composition, CH_2O. Goethite (FeOOH) was added as a possible secondary mineral phase.

Table 5.3. List of Initial Mineral Volume Fractions, Potential Secondary Mineral Phases and Kinetic Parameters

Mineral	Composition	Volume %	Surface Area cm^2/g	k_{25}, $mol/m^2\ s$	E_a, kJ/mol
Quartz	SiO_2	71.28	2.69e+1	1.26e-14	87.5
Glauconite	$K_{1.5}Mg_{0.5}Fe_{2.5}Fe_{0.5}AlS_{7.5}O_{20}(OH)_4$	4.4	1.60e+1	1.00e-14	58.62
Illite	$K_{0.6}Mg_{0.25}Al_{1.8}(Al_{0.5}Si_{3.5}O_{10})(OH)_2$	2.64	9.60e+2	1.00e-14	58.62
Organic	CH_2O	2.64	2.64e+0	1.00e-13	0
Kaolinite	$Al_2Si_2)_5(OH)_4$	1.76	6.88e-1	1.00e-13	62.76
K-feldspar	$KAlSi_3O_8$	1.76	6.80e+2	1.00e-12	67.83
Calcite	$CaCO_3$	0.88	3.25e-1	1.60e-9	41.87
Dolomite	$CaMg(CO_3)_2$	0.88	3.08e-1	6.00e-10	41.87
Oligoclase	$CaNa_4Al_6Si_{14}O_{40}$	0.88	3.19e-1	1.00e-12	67.83
Siderite	$FeCO_3$	0.88	2.23e-1	6.00e-10	41.87
Albite-low	$NaAlSi_3O_8$	0.0	9.54e-1	1.00e-12	67.83
Smectite-Ca	$Ca_{0.145}Mg_{0.26}Al_{1.77}Si_{3.97}O_{10}(OH)_2$	0.0	1.14e+2	1.00e-14	58.62
Smectite-Na	$Na_{0.29}Mg_{0.26}Al_{1.77}Si_{3.97}O_{10}(OH)_2$	0.0	1.00e+2	1.00e-14	58.62
Goethite	FeOOH	0.0	1.14e+2	1.00e-14	58.62

This problem uses the TST rate equation (Lasaga 1984; Steefel and Lasaga 1994), which is expressed as

$$R_k = A_m\, k \left[1 - \frac{Q}{K_{eq}} \right] \tag{5.11}$$

where *m* is mineral index, R_k is the dissolution/precipitation rate (positive values indicate dissolution, and negative values precipitation), A_m is the specific reactive surface area, *k* is the rate constant (moles per unit mineral surface area and unit time) which is temperature dependent, K_{eq} is the equilibrium constant for the mineral-water reaction written for the destruction of one mole of mineral *m*, and *Q* is ion activity product. The temperature dependence of the reaction rate constant can be expressed reasonably well via an Arrhenius equation (Lasaga 1984; Steefel and Lasaga 1994). Since many rate constants are reported at 25°C, it is convenient to approximate rate constant dependency as a function of temperature,

$$k = k_{25} \exp\left[\left(\frac{-E_a}{R} \right) \left(\frac{1}{T(K)} - \frac{1}{298.15\, K} \right) \right] \tag{5.12}$$

5.4.2 Mineral Trapping Input Description

This simulation is concerned with the geochemical effects of CO_2 into a glauconitic sandstone, not the migration of injected CO_2. Whereas CO_2 migration behavior is not a component of this simulation, parameters for transport properties must be defined, including the *Mechanical Properties Card*, *Hydraulic Properties Card*, *Saturation Function Card*, *Aqueous Relative Permeability Card*, *Gas Relative Permeability Card*, and *Salt Transport Card*. With the focus on geochemistry, the simulation considers reactions for a single node. The reactive transport cards required for this simulation are described in the EKEChem Addendum to the STOMP guides (White and McGrail, 2005). Development of the input file for this STOMP-CO2 simulation is described here. The STOMP-CO2 input file is shown in Section 5.4.2.1; where indented lines indicate a continuation of a previous line. Narratives for selected input cards follow.

5.4.2.1 Mineral Trapping Input File

```
~Simulation Title Card
3.2,
GeoSeq Problem 5,
DH Bacon,
Pacific Northwest Laboratory,
15 Jun 2010,
07:19 PM PDT,
3,
This problem addresses geochemical effects of CO2 injection into a glauconitic
sandstone aquifer, and analyzes the impact of CO2 immobilization through
carbonate precipitation.

~Solution Control Card
Normal,
STOMP-CO2 w/ ECKEChem,
1,
0,s,50000,year,1,s,10,year,1.01,16,1.e-06,
10000000,
Variable Aqueous Diffusion,
Variable Gas Diffusion,
0,

~Grid Card
Uniform Cartesian,
1,1,1,
1.0,m,
1.0,m,
1.0,m,

~Rock/Soil Zonation Card
1,
Aquifer,1,1,1,1,1,1,

~Solute/Porous Media Interaction Card
Aquifer,,,,,

~Mechanical Properties Card
Aquifer,2650,kg/m^3,0.12,0.12,Compressibility,4.5e-10,1/Pa,100.0,bar,Millington and
    Quirk,

~Hydraulic Properties Card
Aquifer,1.e-10,m^2,1.e-10,m^2,1.e-10,m^2,0.8,0.8,

~Saturation Function Card
Aquifer,Brooks and Corey Extended,54.0,cm,4.033,0.01,,

~Aqueous Relative Permeability Card
Aquifer,Mualem,,

~Gas Relative Permeability Card
Aquifer,Mualem,,

~Salt Transport Card
Aquifer,0.2,m,0.2,m,
~Aqueous Species Card
37,1.e-9,cm^2/s,Bdot,1.0,
Al(OH)2+,1.0,4.0,A,60.996,kg/kmol,
Al+++,3.0,9.0,A,26.982,kg/kmol,
AlO2-,-1.0,4.0,A,58.980,kg/kmol,
AlOH++,2.0,4.5,A,43.989,kg/kmol,
CO(aq),0.0,3.0,A,28.010,kg/kmol,
CO2(aq),0.0,3.0,A,44.010,kg/kmol,
Ca++,2.0,6.0,A,40.078,kg/kmol,
CaCl+,1.0,4.0,A,75.531,kg/kmol,
CaCl2(aq),0.0,3.0,A,110.983,kg/kmol,
CaHCO3+,1.0,4.0,A,101.095,kg/kmol,
Cl-,-1.0,3.0,A,35.453,kg/kmol,
Ethane(aq),0.0,3.0,A,30.070,kg/kmol,
Fe(OH)2+,1.0,4.0,A,89.862,kg/kmol,
Fe(OH)3(aq),0.0,3.0,A,106.869,kg/kmol,
Fe++,2.0,6.0,A,55.847,kg/kmol,
Fe+++,3.0,9.0,A,55.847,kg/kmol,
FeCl+,1.0,4.0,A,91.300,kg/kmol,
FeCl++,2.0,4.5,A,91.300,kg/kmol,
FeCl2+,1.0,4.0,A,126.752,kg/kmol,
FeHCO3+,1.0,4.0,A,116.864,kg/kmol,
FeOH++,2.0,4.5,A,72.854,kg/kmol,
H+,1.0,9.0,A,1.008,kg/kmol,
HAlO2(aq),0.0,3.0,A,59.988,kg/kmol,
HCO3-,-1.0,4.0,A,61.017,kg/kmol,
K+,1.0,3.0,A,39.098,kg/kmol,
KCl(aq),0.0,3.0,A,74.551,kg/kmol,
Methane(aq),0.0,3.0,A,16.043,kg/kmol,
Mg++,2.0,8.0,A,24.305,kg/kmol,
MgCl+,1.0,4.0,A,59.758,kg/kmol,
MgHCO3+,1.0,4.0,A,85.322,kg/kmol,
Na+,1.0,4.0,A,22.990,kg/kmol,
NaAlO2(aq),0.0,3.0,A,81.970,kg/kmol,
NaCl(aq),0.0,3.0,A,58.442,kg/kmol,
NaHCO3(aq),0.0,3.0,A,84.007,kg/kmol,
NaHSiO3(aq),0.0,3.0,A,100.081,kg/kmol,
O2(aq),0.0,3.0,A,31.999,kg/kmol,
SiO2(aq),0.0,3.0,A,60.084,kg/kmol,

~Solid Species Card
14,
Albite_low,2.62,g/cm^3,262.223,kg/kmol,
Calcite,2.71,g/cm^3,100.087,kg/kmol,
Dolomite,2.86,g/cm^3,184.401,kg/kmol,
Glauconite,2.75,g/cm^3,383.901,kg/kmol,
Goethite,4.27,g/cm^3,88.854,kg/kmol,
Illite,2.75,g/cm^3,383.901,kg/kmol,
K-Feldspar,2.56,g/cm^3,278.332,kg/kmol,
Kaolinite,2.59,g/cm^3,258.160,kg/kmol,
Oligoclase,2.76,g/cm^3,278.207,kg/kmol,
Organic_matter,1.00,g/cm^3,30.026,kg/kmol,
Quartz,2.65,g/cm^3,60.084,kg/kmol,
Siderite,3.94,g/cm^3,115.856,kg/kmol,
Smectite-Ca,2.20,g/cm^3,366.043,kg/kmol,
Smectite-Na,2.50,g/cm^3,367.017,kg/kmol,
~Lithology Card
Aquifer,14,
Albite_low,9.54E-01,cm^2/g,0,
Calcite,3.25E-01,cm^2/g,0.0088,
Dolomite,3.08E-01,cm^2/g,0.0088,
Glauconite,1.60E+01,cm^2/g,0.044,
Goethite,5.85E+01,cm^2/g,0,
Illite,9.60E+02,cm^2/g,0.0264,
K-feldspar,6.80E+02,cm^2/g,0.0176,
Kaolinite,6.88E-01,cm^2/g,0.0176,
Oligoclase,3.19E-01,cm^2/g,0.0088,
Organic_matter,2.64E+00,cm^2/g,0.0264,
Quartz,2.69E+01,cm^2/g,0.7128,
Siderite,2.23E-01,cm^2/g,0.0088,
Smectite-Ca,1.14E+02,cm^2/g,0,
Smectite-Na,1.00E+02,cm^2/g,0,

~Species Link Card
2,
H+,pH,
Total_CO2(aq),Aqueous CO2,

~Conservation Equations Card
11,
Total_Al+++,14,Al+++,1,Al(OH)2+,1,AlO2-
    ,1,AlOH++,1,Albite_low,1,Glauconite,1,HAlO2(aq),1,Illite,2.3,K-
    Feldspar,1,Kaolinite,2,NaAlO2(aq),1,Oligoclase,6,Smectite-Ca,1.77,Smectite-
    Na,1.77,
Total_CO2(aq),13,CO2(aq),1,CO(aq),1,CaHCO3+,1,Calcite,1,Dolomite,2,Ethane(aq),2,FeHCO3
    +,1,HCO3-,1,Methane(aq),1,MgHCO3+,1,NaHCO3(aq),1,Organic_matter,2,Siderite,1,
Total_Ca++,8,Ca++,1,CaCl+,1,CaCl2(aq),1,CaHCO3+,1,Calcite,1,Dolomite,1,Oligoclase,1,Sm
    ectite-Ca,1.45e-01,
Total_Cl-,9,Cl-
    ,1,CaCl+,1,CaCl2(aq),2,FeCl+,1,FeCl++,1,FeCl2+,2,KCl(aq),1,MgCl+,1,NaCl(aq),1,
Total_Fe++,12,Fe++,1,Fe(OH)2+,1,Fe(OH)3(aq),1,Fe+++,1,FeCl+,1,FeCl++,1,FeCl2+,1,FeHCO3
    +,1,FeOH++,1,Glauconite,3,Goethite,1,Siderite,1,
Total_H+,29,H+,1,Al(OH)2+,-2,AlO2-,-4,AlOH++,-1,Albite_low,-4,CaHCO3+,-1,Calcite,-
    2,Dolomite,-4,Fe(OH)2+,-1,Fe(OH)3(aq),-2,Fe+++,1,FeCl++,1,FeCl2+,1,FeHCO3+,-
    1,Glauconite,-1.15e+01,Goethite,-2,HAlO2(aq),-3,HCO3-,-1,Illite,-8,K-Feldspar,-
    4,Kaolinite,-6,MgHCO3+,-1,NaAlO2(aq),-4,NaHCO3(aq),-1,NaHSiO3(aq),-1,Oligoclase,-
    2.4e+01,Siderite,-2,Smectite-Ca,-6.12,Smectite-Na,-6.12,
Total_K+,5,K+,1,Glauconite,1.5,Illite,6.e-01,K-Feldspar,1,KCl(aq),1,
Total_Mg++,8,Mg++,1,Dolomite,1,Glauconite,5.e-01,Illite,2.5e-
    01,MgCl+,1,MgHCO3+,1,Smectite-Ca,2.6e-01,Smectite-Na,2.6e-01,
Total_Na+,8,Na+,1,Albite_low,1,NaAlO2(aq),1,NaCl(aq),1,NaHCO3(aq),1,NaHSiO3(aq),1,Olig
    oclase,4,Smectite-Na,2.9e-01,
Total_O2(aq),13,O2(aq),1,CO(aq),-5.e-01,Ethane(aq),-3.5,Fe(OH)2+,2.5e-
    01,Fe(OH)3(aq),2.5e-01,Fe+++,2.5e-01,FeCl++,2.5e-01,FeCl2+,2.5e-01,FeOH++,2.5e-
    01,Glauconite,6.25e-01,Goethite,2.5e-01,Methane(aq),-2,Organic_matter,-2,
Total_SiO2(aq),11,SiO2(aq),1,Albite_low,3,Glauconite,7.5,Illite,3.5,K-
    Feldspar,3,Kaolinite,2,NaHSiO3(aq),1,Oligoclase,1.4e+01,Quartz,1,Smectite-
    Ca,3.97,Smectite-Na,3.97,

~Equilibrium Reactions Card
26,
EqRc-1,0.0,-9.040,0.0,0.0,0.0,1/mol,
EqRc-2,0.0,-20.084,0.0,0.0,0.0,1/mol,
EqRc-3,0.0,-4.157,0.0,0.0,0.0,1/mol,
EqRc-4,0.0,-42.049,0.0,0.0,0.0,1/mol,
EqRc-5,0.0,-6.265,0.0,0.0,0.0,1/mol,
EqRc-6,0.0,-0.616,0.0,0.0,0.0,1/mol,
EqRc-7,0.0,-0.649,0.0,0.0,0.0,1/mol,
EqRc-8,0.0,-5.136,0.0,0.0,0.0,1/mol,
EqRc-9,0.0,-232.518,0.0,0.0,0.0,1/mol,
EqRc-10,0.0,-2.525,0.0,0.0,0.0,1/mol,
EqRc-11,0.0,-8.855,0.0,0.0,0.0,1/mol,
EqRc-12,0.0,3.145,0.0,0.0,0.0,1/mol,
EqRc-13,0.0,-0.066,0.0,0.0,0.0,1/mol,
EqRc-14,0.0,2.961,0.0,0.0,0.0,1/mol,
EqRc-15,0.0,5.275,0.0,0.0,0.0,1/mol,
EqRc-16,0.0,-3.545,0.0,0.0,0.0,1/mol,
EqRc-17,0.0,0.955,0.0,0.0,0.0,1/mol,
EqRc-18,0.0,-14.154,0.0,0.0,0.0,1/mol,
EqRc-19,0.0,-1.262,0.0,0.0,0.0,1/mol,
EqRc-20,0.0,-127.710,0.0,0.0,0.0,1/mol,
EqRc-21,0.0,-0.081,0.0,0.0,0.0,1/mol,
EqRc-22,0.0,-5.134,0.0,0.0,0.0,1/mol,
EqRc-23,0.0,-20.649,0.0,0.0,0.0,1/mol,
EqRc-24,0.0,-0.676,0.0,0.0,0.0,1/mol,
EqRc-25,0.0,-6.331,0.0,0.0,0.0,1/mol,
EqRc-26,0.0,-8.098,0.0,0.0,0.0,1/mol,

~Equilibrium Equations Card
26,
3,Al(OH)2+,Al+++,1,H+,-2,EqRc-1,1.0,
3,AlO2-,Al+++,1,H+,-4,EqRc-2,1.0,
3,AlOH++,Al+++,1,H+,-1,EqRc-3,1.0,
3,CO(aq),CO2(aq),1,O2(aq),-5.e-01,EqRc-4,1.0,
3,HCO3-,CO2(aq),1,H+,-1,EqRc-5,1.0,
3,CaCl+,Ca++,1,Cl-,1,EqRc-6,1.0,
3,CaCl2(aq),Ca++,1,Cl-,2,EqRc-7,1.0,
4,CaHCO3+,CO2(aq),1,Ca++,1,H+,-1,EqRc-8,1.0,
3,Ethane(aq),CO2(aq),2,O2(aq),-3.5,EqRc-9,1.0,
4,Fe(OH)2+,Fe++,1,H+,-1,O2(aq),2.5e-01,EqRc-10,1.0,
4,Fe(OH)3(aq),Fe++,1,H+,-2,O2(aq),2.5e-01,EqRc-11,1.0,
4,Fe+++,Fe++,1,H+,1,O2(aq),2.5e-01,EqRc-12,1.0,
3,FeCl+,Cl-,1,Fe++,1,EqRc-13,1.0,
5,FeCl++,Cl-,1,Fe++,1,H+,1,O2(aq),2.5e-01,EqRc-14,1.0,
5,FeCl2+,Cl-,2,Fe++,1,H+,1,O2(aq),2.5e-01,EqRc-15,1.0,
4,FeHCO3+,CO2(aq),1,Fe++,1,H+,-1,EqRc-16,1.0,
3,FeOH++,Fe++,1,O2(aq),2.5e-01,EqRc-17,1.0,
3,HAlO2(aq),Al+++,1,H+,-3,EqRc-18,1.0,
3,KCl(aq),Cl-,1,K+,1,EqRc-19,1.0,
3,Methane(aq),CO2(aq),1,O2(aq),-2,EqRc-20,1.0,
3,MgCl+,Cl-,1,Mg++,1,EqRc-21,1.0,
4,MgHCO3+,CO2(aq),1,H+,-1,Mg++,1,EqRc-22,1.0,
4,NaAlO2(aq),Al+++,1,H+,-4,Na+,1,EqRc-23,1.0,
3,NaCl(aq),Cl-,1,Na+,1,EqRc-24,1.0,
4,NaHCO3(aq),CO2(aq),1,H+,-1,Na+,1,EqRc-25,1.0,
4,NaHSiO3(aq),H+,-1,Na+,1,SiO2(aq),1,EqRc-26,1.0,

~Kinetic Reactions Card
14,
KnRc-
    26,TST,Albite_low,3,Al+++,1.0000,Na+,1.0000,SiO2(aq),3.0000,2,Albite_low,1.0000,H+
    ,4.0000,
1.00000e-12,mol/m^2 s,67.83,kJ/mol,25,C,
,1.790700,,,,
KnRc-27,TST,Calcite,2,Ca++,1.0000,HCO3-,1.0000,2,Calcite,1.0000,H+,1.0000,
1.60000e-09,mol/m^2 s,41.87,kJ/mol,25,C,
,1.420000,,,,
KnRc-28,TST,Dolomite,3,Ca++,1.0000,Mg++,1.0000,HCO3-
    ,2.0000,2,Dolomite,1.0000,H+,2.0000,
0.60000e-09,mol/m^2 s,41.87,kJ/mol,25,C,
,1.527900,,,,
KnRc-
    29,TST,Glauconite,6,Fe++,0.5000,Fe+++,2.5000,Mg++,0.5000,K+,1.5000,Al+++,1.0000,Si
    O2(aq),7.5000,2,Glauconite,1.0000,H+,14.0000,
1.00000e-14,mol/m^2 s,58.62,kJ/mol,25,C,
,6.142100,,,,
KnRc-30,TST,Goethite,1,Fe+++,1.0000,2,Goethite,1.0000,H+,3.0000,
1.00000e-14,mol/m^2 s,58.62,kJ/mol,25,C,
,-0.4240000,,,,
KnRc-
    31,TST,Illite,4,Mg++,0.2500,K+,0.6000,Al+++,2.3000,SiO2(aq),3.5000,2,Illite,1.0000
    ,H+,8.0000,
1.00000e-14,mol/m^2 s,58.62,kJ/mol,25,C,
,6.142100,,,,
KnRc-32,TST,K-Feldspar,3,Al+++,1.0000,K+,1.0000,SiO2(aq),3.0000,2,K-
    Feldspar,1.0000,H+,4.0000,
1.00000e-12,mol/m^2 s,67.83,kJ/mol,25,C,
,-0.8170000,,,,
KnRc-33,TST,Kaolinite,2,Al+++,2.0000,SiO2(aq),2.0000,2,Kaolinite,1.0000,H+,6.0000,
1.00000e-13,mol/m^2 s,62.76,kJ/mol,25,C,
,4.334400,,,,
KnRc-
    34,TST,Oligoclase,4,Ca++,1.0000,Al+++,6.0000,SiO2(aq),14.0000,Na+,4.0000,2,Oligocl
    ase,1.0000,H+,24.0000,
1.00000e-12,mol/m^2 s,67.83,kJ/mol,25,C,
,21.73720,,,,
KnRc-35,TST,Organic_matter,3,Methane(aq),1.0000,H+,1.0000,HCO3-
    ,1.0000,1,Organic_matter,2.0000,
1.00000e-13,mol/m^2 s,0.0,kJ/mol,25,C,
,10.,,,,
KnRc-36,TST,Quartz,1,SiO2(aq),1.0000,1,Quartz,1.0000,
1.25890e-14,mol/m^2 s,87.5,kJ/mol,25,C,
,-3.543400,,,,
KnRc-37,TST,Siderite,2,Fe++,1.0000,HCO3-,1.0000,2,Siderite,1.0000,H+,1.0000,
0.60000e-09,mol/m^2 s,41.87,kJ/mol,25,C,
,-0.7248000,,,,
KnRc-38,TST,Smectite-
    Ca,4,Ca++,0.1450,Mg++,0.2600,Al+++,1.7700,SiO2(aq),3.9700,2,Smectite-
    Ca,1.0000,H+,6.1200,
1.00000e-14,mol/m^2 s,58.62,kJ/mol,25,C,
,0.7022000,,,,
KnRc-39,TST,Smectite-
    Na,4,Mg++,0.2600,Na+,0.2900,Al+++,1.7700,SiO2(aq),3.9700,2,Smectite-
    Na,1.0000,H+,6.1200,
1.00000e-14,mol/m^2 s,58.62,kJ/mol,25,C,
,0.8023000,,,,

~Kinetic Equations Card
14,
Kinetic_Albite_low,1,Albite_low,1,
1,KnRc-26,1,
Kinetic_Calcite,1,Calcite,1,
1,KnRc-27,1,
Kinetic_Dolomite,1,Dolomite,1,
1,KnRc-28,1,
Kinetic_Glauconite,1,Glauconite,1,
1,KnRc-29,1,
Kinetic_Goethite,1,Goethite,1,
1,KnRc-30,1,
Kinetic_Illite,1,Illite,1,
1,KnRc-31,1,
Kinetic_K-Feldspar,1,K-Feldspar,1,
1,KnRc-32,1,
Kinetic_Kaolinite,1,Kaolinite,1,
1,KnRc-33,1,
Kinetic_Oligoclase,1,Oligoclase,1,
1,KnRc-34,1,
Kinetic_Organic_matter,1,Organic_matter,1,
1,KnRc-35,1,
Kinetic_Quartz,1,Quartz,1,
1,KnRc-36,1,
Kinetic_Siderite,1,Siderite,1,
1,KnRc-37,1,
Kinetic_Smectite-Ca,1,Smectite-Ca,1,
1,KnRc-38,1,
Kinetic_Smectite-Na,1,Smectite-Na,1,
1,KnRc-39,1,

~Initial Conditions Card
Gas Pressure,Aqueous Pressure,
8,
Gas Pressure,260,bar,,,,,,,1,1,1,1,1,1,
Aqueous Pressure,148.80475,bar,,,,,,,1,1,1,1,1,1,
Temperature,54.0,C,,,,,,,1,1,1,1,1,1,
Salt Mass Fraction,0.1,,,,,,,,1,1,1,1,1,1,
Species Aqueous Volumetric,Cl-,1.00e+00,mol/liter,,,,,,,1,1,1,1,1,1,
Species Aqueous Volumetric,pH,7.0,,,,,,,,1,1,1,1,1,1,
Species Aqueous Volumetric,Na+,1.00e+00,mol/liter,,,,,,,1,1,1,1,1,1,
Species Aqueous Volumetric,O2(aq),1.00e-30,mol/liter,,,,,,,1,1,1,1,1,1,

~Boundary Conditions Card
1,
Top,Aqueous Dirichlet,Gas Dirichlet,Aqueous Mass Fraction,Aqu. Species Zero Flux,Gas
    Species Zero Flux,
0,
1,1,1,1,1,1,1,
0,s,148.80475,bar,1,260,bar,1,0.1,,

~Output Options Card
1,
1,1,1,
1,1,yr,m,6,6,6,
62,
Aqueous Saturation,,
Gas Saturation,,
Salt Saturation,,
Salt Aqueous Mass Fraction,,
CO2 Aqueous Mass Fraction,,
CO2 Gas Mass Fraction,,
Gas Pressure,Pa,
Diffusive Porosity,,
Integrated CO2 Mass,kg,
Integrated Aqueous CO2 Mass,kg,
Integrated Gas CO2 Mass,kg,
Species Aqueous Concentration,Al(OH)2+,mol/liter,
Species Aqueous Concentration,Al+++,mol/liter,
Species Aqueous Concentration,AlO2-,mol/liter,
Species Aqueous Concentration,AlOH++,mol/liter,
Species Aqueous Concentration,CO(aq),mol/liter,
Species Aqueous Concentration,CO2(aq),mol/liter,
Species Aqueous Concentration,Ca++,mol/liter,
Species Aqueous Concentration,CaCl+,mol/liter,
Species Aqueous Concentration,CaCl2(aq),mol/liter,
Species Aqueous Concentration,CaHCO3+,mol/liter,
Species Aqueous Concentration,Cl-,mol/liter,
Species Aqueous Concentration,Ethane(aq),mol/liter,
Species Aqueous Concentration,Fe(OH)2+,mol/liter,
Species Aqueous Concentration,Fe(OH)3(aq),mol/liter,
Species Aqueous Concentration,Fe++,mol/liter,
Species Aqueous Concentration,Fe+++,mol/liter,
Species Aqueous Concentration,FeCl+,mol/liter,
Species Aqueous Concentration,FeCl++,mol/liter,
Species Aqueous Concentration,FeCl2+,mol/liter,
Species Aqueous Concentration,FeHCO3+,mol/liter,
Species Aqueous Concentration,FeOH++,mol/liter,
Species Aqueous Concentration,H+,mol/liter,
Species Aqueous Concentration,HAlO2(aq),mol/liter,
Species Aqueous Concentration,HCO3-,mol/liter,
Species Aqueous Concentration,K+,mol/liter,
Species Aqueous Concentration,KCl(aq),mol/liter,
Species Aqueous Concentration,Methane(aq),mol/liter,
Species Aqueous Concentration,Mg++,mol/liter,
Species Aqueous Concentration,MgCl+,mol/liter,
Species Aqueous Concentration,MgHCO3+,mol/liter,
Species Aqueous Concentration,Na+,mol/liter,
Species Aqueous Concentration,NaAlO2(aq),mol/liter,
Species Aqueous Concentration,NaCl(aq),mol/liter,
Species Aqueous Concentration,NaHCO3(aq),mol/liter,
Species Aqueous Concentration,NaHSiO3(aq),mol/liter,
Species Aqueous Concentration,O2(aq),mol/liter,
Species Aqueous Concentration,SiO2(aq),mol/liter,
Species Volumetric Concentration,Albite_low,mol/liter,
Species Volumetric Concentration,Calcite,mol/liter,
Species Volumetric Concentration,Dolomite,mol/liter,
Species Volumetric Concentration,Glauconite,mol/liter,
Species Volumetric Concentration,Goethite,mol/liter,
Species Volumetric Concentration,Illite,mol/liter,
Species Volumetric Concentration,K-Feldspar,mol/liter,
Species Volumetric Concentration,Kaolinite,mol/liter,
Species Volumetric Concentration,Oligoclase,mol/liter,
Species Volumetric Concentration,Organic_matter,mol/liter,
Species Volumetric Concentration,Quartz,mol/liter,
Species Volumetric Concentration,Siderite,mol/liter,
Species Volumetric Concentration,Smectite-Ca,mol/liter,
Species Volumetric Concentration,Smectite-Na,mol/liter,
23,
0,s,
1,s,
1,min,
1,hr,
1,day,
10,day,
100,day,
1,year,
2,year,
5,year,
10,year,
20,year,
50,year,
100,year,
200,year,
500,year,
1000,year,
2000,year,
5000,year,
10000,year,
20000,year,
50000,year,
100000,year,
62,
Aqueous Saturation,,
Gas Saturation,,
Salt Saturation,,
Salt Aqueous Mass Fraction,,
CO2 Aqueous Mass Fraction,,
CO2 Gas Mass Fraction,,
Gas Pressure,Pa,
Diffusive Porosity,,
Gas Density,kg/m^3,
Aqueous Density,kg/m^3,
Species Aqueous Concentration,Al(OH)2+,mol/liter,
Species Aqueous Concentration,Al+++,mol/liter,
Species Aqueous Concentration,AlO2-,mol/liter,
Species Aqueous Concentration,AlOH++,mol/liter,
Species Aqueous Concentration,CO(aq),mol/liter,
Species Aqueous Concentration,CO2(aq),mol/liter,
Species Aqueous Concentration,Ca++,mol/liter,
Species Aqueous Concentration,CaCl+,mol/liter,
Species Aqueous Concentration,CaCl2(aq),mol/liter,
Species Aqueous Concentration,CaHCO3+,mol/liter,
Species Aqueous Concentration,Cl-,mol/liter,
Species Aqueous Concentration,Ethane(aq),mol/liter,
Species Aqueous Concentration,Fe(OH)2+,mol/liter,
Species Aqueous Concentration,Fe(OH)3(aq),mol/liter,
Species Aqueous Concentration,Fe++,mol/liter,
Species Aqueous Concentration,Fe+++,mol/liter,
Species Aqueous Concentration,FeCl+,mol/liter,
Species Aqueous Concentration,FeCl++,mol/liter,
Species Aqueous Concentration,FeCl2+,mol/liter,
Species Aqueous Concentration,FeHCO3+,mol/liter,
Species Aqueous Concentration,FeOH++,mol/liter,
Species Aqueous Concentration,H+,mol/liter,
Species Aqueous Concentration,HAlO2(aq),mol/liter,
Species Aqueous Concentration,HCO3-,mol/liter,
Species Aqueous Concentration,K+,mol/liter,
Species Aqueous Concentration,KCl(aq),mol/liter,
Species Aqueous Concentration,Methane(aq),mol/liter,
Species Aqueous Concentration,Mg++,mol/liter,
Species Aqueous Concentration,MgCl+,mol/liter,
Species Aqueous Concentration,MgHCO3+,mol/liter,
Species Aqueous Concentration,Na+,mol/liter,
Species Aqueous Concentration,NaAlO2(aq),mol/liter,
Species Aqueous Concentration,NaCl(aq),mol/liter,
Species Aqueous Concentration,NaHCO3(aq),mol/liter,
Species Aqueous Concentration,NaHSiO3(aq),mol/liter,
Species Aqueous Concentration,O2(aq),mol/liter,
Species Aqueous Concentration,SiO2(aq),mol/liter,
Species Volumetric Concentration,Albite_low,mol/liter,
Species Volumetric Concentration,Calcite,mol/liter,
Species Volumetric Concentration,Dolomite,mol/liter,
Species Volumetric Concentration,Glauconite,mol/liter,
Species Volumetric Concentration,Goethite,mol/liter,
Species Volumetric Concentration,Illite,mol/liter,
Species Volumetric Concentration,K-Feldspar,mol/liter,
Species Volumetric Concentration,Kaolinite,mol/liter,
Species Volumetric Concentration,Oligoclase,mol/liter,
Species Volumetric Concentration,Organic_matter,mol/liter,
Species Volumetric Concentration,Quartz,mol/liter,
Species Volumetric Concentration,Siderite,mol/liter,
Species Volumetric Concentration,Smectite-Ca,mol/liter,
Species Volumetric Concentration,Smectite-Na,mol/liter,
No Restart,,

~Surface Flux Card
3,
Total CO2 Flux,kg/s,kg,Bottom,1,1,1,1,1,1,
Total CO2 Flux,kg/s,kg,Top,1,1,1,1,1,1,
Aqueous Mass Flux,kg/s,kg,Top,1,1,1,1,1,1,
```

5.4.2.2 Solution Control Card

This simulation considers geochemical reactions over a 50,000-year period, starting with a 1-second time step. The time step is allowed to increase slowly with a 1.01 acceleration factor to a maximum of 10 years.

5.4.2.3 Solid Species Card

The density and molecular weight of each mineral species from Table 5.3 are listed in the *Solid Species Card*. The species name must be unique and distinct from aqueous and gas species names (e.g., FeCO3(s), FeCO3_solid, solid FeCO3, FeCO3s).

5.4.2.4 Lithology Card

The specific surface area and volume fraction of each mineral are listed in the *Lithology Card*.

5.4.2.5 Kinetic Reactions Card

The TST rate parameters and equilibrium coefficients for each mineral are listed in the *Kinetic Reactions Card*, along with all aqueous species involved in the dissolution/precipitation reaction. Most equilibrium coefficients for minerals were taken from the EQ3/6 v8.0 database (Wolery and Jarek 2003) with the exception of organic matter, which is assumed to degrade at the forward rate.

5.4.2.6 Aqueous Species Card

The relevant aqueous species for this simulation were determined using EQ3/6 v8.0 (Wolery and Jarek 2003), and must be defined in the *Aqueous Species Card*. 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). 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, molecular weight inputs are not required. This problem uses the B-dot (Helgeson 1969) activity coefficient model.

5.4.2.7 Equilibrium Equations Card

Geochemical models usually assume some reactions to be in equilibrium. This assumption is often justified for some reactions, especially those involving only aqueous species. Equilibrium reactions are not zero-rate reactions but have high reaction rates and reach equilibrium quickly when transport, other reactions, or changes in physical chemical conditions disturb it. Specifically, if the rate of a reaction is much greater than the characteristic time of the problem being solved, it should be classified as an equilibrium reaction. Aqueous species are associated with the defined equilibrium reactions via the *Equilibrium Equations Card*. 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 is distinguished from the other species in the equilibrium equation by being the first species listed for the equilibrium equation. For example, Eqn (5.7), (5.8) and (5.9) are equilibrium reactions; Eqn (5.9) is labeled below as EqRc-8.

5.4.2.8 Equilibrium Reactions Card

The *Equilibrium Reactions Card* specifies the equilibrium reaction constants to be considered in the simulation. 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). This example uses equilibrium coefficients, calculated at 54°C using the EQ3/6 v8.0 database (Wolery and Jarek 2003). Equation 5.7, which defines the dissolution of supercritical or gas phase CO_2 in brine, does not have to be explicitly defined in the *Equilibrium Equations Card*, as it is calculated according to the phase equilibria (Section 2.2.3).

5.4.2.9 Species Link Card

The aqueous CO_2 mass fraction calculated in the coupled flow and transport may be associated with the aqueous species CO2(aq) via the *Species Link Card*. This card 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 CO2, gas CO2, 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 CO2, Gas CO2).

5.4.2.10 Conservation Equations Card

This card specifies the conservation equations to be considered in the simulation. Conservation equations have the following general form:

$$\frac{d\sum\left(a_i\, C_i\right)}{dt} = 0 \qquad (5.13)$$

where C_i is the concentration of species *i* (expressed as aqueous molar concentration), and a_i is the stochiometric coefficient of species *i*, and a_i is the 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 with the species name of a reactive species in the conservation equation (e.g., Total_CO2, Total_H2CO3, Total_H+). This name specification is critical in that it links the named species with the conservation equation, making the concentration for that species the primary unknown for the conservation equation.

5.4.2.11 Initial Conditions Card

The geochemical simulations consider 1 m^3 water-saturated medium. The simulation is assumed to start immediately after injection of CO_2. The CO_2 injection pressure was set at 260 bar. The formation pressure was calculated assuming a depth of 1500 meters and a hydrostatic gradient of 0.0992 bar/m. The initial water chemistry used in the simulation is a pure 1.0 M solution of sodium chloride reacting with the primary minerals listed in Table 5.3 at a temperature of 54 ℃, a pH of 7, and an Eh of -0.1. The dissolved oxygen content calculated by EQ3/6 v8.0 (Wolery and Jarek 2003) for these initial conditions is less than 10^{-57}, and so is not specified in the initial conditions. Any aqueous species not specified in the *Initial Conditions Card* are assumed to be less than 10^{-30}.

5.4.3 Mineral Trapping Simulation Results

The reactant minerals dissolve progressively into the formation water, modifying the water composition and leading to precipitation of product phases, with sequestration of CO_2 within precipitated carbonates. The pH increases with time as primary minerals dissolve and consume H+; dissolved oxygen concentrations remain low with time due to degradation of organic matter that consumes oxygen (Figure 5.15). Due to the precipitation of dolomite and siderite, and moderated by the dissolution of calcite, the total amount of CO_2 sequestered as carbonate minerals in 50,000 years is 3.35 kg/m^3 of formation (Figure 5.16). The precipitation of carbonate minerals is driven by the dissolution of primary silicate minerals illite, glauconite, and oligoclase (Figure 5.16). The dissolution of illite provides Mg^{2+} for the formation of dolomite, glauconite provides Mg^{2+} and Fe^{2+} for the formation of siderite and dolomite, and oligoclase provides Ca^{2+} for the formation of dolomite.

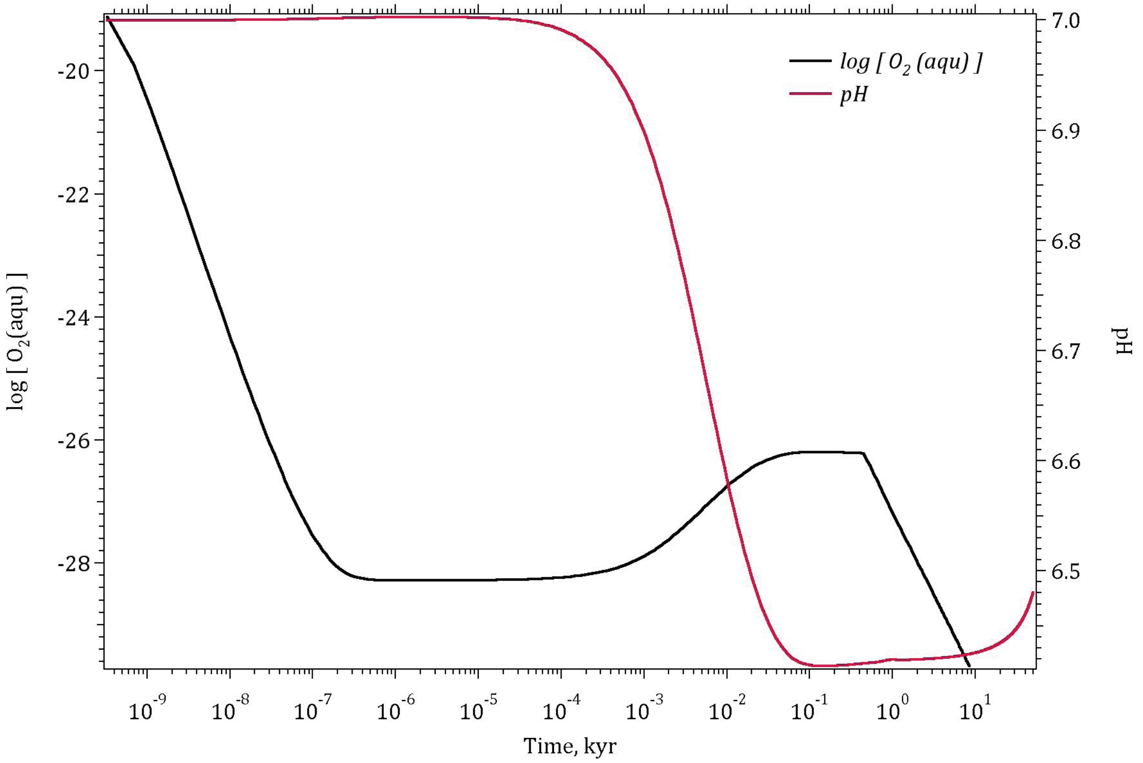

Figure 5.15. Change in pH and Dissolved Oxygen as a Function of Time

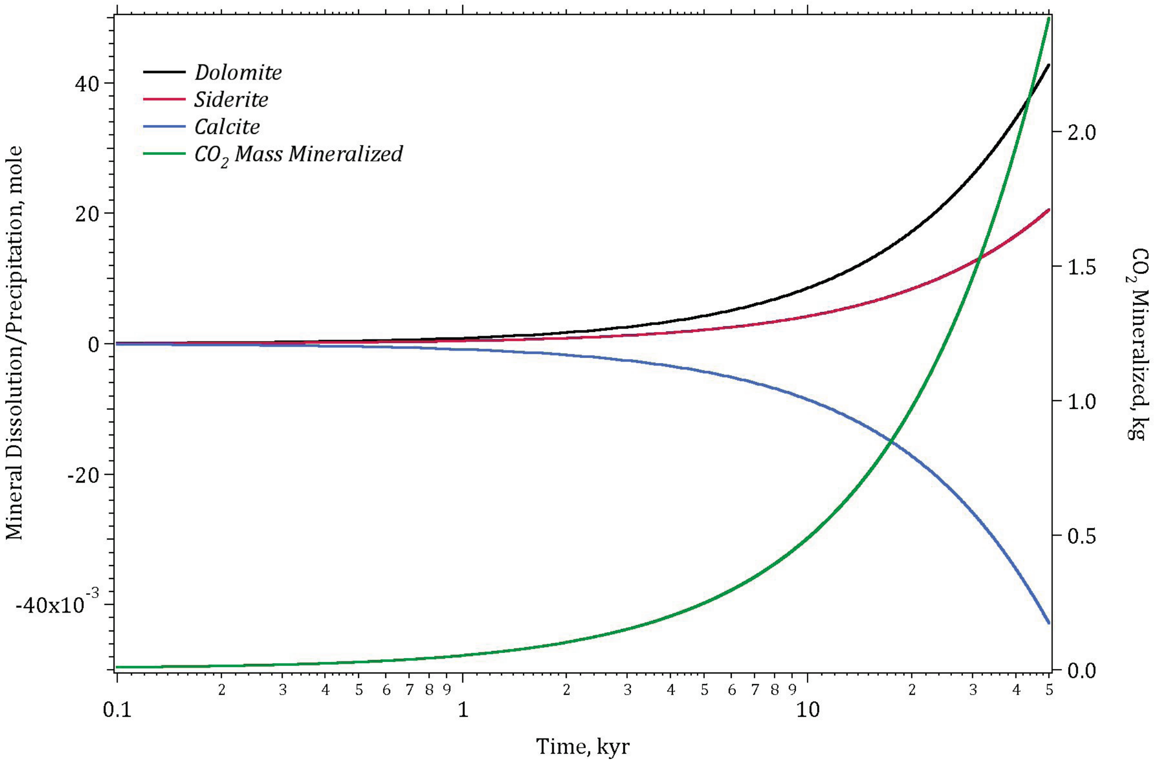

Figure 5.16. Change in Carbonate Minerals as a Function of Time

5.5 CO_2 Injection into a 2-Dimensional Layered Brine Formation

Pressure and buoyancy driven migration of CO_2 injected into a layered formation that is representative of the Sleipner Vest field in the Norwegian sector of the North Sea is investigated. This problem is identical to Problem 7 of the code intercomparison problems developed under the GeoSeq Project (Pruess et al. 2002). A key assumption for the problem, as posed, was isothermal conditions at the formation temperature of 37°C; therefore, STOMP-CO2 was executed for these simulations. The problem involves a constant mass rate injection of $scCO_2$ into a layered saline formation comprising sands and shales. There are five sand layers and four thinner shale layers, whose intrinsic permeability is lower than those of the sands, as shown in Figure 5.17.

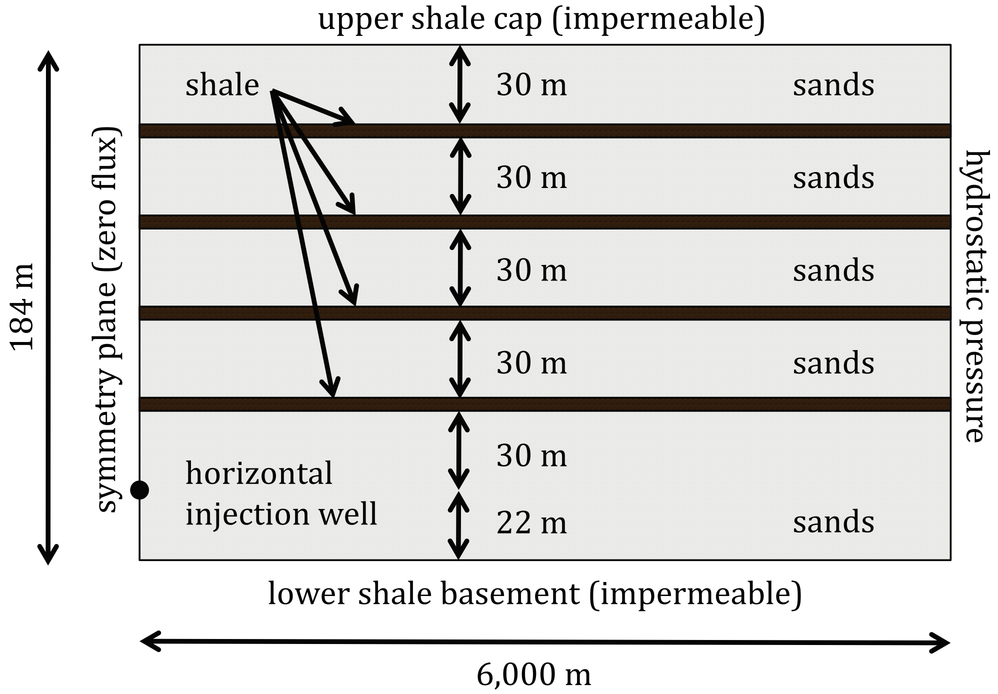

Figure 5.17. Schematic of Injection Reservoir, Showing Location of The Injection Well and Lithology

The system is initialized under hydrostatic conditions with a reference pressure of 110 bar at the well elevation, which are then held throughout the simulation along the right vertical boundary surface (i.e., STOMP-CO2 east boundary). Zero flux boundary surfaces are assumed for the upper horizontal shale cap (i.e., STOMP-CO2 top boundary), lower horizontal shale basement (i.e., STOMP-CO2 bottom boundary), and symmetry plane on the left vertical surface (i.e., STOMP-CO2 west boundary). The gravitational vector is assumed to be pointed vertically down. The domain is 6,000 m in length, 184 m in height, and 1 m in depth. The system was designed to simulate a unit length of a 100-m horizontal injection well where a symmetry plane was assumed in the vertical direction through the center of the well. $scCO_2$ is injected for a 2-yr period at a rate of 0.1585 kg/s, representing a total injection rate for the 100 m of injection well of 1 MMT/yr. Results to be calculated are the distribution of CO_2 mass in the sand layers at 30 days, 1 year, and 2 years, the distribution of CO_2 between the gas and aqueous phases, and the fluxes of CO_2 across the shale layers.

The capillary pressure-saturation relation is described using the van Genuchten formulation (van Genuchten 1980):

$$\bar{s}_l = \frac{s_l - s_{lr}}{1 - s_{lr}} = \left[1 + \left(\alpha\, \beta_{gl}\, h_{gl}\right)^n\right]^{-m} \;;\; m = 1 - \frac{1}{n} \qquad (5.14)$$

The aqueous relative permeability relation is described using the van Genuchten capillary pressure function with the Mualem porosity distribution function (van Genuchten 1980):

$$k_{rl} = \left(\bar{s}_l\right)^{1/2} \left[1 - \left(1 - \left(\bar{s}_l\right)^{1/m}\right)^m\right]^2 \qquad (5.15)$$

The gas relative permeability relation is described using the van Genuchten capillary pressure function with the Mualem porosity distribution function (van Genuchten 1980), with an additional residual gas saturation:

$$k_{rg} = \left(\bar{s}_{gf}\right)^{1/2} \left[\left(1 - \left(1 - \bar{s}_{gf}\right)^{1/m}\right)^m\right]^2 \;;\; s_{gf} = \left(1 - \bar{\bar{s}}_l\right)\left(1 - s_{lr}\right);\; \bar{s}_{gf} = \frac{s_{gf} - s_{gr}}{1 - s_{gr}} \qquad (5.16)$$

Equation 5.16 is the corrected form of the version shown in Table G.2 of the GeoSeq documentation (Pruess et al. 2002). Simulation parameters are shown in Table 5.4.

Table 5.4. Simulation Parameter Values

Parameter Description		Parameter Value
Sands Intrinsic Permeability		3 x 10^{-13} m^2
Shale Intrinsic Permeability		1 x 10^{-14} m^2
Sands Porosity		0.35
Shale Porosity		0.1025
Pore Compressibility		4.5 x 10^{-10} Pa^{-1}
Saturation Function	s_{lr}	0.20
Saturation Function	n	1.667
Sands Saturation Function	α	2.735 m^{-1}
Shale Saturation Function	α	0.158 m^{-1}
Aqu. Rel. Perm.	s_{lr}	0.20
Aqu. Rel. Perm.	m	0.4

Gas. Rel. Perm.	s_{gr}	0.05
Gas Rel. Perm.	s_{lr}	0.4
Initial Aquifer Pressure		hydrostatic w/ 100 bar at top
Initial Aquifer Temperature		37°C
Initial Aquifer Salinity		0.032 wt.% NaCl
CO_2 Injection Rate		0.1585 kg/s

Time stepping and grid spacing were not specified as part of the original GeoSeq problem description, but were left to the discretion of the modeler. For this problem two grid systems were developed: 1) 78z and 2) 193z, with the 78z grid using 78 nodes in the vertical direction and the193z grid using 193 nodes in the vertical direction. Both grids used nonuniform spacing in the horizontal grid using a grid spacing that increased geometrically, starting with an initial spacing of 2 m. An initial time step of 1 s was used, with a time-step acceleration factor of 1.25 for the 2-year simulation period.

5.5.1 78z Input

Because this problem involves isothermal conditions, STOMP-CO2 was selected for the simulations. This section describes the input file for the lower resolution 78z grid. The complete 78z input file is shown in Section 5.5.1.6. A narrative of selected input cards in the sections that follow.

5.5.1.1 Solution Control Card

Hydrostatic conditions were established with the gradient capabilities of the *Initial Condition Card*, without conducting a preliminary simulation to establish hydrostatic conditions. A *Normal* simulation was conducted with an initial time step of 1 s and a maximum time step of 0.1 years. The Newton-Raphson iteration limit was set to 16, and no minimum time step level was set, which implies that after four successive convergence failures the simulator would halt. However, successive convergence failures did not occur in this simulation.

5.5.1.2 Grid Card

The 78z grid used maximum vertical node spacings of 3 m in the sands, and 1 m in the shale. The vertical node spacing in the sands was reduced to 2 m at the shale interface.

5.5.1.3 Initial Conditions Card

The pressure gradients in the *Initial Conditions Card* were established by conducting a zero time step simulation at temperature and pressure conditions of 37°C and 110 bar and an aqueous salt mass fraction of 0.032. This yielded an aqueous density of 1020.61 kg/m^3. This density was then converted to a pressure gradient by multiplying by the acceleration of gravity (i.e., 9.81 m/s^2) and converting to bar/m.

5.5.1.4 Boundary Condition Card

All boundary surfaces except for the right vertical boundary were zero flux boundaries. The right boundary was maintained under hydrostatic conditions by using the initial hydrostatic conditions as the boundary condition on the vertical surface, by specifying *Initial Condition* boundary conditions for the *Aqueous*, *Gas*, and *Salt Boundary Types*.

5.5.1.5 Surface Flux Card

The *Integrated CO2 Mass* output provides data about the integrated CO_2 mass in the entire domain. To determine the amount of CO_2 mass in each of the sands horizons the *Surface Flux Card* was used to track CO_2 mass crossing each of the shale layers. The surface normal for a top surface is in the positive *z* direction, which means that upward fluxes are positive. The first two defined surfaces track the rates and integrated amounts of CO_2 crossing the lower and upper surfaces of the lowest shale layer.

5.5.1.6 78z Input File

```
~Simulation Title Card
1,
Problem 7,
M.D. White,
Pacific Northwest Laboratory,
24 January 2012,
09:11 PST,
20,
Intercomparison of simulation models for CO2 disposal in
underground storage reservoirs.
Test Problem 7: CO2 Injection into a 2D Layered Brine Formation
This test problem is patterned after the CO2 injection project
at the Sleipner Vest field in the Norwegian sector of the North Sea,
and is intended to investigate the dominant physical processes
associated with the injection of supercritical CO2 into a layered
medium. Significant simplifications have been made, the most important
of which is the assumption of isothermal conditions (37 °C, the
ambient temperature of the formation). CO2 injection rates
(1,000,000 tonnes per year), system geometry, and system permeabilities
correspond approximately to those at Sleipner, although no attempt was
made to represent details of the permeability structure within the
host formation. Injection of the supercritical CO2, which is less
dense than the saline formation waters into which it is injected,
causes it to rise through the formation. Its rate of ascent, however,
is limited by the presence of four relatively low permeability shales.
The top and bottom of the formation is assumed to be impermeable.
The only reactive chemistry considered in this problem is the
dissolution of CO2 in the aqueous phase. (Pruess and Garcia, 2000).

~Solution Control Card
Normal,
STOMP-CO2,
1,
0,yr,2,yr,1.0,s,0.1,yr,1.25,16,1.e-06,
10000,
Variable Aqueous Diffusion,
Variable Gas Diffusion,
0,

~Grid Card
Cartesian,
100,1,78,
0.0,m,2.0,m,4.1,m,6.3,m,8.6,m,11.1,m,
13.7,m,16.4,m,19.2,m,22.2,m,25.4,m,28.7,m,
32.2,m,35.8,m,39.7,m,43.8,m,48.0,m,52.5,m,
57.2,m,62.2,m,67.4,m,72.9,m,78.7,m,84.8,m,
91.2,m,97.9,m,105.0,m,112.4,m,120.2,m,128.5,m,
137.1,m,146.2,m,155.8,m,165.9,m,176.5,m,187.6,m,
199.4,m,211.7,m,224.7,m,238.3,m,252.7,m,267.8,m,
283.6,m,300.3,m,317.9,m,336.4,m,355.8,m,376.3,m,
397.8,m,420.4,m,444.2,m,469.2,m,495.5,m,523.2,m,
552.3,m,582.9,m,615.1,m,649.0,m,684.7,m,722.2,m,
761.6,m,803.1,m,846.7,m,892.6,m,940.8,m,991.6,m,
1045.0,m,1101.1,m,1160.2,m,1222.3,m,1287.7,m,1356.4,m,
1428.7,m,1504.8,m,1584.8,m,1668.9,m,1757.4,m,1850.5,m,
1948.4,m,2051.4,m,2159.7,m,2273.7,m,2393.5,m,2519.6,m,
2652.2,m,2791.6,m,2938.3,m,3092.6,m,3254.9,m,3425.6,m,
3605.2,m,3794.0,m,3992.7,m,4201.6,m,4421.4,m,4652.5,m,
4895.7,m,5151.4,m,5420.4,m,5703.4,m,6000.0,m,
0.0,m,1.0,m,
0.0,m,7@3.0,m,1@2.0,m,9@3.0,m,1@2.0,m,
3@1.0,m,
1@1.0,m,1@2.0,m,8@3.0,m,1@2.0,m,1@1.0,m,
3@1.0,m,
1@1.0,m,1@2.0,m,8@3.0,m,1@2.0,m,1@1.0,m,
3@1.0,m,
1@1.0,m,1@2.0,m,8@3.0,m,1@2.0,m,1@1.0,m,
3@1.0,m,
1@1.0,m,1@2.0,m,8@3.0,m,1@2.0,m,1@1.0,m,

~Rock/Soil Zonation Card
9,
Sands,1,100,1,1,1,18,
Shale,1,100,1,1,19,21,
Sands,1,100,1,1,22,33,
Shale,1,100,1,1,34,36,
Sands,1,100,1,1,37,48,
Shale,1,100,1,1,49,51,
Sands,1,100,1,1,52,63,
Shale,1,100,1,1,64,66,
Sands,1,100,1,1,67,78,

~Mechanical Properties Card
Sands,2650,kg/m^3,0.35,0.35,Compressibility,4.5e-10,1/Pa,,,Millington and Quirk,
Shale,2650,kg/m^3,0.1025,0.1025,Compressibility,4.5e-10,1/Pa,,,Millington and Quirk,

~Hydraulic Properties Card
Sands,3.e-12,m^2,3.e-12,m^2,3.e-12,m^2,
Shale,1.e-14,m^2,1.e-14,m^2,1.e-14,m^2,

~Saturation Function Card
Sands,van Genuchten,2.735,1/m,1.667,0.20,0.4,0.0,
Shale,van Genuchten,0.158,1/m,1.667,0.20,0.4,0.0,

~Aqueous Relative Permeability Card
Sands,Mualem Irreducible,0.4,0.20,
Shale,Mualem Irreducible,0.4,0.20,

~Gas Relative Permeability Card
Sands,van Genuchten,0.4,0.05,
Shale,van Genuchten,0.4,0.05,

~Salt Transport Card
Sands,0.0,m,0.0,m,
Shale,0.0,m,0.0,m,

~Initial Conditions Card
Gas Pressure,Aqueous Pressure,
4,
Gas Pressure,112.0525,Bar,,,,,-0.1001218,1/m,1,100,1,1,1,78,
Aqueous Pressure,112.0525,Bar,,,,,-0.1001218,1/m,1,100,1,1,1,78,
Temperature,37.0,C,,,,,,,1,100,1,1,1,78,
Salt Mass Fraction,0.032,,,,,,,,1,100,1,1,1,78,
~Boundary Conditions Card
1,
East,Aqueous Initial Condition,Gas Initial Condition,Aqueous Initial Condition,
100,100,1,1,1,78,1,
0,s,,,,,,,,

~Source Card
1,
Gas Mass Rate,Water-Vapor Mass Fraction,1,1,1,1,8,8,1,
0,s,112.0525,bar,0.1585,kg/s,0.0,

~Output Options Card
4,
1,1,8,
1,1,18,
1,1,33,
1,1,48,
1,1,s,m,6,6,6,
8,
Gas Saturation,,
CO2 Gas Mass Fraction,,
CO2 Aqueous Mass Fraction,,
Gas Pressure,Pa,
Aqueous Density,kg/m^3,
Integrated CO2 Mass,kg,
Integrated Aqueous CO2 Mass,kg,
Integrated Gas CO2 Mass,kg,
2,
30,day,
1,year,
7,
Gas Saturation,,
CO2 Gas Mass Fraction,,
CO2 Aqueous Mass Fraction,,
Gas Pressure,Pa,
Diffusive Porosity,,
Gas Density,kg/m^3,
Aqueous Density,kg/m^3,

~Surface Flux Card
5,
Total CO2 Flux,kg/s,kg,Top,1,100,1,1,18,18,
Total CO2 Flux,kg/s,kg,Top,1,100,1,1,21,21,
Total CO2 Flux,kg/s,kg,Top,1,100,1,1,36,36,
Total CO2 Flux,kg/s,kg,Top,1,100,1,1,51,51,
Total CO2 Flux,kg/s,kg,Top,1,100,1,1,66,66,
```

5.5.2 78z Results

Simulation results from other modeling groups and numerical simulators for this problem are reported in Pruess et al. (2002). scCO_2 injected into the system enters the domain beneath the first shale layer and then migrates both horizontally and vertically under pressure gradient and buoyancy forces. The migration pattern is controlled by both the pressure and buoyancy driving forces and the contrast in entry pressure and intrinsic permeability between the sand and shale layers. Gas saturation profiles at 30 days, 1 year and 2 years after the start of injection are shown in Figure 5.18. Saturation profiles reported in Pruess et al. (2002) for the same points in time are shown in Figure 5.19. The GeoSeq results show slightly more extended plumes beneath each of the shale layers compared with the STOMP-CO2 results. Otherwise there is good agreement between the simulations. The pressure distribution in the domain is a function of the CO_2 injection rate, the overall resistance of the injected CO_2 to displace the formation brine, the phase relative permeabilities, and formation intrinsic permeabilities. The pressure distribution after two years is shown in Figures 5.20 and 5.21 for the STOMP-CO2 and TOUGH2/ECO2 simulations respectively. The distribution of CO_2 mass in the domain over time is shown in Figure 5.22. At the end of the simulation, 0.149 of the CO_2 mass occurs as dissolved in the aqueous phase. The TOUGH2/ECO2 simulator predicted 0.215 of the CO_2 mass to be dissolved in the aqueous phase. The higher aqueous mass in the TOUGH2/ECO2 simulation is probably due to the larger lateral spread beneath the shale layers. The amount of CO_2 mass in each of the sand layers is shown in Figures 5.23 and 5.24 for the STOMP-CO2 and GeoSeq simulations. There is good agreement between the total amounts of CO_2 in the sand levels, but the arrival times are advanced in the STOMP-CO2 simulations. The arrival times were determined to be dependent on the gas relative permeability model. The model function reported in Pruess et al. (2002) was not a standard form. When this equation was implemented in STOMP-CO2 the simulation results showed poor agreement with those for the GeoSeq simulations (Pruess et al. 2002). As a result, the more conventional form of the gas relative permeability function shown in Eqn. (5.17) was implemented for the simulation results shown.

$$k_{rg} = \left(\bar{s}_g\right)^{1/2} \left[\left(1 - \left(1 - \bar{s}_g\right)^{1/m}\right)^m\right]^2 \; ; \; \bar{s}_g = \frac{s_g - s_{gr}}{1 - s_{gr}} \qquad (5.17)$$

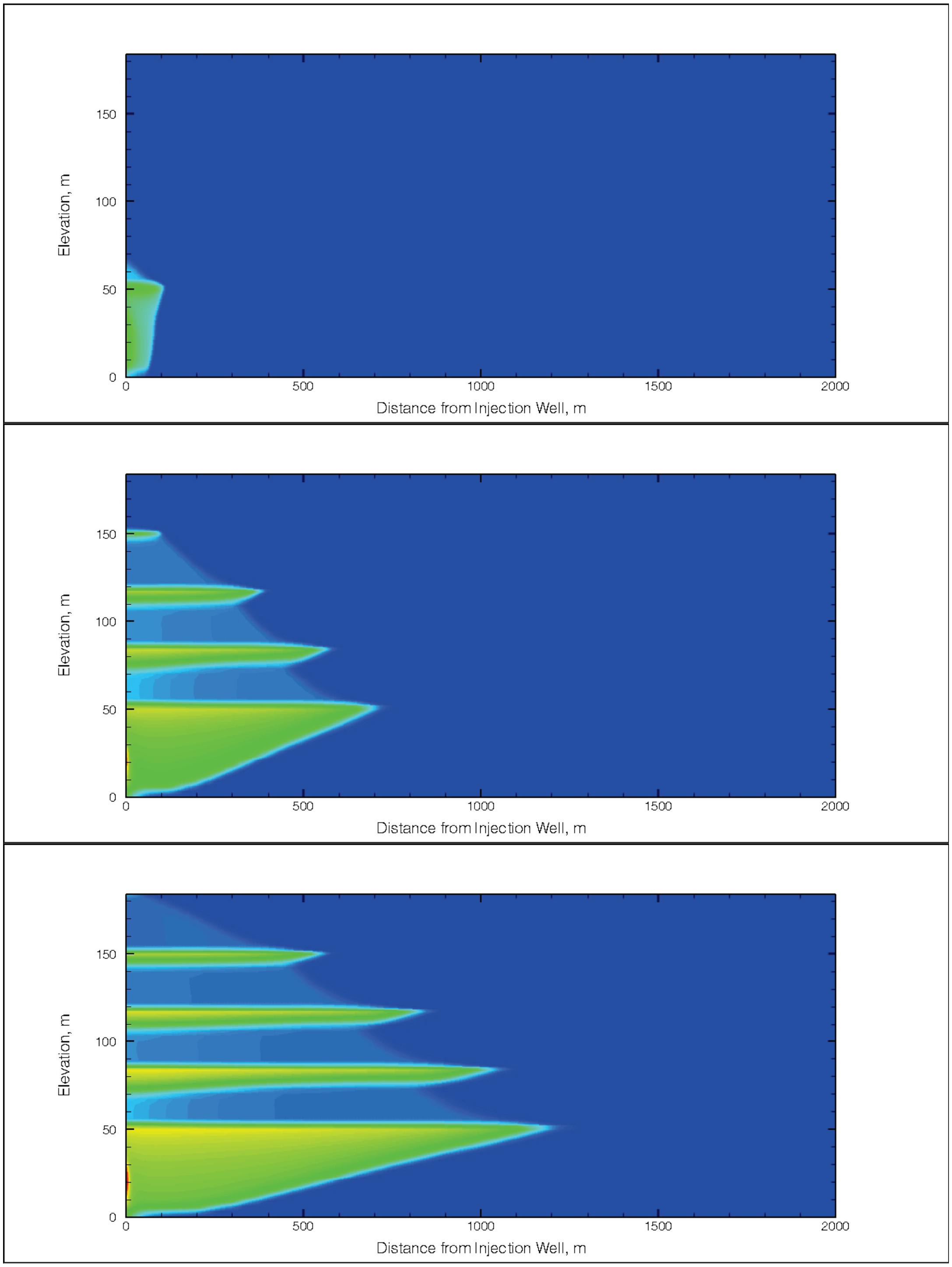

Figure 5.18. Gas Saturation from STOMP-CO2 at 30 Days, 1 Year, and 2 Years

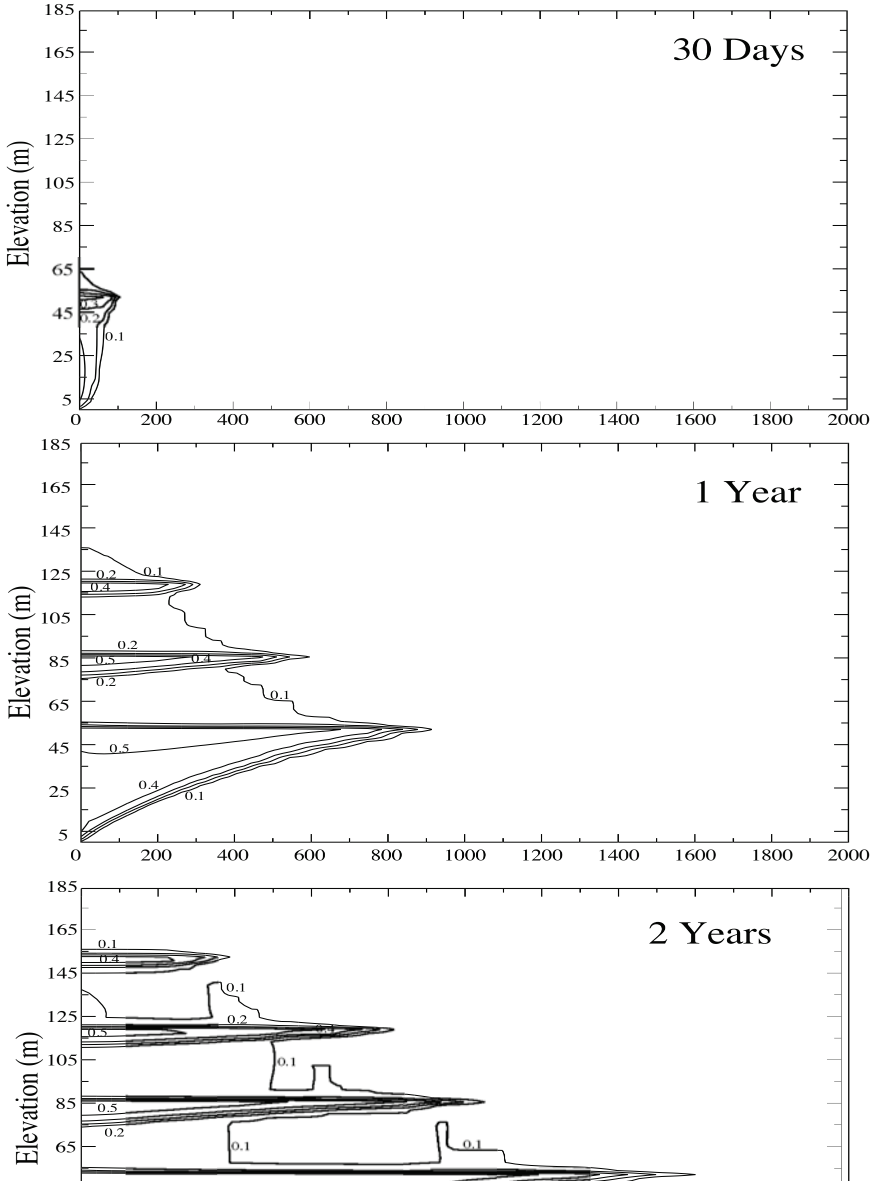

Figure 5.19. Gas Saturation from TOUGH2 at 30 Days, 1 Year, and 2 Years from Pruess et el. (2002)

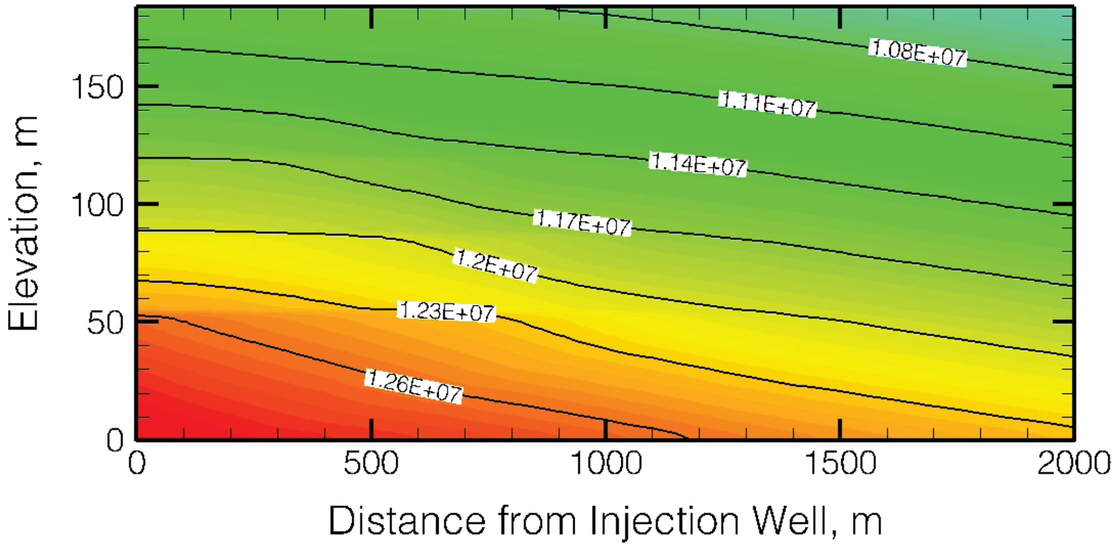

Figure 5.20. Pressure Distribution from STOMP-CO2 after 2 Years of CO_2 Injection

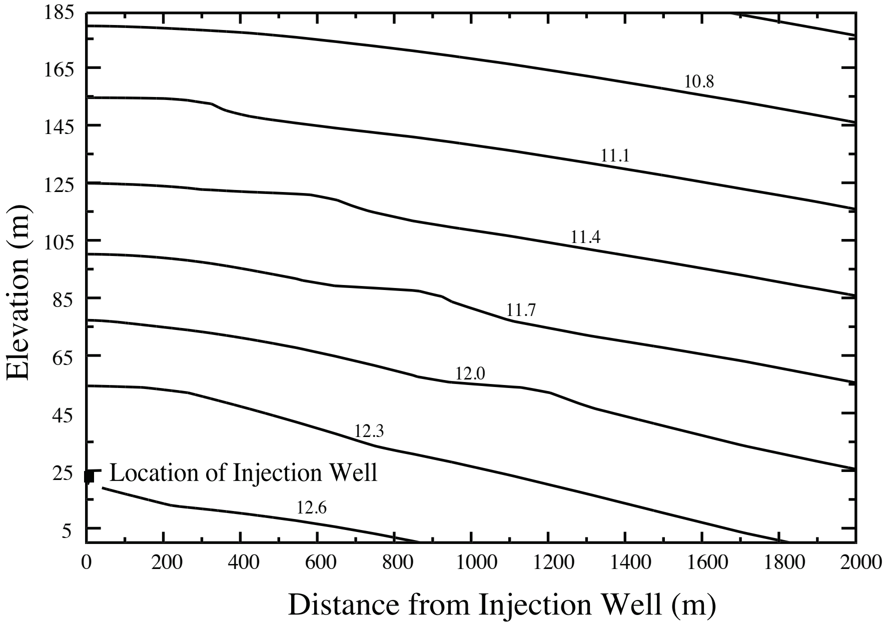

Figure 5.21. Pressure Distribution from TOUGH2 after 2 Years of CO_2 Injection from Pruess et al. (2002)

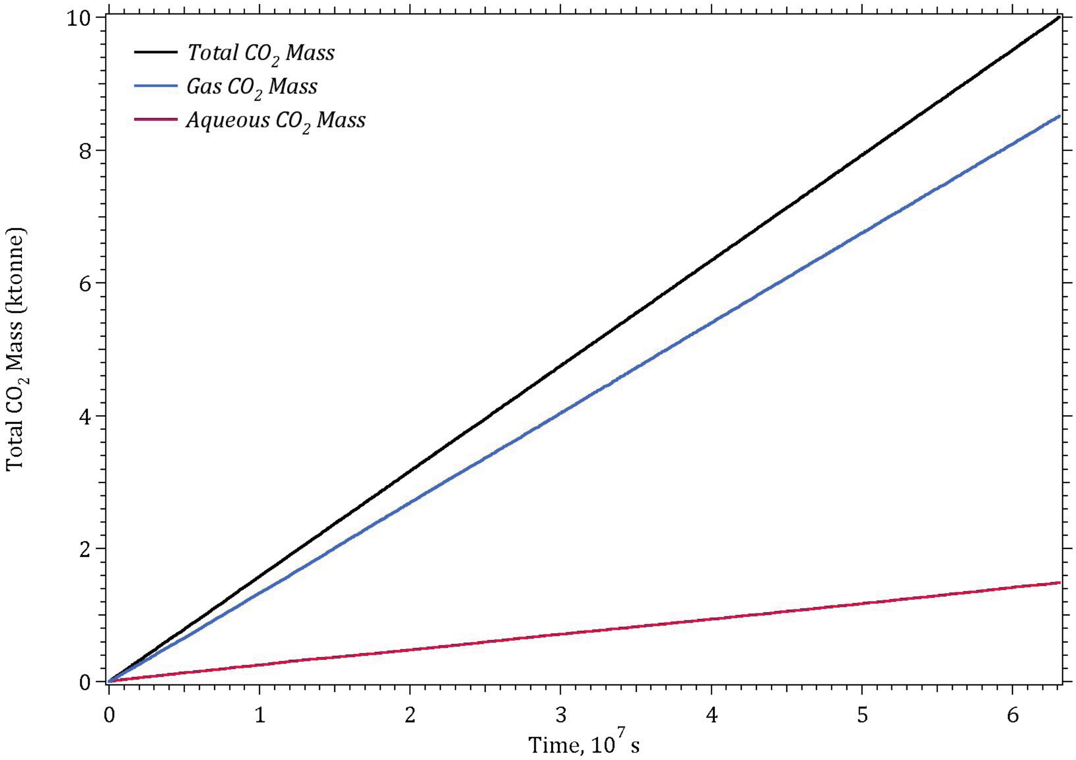

Figure 5.22. Injected CO_2 Mass Distribution from STOMP-CO2 as a Function of Time

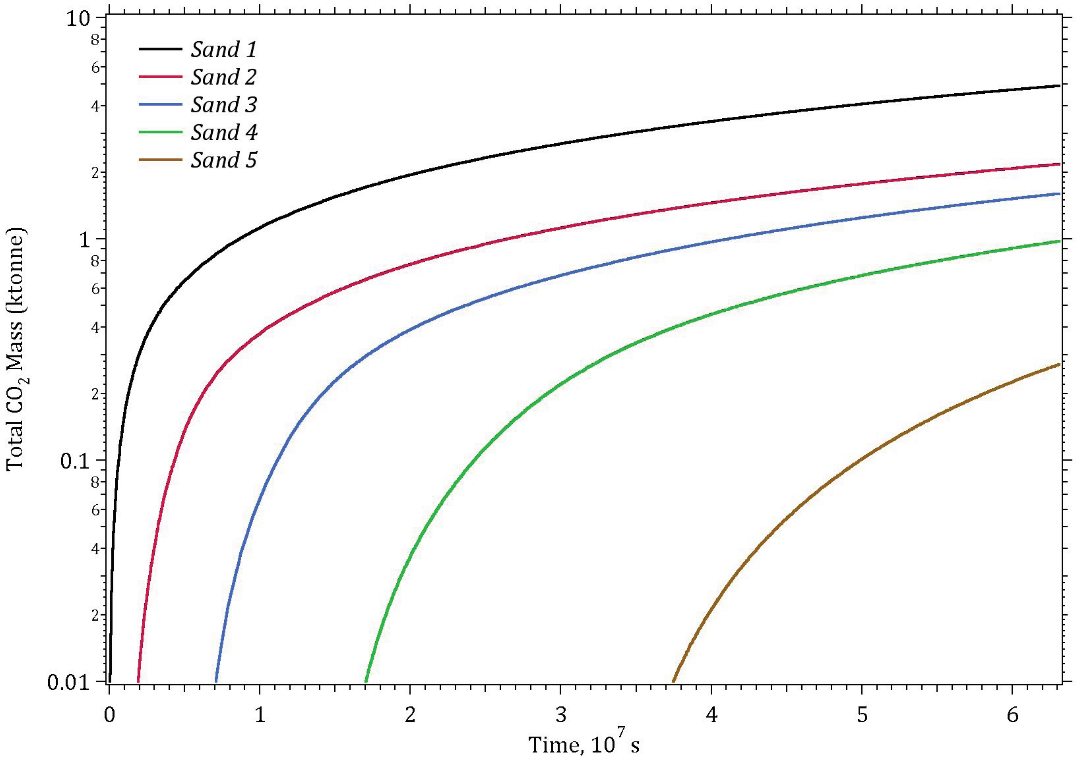

Figure 5.23. Total CO_2 in Sand Horizons from STOMP-CO2 as a Function of Time

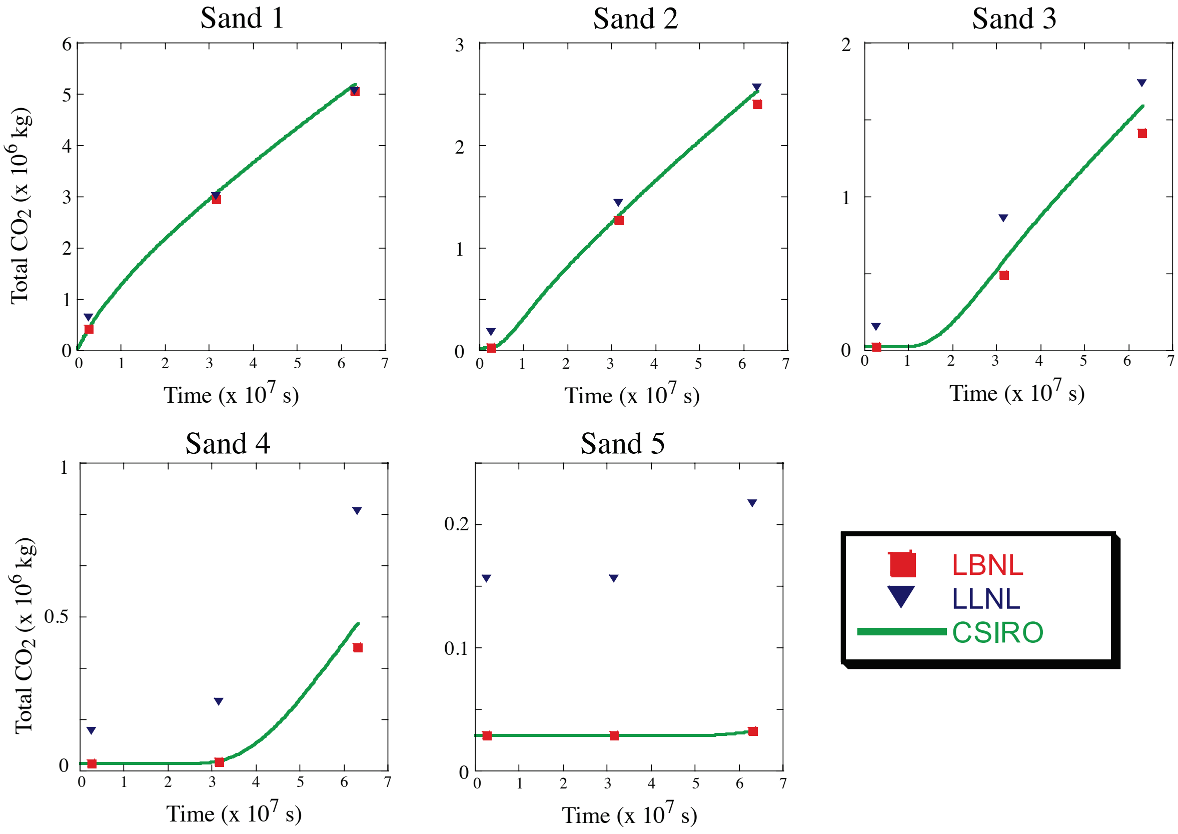

Figure 5.24. Total CO_2 in Sand Horizons from LBNL (TOUGH2/ECO2), LLNL (NUFT), and CSIRO (TOUGH2/ECO2) from Pruess et al. (2002)

5.5.3 193z Input

To examine whether the simulation results were dependent on the selected vertical grid spacing, a second simulation was conducted that used more nodes in the vertical direction. This domain used uniform 1-m vertical grid spacing with 0.5-m grid spacing at the sand-shale interfaces. The complete 193z input file is shown in Section 5.5.3.1.

5.5.3.1 193z Input File

```
~Simulation Title Card
1,
Problem 7,
M.D. White,
Pacific Northwest Laboratory,
24 January 2012,
09:11 PST,
20,
Intercomparison of simulation models for CO2 disposal in
underground storage reservoirs.
Test Problem 7: CO2 Injection into a 2D Layered Brine Formation
This test problem is patterned after the CO2 injection project
at the Sleipner Vest field in the Norwegian sector of the North Sea,
and is intended to investigate the dominant physical processes
associated with the injection of supercritical CO2 into a layered
medium. Significant simplifications have been made, the most important
of which is the assumption of isothermal conditions (37  ℃, the
ambient temperature of the formation). CO2 injection rates
(1,000,000 tonnes per year), system geometry, and system permeabilities
correspond approximately to those at Sleipner, although no attempt was
made to represent details of the permeability structure within the
host formation. Injection of the supercritical CO2, which is less
dense than the saline formation waters into which it is injected,
causes it to rise through the formation. Its rate of ascent, however,
is limited by the presence of four relatively low permeability shales.
The top and bottom of the formation is assumed to be impermeable.
The only reactive chemistry considered in this problem is the
dissolution of CO2 in the aqueous phase. (Pruess and Garcia, 2000).

~Solution Control Card
Normal,
STOMP-CO2,
1,
0,yr,2,yr,1.0,s,0.1,yr,1.25,16,1.e-06,
10000,
Variable Aqueous Diffusion,
Variable Gas Diffusion,
0,

~Grid Card
Cartesian,
100,1,193,
0.0,m,2.0,m,4.1,m,6.3,m,8.6,m,11.1,m,
13.7,m,16.4,m,19.2,m,22.2,m,25.4,m,28.7,m,
32.2,m,35.8,m,39.7,m,43.8,m,48.0,m,52.5,m,
57.2,m,62.2,m,67.4,m,72.9,m,78.7,m,84.8,m,
91.2,m,97.9,m,105.0,m,112.4,m,120.2,m,128.5,m,
137.1,m,146.2,m,155.8,m,165.9,m,176.5,m,187.6,m,
199.4,m,211.7,m,224.7,m,238.3,m,252.7,m,267.8,m,
283.6,m,300.3,m,317.9,m,336.4,m,355.8,m,376.3,m,
397.8,m,420.4,m,444.2,m,469.2,m,495.5,m,523.2,m,
552.3,m,582.9,m,615.1,m,649.0,m,684.7,m,722.2,m,
761.6,m,803.1,m,846.7,m,892.6,m,940.8,m,991.6,m,
1045.0,m,1101.1,m,1160.2,m,1222.3,m,1287.7,m,1356.4,m,
1428.7,m,1504.8,m,1584.8,m,1668.9,m,1757.4,m,1850.5,m,
1948.4,m,2051.4,m,2159.7,m,2273.7,m,2393.5,m,2519.6,m,
2652.2,m,2791.6,m,2938.3,m,3092.6,m,3254.9,m,3425.6,m,
3605.2,m,3794.0,m,3992.7,m,4201.6,m,4421.4,m,4652.5,m,
4895.7,m,5151.4,m,5420.4,m,5703.4,m,6000.0,m,
0.0,m,1.0,m,
0.0,m,1@0.5,m,51@1.0,m,1@0.5,m,
1@0.5,m,2@1.0,m,1@0.5,m,
1@0.5,m,29@1.0,m,1@0.5,m,
1@0.5,m,2@1.0,m,1@0.5,m,
1@0.5,m,29@1.0,m,1@0.5,m,
1@0.5,m,2@1.0,m,1@0.5,m,
1@0.5,m,29@1.0,m,1@0.5,m,
1@0.5,m,2@1.0,m,1@0.5,m,
1@0.5,m,29@1.0,m,1@0.5,m,

~Rock/Soil Zonation Card
9,
Sands,1,100,1,1,1,53,
Shale,1,100,1,1,54,57,
Sands,1,100,1,1,58,88,
Shale,1,100,1,1,89,92,
Sands,1,100,1,1,93,123,
Shale,1,100,1,1,124,127,
Sands,1,100,1,1,128,158,
Shale,1,100,1,1,159,162,
Sands,1,100,1,1,163,193,

~Mechanical Properties Card
Sands,2650,kg/m^3,0.35,0.35,Compressibility,4.5e-10,1/Pa,,,Millington and Quirk,
Shale,2650,kg/m^3,0.1025,0.1025,Compressibility,4.5e-10,1/Pa,,,Millington and Quirk,

~Hydraulic Properties Card
Sands,3.e-12,m^2,3.e-12,m^2,3.e-12,m^2,
Shale,1.e-14,m^2,1.e-14,m^2,1.e-14,m^2,

~Saturation Function Card
Sands,van Genuchten,2.735,1/m,1.667,0.20,0.4,0.0,
Shale,van Genuchten,0.158,1/m,1.667,0.20,0.4,0.0,

~Aqueous Relative Permeability Card
Sands,Mualem Irreducible,0.4,0.20,
Shale,Mualem Irreducible,0.4,0.20,

~Gas Relative Permeability Card
Sands,van Genuchten,0.4,0.05,
Shale,van Genuchten,0.4,0.05,

~Salt Transport Card
Sands,0.0,m,0.0,m,
Shale,0.0,m,0.0,m,

~Initial Conditions Card
Gas Pressure,Aqueous Pressure,
4,
Gas Pressure,112.17765,Bar,,,,,-0.1001218,1/m,1,100,1,1,1,193,
Aqueous Pressure,112.17765,Bar,,,,,-0.1001218,1/m,1,100,1,1,1,193,
Temperature,37.0,C,,,,,,,1,100,1,1,1,193,
Salt Mass Fraction,0.032,,,,,,,,1,100,1,1,1,193,

~Boundary Conditions Card
1,
East,Aqueous Initial Condition,Gas Initial Condition,Aqueous Initial Condition,
100,100,1,1,1,193,1,
0,s,,,,,,,,,

~Source Card
1,
Gas Mass Rate,Water-Vapor Mass Fraction,1,1,1,1,23,23,1,
0,s,110,bar,0.1585,kg/s,0.0,
~Output Options Card
4,
1,1,23,
1,1,53,
1,1,54,
1,1,88,
1,1,s,m,6,6,6,
8,
Gas Saturation,,
CO2 Gas Mass Fraction,,
CO2 Aqueous Mass Fraction,,
Gas Pressure,Pa,
Aqueous Density,kg/m^3,
Integrated CO2 Mass,kg,
Integrated Aqueous CO2 Mass,kg,
Integrated Gas CO2 Mass,kg,
2,
30,day,
1,year,
7,
Gas Saturation,,
CO2 Gas Mass Fraction,,
CO2 Aqueous Mass Fraction,,
Gas Pressure,Pa,
Diffusive Porosity,,
Gas Density,kg/m^3,
Aqueous Density,kg/m^3,

~Surface Flux Card
5,
Total CO2 Flux,kg/s,kg,Top,1,100,1,1,53,53,
Total CO2 Flux,kg/s,kg,Top,1,100,1,1,57,57,
Total CO2 Flux,kg/s,kg,Top,1,100,1,1,92,92,
Total CO2 Flux,kg/s,kg,Top,1,100,1,1,127,127,
Total CO2 Flux,kg/s,kg,Top,1,100,1,1,162,162,
```

5.5.4 193z Results

The gas saturation distribution at 30 days, 1 year and 2 years are shown in Figure 5.25. Comparing these profiles with those from the 78z simulation reveals only slight changes in the gas saturation distribution, with more extension of the gas phase beneath the shale layers. Simulation results for the distribution of CO_2 mass between phases and sands layers also show only slight differences compared to the 78z simulation. As the differences in the simulation results between the 78z and 193z simulations are only slight, the 78z simulations are considered to have sufficient grid resolution.

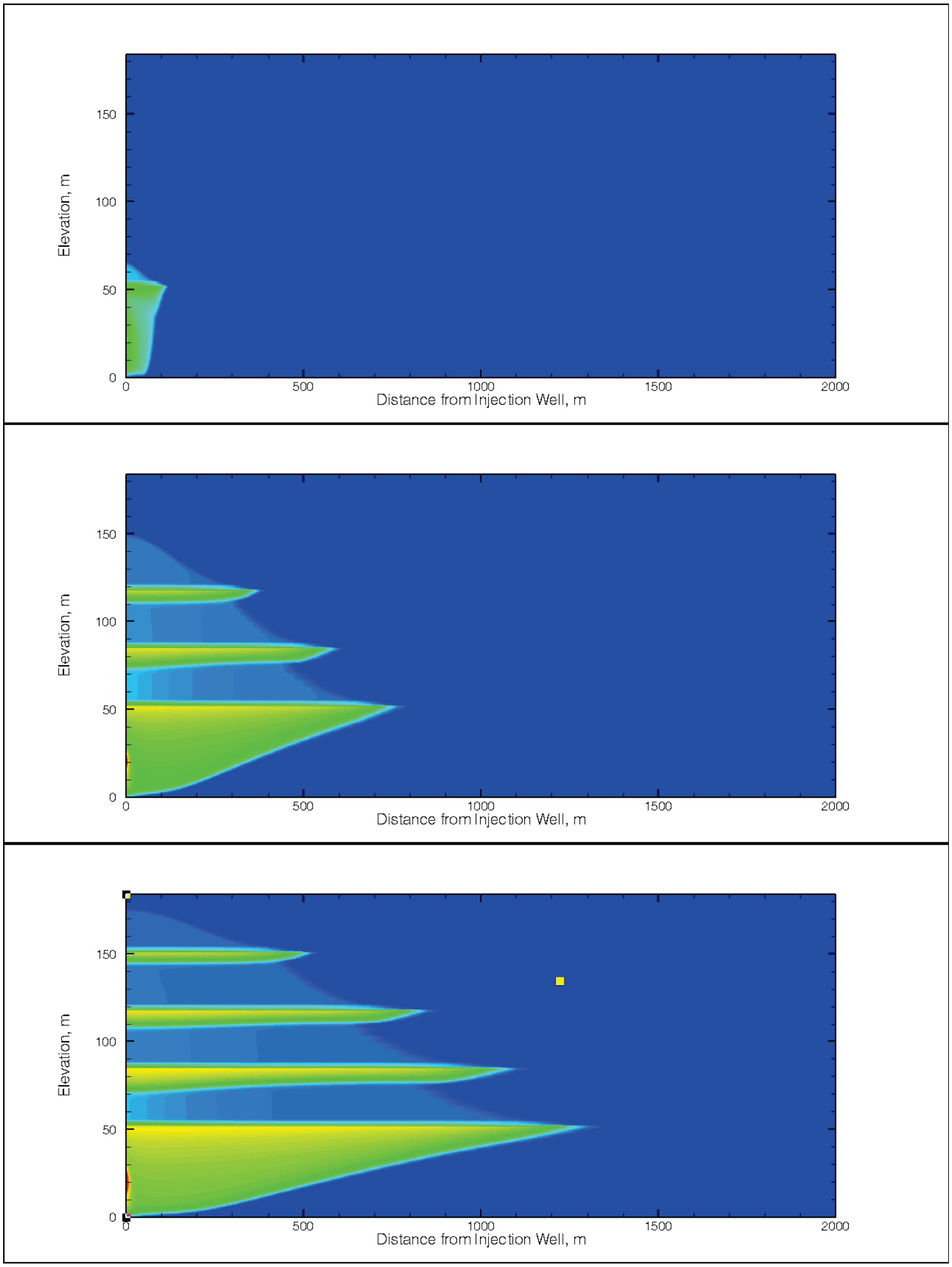

Figure 5.25. Gas Saturation from STOMP-CO2 at 30 Days, 1 Year, and 2 Years

5.6 CO_2 Plume Evolution and Leakage through an Abandoned Well

This problem involves the injection of $scCO_2$ into a saline formation with an abandoned well that provides a conduit for CO_2 migration between lower and upper permeable layers. Two scenarios are considered: 1) deep conditions where the entire domain remains under supercritical temperature and pressure conditions for CO_2, and 2) shallow conditions where the upper portion of the domain is outside of supercritical temperature and pressure conditions for CO_2. This problem is identical to Problem 1 from the series of problems developed at the University of Stuttgart (Ebigbo et al., 2007a,b), entitled "Numerical Investigations of CO_2 Sequestration in Geological Formations: Problem-Oriented Benchmarks." This problem was developed using analytical and semi-analytical solutions published by Nordbotten et al. (2004, 2005a,b). The deep scenario involves isothermal conditions and was executed with STOMP-CO2. The shallow scenario considered nonisothermal effects and was executed with STOMP-CO2e.

5.6.1 Problem Description

CO_2 is injected into a saline reservoir; spreads within the reservoir under pressure gradient and buoyancy forces and, upon reaching a leaky well, rises up to a more shallow, saline aquifer. The goal of the simulation is to quantify the leakage rate, which depends on the pressure build-up in the aquifer due to injection and on the plume evolution. This scenario is shown in Figure 5.26. The simulation domain has a lateral extent of 1,000 m x 1,000 m. At the lateral boundaries, constant boundary conditions are imposed on the system. The leaky well is at the center of the domain and the injection well is 100 m away. Both aquifers are 30 m thick and the aquitard has a thickness of 100 m. The leaky well is modeled as a porous medium with a higher permeability than the formation. The vertical plane passing through the injection and leaky well is a plane of symmetry for the problem, which allows the problem to be modeled using a half domain. For the deep domain problem the domain top is at a depth of 2840 m, and for the shallow domain problem, the domain top is at a depth of 640 m. An overview of the domain geometries for the deep and shallow domains is given in Table 5.5.

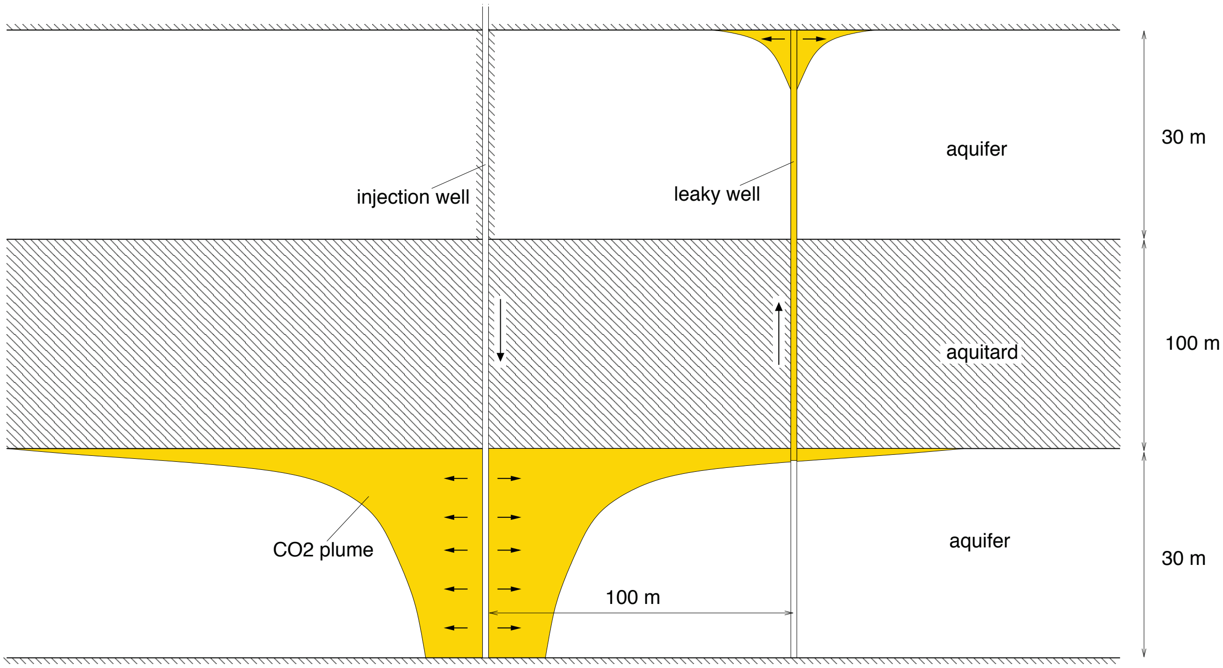

Figure 5.26. Problem Domain from Ebigbo et al. (2007a,b)

Table 5.5. Domain Geometry

Parameter	Deep Domain	Shallow Domain
Domain Depth Range	2840-3000 m	640-800 m
Aquifer Thickness	30 m	30 m
Aquitard Thickness	100 m	100 m
Computational Domain	1000 m x 500 m x 160 m	1000 m x 500 m x 160 m
Distance between Wells	100 m	100 m
Well Radii	0.15 m	0.15 m

For the deep domain problem a number of simplifying assumptions were made. The most significant being constant gas and aqueous densities and viscosities, isothermal conditions, and zero solubility for CO_2 in the aqueous phase and water in the gas phase. Secondary assumptions were zero capillary pressure. The STOMP-CO2 and –CO2e simulators use phase pressure primary variables which makes a capillary pressure saturation function necessary for computing phase saturations. The Brooks and Corey (1966) saturation function (Eqn. 2.44) was used for both the deep and shallow domain problems. A linear relative permeability function was used for the deep domain problem:

$$\begin{aligned} k_{rg} &= s_g \\ k_{rl} &= s_l \end{aligned} \tag{5.18}$$

For the shallow domain problem the Burdine (1953) porosity distribution model and Brooks and Corey saturation function was used for the phase relative permeability model:

$$\begin{aligned} &k_{rl} = \left(\bar{s}_l\right)^{\left(3+\frac{2}{\lambda}\right)}; \; \bar{s}_l = \frac{s_l - s_{lr}}{1 - s_{lr}} \\ &k_{rg} = \left(\bar{s}_g\right)^2 \left[1 - \left(\bar{s}_l\right)^{\left(1+\frac{2}{\lambda}\right)}\right]; \; \bar{s}_l = \frac{s_l - s_{lr}}{1 - s_{lr}}; \; \bar{s}_g = \frac{s_g - s_{gr}}{1 - s_{gr}} \end{aligned} \tag{5.19}$$

Intrinsic properties for the aquifer and leaky well are given in Table 5.6. The aquitard was considered to be impermeable and was modeled using inactive nodes.

Table 5.6. Porous Media Intrinsic Properties

Parameter	Deep Domain	Shallow Domain
Porosity	0.15	0.15
Aquifer Permeability	2 x 10^{-14} m^2	2 x 10^{-14} m^2
Leaky Well Permeability	1 x 10^{-12} m^2	1 x 10^{-12} m^2
Brooks-Corey Entry Pressure	10^4 Pa	10^4 Pa
Brooks-Corey λ	2.0	2.0
Residual Aqueous Saturation	0.0	0.2
Residual Gas Saturation	0.0	0.05

The STOMP-CO2 and -CO2e simulators computed gas and aqueous density, viscosity, enthalpy, and component concentrations according to the algorithms in Section 2. In the deep domain problem, constant gas and aqueous density and viscosity were specified as shown in Table 5.7. To bypass the standard fluid property routines, special options were invoked to use constant properties. Specification of these options is given below in the input card descriptions.

Table 5.7. Fluid Properties

Parameter	Deep Domain	Shallow Domain
Gas Density	479 kg/m^3	Eqn. (2.21)
Aqueous Density	1045 kg/m^3	Eqn. (2.20)
Gas Viscosity	3.950 x 10^{-5} Pa s	Eqn. (2.34)
Aqueous Viscosity	2.535 x 10^{-4} Pa s	Eqn. (2.33)
Gas Enthalpy	isothermal	Eqn. (2.27)
Aqueous Enthalpy	isothermal	Eqn. (2.24)
Gas H_2O Fraction	0.0	Eqn. (2.9)
Aqueous CO_2 Fraction	0.0	Eqn. (2.9)

Injection of CO_2 into the system was specified using the coupled well model (Section 2.8), which allows the user to specify both an injection rate and a maximum injection pressure. This well model will solve for the injection pressure if the injection rate can be met without exceeding the maximum injection pressure. Otherwise, the well is considered to be pressure controlled and the injection rate becomes the unknown at the maximum injection pressure. An injection rate of 4.435 kg/s was specified, which is one half of the full-domain injection rate. The maximum injection pressure of 45 MPa was sufficiently high to avoid pressure-controlled conditions in the well. Simulation time for the deep domain problem was 1,000 days and 2,000 days for the shallow domain problem.

The problems are initialized with an aqueous hydrostatic pressure distribution. In the deep domain problem the aqueous hydrostatic pressure gradient is linear, as the aqueous density is constant. The initial pressure at the bottom of the domain (at 3000-m depth) for deep domain problem was 30.86 MPa, and for the shallow domain problem 84.99 MPa at 800-m depth. A linear temperature gradient was used for the shallow domain problem; where the formation temperature at the bottom of the domain (at 800-m depth) was 34˚C and a linear geothermal temperature gradient of 0.03 K/m was assumed. For the deep domain problem the temperature was held at 34˚C. For the shallow domain problem CO_2 was injected at 33.6˚C.

5.6.2 Deep Domain Input

As this problem involves isothermal conditions, STOMP-CO2 was selected for the simulations. Time stepping and grid resolution were not specified in the problem description (Ebigbo et al., 2007a). Before developing the simulation grid, a two-dimensional study was conducted to determine the vertical grid resolution. This study indicated that the arrival time at the leaky well of the gas plume was strongly dependent on having smaller vertical grid spacing immediately beneath the aquitard. The complete deep domain input file is shown in Section 5.6.2.1. A narrative of selected input cards in the sections that follow.

5.6.2.1 Deep Domain Input File

```
~Simulation Title Card
1,
Stuttgart 1.1,
Mark White,
Pacific Northwest Laboratory,
01 June 2011,
09:37 PDT,
8,
2.1 Definition of benchmark problem 1: CO2 plume evolution and
leakage through an abandoned well
2.1.1 Formulated by A. Ebigbo, J. Nordbotten and H. Class
Problem description CO2 is injected into an aquifer; spreads within the
aquifer and, upon reaching a leaky well, rises up to a shallower
aquifer. A quantification of the leakage rate which depends on the
pressure build- up in the aquifer due to injection and on the plume
evolution is the goal of the simulation.

~Solution Control Card
Normal,
STOMP-CO2 w/ Invariant Fluid Density and Viscosity w/ Fractional CO2 Solubility,
1,
0,day,1000,day,1,s,10,day,1.25,16,1.e-06,0.001,s,0.2,
10000,
Variable Aqueous Diffusion,
Variable Gas Diffusion,
1045,kg/m^3,2.535e-4,Pa s,479,kg/m^3,3.950e-5,Pa s,1.e-3,
0,

~Grid Card
Cartesian,
73,32,44,
-500.000,m,-475.000,m,-450.000,m,-425.000,m,-400.000,m,-375.000,m,
-350.000,m,-325.000,m,-300.000,m,-275.000,m,-255.000,m,-235.000,m,
-215.000,m,-200.000,m,-185.000,m,-170.000,m,-160.000,m,-150.000,m,
-140.000,m,-130.000,m,-122.500,m,-115.000,m,-107.500,m,-102.500,m,
-97.500,m,-92.500,m,-85.000,m,-77.500,m,-70.000,m,-60.000,m,
-50.000,m,-40.000,m,-30.000,m,-20.000,m,-12.500,m,-8.000,m,
-5.000,m,-2.800,m,-1.500,m,-0.800,m,-0.400,m,-0.133,m,
0.133,m,0.400,m,0.800,m,1.500,m,2.800,m,5.000,m,
8.000,m,12.500,m,20.000,m,30.000,m,40.000,m,50.000,m,
60.000,m,70.000,m,85.000,m,100.000,m,125.000,m,150.000,m,
175.000,m,200.000,m,225.000,m,250.000,m,275.000,m,300.000,m,
325.000,m,350.000,m,375.000,m,400.000,m,425.000,m,450.000,m,
475.000,m,500.000,m,
0.000,m,0.133,m,0.400,m,0.800,m,1.500,m,2.800,m,
5.000,m,8.000,m,12.500,m,20.000,m,30.000,m,40.000,m,
50.000,m,60.000,m,70.000,m,85.000,m,100.000,m,125.000,m,
150.000,m,175.000,m,200.000,m,225.000,m,250.000,m,275.000,m,
300.000,m,325.000,m,350.000,m,375.000,m,400.000,m,425.000,m,
450.000,m,475.000,m,500.000,m,
0.0,m,4.41,m,8.32,m,11.80,m,14.90,m,17.65,m,20.09,m,22.27,m,24.20,m,
25.92,m,27.44,m,28.80,m,30.0,m,20@5.000,m,12@2.500,m,

~Inactive Nodes Card
3,
1,41,1,32,13,32,
43,73,1,32,13,32,
42,42,2,32,13,32,
~Rock/Soil Zonation Card
2,
aquifer,1,73,1,32,1,44,
leaky well,42,42,1,1,1,44,

~Mechanical Properties Card
aquifer,2650,kg/m^3,0.15,0.15,Compressibility,1.e-9,1/psi,,,constant,1.0,1.0,
leaky well,2650,kg/m^3,0.15,0.15,Compressibility,1.e-9,1/psi,,,constant,1.0,1.0,

~Hydraulic Properties Card
aquifer,2.e-14,m^2,2.e-14,m^2,2.e-14,m^2,
leaky well,1.e-12,m^2,1.e-12,m^2,1.e-12,m^2,

~Saturation Function Card
aquifer,Brooks and Corey,0.1,m,2.0,,,
leaky well,Brooks and Corey,0.1,m,2.0,,,

#aquifer,Brooks and Corey,1,m,2.0,0.2,0.05,
#leaky well,Brooks and Corey,1,m,2.0,0.2,0.05,

~Aqueous Relative Permeability Card
aquifer,tabular,2,
0.0,0.0,
1.0,1.0,
leaky well,tabular,2,
0.0,0.0,
1.0,1.0,

aquifer,burdine,2.0,
leaky well,burdine,2.0,

~Gas Relative Permeability Card
aquifer,tabular,2,
0.0,0.0,
1.0,1.0,
leaky well,tabular,2,
0.0,0.0,
1.0,1.0,

aquifer,burdine,2.0,
leaky well,burdine,2.0,

~Salt Transport Card
aquifer,0.0,ft,0.0,ft,
leaky well,0.0,ft,0.0,ft,

~Initial Conditions Card
Gas Pressure,Aqueous Pressure,
3,
Gas Pressure,30.81955,MPa,,,,,-0.01025,1/m,1,73,1,32,1,44,
Aqueous Pressure,30.81955,MPa,,,,,-0.01025,1/m,1,73,1,32,1,44,
Temperature,34,C,,,,,,,1,73,1,32,1,44,

~Boundary Conditions Card
3,
West,Aqu. Initial Condition,Gas Initial Condition,Aqu. Mass Frac.,
1,1,1,32,1,44,1,
0,s,,,,,,,0.0,,,,
East,Aqu. Initial Condition,Gas Initial Condition,Aqu. Mass Frac.,
73,73,1,32,1,44,1,
0,s,,,,,,,0.0,,,,
North,Aqu. Initial Condition,Gas Initial Condition,Aqu. Mass Frac.,
1,73,32,32,1,44,1,
0,s,,,,,,,0.0,,,,
~Coupled Well Card
1,
CO2 Injection Well,Water Relative Saturation,1.0,0.5,1.0,0.383184,MMT,
1,
-100.0,m,0.075,m,30.0,m,-100.0,m,0.075,m,0.0,m,0.15,m,0.0,screened,
1,
0.0,hr,4.435,kg/s,45,MPa,0.0,

~Output Options Card
5,
24,1,1,
24,1,12,
42,1,12,
42,1,22,
42,1,33,
1,1,day,m,6,6,6,
16,
Phase Condition,,
Gas Saturation,,
Gas Relative Permeability,,
Integrated CO2 Mass,kg,
Integrated CO2 Aqueous Mass,kg,
Integrated CO2 Gas Mass,kg,
Integrated CO2 Trapped-Gas Mass,kg,
Salt Aqueous Mass Fraction,,
CO2 Aqueous Mass Fraction,,
Gas Density,kg/m^3,
Aqueous Density,kg/m^3,
Gas Viscosity,Pa s,
Aqueous Viscosity,Pa s,
Gas Pressure,MPa,
Aqueous Pressure,MPa,
Diffusive Porosity,,
8,
0.1,day,
0.5,day,
1.0,day,
5.0,day,
10.0,day,
50.0,day,
100.0,day,
500.0,day,
13,
Rock/Soil Type,,
Gas Saturation,,
Trapped Gas Saturation,,
Salt Saturation,,
CO2 Aqueous Concentration,gm/cm^3,
Salt Aqueous Mass Fraction,,
CO2 Aqueous Mass Fraction,,
Gas Pressure,MPa,
Aqueous Pressure,MPa,
Diffusive Porosity,,
Gas Density,kg/m^3,
Aqueous Density,kg/m^3,
Phase Condition,,

~Surface Flux Card
1,
CO2 Mass Flux,kg/s,kg,top,42,42,1,1,22,22,
```

5.6.2.2 Solution Control Card

The deep domain problem specified constant fluid properties. This capability is invoked in STOMP-CO2 by adding the keyword *Invariant* to the *Operational Mode Option* line of the input. This triggers the reading of an additional input line in the *Solution Control Card*, just after the specification of the *Gas Diffusion Option* line. This additional line allows for input of the aqueous density, aqueous viscosity, gas density and gas viscosity, whose values will be maintained throughout the simulation. The deep domain problem also specified zero solubility of CO_2 in the aqueous phase and zero solubility of H_2O in the gas phase. STOMP-CO2 does not currently have a zero mutual solubility option, but will allow the user to reduce the CO_2 solubility in the aqueous phase by a factor, by adding the keyword *Fractional* to the *Operational Mode Option* line of the input. This triggers reading the fractional factor in the same line as the fluid densities and viscosities.

5.6.2.3 Grid Card

The deep domain problem was spatially discretized using a three-dimensional Cartesian domain (73 x 32 x 44) with grid resolution in the horizontal directions around the leaking well, and in the vertical direction beneath the aquitard, which represented ½ of the full problem domain. This domain took advantage of the vertical plane of symmetry passing through the injection and leaky wells. A schematic of the grid is shown in Figure 5.27.

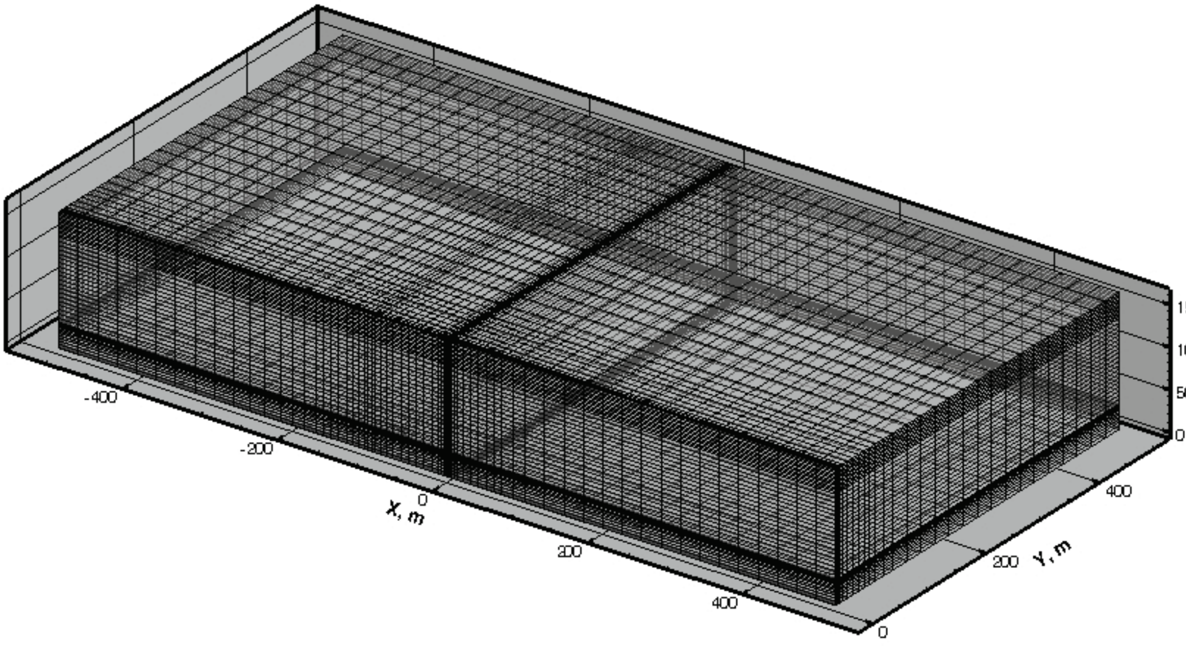

Figure 5.27. Computational Grid with Horizontal Resolution Near the Leaky Well

5.6.2.4 Inactive Nodes Card

Inactive nodes were used to eliminate the impermeable aquitard nodes from the computational domain. The single-node column of leaky well nodes then became the only hydrologic connection between the lower and upper aquifer.

5.6.2.5 Saturation Function Card

The deep domain problem specified zero capillary pressure. To approximate zero capillary pressure conditions a Brooks-Corey saturation function was chosen with a small entry pressure. The gas and aqueous residual saturations were set to zero.

5.6.2.6 Aqueous Relative Permeability Card

A linear aqueous relative permeability function was specified by using a two-point tabular function. Linear interpolation was used to determine the aqueous relative permeability with the aqueous saturation.

5.6.2.7 Gas Relative Permeability Card

A linear gas relative permeability function was specified by using a two-point tabular function. Linear interpolation was used to determine the gas relative permeability with the gas saturation.

5.6.2.8 Initial Conditions Card

Aqueous saturated hydrostatic conditions were created for the initial conditions by specifying the aqueous and gas pressures to have a linear gradient in the *z* direction. The specified aqueous density of 1045 kg/m^3 yields a pressure gradient of 0.0102515 MPa/m. The pressure at the bottom of the domain was specified as being 30.86 MPa, which makes the pressure at the lowest node centroid (2.205 m from the bottom of the domain) equal to 30.81955 MPa.

5.6.2.9 Boundary Condtions Card

The problem description called for the lateral boundaries to be constant Dirichlet type boundaries, equal to the initial pressures, and all horizontal boundaries to be no-flow type boundaries. The symmetry plane for the domain used for the STOMP-CO2 simulation was set on the *south* lateral boundary surface, making that a no-flow type boundary also. Initial condition type boundary conditions were assigned to the *west*, *east*, and *north* lateral boundaries. The *south*, *bottom*, and *top* boundaries were defaulted to zero flux type boundary conditions.

5.6.2.10 Coupled Well Card

scCO_2 was injected via a coupled well. The problem description only specified an overall mass injection rate for the CO_2, without indicating the distribution of CO_2 mass along the screened interval of the well. To ensure that the injection well did not become flow controlled, a high (45 MPa) upper injection pressure limit was assigned. An injection rate of 4.435 kg/s over the full screened interval of the lower aquifer was specified, which was ½ of the injection rate for the full domain (i.e., 8.87 kg/s).

5.6.2.11 Surface Flux Card

The CO_2 mass flux midway between the top and bottom aquifers was tracked by requesting a *CO2 Mass Flux* as a *Surface Flux Type Option* for a surface within the leaky well at the midway point.

5.6.3 Deep Domain Results

The leakage rate of CO_2 mass passing the midway point between the lower and upper aquifer in the leaky well, expressed as a percent of the total injection rate, is shown in Figure 5.28. Results from 13 modeling groups, using 10 different numerical simulators, is shown in Figure 5.29 (Class et al., 2009). The STOMP-CO2 results show good agreement with those from these various modeling groups. The maximum leakage rate was 0.214%, which occurred at 50 days. This compares to values of 0.212% at 46 days, reported for TOUGH2/ECO2N (Class et al., 2009). The leakage rate at 1,000 days was 0.118%, which compares with 0.115%, reported for TOUGH2/ECO2N (Class et al., 2009). Color-scaled plots of gas saturation at 50, 100, and 500 days are shown in Figures 5.30 through 5.32, respectively. At 50 days, Figure 5.30, the leakage rate is at a maximum, with visual distorting of the gas plume in the lower aquifer, and the gas plume reaching the cap of the upper aquifer. Whereas the driving forces for gas migration in the lower aquifer are a combination of pressure gradient and buoyancy derived, in the upper aquifer, buoyancy is the principal driving force for gas migration. By 500 days the gas plume has contacted the vertical lateral boundaries in the lower aquifer, indicating that CO_2 is migrating out of the domain.

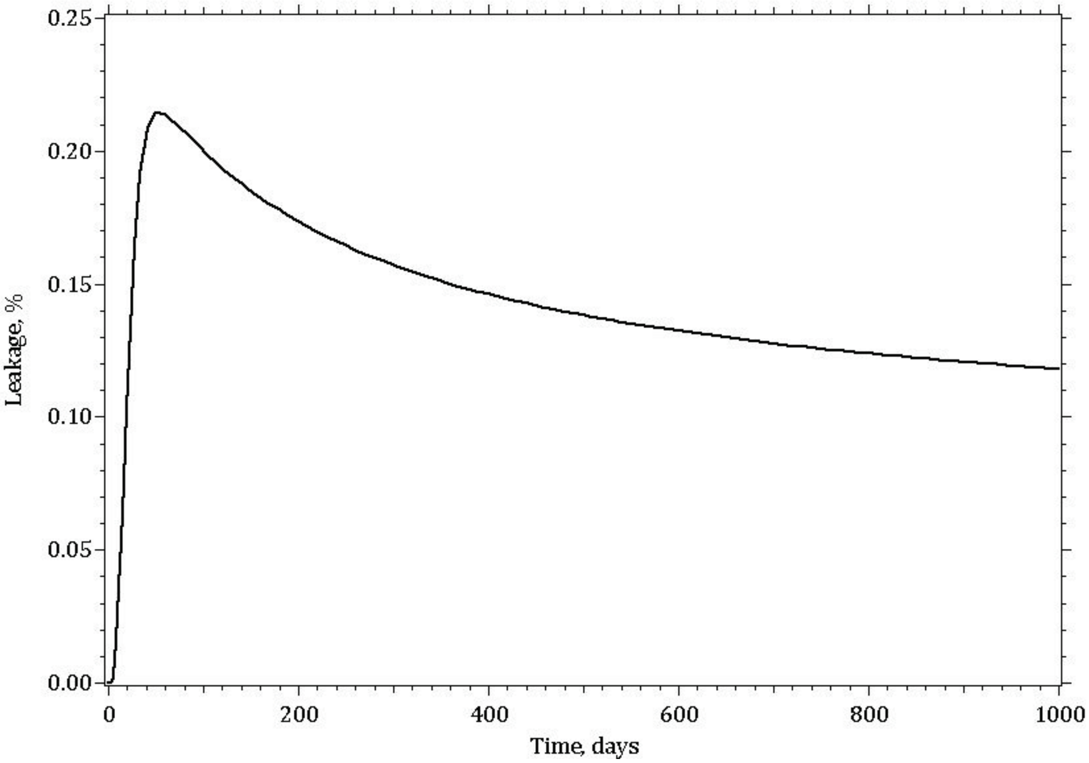

Figure 5.28. CO_2 Leakage Rate at the Midpoint of The Leaky Well

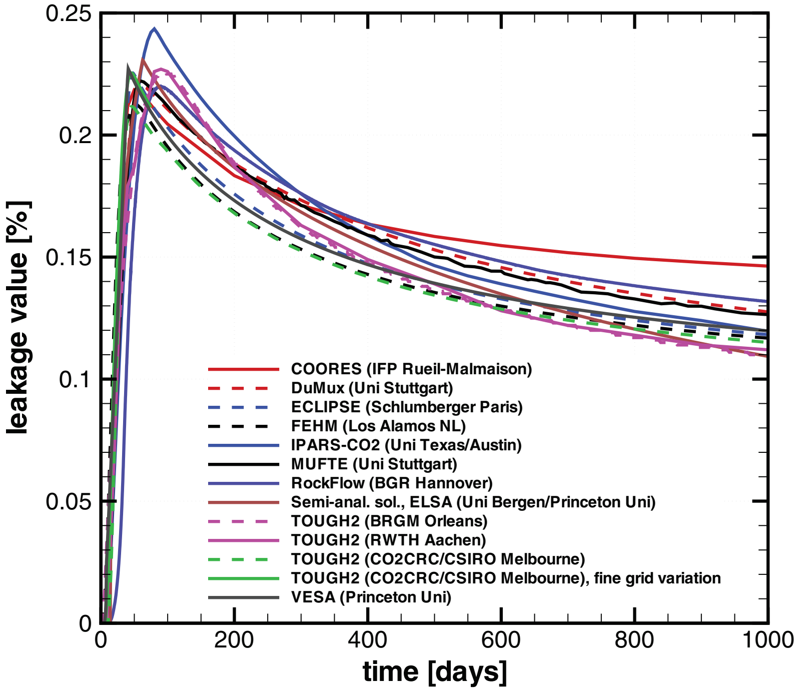

Figure 5.29. CO_2 Leakage Rate at the Midpoint of the Leaky Well from Cass et al. (2009)

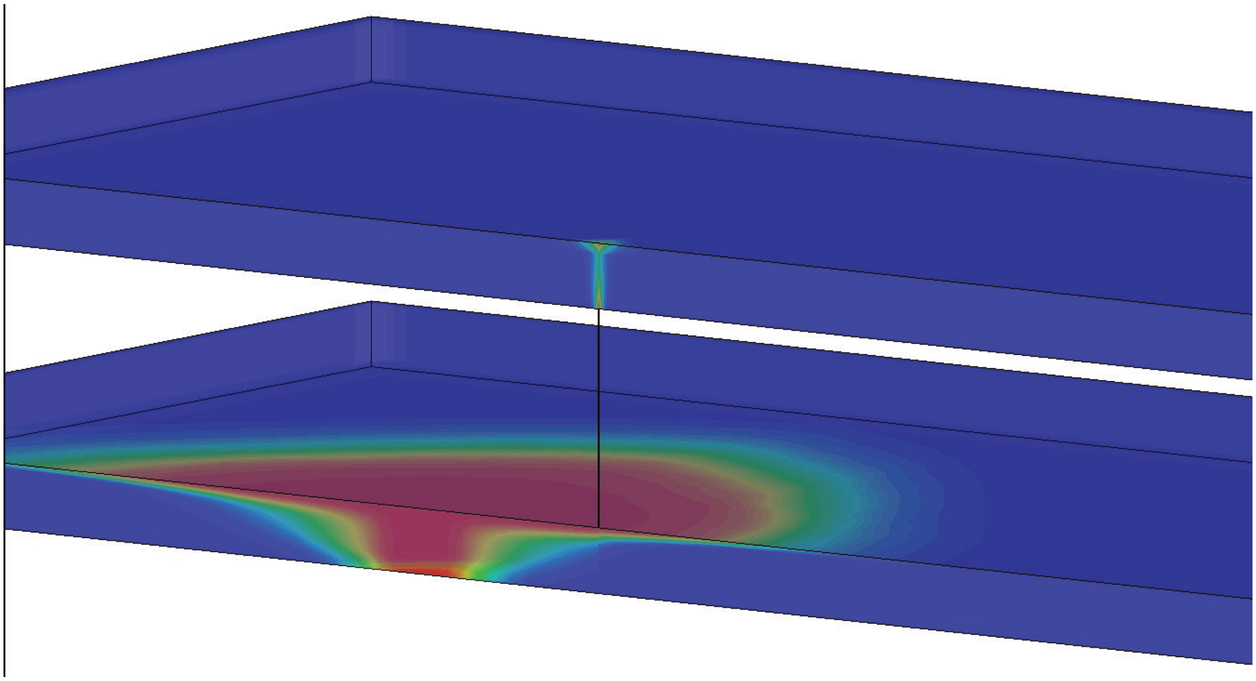

Figure 5.30. Gas Saturation Profile at 50 Days after the Start of Injection

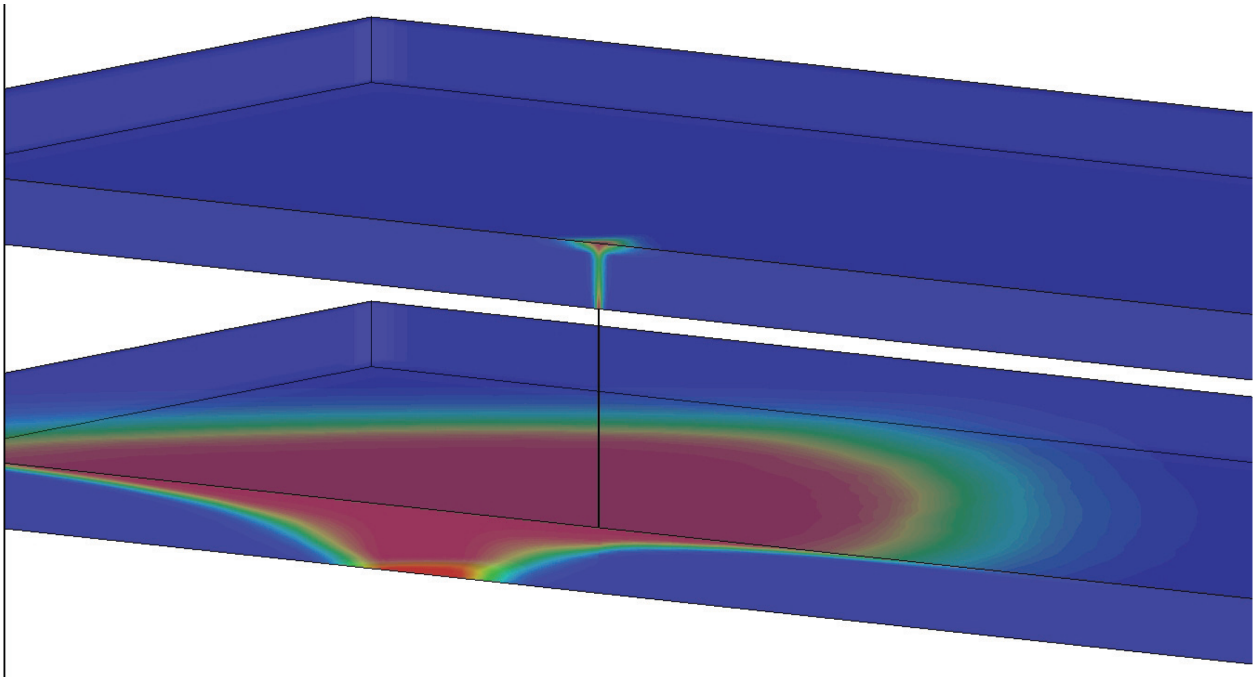

Figure 5.31. Gas Saturation Profile at 100 Days after the Start of Injection

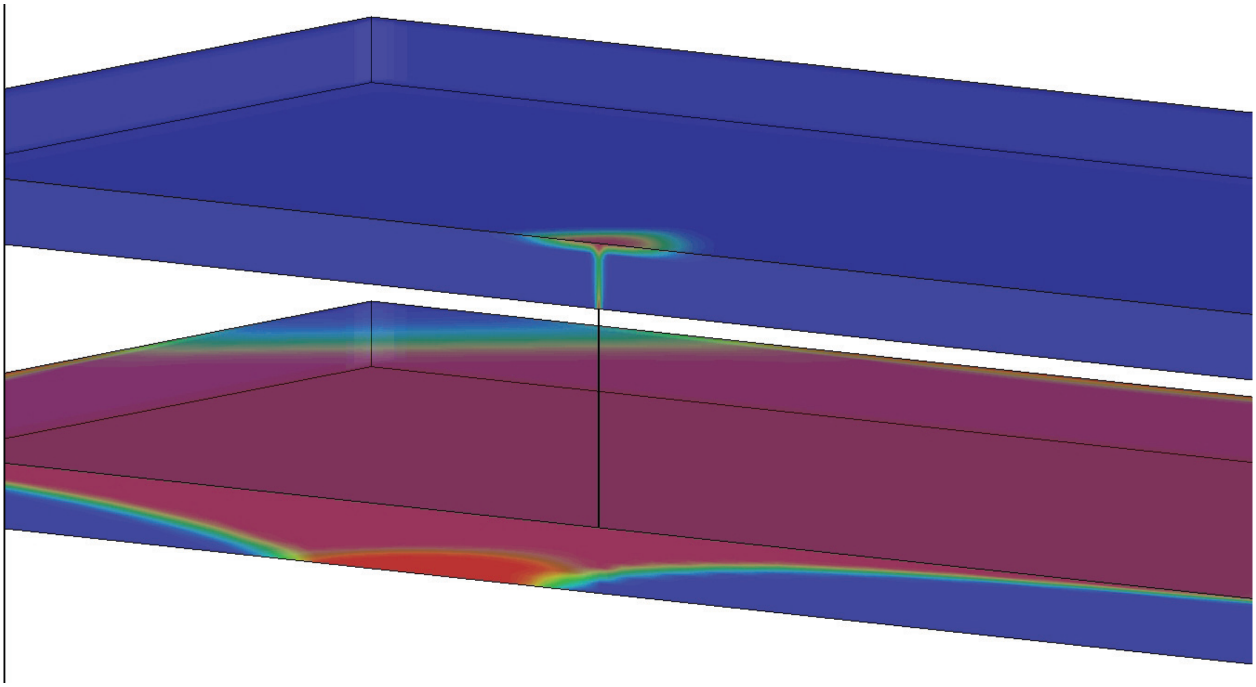

Figure 5.32. Gas Saturation Profile at 500 Days after the Start of Injection

6.0 References

Anderson, G.M., and D.A. Crerar. 1992. Thermodynamics in Geochemistry: The Equilibrium Model, Oxford University Press.

Battistelli, A., C. Claudio, and K. Pruess. 1997. "The simulator TOUGH2/EWASG for modelling geothermal reservoirs with brines and gas." Geothermics, 26(4): 437-464.

Bramley, R. and X. Wang. 1995. SPLIB: A Library of Iterative Methods for Sparse Linear Systems. Department of Computer Science, Indiana University, Bloomington, Indiana.

Bromley, L.A. 1973. "Thermodynamic properties of strong electrolytes in aqueous solutions." AIChE Journal, 19(2):313-320

Brooks, R. H., and A. T. Corey. 1966. "Properties of porous media affecting fluid flow." Journal of Irrigation and Drainage Division 93(3):61-88.

Burdine, N. T. 1953. "Relative permeability calculations from pore-size distribution data." Petroleum Trans. 198:71-77.

Class, H. A. Ebigbo, R. Helmig, H. K. Dahle, J. M. Nordbotten, M. A. Celia, P. Audigane, M. Darcis, J. Ennis-King, Y. Fan, B. Flemisch, S. E. Gasda, M. Jin, S. Krug, D. Labregere, A. N. Beni, R. J. Pawar, A. Sbai, S. G. Thomas, L. Trenty, L. Wei. 2009. "A benchmark study on problems related to CO2 storage in geologic formations. Summary and discussion of results." Comput. Geosci., 13:409-434.

Corey, A. T. 1977. "Mechanics of heterogeneous fluids in porous media." Water Resources Publications, Fort Collins, Colorado.

De Vries, D. A. 1966. "Thermal Properties of Soils." In Physics of plant environment, edited by W. R. Van Wijk, pp. 210-235, North-Holland, Amsterdam.Fayer, M.J. and C.S. Simmons. 1995. "Modified soil water retention functions for all matric suctions." Water Resour. Res. 31:1233-1238.

Doughty, C. and. K. Pruess. 1992. "A similarity solution for two-phase water, air and heat flow near a linear heat source in porous medium." Journal of Geophysical Research. 97(B2):1821-1838.

Ebigbo, A., J. M. Nordbotten., and H. Class. 2007a. "Problem 1: CO2 Plume Evolution and Leakage through an Abandoned Well," Numerical Investigations of CO2 Sequestration in Geological Formations: Problem-Oriented Benchmarks, Universität Stuttgart, www.hydrosys.uni-stuttgart.de/co2-workshop/Problem1.pdf.

Ebigbo, A., Class, H., Helmig, R., 2007b. CO2 leakage through an abandoned well: problem- oriented benchmarks. Computational Geoscience, 11:103–115.

Fatt, I. and W.A. Klickoff. 1959. "Effect of fractional wettability on multiphase flow through porous media." AIME (Am. Inst. Min. Metall. Eng.) Trans., 216:246–250.

Fayer, M.J. and C.S. Simmons. 1995. "Modified soil water retention functions for all matric suctions." Water Resour. Res. 31:1233-1238.

Fenghour, A., W. A. Wakeham, V. Vesovic. 1998. "The viscosity of carbon dioxide." J. Phys. Chem. Ref. Data, 27(1):31-41.

Freedman V. L., and M. D. White. 2007. STOMP Software Test Plan Rev. 1.0, PNNL-SA-54022, Pacific Northwest National Laboratory, Richland, WA.

W. Gropp, E. Lusk, and A. Skjellum, Using MPI: Portable Parallel Programming with the Message-Passing Interface, MIT Press, 1999.

Gudmundsson, J.S., and H. Thrainsson. 1989. "Power potential of two-phase geothermal wells." Geothermics, 18(3):357-366.

Gunter W.D., B. Wiwchar, and E.H. Perkins. 1997. "Aquifer disposal of CO2-rich greenhouse gases: Extension of the time scale of experiment for CO2-sequestering reactions by geochemical modelling." Mineralogy and Petrology, 59(1-2):121-140.

Haas Jr., J.L. 1976. Physical Properties of the Coexisting Phases and Thermochemical Properties of the H2O Component in Boiling NaCl Solutions. Preliminary Steam Tables for NaCl Solutions. Geological Survey Bulletin 1421-A.

Haverkamp R, M. Vaclin, J. Touma, P.J. Wierenga, and G. Vachaud. 1977. "A comparison of numerical simulation models for one-dimensional infiltration," Soil Sci. Soc. Am. J. 41:285-294.

Helgeson H.C. 1969. "Thermodynamics of hydrothermal systems at elevated temperatures and pressures." American Journal of Science, 267(7):729-804.

Kaluarachchi, J.J. and J.C. Parker. 1992. "Multiphase flow with a simplified model for oil entrapment." Transport in Porous Media 7:1-14.

Klavetter, E.A., and R. R. Peters. 1986. Estimation of hydrologic properties of unsaturated fractured rock mass, SAND84-2642, Sandia National Laboratories, Albuquerque, New Mexico.

Lasaga A.C. 1984. "Chemical kinetics of water-rock interactions." Journal of Geophysical Research, 89(NB6):4009-4025.

Land, C. S. 1968. "Calculation of imbibition relative permeability for two- and three-phase flow from rock properties." Trans. Am. Inst. Min. Metall. Pet. Eng. 243:149-156.

Lenhard, R. J., and J. C. Parker. 1987. "A model for hysteretic constitutive relations governing multiphase flow 2. Permeability-saturation relations." Water Resour.s Res. 23(12):2197-2206.

Lide, D.R. and H.V. Kehiaian. 1994. CRC Handbook of Thermophysical and Thermochemical Data, CRC Press, Inc., Boca Raton, Florida, pp. 97-98.

Meyer, C.A., R.B. McClintock, G.J. Silvestri, and R.C. Spencer. 1993. ASME Steam Tables, The American Society of Mechanical Engineers, New York.

Millington, R. J. and J. P. Quirk. 1959. "Permeability of porous media." Nature 183:387-388.

Mualem, Y. 1976. "A new model for predicting the hydraulic conductivity of unsaturated porous media." Water Resources Research., 12:513-522.

Nickalls, R. W. D. 1993 "A new approach to solving the cubic: Cardan's solution revealed." Math. Gazette, 77:354-359.

Nichols W. E., and M. D. White. 2007a. Project Management Plan For Subsurface Transport Over Multiple Phases (STOMP) Software Maintenance and Development, PNNL-SA-54024, Pacific Northwest National Laboratory, Richland, WA.

Nichols W. E., and M. D. White. 2007b. Software Design Description For Subsurface Transport Over Multiple Phases (STOMP) Software, PNNL-SA-54078, Pacific Northwest National Laboratory, Richland, WA.

Nitao, J. J., 1988. Numerical Modeling of the Thermal and Hydrological Environment around a Nuclear Waste Package Using the Equivalent Continuum Approximation: Horizontal Emplacement, UCID-2144, Lawrence Livermore National Laboratory, Livermore, California.

Nordbotten, J., M. Celia, and S. Bachu. 2004. "Analytical Solutions for Leakage Rates through Abandoned Wells," Water Resources Research, 40(4), W04204.

Nordbotten, J., M. Celia and S. Bachu. 2005a. "Injection and Storage of CO2 in Deep Saline Aquifers: Analytical Solution for CO2 Plume Evolution During Injection." Transport in Porous Media, 58(3):339–360.

Nordbotten, J., M. Celia, S. Bachu and H. Dahle. 2005b. "Semi-Analytical Solution for CO2 Leakage through an Abandoned Well." Environmental Science and Technology, 39(2):602– 611.

O'Sullivan, M. J. 1981. "A similarity method for geothermal well test analysis." Water Resources Research, 17(2):390-398.

Ozbek, H. and S.L. Phillips. 1980. "Thermal conductivity of aqueous sodium chloride solutions from 20 to 330 C." J. Chem. Engr. Data, 25:263-267.

Parker, J. C. and R. J. Lenhard. 1987. "A model for hysteretic constitutive relations governing multiphase flow 1. Saturation-pressure relations." Water Resources Research, 23(12):2187- 2196.

Peaceman, D.W. 1977. "Interpretation of Well-Block Pressures in Numerical Reservoir Simulation." Paper SPE-6893, presented at the SPE-AIME 52nd Annual Fall Technical Conference and Exhibition, Denver, Oct. 9-12, 1977.

Peaceman, D.W. 1982. "Interpretation of Well-Block Pressures in Numerical Reservoir Simulation With Nonsquare Grid Blocks and Anisotropic Permeability." Paper SPE-10528, presented at the 1982 SPE Symposium on Reservoir Simulation, New Orleans, Jan. 31-Feb. 3, 1982.

Peaceman, D.W. 1987. "Interpretation of Wellblock Pressures in Numerical Reservoir Simulation: Part3 – Off-Center and Multiple Wells Within a Wellblock." Paper SPE-16976, presented at the 1987 SPE Annual Technical Conference and Exhibition, Dallas, Sept. 27-30, 1987.

Phillips, S.L., A. Igbene, J.A. Fair, H. Ozbek and M. Tavana. 1981. A Technical Databook for Geothermal Energy Utilization, LBL-12810, UC-66a, Lawrence Berkeley Laboratory, University of California, Berkeley, California.

Prausnitz J. M., R. N. Lichtenthaller, and E. G. De Azedevo. 1986. Molecular Thermodynamics of Fluid Phase Equilibria. Prentice Hall, New York.

Pruess, K. and J. Garcia. 2002. “Multiphase flow dynamics during CO2 disposal into saline aquifers.” Environmental Geology, 42(2-3):282-295. doi: 10.1007/s00254-001-0498-3.

Pruess, K., J. Garcia, T. Kovscek, C. Oldenburg, J. Rutqvist, C. Steefel, and T. Xu. 2002. Intercomparison of Numerical Simulation Codes for Geologic Disposal of CO2, Lawrence Berkeley National Laboratory, LBNL-51813, Berkeley, California.

Reid, R.C., J.M. Prausnitz, and B.E. Poling. 1987. The Properties of Gases and Liquids. pp. 474-475. McGraw-Hill, New York, New York

Renner, T.A. 1988. "Measurement and correlation of diffusion for CO2 and rich-gas applications." SPE Reservoir Engineering, pp:517-523.

Shu, J. 2005. Comparison of Various Techniques for Computing Well Index, Master of Science Thesis, Stanford University, Stanford, California.

Somerton, W. H., A. H. El-shaarani, and S. M. Mobarak. 1974. “High temperature behavior of rocks associated with geothermal type reservoirs.” Paper SPE-4897, presented at the 44th Annual California Regional Meeting of the Society of Petroleum Engineers, San Francisco, California.

Somerton, W. H., J. A. Keese, and S. L. Chu. 1973. “Thermal behavior of unconsolidated oil sands.” Paper SPE-4506, presented at the 48th Annual Fall Meeting of the Society of Petroleum Engineers, Las Vegas, Nevada.

Span, R. and W. Wagner. 1996. A new equation of state for carbon dioxide covering the fluid region from the triple-point to 1100 K at pressures up to 800 MPa. Journal of Physical and Chemical Reference Data, 25(6):1509-1588.

Spycher, N., K. Pruess, and J. Ennis-King. 2003. “CO2-H2O mixtures in geological sequestration of CO2. I. Assessment and calculation of mutual solubilities from 12 to 100°C and up to 600 bar.” Geochimica et Cosmoschimica Acta, 67(16):3015-3031, doi:10.1016/s0016-7037(03)00273-4.

Spycher, N. and K. Pruess. 2010. “A phase-partitioning model for CO2-brine mixtures at elevated temperatures and pressures: Application to CO2-enhanced geothermal systems.” Transport in Porous Media, 82:173-196, doi:10.1007/s11242-009-9425-y

Steefel C.I. and A.C. Lasaga. 1994. “A coupled model for transport of multiple chemical species and kinetic precipitation/dissolution reactions with applications to reactive flow in single phase hydrothermal system.” American Journal of Science, 294(5):529-592.

Touma, J. and M. Vauclin. 1986. “Experimental and numerical analysis of two-phase infiltration in a partially saturated soil.” Transport in Porous Media, 1:27-55.

van Genuchten, M. Th. 1980. “A closed-form equation for predicting the hydraulic conductivity of unsaturated soils.” Soil Sci. Soc. Am. J., 44:892-898.

Verma, A., and K. Pruess. 1988. "Thermohydrological conditions and silica redistribution near high-level nuclear wastes emplaced in saturated geological formations." Journal of Geophysical Research, 93(B2):1159-1173.

Webb, S. W., 2000. “A simple extension of two-phase characteristic curves to include the dry region.” Water Resources Research, 36(6):1425-1430.

White M. D., and V. L. Freedman. 2007. STOMP Software Configuration Management Plan, PNNL-SA-54023, Pacific Northwest National Laboratory, Richland, WA.

White, M. D., and M. Oostrom. 2006. STOMP Subsurface Transport Over Multiple Phases, Version 4.0, User's Guide, PNNL-15782, Pacific Northwest National Laboratory, Richland, WA.

White, M. D. and B. P. 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, USA.

White MD and M Oostrom. 2000. STOMP Subsurface Transport Over Multiple Phases Version 2.0 Theory Guide. PNNL-12030 UC-2010, Pacific Northwest National Laboratory, Richland, Washington, USA.

Wolery T. W. and R. L. Jarek. 2003. Software User's Manual, EQ3/6, Version 8.0. Sandia National Laboratories, Albuquerque, New Mexico.

Xu T, J. A. Apps and K. Pruess. 2001. Analysis of Mineral Trapping for CO2 Disposal in Deep Aquifers. LBNL-46992, Lawrence Berkeley National Laboratory, Berkeley, California.

Appendix A

Parameter Data

Appendix A: Parameter Data

Table A.1. Water Saturation Line Constants, Eqn. (2.7)

k_1	-7.691234564e+0
k_2	-2.608023696e+1
k_3	-1.681706546e+2
k_4	6.423285504e+1
k_5	-1.189646225e+2
k_6	4.167117320e+0
k_7	2.097506760e+1
k_8	1.e+9
k_9	6.e+0

Table A.2. CO_2 Henry's Coefficient Constants, Eqn. (2.24)

b_0	7.83666e+7	c_0	1.19784e-1
b_1	1.96025e+6	c_1	-7.17823e-4
b_2	8.20574e+4	c_2	4.93854e-6
b_3	-7.40674e+2	c_3	-1.03826e-8
b_4	2.18380e+0	c_4	1.08233e-11
b_5	-2.20999e-3		

Table A.3. Brine Enthalpy Constants, Eqn. (2.25)

$a_{0,0}$	9633.6e+0	b_1	-25.9293e+0
$a_{0,1}$	-4080.0e+0	b_2	0.16792e+0
$a_{0,2}$	286.49e+0	b_3	-0.83624e-3
$a_{1,0}$	166.58e+0		
$a_{1,1}$	68.577e+0		
$a_{1,2}$	-4.6856e+0		
$a_{2,0}$	-0.90963e+0		
$a_{2,1}$	-0.36524e+0		
$a_{2,2}$	0.249667e-1		
$a_{3,0}$	0.17965e-2		
$a_{3,1}$	0.71924e-3		
$a_{3,2}$	-0.4900e-4		

Table A.4. Pure-Water Liquid Enthalpy Constants, Eqn. (2.26)

A_0	6.824687741e+3	A_{12}	-2.616571843e-2	a_1	8.438375405e-1
A_1	-5.422063673e+2	A_{13}	1.522411790e-3	a_2	5.362162162e-4
A_2	-2.096666205e+4	A_{14}	2.284279054e-2	a_3	1.720000000e+0
A_3	3.941286787e+4	A_{15}	2.421647003e+2	a_4	7.342278489e-2
A_4	-6.733277739e+4	A_{16}	1.269716088e-10	a_5	4.975858870e-2
A_5	9.902381028e+4	A_{17}	2.074838328e-7	a_6	6.537154300e-1
A_6	-1.093911774e+5	A_{18}	2.174020350e-8	a_7	1.150000000e-6
A_7	8.590841667e+4	A_{19}	1.105710498e-9	a_8	1.150800000e-5
A_8	-4.511168742e+4	A_{20}	1.293441934e+1	a_9	1.418800000e-1
A_9	1.418138926e+4	A_{21}	1.308119072e-5	a_{10}	7.002753165e+0
A_{10}	-2.017271113e+3	A_{22}	6.047626338e-14	a_{11}	2.995284926e-4
A_{11}	7.982692717e+0			a_{12}	2.040000000e-1

Table A.5. Pure-Water Vapor Enthalpy Constants, Eqn. (2.28)

b	7.633333333D-1	$B_{7,1}$	1.683998803e-1	n_7	2
$b_{6,1}$	4.006073948D-1	$B_{7,2}$	-5.809438001e-2	n_8	2
$b_{7,1}$	8.636081627D-2	$B_{8,1}$	6.552390126e-3		
$b_{8,1}$	-8.532322921D-1	$B_{8,2}$	5.710218649e-4	$x_{6,1}$	14
$b_{8,2}$	3.460208861D-1	$B_{9,0}$	1.936587558e+2	$x_{7,1}$	19
		$B_{9,1}$	-1.388522425e+3	$x_{8,1}$	54
B_0	1.683599274e+1	$B_{9,2}$	4.126607219e+3	$x_{8,2}$	27
$B_{0,1}$	2.856067796e+1	$B_{9,3}$	-6.508211677e+3		
$B_{0,2}$	-5.438923329e+1	$B_{9,4}$	5.745984054e+3	$z_{1,1}$	13
$B_{0,3}$	4.330662834e-1	$B_{9,5}$	-2.693088365e+3	$z_{2,1}$	18
$B_{0,4}$	-6.547711697e-1	$B_{9,6}$	5.235718623e+2	$z_{3,1}$	18
$B_{0,5}$	8.565182058e-2			$z_{4,1}$	25
$B_{1,1}$	6.670375918e-2	l_6	1	$z_{5,1}$	32
$B_{1,2}$	1.388983801e+0	l_7	1	$z_{6,1}$	12
$B_{2,1}$	8.390104328e-2	l_8	2	$z_{7,1}$	24
$B_{2,2}$	2.614670893e-2			$z_{8,1}$	24
$B_{2,3}$	-3.373439453e-2	L_0	1.574373327e+1	$z_{1,2}$	3
$B_{3,1}$	4.520918904e-1	L_1	-3.417061978e+1	$z_{2,2}$	2
$B_{3,2}$	1.069036614e-1	L_2	1.931380707e+1	$z_{3,2}$	10
$B_{4,1}$	-5.975336707e-1			$z_{4,2}$	14
$B_{4,2}$	-8.847535804e-2	n_1	2	$z_{5,2}$	28
$B_{5,1}$	5.958051609e-1	n_2	3	$z_{6,2}$	11
$B_{5,2}$	-5.159303373e-1	n_3	2	$z_{7,2}$	18
$B_{5,3}$	2.075021122e-1	n_4	2	$z_{8,2}$	14
$B_{6,1}$	1.190610271e-1	n_5	3	$z_{5,3}$	24
$B_{6,2}$	-9.867174132e-2	n_6	2		

Table A.6. Precipitated Salt Enthalpy Constants, Eqn. (2.30)

A_0	-1.24858e-4
A_1	25.19e+0
A_2	0.1973e+0
A_3	-6.0114e-4
A_4	8.81505e-7
A_5	-4.765e-10

Table A.7. Pure Liquid Water Viscosity Constants, Eqn. (2.31)

h_0	1.0	$h_{3,0}$	1.778064e-1	$h_{5,0}$	0.0
h_1	9.78197e-1	$h_{3,1}$	4.605040e-1	$h_{5,1}$	-1.578386e-2
h_2	5.79829e-1	$h_{3,2}$	2.340379e-1	$h_{5,2}$	0.0
h_3	-2.02354e-1	$h_{3,3}$	-4.924179e-1	$h_{5,3}$	0.0
$h_{0,0}$	5.132047e-1	$h_{3,4}$	0.0	$h_{5,4}$	0.0
$h_{0,1}$	3.205656e-1	$h_{3,5}$	0.0	$h_{5,5}$	0.0
$h_{0,2}$	0.0	$h_{4,0}$	-4.176610e-2	$h_{6,0}$	0.0
$h_{0,3}$	0.0	$h_{4,1}$	0.0	$h_{6,1}$	0.0
$h_{0,4}$	-7.782567e-1	$h_{4,2}$	0.0	$h_{6,2}$	0.0
$h_{0,5}$	1.885447e-1	$h_{4,3}$	1.600435e-1	$h_{6,3}$	-3.629481e-3
$h_{1,0}$	2.151778e-1	$h_{4,4}$	0.0	$h_{6,4}$	0.0
$h_{1,1}$	7.317883e-1	$h_{4,5}$	0.0	$h_{6,5}$	0.0
$h_{1,2}$	1.241044e+0				
$h_{1,3}$	1.476783e+0				
$h_{1,4}$	0.0				
$h_{1,5}$	0.0				
$h_{2,0}$	-2.818107e-1				
$h_{2,1}$	-1.070786e+0				
$h_{2,2}$	-1.263184e+0				
$h_{2,3}$	0.0				
$h_{2,4}$	0.0				
$h_{2,5}$	0.0				

Table A.8. Brine Viscosity Constants, Eqn. (2.32)

a_1	0.0816
a_2	0.0122
a_3	0.000128
a_4	0.000629
a_5	-0.7

Table A.9. Pure CO_2 Viscosity Constants, Eqn. (2.35)

a_0	0.235156e+0	b_1	0.4071119e-2
a_1	-0.491266e+0	b_2	0.7198037e-4
a_2	5.211155e-2	b_3	0.2411697e-16
a_3	5.347906e-2	b_4	0.2971072e-22
a_4	-1.537102e-2	b_5	-0.1627888e-22

Table A.10. Brine Thermal Conductivity Constants, Eqn. (2.36)

c_1	2.3434e-3
c_2	-7.924e-6
c_3	3.924e-8
c_4	1.06e-5
c_5	-2.e-8
c_6	1.2e-10

Table A.11. Pure-Water Thermal Conductivity Constants Eqn. (2.37)

a_0	0.0102811	B_1	-0.171587	d_1	0.0701309
a_1	0.0299621	B_2	2.392190D+0	d_2	0.0118520
a_2	0.0156146			d_3	0.00169937
a_3	-0.00422464	C_1	0.642857	d_4	-1.0200
		C_2	-4.11717		
		C_3	-6.17937		
b_1	-0.397070	C_4	0.00308976		
b_2	0.400302	C_5	0.0822994		
b_3	1.060000	C_6	10.0932		

Table A.12. Gas Molecular Diffusion Constants, Eqn. (2.41)

A_1	1.06036e+0
A_2	1.5610e-1
A_3	1.9300e-1
A_4	4.7635e-1
A_5	1.03587e+0
A_6	1.52996e+0
A_7	1.76474e+0
A_8	3.89411e+0

Appendix B

Input File Formatting

Appendix B: Input File Formatting

The principal input file for STOMP-CO2 and -CO2e is a text file, named *input*, that comprises *cards*. *Cards* in the *input* file may be sequenced in any order. Secondary input files contain additional data and are named according to user in the *input* file. Input *cards* comprise *lines*, which are delimited with a *line return*. The first *line* of an input *card* contains the *card* title and begins with a ~ (e.g., ~*Simulation Title Card*). Input *lines* comprise *fields*, which are delimited with *commas*. An ending *comma* is required on every *line*. Input *fields* may be one of three types: *character strings*, *integers*, or *real numbers*. *Character string* entries are case insensitive, which means that a *field* entry of "Pressure" is equivalent to that of "pressure". *Integer* entries must be integer numbers without decimals or scientific notation. *Real number* entries can be integers, or numbers with decimals and scientific notation (e.g., 9.9832e+2, 998.32, 99.832E+01). Input formatting is described using a formatting guide and example *cards*. Notation for the formatting guide is shown in Table B.1.

Table B.1. Notation Guide for Input Formatting

Notation	Description
{ Option }	Character string options are indicated by enclosing braces. Options are chosen by entering word(s) within the braces, exactly as shown. Only one option should be chosen for each data entry.
[Optional]	Enclosing brackets indicate optional characters or words. These characters can be included in the input file to improve its readability or to specify optional features.
{{ Contains }}	Indicates the option contains a particular word. For example "Fractured Tuff" contains the word "Fractured" thus indicating a dual-porosity type rock/soil.
< Data types >	Indicates repeated formatting.
Char^{a}	Character string data type, referenced by superscript "a".
Integer^{a}	Integer data type (no character data or decimal points) referenced by superscript "a".
Real^{a}	Real data type (decimal points and exponential notation are acceptable), referenced by superscript "a".
#	A pound symbol in the first column indicates a comment line and will be ignored during execution. Comment lines may be placed inside or outside card structures. All lines outside of the card structures are ignored during execution.
~Card Name	A tilde symbol in the first column indicates the start of a new card.
,	Data entries are comma delimited. Commas shown in the line format structures must be entered as shown, including a closing comma at the end of each line. Characters following the last comma of a data line are ignored during execution.

Units^a (m)	Indicates the SI unit for the input data item referenced by superscript “a”.
|	Indicates a choice between more than one option.
Format:	Indicates line formatting instructions and the beginning of a new input line. Each format statement requires a new input line.
Endcard:	Indicates the end of a card.
For: Integer Instructions **Endfor:** Integer	Indicates instruction looping.
If: Name: Card = { Opt_1 } Instrucitons1 **Elseif:** Name: Card = { Opt_2 } Instructions2 **Elseif:** Instructions3 **Endif:**	Indicates decision logic.
IfDef: Opt_1 Instructions1 **ElseifDef:** Opt_2 Instructions2 **ElseDef:** Instructions3 **EndifDef:**	Indicates C preprocessor options and logic.
Note:	Indicates formatting information.

B.1 Simulation Title Card

Card Title[a] { ~Simulation Title Card }
Format: Char[a]

Version Number[a],
Format: Integer[a],

Simulation Title[a],
Format: Char[a],

User Name[a],
Format: Char[a],

Company Name[a],
Format: Char[a],

Input Creation Date[a],
Format: Char[a],

Input Creation Time[a],
Format: Char[a],

Number of Simulation Note Lines[a],
Format: Integer[a],

For: Number of Simulation Note Lines
Simulation Notes[a]
Format: Char[a] (maximum of 132 characters per line)
Endfor: Number of Simulation Note Lines

Endcard: Simulation Title Card

B.1.1 Simulation Title Card Example

```
~Simulation Title Card
1,
Problem 4,
M.D. White,
Pacific Northwest Laboratory,
26 August 2002,
14:45 AM PST,
10,
Intercomparison of simulation models for CO2 disposal in
underground storage reservoirs.
Test Problem 4: CO2 Discharge Along a Fault Zone
This problem explores CO2 loss from storage through a leaky fault,
using a highly simplified 1-D linear flow geometry. It is envisioned
that an aquifer into which CO2 disposal is made is intersected by a
vertical fault, which establishes a connection through an otherwise
impermeable caprock to another aquifer 500 m above the storage aquifer.
This situation is idealized by assuming 1-D flow geometry and constant
pressure boundary conditions (Pruess and Garcia, 2000).
```

B.1.2 Simulation Title Card Example

```
~Simulation Title Card
1,
Stuttgart 1.1,
Mark White,
Pacific Northwest Laboratory,
01 June 2011,
09:37 PDT,
8,
2.1 Definition of benchmark problem 1: CO2 plume evolution and
leakage through an abandoned well
2.1.1 Formulated by A. Ebigbo, J. Nordbotten and H. Class
Problem description CO2 is injected into an aquifer; spreads within the
aquifer and, upon reaching a leaky well, rises up to a shallower
aquifer. A quantification of the leakage rate which depends on the
pressure build- up in the aquifer due to injection and on the plume
evolution is the goal of the simulation.
```

B.2 Solution Control Card

Card Title[a] {~Solution Control Card }
    Format: Char[a]

Execution Mode Option[a], { Normal | Restart | Restart File | Initial Conditions }

    {{ No Flow }} (i.e., flow solution is only computed once)
    {{ Extended Output }} (i.e., additional convergence failure information)
    {{ Summary }} (i.e., additional linear system solver information)

If: Execution Mode Option = { **Restart File** }
    Restart File Name[b],
    Format: Char[a], Char[b],
Else:
    Format: Char[a],
Endif

Operational Mode Option[a], { STOMP-CO2 | STOMP-CO2e }
    Format: Char[a],

If: Operational Mode Option = { **STOMP-CO2e** }
    {{ Isobrine }} (i.e., constant salt concentration, inactive salt mass equation)
    {{ Isothermal }} (i.e., constant temperature, inactive energy equation)
Endif:

{{ Invariant }} (i.e., aqueous and gas density and viscosity are constant)
{{ Fractional CO2 Solubility }} (i.e., CO_2 solubility in aqueous is reduced)
{{ Transport }} (i.e., solute transport is activated)
{{ ECKEChem }} (i.e., reactive transport is activated)
{{ Initial Geomechanics }} (i.e., initial state geomechanics)
{{ Geomechanics }} (i.e., geomechanics is activated)

If: Operational Mode Option = { **Transport | ECKEChem** }
    {{ Courant }} (i.e., Courant number limiting on transport time step)
    {{ Leonard | TVD }} (i.e., Leonard-TVD solute/species transport scheme)
    {{ Roe | Superbee }} (i.e., Roe's Superbee solute/species transport scheme)
    {{ First-Order | Upwind }} (i.e., Upwind solute/species transport scheme)
    {{ Patankar | null }} (i.e., Patankar solute/species transport scheme)
Endif:

If: Operational Mode Option = { **ECKEChem** }
    {{ Equilibrium Reduced }} (i.e., equilibrium equations eliminated from coupled solve)
    {{ Mixing Coefficient }} (i.e., mixing coefficient read for TST reactions)
    {{ Minimum Concentration }} (i.e., minimum species concentration)
    {{ Log }} (i.e., logarithm formulation)
    {{ Guess }} (i.e., create an initial guess of species concentrations)
    {{ Porosity }} (i.e., porosity alteration with mineral precipitation)
    {{ Area }} (i.e, mineral effective surface area scaled with aqueous saturation)
Endif:

```
Number of Execution Time Periods[a],
Format: Integer[a],

For: Number of Execution Time Periods

    If: Execution Mode Option = { Normal }
        Initial Time[a], Units[b] (s), Final Time[c], Units[d] (s),
        Initial Time Step[e], Units[f] (s),
        Maximum Time Step[g], Units[h] (s),
        Time Step Acceleration Factor[i],
        Maximum Number of Newton-Raphson Iterations[j], Convergence Criterion[k],
        [ Minimum Time Step[l], Minimum Time Step Units[m], Time Step Cut Factor[n] ],
        Format: Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Integer[j],
            Real[k], [ Real[l], Char[m], Real[n], ]
    Elseif: Execution Mode Option = { Restart }
        Initial Time[a], Units[b], (s), Final Time[c], Units[d] (s),
        Initial Time Step[e], Units[f] (s),
        Maximum Time Step[g], Units[h] (s),
        Time Step Acceleration Factor[i],
        Maximum Number of Newton-Raphson Iteratsion[j],
        Convergence Criterion[k], [ Minimum Time Step[l], Minimum Time Step Units[m],
        Time Step Cut Factor[n] ],
        Format: Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Integer[j],
            Real[k], [ Real[l], Char[m], Real[n], ]
    Endif:

Endfor: Number of execution Time Periods


If: Operational Mode Option contains { ECKEChem }
    Maximum Number of Time Steps[a],
    [Number of Reactive Transport Sequence Iterations[b], ]
    Format: Integer[a], [Integer[b],]
Else:
    Maximum Number of Time Steps[a],
    Format: Integer[a],
Endif:

Aqueous Diffusion Option[a], { Zero } { Constant } { Variable }

If: Aqueous Diffusion Option = { Constant }
    Aqueous $CO_2$ Diffusion Coefficient[b], Units[c] (m^2/s),
    Aqueous Salt Diffusion Coefficient[d], Units[e] (m^2/s),
    Format: Char[a], Real[b], Char[c], Real[d],
Endif:

Gas  Diffusion Option[a], { Zero } { Constant } { Variable }

If: Gas Diffusion Option = { Constant }
    Water Vapor Diffusion Coefficient[b], Units[c] (m^2/s),
    Format: Char[a], Real[b], Char[c],
Endif:
```

```
If: Fluid Density and Viscosity = { Invariant }
    Aqueous Density$^a$, Units$^b$ (kg/m^3),
    Aqueous Viscosity$^c$, Units$^d$ (Pa s),
    Gas Density$^e$, Units$^f$ (kg/m^3),
    Gas Viscosity$^g$, Units$^h$ (Pa s),

    If: CO2 Solubility = { Fractional CO2 Solubility }
        Fractional CO2 Solubility Factor$^i$,
        Format: Real$^a$, Char$^b$, Real$^c$, Char$^d$, Real$^e$, Char$^f$, Real$^g$, Char$^h$, Real$^i$,
    Else:
        Format: Real$^a$, Char$^b$, Real$^c$, Char$^d$, Real$^e$, Char$^f$, Real$^g$, Char$^h$,
    Endif:

Else:

    If: CO2 Solubility = { Fractional CO2 Solubility }
        Fractional CO2 Solubility Factor$^a$,
        Format: Real$^a$,
    Endif:

Endif:

Number of Interfacial Averaging Variables$^a$
Format: Integer$^a$,

For: Number of Interfacial Averaging Variables
    Surface Variable Option$^a$,
    { Air Gas Diffusion | Air Aqueous Diffusion |
        Aqueous Density | Aqueous Relative Permeability |
        Aqueous Viscosity | Effective Permeability |
        Gas Density | Gas Relative Permeability |
        Gas Viscosity | Hydraulic Dispersion |
        Intrinsic Permeability | NAPL Density |
        NAPL Relative Permeability | NAPL Viscosity |
        Solute Diffusion | Thermal Conductivity |
        Oil Gas Diffusion | Oil Aqueous Diffusion |
        Salt Aqueous Diffusion | Water Gas Diffusion }
        Interfacial Averaging Scheme Option$^b$
    { Harmonic | Geometric | Arithmetic | Upwind | Downstream |
        Moderated Upwind | Neiber Downstream }

    If: Interfacial Averaging Scheme Option = { Downstream | Neiber Downstream }
        Weighting Factor$^c$,
        Format: Char$^a$, Char$^b$, Real$^c$,
    Elseif: Interfacial Averaging Scheme Option = { Moderated Upwind }
        Moderation Asymptote$^c$,
        Format: Char$^a$, Char$^b$, Real$^c$,
    Else:
        Format: Char$^a$, Char$^b$,
    Endif:

Endfor: Number of Interfacial Averaging Variables
```

If: Operational Mode Option contains { { **ECKEChem** }}
and Operational Mode Option contains {{ **Minimum Concentration** }}
Minimum Aqueous Concentration in ECKEChem[a],
Format: Real[a],
Endif:

Endcard: Solution Control Card

B.2.1 Solution Control Card Example

```
~Solution Control Card
Normal,
STOMP-CO2 w/ Invariant Density and Viscosity w/ Fractional CO2 Solubility,
1,
0,day,1000,day,1,s,10,day,1.25,16,1.e-06,0.001,s,0.2,
10000,
Variable Aqueous Diffusion,
Variable Gas Diffusion,
1045,kg/m^3,2.535e-4,Pa s,479,kg/m^3,3.950e-5,Pa s,1.e-3,
0,
```

B.2.2 Solution Control Card Example

```
~Solution Control Card
Restart File,restart.77,
STOMP-CO2,
1,
0,s,1.e+11,s,1.e+0,s,1.e+11,s,1.25,16,1.e-06,
10000,
Variable Aqueous Diffusion,
Variable Gas Diffusion,
0,
```

B.2.3 Solution Control Card Example

```
~Solution Control Card
Normal,
STOMP-CO2e,
1,
0,s,1000000,yr,1.0,s,1000000,yr,1.25,16,1.e-06,
10000,
Variable Aqueous Diffusion,
Variable Gas Diffusion,
0,
```

B.3 Grid Card

```
Card Title[a]{ ~Grid Card }
    Format: Char[a]

Coordinate System Option[a], { Tilted Cartesian | Uniform Cartesian | Cartesian |
    Uniform Cylindrical | Cylindrical | Orthogonal | Generic Eclipse |
    Sampled Earthvision | Element Vertices }

If: Coordinate System Option = { Tilted Cartesian }
    X-Z Plane Tilt Angle[b], Units[c] (deg),
    Y-Z Plane Tilt Angle[d], Units[e] (deg),

    If: Coordinate System Option = {{ Reference }}
     X Reference Point[f], Units[g] (m), Direction Index[h],
     Y Reference Point[i], Units[j] (m), Direction Index[k],
     Z Reference Point[l], Units[m] (m), Direction Index[n],
     Format: Char[a], Real[b], Char[c], Real[d], Char[e], Real[f], Char[g], Integer[h],
        Real[i], Char[j], Integer[k], Real[l], Char[m], Integer[n],
    Elseif:
     Format: Char[a], Real[b], Char[c], Real[d], Char[e],
    Endif:

Elseif:

    If: Coordinate System Option = {{ Reference }}
     X Reference Point[b], Units[c] (m), Direction Index[d],
     Y Reference Point[e], Units[f] (m), Direction Index[g],
     Z Reference Point[h], Units[i] (m), Direction Index[j],
     Format: Char[a], Real[b], Char[c], Integer[d], Real[e], Char[f], Integer[g], Real[h], Char[i], Integer[j],
    Elseif:
     Format: Char[a],
    Endif:

Endif:

Number of X-Dir. Nodes[a],
Number of Y-Dir. Nodes[b],
Number of Z-Dir. Nodes[c],
External File Name[d], Dimensional Units[e] (m),
Format: Integer[a], Integer[b], Integer[c],Char[d], Char[e],

If: Coordinate System Option = { Tilted Cartesian } { Cartesian }

    For: Number of X-Dir. Nodes + 1
        < Surface Position[a], Units[b] (m), > or < Count[a] @ Node Width[b], Units[c] (m), >
        Format: < Real[a], Char[b], > or < Integer[a]@Real[a], Char[b], >
    Endfor: Number of X-Dir. Nodes
    For: Number of Y-Dir. Nodes + 1
        < Surface Position$^a$, Units$^b$ (m), > or < Count$^a$ @ Node Width$^b$, Units$^c$ (m), >
        Format: < Real$^a$, Char$^b$, > or < Integer$^a$@Real$^a$, Char$^b$, >
    Endfor: Number of Y-Dir. Nodes

    For: Number of Z-Dir. Nodes + 1
        < Surface Position$^a$, Units$^b$ (m), > or < Count$^a$ @ Node Width$^b$, Units$^c$ (m), >
        Format: < Real$^a$, Char$^b$, > or < Integer$^a$@Real$^a$, Char$^b$, >
    Endfor: Number of Z-Dir. Nodes

Elseif: Coordinate System Option = { Cylindrical }

    For: Number of Radial-Dir. Nodes + 1
        < Surface Position$^a$, Units$^b$ (m), > or < Count$^a$ @ Node Width$^b$, Units$^c$ (m), >
        Format: < Real$^a$, Char$^b$, > or < Integer$^a$@Real$^a$, Char$^b$, >
    Endfor: Number of Radial-Dir. Nodes

    For: Number of Azimuthal-Dir. Nodes + 1
        < Surface Position$^a$, Units$^b$ (deg), > or < Count$^a$ @ Node Width$^b$, Units$^c$ (deg), >
        Format: < Real$^a$, Char$^b$, > or < Integer$^a$@Real$^a$, Char$^b$, >
    Endfor: Number of Azimuthal-Dir. Nodes

    For: Number of Z-Dir. Nodes + 1
        < Surface Position$^a$, Units$^b$ (m), > or < Count$^a$ @ Node Width$^b$, Units$^c$ (m), >
        Format: < Real$^a$, Char$^b$, > or < Integer$^a$@Real$^a$, Char$^b$, >
    Endfor: Number of Z-Dir. Nodes

Elseif: Coordinate System Option = { Uniform Cartesian }

    X-Dir. Node Dimension$^a$, Units$^b$ (m)
    Format: Real$^a$, Char$^b$,

    Y-Dir. Node Dimension$^a$, Units$^b$ (m)
    Format: Real$^a$, Char$^b$,

    Z-Dir. Node Dimension$^a$, Units$^b$ (m)
    Format: Real$^a$, Char$^b$,

Elseif: Coordinate System Option = { Uniform Cylindrical }

    Radial-Dir. Node Dimension$^a$, Units$^b$ (m)
    Format: Real$^a$, Char$^b$,

    Azimuthal-Dir. Node Dimension$^a$, Units$^b$ (deg)
    Format: Real$^a$, Char$^b$,

    Z-Dir. Node Dimension$^a$, Units$^b$ (m)
    Format: Real$^a$, Char$^b$,

Elseif: Coordinate System Option = { Generic Eclipse }

    External File Name$^a$, Minimum Z-Direction Spacing$^b$, Dimensional Units$^c$ (m),
```

Format: Char^{a}, Real^{b}, Char^{c},

Elseif: Coordinate System Option = { **Sampled Earthvision** } { **Orthogonal** }

External File Name^{a},
Format: Char^{a},

Elseif: Coordinate System Option = { **Element Vertices** }

External Vertices File Name^{a}, Number of Vertices^{b},
Format: Char^{a}, Integer^{b},
External Element File Name^{a},
Format: Char^{a},

Endif:

Endcard: Grid Card

B.3.1 Grid Card Example

```
~Grid Card
Cylindrical,
100,1,1,
0.3,m,0.34068267,m,0.386882272,m,0.439346951,m,0.498926308,m,0.566585156,m,
0.643419145,m,0.730672508,m,0.829758203,m,0.9422808,m,1.070062462,m,
1.215172455,m,1.379960655,m,1.567095601,m,1.779607712,m,2.020938356,m,
2.294995583,m,2.606217409,m,2.959643684,m,3.360997708,m,3.81677891,m,
4.334368098,m,4.922146988,m,5.589633925,m,6.347638032,m,7.208434242,m,
8.185962079,m,9.296051391,m,10.55667869,m,11.98825828,m,13.61397279,m,
15.46014866,m,17.55668242,m,19.9375248,m,22.64123061,m,25.71158298,m,
29.19830246,m,33.15785213,m,37.65435198,m,42.76061723,m,48.55933748,m,
55.14441582,m,62.62248937,m,71.11465626,m,80.75843656,m,91.70999929,m,
104.1466914,m,118.2699096,m,134.308362,m,152.5217712,m,173.2050808,m,
196.6932312,m,223.3665839,m,253.6570806,m,288.0552382,m,327.1180922,m,
371.4782167,m,421.853969,m,479.0611216,m,544.0260733,m,617.8008506,m,
701.5801442,m,796.7206557,m,904.7630673,m,1027.456991,m,1166.789304,m,
1325.016317,m,1504.700323,m,1708.751078,m,1940.472932,m,2203.618331,m,
2502.448588,m,2841.802888,m,3227.176651,m,3664.810527,m,4161.79145,m,
4726.16741,m,5367.077773,m,6094.901285,m,6921.424143,m,7860.030856,m,
8925.920993,m,10136.35532,m,11510.93531,m,13071.92059,m,14844.58935,m,
16857.64778,m,19143.69485,m,21739.75025,m,24687.85387,m,28035.74657,m,
31837.64332,m,36155.1111,m,41058.06594,m,46625.90509,m,52948.79278,m,
60129.12031,m,68283.16417,m,77542.96894,m,88058.48564,m,100000,m,
0.0,deg,45.0,deg,
0.0,m,100.0,m,
```

B.3.2 Grid Card Example

```
~Grid Card
Cartesian,
100,1,193,
0.0,m,2.0,m,4.1,m,6.3,m,8.6,m,11.1,m,
13.7,m,16.4,m,19.2,m,22.2,m,25.4,m,28.7,m,
32.2,m,35.8,m,39.7,m,43.8,m,48.0,m,52.5,m,
57.2,m,62.2,m,67.4,m,72.9,m,78.7,m,84.8,m,
91.2,m,97.9,m,105.0,m,112.4,m,120.2,m,128.5,m,
137.1,m,146.2,m,155.8,m,165.9,m,176.5,m,187.6,m,
199.4,m,211.7,m,224.7,m,238.3,m,252.7,m,267.8,m,
283.6,m,300.3,m,317.9,m,336.4,m,355.8,m,376.3,m,
397.8,m,420.4,m,444.2,m,469.2,m,495.5,m,523.2,m,
552.3,m,582.9,m,615.1,m,649.0,m,684.7,m,722.2,m,
761.6,m,803.1,m,846.7,m,892.6,m,940.8,m,991.6,m,
1045.0,m,1101.1,m,1160.2,m,1222.3,m,1287.7,m,1356.4,m,
1428.7,m,1504.8,m,1584.8,m,1668.9,m,1757.4,m,1850.5,m,
1948.4,m,2051.4,m,2159.7,m,2273.7,m,2393.5,m,2519.6,m,
2652.2,m,2791.6,m,2938.3,m,3092.6,m,3254.9,m,3425.6,m,
3605.2,m,3794.0,m,3992.7,m,4201.6,m,4421.4,m,4652.5,m,
4895.7,m,5151.4,m,5420.4,m,5703.4,m,6000.0,m,
0.0,m,1.0,m,
0.0,m,1@0.5,m,51@1.0,m,1@0.5,m,
1@0.5,m,2@1.0,m,1@0.5,m,
1@0.5,m,29@1.0,m,1@0.5,m,
1@0.5,m,2@1.0,m,1@0.5,m,
1@0.5,m,29@1.0,m,1@0.5,m,
1@0.5,m,2@1.0,m,1@0.5,m,
1@0.5,m,29@1.0,m,1@0.5,m,
1@0.5,m,2@1.0,m,1@0.5,m,
1@0.5,m,29@1.0,m,1@0.5,m,
```

B.3.3 Grid Card Example

```
~Grid Card
Uniform Cartesian,
1,1,100,
25.0,m,
1.0,m,
5.0,m,
```

B.3.4 Grid Card Example

```
~Grid Card
Generic Eclipse Grid,
51,50,16,
grid.txt,0.1,m,
```

B.3.5 Grid Card Example

```
~Grid Card
Cartesian,
73,32,44,
-500.000,m,-475.000,m,-450.000,m,-425.000,m,-400.000,m,-375.000,m,
-350.000,m,-325.000,m,-300.000,m,-275.000,m,-255.000,m,-235.000,m,
-215.000,m,-200.000,m,-185.000,m,-170.000,m,-160.000,m,-150.000,m,
-140.000,m,-130.000,m,-122.500,m,-115.000,m,-107.500,m,-102.500,m,
-97.500,m,-92.500,m,-85.000,m,-77.500,m,-70.000,m,-60.000,m,
-50.000,m,-40.000,m,-30.000,m,-20.000,m,-12.500,m,-8.000,m,
-5.000,m,-2.800,m,-1.500,m,-0.800,m,-0.400,m,-0.133,m,
0.133,m,0.400,m,0.800,m,1.500,m,2.800,m,5.000,m,
8.000,m,12.500,m,20.000,m,30.000,m,40.000,m,50.000,m,
60.000,m,70.000,m,85.000,m,100.000,m,125.000,m,150.000,m,
175.000,m,200.000,m,225.000,m,250.000,m,275.000,m,300.000,m,
325.000,m,350.000,m,375.000,m,400.000,m,425.000,m,450.000,m,
475.000,m,500.000,m,
0.000,m,0.133,m,0.400,m,0.800,m,1.500,m,2.800,m,
5.000,m,8.000,m,12.500,m,20.000,m,30.000,m,40.000,m,
50.000,m,60.000,m,70.000,m,85.000,m,100.000,m,125.000,m,
150.000,m,175.000,m,200.000,m,225.000,m,250.000,m,275.000,m,
300.000,m,325.000,m,350.000,m,375.000,m,400.000,m,425.000,m,
450.000,m,475.000,m,500.000,m,
0.0,m,4.41,m,8.32,m,11.80,m,14.90,m,17.65,m,20.09,m,22.27,m,24.20,m,
25.92,m,27.44,m,28.80,m,30.0,m,20@5.000,m,12@2.500,m,
```

B.3.6 Grid Card Example

```
~Grid Card
#Boundary Fitted,
earthvision sampled input,
81,79  ,52  ,
fg_uic.dat,ft,
```

B.3.7 Grid Card Example

```
~Grid Card
Element and Vertices,
78,78,9,
vertices_johansen_stomp.dat,438048,
elements_johansen_stomp.dat,
```

B.4 Internal Boundary Surfaces Card

```
Card Title[a] {~Internal Boundary [ Surfaces Card ] }
Format: Char[a]

Number of Internal Boundary Condition Domains[a],
Format: Integer[a],

For: Number of Boundary Condition Domains

    Boundary Surface Direction Option[a]
    { Bottom } { South } { West }{ East }{ North }{ Top }{ File }

    If: Boundary Surface Direction Option=
        { Bottom } { South } { West } { East } { North } { Top }
        Format: Integer[a],
    Elseif: Boundary Surface Direction Option={ File }
        Boundary Surface Direction Option={ File }
        Format: Char[a],
   Endif:

    If: Boundary Surface Direction Option=
        { Bottom } { South } { West } { East } { North } { Top }
        I-Start Index[a], I-End Index[b], J-Start Index[c], J-End Index[d],
        K-Start Index[e], K-End Index[f],
        Format: Integer[a], Integer[b], Integer[c], Integer[d], Integer[e], Integer[f],
   Endif:


Endfor:
```

B.4.1 Internal Boundary Surfaces Card Example

```
~Boundary Conditions Card
2,
East,
17,17,1,57,20,41,
West,
38,38,1,57,20,41,
```

B.4.2 Internal Boundary Surfaces Card Example*

```
~Boundary Conditions Card
4,
file,east_internal.dat,
file,west_internal.dat,
file,top_internal.dat,
file,bottom_internal.dat,
```

*file format: I-Index, J-Index, K-Index, Boundary Direction Index {Bottom = -3, South = -2, West = -1, East = 1, North = 2, Top = 3}

B.5 Inactive Nodes Card

```
Card Title[a] { ~Inactive [ Nodes Card ] }
Format: Char[a]

Inactive Domain Input Option[a],
    { [ Rock | Soil ] | Zonation File [ Formatted ] | File | Integer }

If: Inactive Domain Input Option = { [ Rock | Soil ] }
    Number of Rock/Soil Type Lines[b],
    Format: Char[a], Integer[b],

    For: Number of Rock/Soil Type Lines
        Inactive Rock/Soil Types[a],
        Format: Char[a],
    Endfor: Number of Rock/Soil Type Lines

Elseif: Inactive Domain Input Option = { Zonation File }
    Zonation File Name[b],
    Format: Char[a], Char[b],

Elseif: Inactive Domain Input Option = { File }
    Inactive Node File Name[b],
    Format: Char[a], Char[b],

Elseif: Inactive Domain Input Option = { Integer }
    Number of Inactive Node Domains[a],
    Format: Integer[a],

    For: Number of Inactive Node Domains
        I-Start Index[a], I-End Index[b],
        J-Start Index[c], J-End Index[d],
        K-Start Index[e], K-End Index[f],
        Format: Integer[a], Integer[b], Integer[c], Integer[d], Integer[e], Integer[f],
    Endfor: Number of Inactive Node Domains

Endif:

Endcard: Inactive Nodes Card
```

B.5.1 Inactive Nodes Card Example

```
~Inactive Nodes Card
3,
1,41,1,32,13,32,
43,73,1,32,13,32,
42,42,2,32,13,32,
```

B.5.2 Inactive Nodes Card Example

```
~Inactive Nodes Card
file,inactive.dat,
```

B.5.3 Inactive Nodes Card Example

```
~Inactive Nodes Card
zonation file, zonation_4,
```

B.5.4 Inactive Nodes Card Example

```
~Inactive Nodes Card
Rock/Soil Types,2,
fill material,
engineered structure,
```

B.6 Rock/Soil Zonation Card

```
Card Title[a] { ~Rock/Soil [ Zonation Card ] }
Format: Char[a]

Rock/Soil Zonation Input Option[a],
    {  File  |  [ Formatted  |  Unformatted ] Zonation File  |  [ IJK  |  JKI  |  KIJ ] Indexing  |  Integer }

If: Rock/Soil Zonation Input Option = { [ Formatted  |  Unformatted ] Zonation File }
      Rock/Soil Zonation File Name[b],
      Format: Char[a], Char[b],
      For: Number of Rock/Soil Zonation Domains (defined in external file)
          Rock/Soil or Scaling Group Name[a],
          Format: Char[a],
      Endfor: Number of Rock/Soil Zonation Domains

Elseif:  Rock/Soil Zonation Input Option = { File }
     External File Name[b],Number of Zonation Lines[c],
     For: Number of Zonation Lines
          Rock/Soil or Scaling Group name[d],
      Endfor:
      Format: Char[a], Char[b], Integer[c], <Char[d]>,

Elseif: Rock/Soil Zonation Input Option = { Indexing }

      Note: Each node is assigned a different Rock/Soil Zonation index according to the indexing scheme
      chosen (i.e., IJK, JKI, or KIJ).  This option is useful for stochastic realizations.

Elseif: Rock/Soil Zonation Input Option = { Integer }

      For: Number of Rock/Soil Zonation Domains
          Rock/Soil or Scaling Group Name[a],
          I-Start Index[b], I-End Index[c],
          J-Start Index[d], J-End Index[e],
          K-Start Index[f], K-End Index[g],
          Format: Char[a], Integer[b], Integer[c], Integer[d], Integer[e], Integer[f], Integer[g],
      Endfor: Number of Rock/Soil Zonation Domains

Endif:

Endcard: Rock/Soil Zonation Card
```

B.6.1 Rock/Soil Zonation Card Example

```
~Rock/Soil Zonation Card
1,
Aquifer,1,100,1,1,1,1,
```

B.6.2 Rock/Soil Zonation Card Example

```
~Rock/Soil Zonation Card
2,
aquifer,1,73,1,32,1,44,
leaky well,42,42,1,1,1,44,
```

B.6.3 Rock/Soil Zonation Card Example

```
~Rock/Soil Zonation Card
9,
Sands,1,100,1,1,1,18,
Shale,1,100,1,1,19,21,
Sands,1,100,1,1,22,33,
Shale,1,100,1,1,34,36,
Sands,1,100,1,1,37,48,
Shale,1,100,1,1,49,51,
Sands,1,100,1,1,52,63,
Shale,1,100,1,1,64,66,
Sands,1,100,1,1,67,78,
```

B.6.4 Rock/Soil Zonation Card Example

```
~Rock/Soil Zonation Card
IJK Indexing,
```

B.6.5 Rock/Soil Zonation Card Example

```
~Rock/Soil Zonation Card
formatted file, hg.dat,
backfill,
hanford sand,
plio-pleistocene,
upper ringold,
middle ringold,
```

B.6.6 Rock/Soil Zonation Card Example

```
~Rock/Soil Zonation Card
zonation file formatted,c_geology.dat,
Backfill,
H2 Sand,
H3 Gravelly Sand,
H1 Gravelly Sand,
PPlgR,
Aquifer,
```

B.7 Vertical Equilibrium Card

Card Title^{a} { ~Vertical Equilibrium Card }
Format: Char^{a}

Number of Vertical Equilibrium Domains^{a},
Format: Integer^{a}

For: Number of Vertical Equilibrium Domains
    Starting I Index^{a},
    Ending I Index^{b},
    Starting J Index^{c},
    Ending J Index^{d},
    Starting K Index^{e},
    Ending K Index^{f},
Endfor:
Format: Integer^{a}, Integer^{b}, Integer^{c}, Integer^{d}, Integer^{e}, Integer^{f},

Endcard: Vertical Equilibrium Card

B.7.1 Vertical Equilibrium Card Example

```
~Vertical Equilibrium Card
1,
1,100,1,75,1,50,
```

B.7.2 Vertical Equilibrium Card Example

```
~Vertical Equilibrium Card
2,
1,100,1,75,1,25,
1,100,1,75,26,50,
```

B.8 Mechanical Properties Card

```
Card Title[a] { ~Mechanical [ Properties Card ] }
Format: Char[a]

If: Rock/Soil or Scaling Group Name = { IJK | JKI | KIJ } Indexing

    Note: Parameter input can be replaced with an external file using the
    following formatting for ASCII files:

        file: filename

    or the following formattings for binary files:

        binary file: filename

    where; the external file will contain unique parameter values for each node
    (active or inactive) arranged according to the indexing scheme (i.e., IJK, JKI, or KIJ).
    Applicable units will be applied to all parameter values in the external file.

Elseif:

For: Number of Rock/Soil or Scaling-Group Types

    Rock/Soil or Scaling-Group Name[a], Particle Density[b] (2650.0), Units[c] (kg/m^3),
    If: Rock/Soil or Scaling-Group Name = {{ Fractured }}
        Total Porosity[d], Diffusive Porosity[e],
        Fracture Total Porosity[f], Fracture Diffusive Porosity[g],
        If: Compressibility
            {{ [ Pore ] Compressibility }}[h], Compressibility[i], Units[j],
            Fracture Compressibility[k], Units[l],
            Compressibility Reference Pressure[m], Units[n],
            Tortuosity Function Option[o],
                { Constant  | Constant-Aqueous Millington-Gas | Millington and Quirk |
                 Marshall | Free Gas }
            If: Tortuosity Function Option = { Constant }
                Aqueous-Phase Tortuosity[p], Gas-phase Tortuosity[q],
            Format: Char[a], Real[b], Char[c], Real[d], Real[e], Real[f], Real[g], Char[h], Real[i],
                        Char[j], Real[k], Char[l], Real[m], Char[n], Char[o], Real[p], Real[q],
            Elseif: Tortuosity Function Option = { Constant-Aqueous Millington-Gas }
                Aqueous-Phase Tortuosity[p],
                Format: Char[a], Real[b], Char[c], Real[d], Real[e], Real[f], Real[g], Char[h], Real[i],
                        Char[j], Real[k], Char[l], Real[m], Char[n], Char[o], Real[p],
            Else:
                Format: Char[a], Real[b], Char[c], Real[d], Real[e], Real[f], Real[g], Char[h], Real[i],
                        Char[j], Real[k], Char[l], Real[m], Char[n], Char[o],
            Endif:
        Else:
            Specific Storativity[h], Units[i] (1/m),
            Fracture Specific Storativity[j], Units[k] (1/m),
            Tortuosity Function Option[l],
                { Constant  | Constant-Aqueous Millington-Gas | Millington and Quirk |
                 Marshall | Free Gas }
            If: Tortuosity Function Option = { Constant }
                Aqueous-Phase Tortuosity[m], Gas-Phase Tortuosity[n],
                Format: Char[a], Real[b], Char[c], Real[d], Real[e], Real[f],
                        Real[g], Real[h], Char[i], Real[j], Char[k], Char[l], Real[m], Real[n],
            Elseif: Tortuosity Function Option = { Constant-Aqueous Millington-Gas }
                Aqueous-Phase Tortuosity[m],
                Format: Char[a], Real[b], Char[c], Real[d], Real[e], Real[f],
                        Real[g], Real[h], Char[i], Real[j], Char[k], Char[l], Real[m],
            Else:
                Format: Char[a], Real[b], Char[c], Real[d], Real[e], Real[f],
                        Real[g], Real[h], Char[i], Real[j], Char[k], Char[l],
            Endif:
        Endif:
    Else:
        Total Porosity[d], Diffusive Porosity[e],
        If: Compressibility
            {{ [ Pore ] Compressibility }}[f], Compressibility[g], Units[h],
            Compressibility Reference Pressure[i], Units[j],
            Tortuosity Function Option[k],
                { Constant  | Constant-Aqueous Millington-Gas | Millington and Quirk |
                 Marshall | Free Gas }
            If: Tortuosity Function Option = { Constant }
                Aqueous-Phase Tortuosity[l], Gas-phase Tortuosity[m],
            Format: Char[a], Real[b], Char[c], Real[d], Real[e], Char[f], Real[g],
                        Char[h], Real[i], Char[j], Char[k], Real[l], Real[m],
            Elseif: Tortuostiy Function Option = { Constant-Aqueous Millington-Gas }
                Aqueous-Phase Tortuosity[l],
                Format: Char[a], Real[b], Char[c], Real[d], Real[e], Char[f], Real[g],
                        Char[h], Real[i], Char[j], Char[k], Real[l],
            Else:
                Format: Char[a], Real[b], Char[c], Real[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j],
                         Char[k],
            Endif:
        Else:
            Specific Storativity[f], Units[g] (1/m),
            Tortuosity Function Option[h],
                { Constant  | Constant-Aqueous Millington-Gas | Millington and Quirk |
                 Marshall | Free Gas }
            If: Tortuosity Function Option = { Constant }
                Aqueous-Phase Tortuostiy[i], Gas-Phase Tortuosity[j],
                Format: Char[a], Real[b], Char[c], Real[d], Real[e], Real[f], Char[g], Char[h], Real[i], Real[j],
            Elseif: Tortuostiy Function Option = { Constant-Aqueous Millington-Gas }
                Aqueous-Phase Tortuosity[i],
                Format: Char[a], Real[b], Char[c], Real[d], Real[e], Real[f], Char[g], Char[h], Real[i],
            Else:
                Format: Char[a], Real[b], Char[c], Real[d], Real[e], Real[f], Char[g], Char[h],
            Endif:
        Endif:
    Endif:
```

Endfor: Number of Rock/Soil or Scaling Group Types

Endcard: Mechanical Properties Card

B.8.1 Mechanical Properties Card Example

```
~Mechanical Properties Card
Sands,2650,kg/m^3,0.35,0.35,Compressibility,4.5e-10,1/Pa,,,Millington and
Quirk,
Shale,2650,kg/m^3,0.1025,0.1025,Compressibility,4.5e-10,1/Pa,,,Millington and
Quirk,
```

B.8.2 Mechanical Properties Card Example

```
~Mechanical Properties Card
aquifer,2650,kg/m^3,0.15,0.15,Compressibility,1.e-9,1/psi,,,constant,1.0,1.0,
leaky well,2650,kg/m^3,0.15,0.15,Compressibility,1.e-9,1/psi,,,constant,
1.0,1.0,
```

B.8.3 Mechanical Properties Card Example

```
~Mechanical Properties Card
SP1,2.63,g/cm^3,0.19,0.19,,,,
SP2,2.63,g/cm^3,0.24,0.24,,,,
SM-ML1,2.63,g/cm^3,0.35,0.35,,,,
SM-SP1,2.63,g/cm^3,0.37,0.37,,,,
SP3,2.63,g/cm^3,0.27,0.27,,,,
SW1,2.63,g/cm^3,0.28,0.28,,,,
US,2.63,g/cm^3,0.96,0.96,,,,
```

B.8.4 Mechanical Properties Card Example

```
~Mechanical Properties Card
IJK Indexing,2690,kg/m^3,file:por.dat,file:por.dat,,1/m,Millington and Quirk,
```

B.9 Hydraulic Properties Card

Card Title[a] { ~Hydraulic [Properties Card] }
Format: Char[a]

If: Rock/Soil or Scaling Group Name = { IJK | JKI | KIJ } Indexing

> **Note:** The parameter input can be replaced with an external file using the following formatting for ASCII files:
>
> file: filename
>
> or the following formatting for binary files:
>
> binary file: filename
>
> where; the external file will contain unique parameter values for each node (active or inactive) arranged according to the indexing scheme (i.e., IJK, JKI, or KIJ). Applicable units will be applied to all parameter values in the external file. An example input card is included in section 4.3.8.1.

Elseif:

For: Number of Rock/Soil or Scaling-Group Types

Rock/Soil or Scaling-Group Name[a],

If: Rock/Soil or Scaling-Group Name = {{ Fractured }} {{ DP }}

X-Dir. (Radial-Dir.) Matrix Intrinsic Permeability[b], Units[c] (m^2),
or X-Dir. (Radial-Dir.) Matrix Hydraulic Conductivity[b], Units[c] (hc m/s),

Y-Dir. (Azimuthal-Dir.) Matrix Intrinsic Permeability[d], Units[e] (m^2),
or Y-Dir. (Azimuthal-Dir.) Matrix Hydraulic Conductivity[d], Units[e] (hc m/s),

Z-Dir. Matrix Intrinsic Permeability[f], Units[g] (m^2),
or Z-Dir. Matrix Hydraulic Conductivity[f], Units[g] (hc m/s),

X-Dir. (Radial-Dir.) Fracture Intrinsic Permeability[h], Units[i] (m^2),
or X-Dir. (Radial-Dir.) Fracture Hydraulic Conductivity[h], Units[i] (hc m/s),

Y-Dir. (Azimuthal-Dir.) Fracture Intrinsic Permeability[j], Units[l] (m^2),
or Y-Dir. (Azimuthal-Dir.) Fracture Hydraulic Conductivity[j], Units[k] (hc m/s),

Z-Dir. Fracture Intrinsic Permeability[l], Units[m] (m^2),
or Z-Dir. Fracture Hydraulic Conductivity[l], Units[m] (hc m/s),

If: NaCl precipitation is considered,
Pore-body Fractional Length[o], Fractional Critical Porosity[p],
If: Kozeny & Carmen Intrinsic Permeability is used
{kozeny}[q],

```
            Format:  Char[a] Char[a], Real[b], Char[c], Real[d], Char[e], Real[f], Char[g], Real[h], Char[i], Real[j],
                Char[k], Real[l], Char[m], Real[o], Real[p],Char[q],
        Else:
            Format: Char[a], Real[b], Char[c], Real[d], Char[e], Real[f], Char[g], Real[h], Char[i], Real[j],
                Char[k], Real[l], Char[m], Real[o], Real[p],
        Endif:
    Else:
    If: Kozeny & Carmen Intrinsic Permeability is used
            {kozeny}[n],
            Format:  Char[a], Real[b], Char[c], Real[d], Char[e], Real[f], Char[g], Real[h], Char[i], Real[j],
                Char[k], Real[l], Char[m], Char[n],
        Else:
            Format: Char[a], Real[b], Char[c], Real[d], Char[e], Real[f], Char[g], Real[h], Char[i], Real[j],
                Char[k], Real[l], Char[m],
        Endif:
    Endif:

Else:

    X-Dir. (Radial-Dir.) Intrinsic Permeability[b], Units[c] (m^2),
    or X-Dir. (Radial-Dir.) Hydraulic Conductivity[b], Units[c] (hc m/s),

    Y-Dir. (Azimuthal-Dir.) Intrinsic Permeability[d], Units[e] (m^2),
    or Y-Dir. (Azimuthal-Dir.) Hydraulic Conductivity[d], Units[e] (hc m/s),

    Z-Dir. Intrinsic Permeability[f], Units[g] (m^2),
    or Z-Dir. Hydraulic Conductivity[f], Units[g] (hc m/s),

    If: NaCl precipitation is considered,
        Pore-body Fractional Length[h], Fractional Critical Porosity[i],
        If: Kozeny & Carmen Intrinsic Permeability is used
            {kozeny}[j],
            Format:  Char[a], Real[b], Char[c], Real[d], Char[e], Real[f], Char[g], Real[h], Real[i],Char[j],
        Else:
            Format: Char[a], Real[b], Char[c], Real[d], Char[e], Real[f], Char[g], Real[h], Real[i],
        Endif:
    Else:
        If: Kozeny & Carmen Intrinsic Permeability is used
            {kozeny}[h],
            Format:  Char[a], Real[b], Char[c], Real[d], Char[e], Real[f], Char[g], Char[h],
        Else:
            Format: Char[a], Real[b], Char[c], Real[d], Char[e], Real[f], Char[g],
        Endif:
    Endif:

Endif:

Endfor: Number of Rock/Soil or Scaling-Group Types

Endcard: Hydraulic Properties Card
```

B.9.1 Hydraulic Properties Card Example

```
#NaCl Precipitation Considered
~Hydraulic Properties Card
Aquifer,1.e-13,m^2,,,,,0.8,0.8,
```

B.9.2 Hydraulic Properties Card Example

```
~Hydraulic Properties Card
Sands,3.e-12,m^2,3.e-12,m^2,3.e-12,m^2,
Shale,1.e-14,m^2,1.e-14,m^2,1.e-14,m^2,
```

B.9.3 Hydraulic Properties Card Example

```
~Hydraulic Properties Card
IJK Indexing, file:ksx.dat,hc:cm/s, file:ksy.dat,hc:cm/s,
file:ksz.dat,hc:cm/s,
```

B.9.4 Hydraulic Properties Card Example

```
#R1 is a scaling group
~Hydraulic Properties Card
R1,477.09,hc:cm/day,477.09,hc:cm/day,477.09,hc:cm/day,
```

B.10 Thermal Properties Card

Card Title[a] { ~Thermal [Properties Card] }
Format: Char[a]

If: Operational Mode Option = **{ STOMP-CO2e }**
If: Rock/Soil or Scaling Group Name = { IJK | JKI | KIJ } Indexing

Note: The parameter input can be replaced with an external file using the following formatting for ASCII files:

file: filename

or the following formattings for binary files:

binary file: filename

where; the external file will contain unique parameter values for each node (active or inactive) arranged according to the indexing scheme (i.e., IJK, JKI, or KIJ). Applicable units will be applied to all parameter values in the external file.

Elseif:

For: Number of Rock/Soil Types
Rock/Soil Name[a],
Thermal Conductivity Function Option[b],
{ Constant } { Parallel } { Linear } { Somerton } { Campbell }
{ Jame and Norium } { Cass }

If: Thermal Conductivity Function Option = { Constant }
X-Dir. Thermal Conductivity[c], Units[d] (W/m K),
Y-Dir. Thermal Conductivity[e], Units[f] (W/m K),
Z-Dir. Thermal Conductivity[g], Units[h] (W/m K), Specific Heat[i], Units[j] (J/kg K),
Format: Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j],
Elseif: Thermal Conductivity Function Option = { Parallel }
X-Dir. Rock/Soil Grain Thermal Conductivity[c], Units[d] (W/m K),
Y-Dir. Rock/Soil Grain Thermal Conductivity[e], Units[f] (W/m K),
Z-Dir. Rock/Soil Grain Thermal Conductivity[g], Units[h] (W/m K),
Specific Heat[i], Units[j] (J/kg K),
Format: Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j],
Elseif: Thermal Conductivity Function Option = { Linear }
X-Dir. Rock/Soil Unsaturated Thermal Conductivity[c], Units[d] (W/m K),
Y-Dir. Rock/Soil Unsaturated Thermal Conductivity[e], Units[f] (W/m K),
Z-Dir. Rock/Soil Unsaturated Thermal Conductivity[g], Units[h] (W/m K),
X-Dir. Rock/Soil Water Saturated Thermal Conductivity[i], Units[j] (W/m K),
Y-Dir. Rock/Soil Water Saturated Thermal Conductivity[k], Units[l] (W/m K),
Z-Dir. Rock/Soil Water Saturated Thermal Conductivity[m], Units[n] (W/m K),
Specific Heat[o], Units[p] (J/kg K),
Format: Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h],
Real[i], Char[j], Real[k], Char[l], Real[m], Char[n], Real[o], Char[p],
Elseif: Thermal Conductivity Function Option = { Somerton }

```
    X-Dir. Rock/Soil Unsaturated Thermal Conductivity[c], Units[d] (W/m K),
    Y-Dir. Rock/Soil Unsaturated Thermal Conductivity[e], Units[f] (W/m K),
    Z-Dir. Rock/Soil Unsaturated Thermal Conductivity[g], Units[h] (W/m K),
    X-Dir. Rock/Soil Water Saturated Thermal Conductivity[i], Units[j] (W/m K),
    Y-Dir. Rock/Soil Water Saturated Thermal Conductivity[k], Units[l] (W/m K),
    Z-Dir. Rock/Soil Water Saturated Thermal Conductivity[m], Units[n] (W/m K),
    Specific Heat[o], Units[p] (J/kg K),
    Format: Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h],
        Real[i], Char[j], Real[k], Char[l], Real[m], Char[n], Real[o], Char[p],
Elseif: Thermal Conductivity Function Option = { Campbell }
    Parameter a[c] (0.734), Units[d] (W/m K), Parameter b[e] (1.45), Units[f] (W/m K),
    Parameter c[g] (2.01), Parameter d[h] (0.204), Units[i] (W/m K),
    Parameter e[j] (4.0), Specific Heat[k], Units[l] (J/kg K),
    Format: Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g],
        Real[h], Char[i], Real[j], Real[k], Char[l],
Elseif: Thermal Conductivity Function Option = { Cass }
    Parameter a[c] (0.6), Units[d] (W/m k), Parameter b[e] (0.7), Units[f] (W/m K),
    Parameter c[g] (8.0), Parameter d[h] (0.26), Units[i] (W/m K),
    Parameter e[i] (3.0), Specific Heat[k], Units[l] (J/kg K),
    Format: Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Real[h],
        Char[i], Real[j], Real[k], Char[l],
Elseif: Thermal Conductivity Function Option = { Jame and Norium }
    Specific Heat[c], Units[d] (J/kg K),
    Format: Char[a], Char[b], Real[c], Char[d],
Endif:

If: Thermal Properties Card Extension Option is specified, the following
    parameters have to be added to the end of the input line as follows:

    [ Thermal Properties Card Extension option
        { Enhanced [ Water-Vapor Diffusion ] } | { [ Ground-Surface ] Albedo }[a],
    If: Thermal Properties Card Extension Option = { Enhanced }
        Parameter a[b] (9.5), Units[c], Parameter b[d] (2.0), Units[e],
        Parameter c[f] (8.0), Units[g], Parameter d[h] (0.5), Units[i],
        Parameter e[j] (3.0), Units[k],
        Format: Char[a], Real[b], Char[c], Real[d], Char[e], Real[f], Char[g], Real[h], Char[i],
                Real[j], Char[k],
    ElseIf: Thermal Properties Card Extension Option = { Albedo }
        Ground-Surface Solar Angle Model[b],
                { Plem | Xiu } | { Wang } | { Briegleb } | { Moisture }
        Dry-Soil Albedo[c], Wet-Soil Albedo[d], Albedo Attenuation Factor[e],

        If: Ground-Surface Solar Angle Model = { Plem | Xiu } |
                Ground-Surface Solar Angle Model = { Moisture }
                    Format: Char[a], Char[b], Real[c], Real[d], Real[e],
        ElseIf: Ground-Surface Solar Angle Model = { Wang } |
                Ground-Surface Solar Angle Model = { Briegleb }
                    Reference Albedo @ Solar Zenith = 60 deg[f],
                If: Ground-Surface Solar Angle Model = { Wang }
                    Format: Char[a], Char[b], Real[c], Real[d], Real[e], Real[f],
                ElseIf: Ground-Surface Solar Angle Model = { Briegleb }
                    Parameter C[f],
                        Format:  Char[a], Char[b], Real[c], Real[d], Real[e], Real[f],
                    Endif:
                Endif:
            Endif:
        Endif:
    Endif:
Endif:

Endcard: Thermal Properties Card
```

B.10.1 Thermal Properties Card Example

```
~Thermal Properties Card
Sand,Somerton,0.228,W/m K,0.228,W/m K,0.228,W/m K,0.55,W/m K,0.55,W/m
K,0.55,W/m K,810,J/kg K,
```

B.10.2 Thermal Properties Card Example

```
~Thermal Properties Card
sand,constant,0.5,W/m K,0.5,W/m K,0.5,W/m K,700,J/kg K,
```

B.10.3 Thermal Properties Card Example

```
~Thermal Properties Card
L1,Cass,0.60,W/m K,0.70,W/m K,8.0,0.26,W/m K,3.0,793.1,J/kg
C,enhanced,9.5,2.0,8.0,0.50,3.0,albedo,Wang,0.267,0.160,3.585,0.04,
L2,Cass,0.60,W/m K,0.70,W/m K,8.0,0.26,W/m K,3.0,793.1,J/kg
C,enhanced,9.5,2.0,8.0,0.50,3.0,albedo,Wang,0.402,0.275,3.585,0.04,
L3,Cass,0.60,W/m K,0.70,W/m K,8.0,0.26,W/m K,3.0,730.6,J/kg
C,enhanced,9.5,2.0,8.0,0.50,3.0,albedo,Wang,0.402,0.275,3.585,0.04,
L4,Cass,0.60,W/m K,0.70,W/m K,8.0,0.26,W/m K,3.0,730.6,J/kg
C,enhanced,9.5,2.0,8.0,0.50,3.0,albedo,Wang,0.402,0.275,3.585,0.04,
```

B.11 Saturation Function Card

Card Title[a] { ~Saturation Function [Card] }
Format: Char[a]

If: Rock/Soil Name = { IJK | JKI | KIJ } Indexing

Note: The parameter input can be replaced with an external file using the following formatting for ASCII files:

File: filename

or the following formatting for binary files:

binary file: filename

where; the external file will contain unique parameter values for each node (active or inactive) arranged according to the indexing scheme (i.e., IJK, JKI, or KIJ). Applicable units will be applied to all parameter values in the external file.

Elseif:

For: Number of Rock/Soil or Scaling-Group Types

Rock/Soil Name[a],
Saturation Function[b],
{ [Entrapment | Extended | [Fractured | DP | Dual]] van Genuchten |
[Entrapment | Extended | Fractured] Brooks and Corey |
Haverkamp | Russo | Tabular [Linear | Spline [Log]] [Water Content |
Saturation] }

If: Saturation Function Option = { van Genuchten }
 Parameter[c], Units[d] (1/m), n Parameter[e], Minimum Saturation[f],
m Parameter[g],
If: Operational Mode Option = { **STOMP-CO2** }
Actual Gas Residual Saturation[h],
Format: Char[a], Char[b], Real[c], Char[d], Real[e], Real[f], Real[g], Real[h],
ElseIf: Operational Mode Option = { **STOMP-CO2e** }
If: Saturation Function Option = { [Extended] van Genuchten }
Format: Char[a], Char[b], Real[c], Char[d], Real[e], Real[f], Real[g],
ElseIf: Saturation Function Option = { Entrapment van Genuchten }
Effective Gas Residual Saturation[h],
Format: Char[a], Char[b], Real[c], Char[d], Real[e], Real[f], Real[g], Real[h],
Endif:
Endif:

Elseif: Saturation Function Option = { Fractured van Genuchten }
Matrix α Parameter[c], Units[d] (1/m), Matrix n Parameter[e],
Matrix Minimum Saturation[f],

```
    Fracture $\alpha$ Parameter$^g$, Units$^h$ (1/m), Fracture n Parameter$^i$,
    Fracture Minimum Saturation$^j$,
    Matrix m Parameter$^k$, Fracture m Parameter$^l$,
    Format: Char$^a$, Char$^b$, Real$^c$, Char$^d$, Real$^e$, Real$^f$, Real$^g$, Char$^h$, Real$^i$, Realj, Real$^k$,
    Real$^l$,

Elseif: Saturation Function Option = { Brooks and Corey }
    Entry Head$^c$, Units$^d$ (m), $\lambda$ Parameter$^e$, Minimum Saturation$^f$,
    If: Operational Mode Option = { STOMP-CO2 }
        Actual Gas Residual Saturation$^g$,
        Format:  Char$^a$, Char$^b$, Real$^c$, Char$^d$, Real$^e$, Real$^f$, Real$^g$,
    ElseIf: Operational Mode Option = { STOMP-CO2e }
        If: Saturation Function Option = { [ Extended ] Brooks and Corey }
            Format:  Char$^a$, Char$^b$, Real$^c$, Char$^d$, Real$^e$, Real$^f$,
        ElseIf: Saturation Function Option = { Entrapment Brooks and Corey }
            Effective Gas Residual Saturation$^g$,
            Format:  Char$^a$, Char$^b$, Real$^c$, Char$^d$, Real$^e$, Real$^f$, Real$^g$,
        Endif:
    Endif:

Elseif: Saturation Function Option = { Fractured Brooks and Corey }
    Matrix Entry Head$^c$, Units$^d$ (m), Matrix $\lambda$ Parameter$^e$,
    Matrix Minimum Saturation$^f$,
    Fracture Entry Head$^g$, Units$^h$ (m), Fracture $\lambda$ Parameter$^i$,
    Fracture Minimum Saturation$^j$,
    Format:  Char$^a$, Char$^b$, Real$^c$, Char$^d$, Real$^e$, Real$^f$, Real$^g$, Char$^h$, Real$^i$, Real$^j$

Elseif: Saturation Function Option = { Haverkamp }
    Entry Head Parameter$^c$, Units$^d$ (m), $\alpha$ Parameter$^e$, $\beta$ Parameter$^f$,
    Minimum Saturation$^g$,
    Format: Char$^a$, Char$^b$, Real$^c$, Char$^d$, Real$^e$, Char$^f$, Real$^g$,

Elseif: Saturation Function Option = { Russo }
    $\alpha$ Parameter$^c$, n Parameter$^d$, Minimum Saturation$^e$,
    Format: Char$^a$, Char$^b$, Real$^c$, Real$^d$, Real$^e$,

Elseif: Saturation Function Option = { Tabular [ Saturation ]
    [Linear | Spline [ Log ]]}
    Number of Table Entries$^c$,
    Format: Char$^a$, Char$^b$, Integer$^c$
    For: Number of Table Entries
        Air-water Capillary Head$^a$, Units$^b$ (m), Aqueous Saturation$^c$,
        Format: Real$^a$, Char$^b$, Real$^c$,
        Endfor: Number of Table Entries

Elseif: Saturation Function Option = { Tabular Water Content
    [ Linear | Spline [ Log ] ] }
    Number of Table Entries$^c$,
    Format: Char$^a$, Char$^b$, Integer$^c$
    For: Number of Table Entries
        Air-Water Capillary Head$^a$, Units$^b$ (m), Water Content$^c$,
            Format: Real[a], Char[b], Real[c],
        Endfor: Number of Table Entries
    Endif:

Endif:

Endfor: Number of Rock/Soil Types

Endcard: Saturation Function Card
```

B.11.1 Saturation Function Card Example

```
~Saturation Function Card
Aquifer,van Genuchten,0.5,1/m,1.84162,0.0,0.457,0.0,
```

B.11.2 Saturation Function Card Example

```
~Saturation Function Card
Aquifer,Brooks and Corey Extended,54.0,cm,4.033,0.01,,
```

B.11.3 Saturation Function Card Example

```
~Saturation Function Card
IJK Indexing, Tabular Log-linear,12,file:tabh.dat,cm,file:tabs.dat,
saturation,
```

B.11.4 Saturation Function Card Example

```
#R1 is a scaling group
~Saturation Function Card
R1,van Genuchten,0.036,1/cm,1.756,0.092417,,
```

B.12 Aqueous Relative Permeability Card

Card Title[a] { ~Aqueous Rel [ative Permeability Card] }
Format: Char[a]

If: Rock/Soil or Scaling Group Name = { IJK | JKI | KIJ } Indexing

Note: A parameter value input can be replaced with an external file using the following formatting for ASCII files:

file: filename

or the following formattings for binary files:

binary file: filename

where; the external file will contain unique parameter values for each node (active or inactive) arranged according to the indexing scheme (i.e., IJK, JKI, or KIJ). Applicable units will be applied to all parameter values in the external file. An example input card is included in section 4.3.11.1.

Elseif:

For: Number of Rock/Soil Types or Scaling Groups

If: Anisotropy Option is specified in combination with the Mualem permeability function option, the following two real parameters have to be added to the end of the input line (see example in 4.3.11.1): Horizontal Pore-Scale Parameter (0.5), Horizontal Pore-Scale Parameter (0.5),
Endif:

If: Polmann Anisotropy Option is specified, the following eight real parameters have to be added to the end of the input line
<LnKs> Mean of lnKs with Ks in cm/s,
σLnKs2 Variance of lnKs with Ks in cm/s,
p Slope of the β versus lnKs regression line with Ks in cm/s,
ζ Parameter with Ks in cm/s,
λ Vertical correlation lengths for lnKs with Ks in cm/s,
Mean slope, β, for lnKs versus ψ with Ks in cm/s,
Upper Anisotropy Ratio Limit,
Lower Anisotropy Ratio Limit,
ElseIf: Pruess Anisotropy option is specified the following 3 parameters have to be added to the end of the input line
Gompertz Function a Parameter,
Gompertz Function b Parameter,
Gompertz Function c Parameter,
Endif:

```
Rock/Soil Name,
Permeability Function Option$^{b}$, (Polmann Anisotropy Option may be specified)
    { Constant | Gardner | Mualem [ Irreducible | Modified | Anisotropy ] | Burdine |
      Fatt and Klikoff | Corey | Free Corey | Haverkamp |
      Touma and Vauclin | Stone | Polynomial |
      Tabular [ Linear | Spline ] [ Water Content | Head [ Log ] | Saturation ] }
If: Permeability Function Option = { Constant }
    If: Rock/Soil Name= {{ Fractured }} {{ DP }}
      Matrix Aqueous Relative Permeability$^{c}$,
      Fracture Aqueous Relative Permeability$^{d}$,
      Format: Char$^{a}$, Char$^{b}$, Real$^{c}$, Real$^{d}$,
   Else:
      Aqueous Relative Permeability$^{c}$,
      Format: Char$^{a}$, Char$^{b}$, Real$^{c}$,
   Endif:
Elseif: Permeability Function Option = { Mualem } { Burdine }
    If: Saturation Function Option = {{ van Genuchten }}
      and Rock/SoilName contains {{ Fractured }} {{ DP }}
      Matrix van Genuchten m parameter$^{c}$,
      Fracture van Genuchten m parameter$^{d}$,
      Format: Char$^{a}$, Char$^{b}$, Real$^{c}$, Real$^{d}$,
    Elseif: Saturation Function Option = {{ Brooks and Corey }}
      and Rock/Soil Name contains {{ Fractured }} {{ DP }}
      Matrix Brooks and Corey λ parameter$^{c}$,
      Fracture Brooks and Corey λ parameter$^{d}$,
      Format: Char$^{a}$, Char$^{b}$, Real$^{c}$, Real$^{d}$,
    Elseif: Saturation Function Option = {{ van Genuchten }}
      van Genuchten m parameter$^{c}$,
      Format: Char$^{a}$, Char$^{b}$, Real$^{c}$,
    Elseif: Saturation Function Option = {{ Brooks and Corey }}
      Brooks and Corey λ parameter$^{c}$,
      Format: Char$^{a}$, Char$^{b}$, Real$^{c}$,
    Endif:
Elseif: Permeability Function Option = { Irreducible Mualem }
    If: Saturation Function Option = {{ van Genuchten }}
      van Genuchten m parameter$^{c}$, Irreducible Saturation$^{d}$,
      Format: Char$^{a}$, Char$^{b}$, Real$^{c}$, Real$^{d}$,
    Elseif: Saturation Function Option = {{ Brooks and Corey }}
      Brooks and Corey λ parameter$^{c}$, Irreducible Saturation$^{d}$,
      Format: Char$^{a}$, Char$^{b}$, Real$^{c}$, Real$^{d}$,
    Endif:
Elseif: Permeability Function Option = { Mualem w/ Anisotropy}
    If: Saturation Function Option = {{ van Genuchten }}
      van Genuchten m parameter$^{c}$, Irreducible Saturation$^{d}$,
          Horizontal Pore-Scale Parameter$^{e}$, Vertical Pore-Scale Parameter$^{f}$,
      Format: Char$^{a}$, Char$^{b}$, Real$^{c}$, Real$^{d}$,Real$^{e}$, Real$^{f}$,
    Elseif: Saturation Function Option = {{ Brooks and Corey }}
      Brooks and Corey λ parameter$^{c}$, Irreducible Saturation$^{d}$,
          Horizontal Pore-Scale Parameter$^{e}$, Vertical Pore-Scale Parameter$^{f}$,
      Format: Char$^{a}$, Char$^{b}$, Real$^{c}$, Real$^{d}$, Real$^{e}$, Real$^{f}$,
    Endif:
```

```
Elseif: Permeability Function Option = { Fatt and Klikoff } { Corey }
    Format: Char[a], Char[b],
Elseif: Permeability Function Option = { Haverkamp }
    A Parameter[c], Units[d] (m), γ Parameter[e], Effective Air Entry Head[f], Units[g] (m),
    Format: Char[a], Char[b], Real[c], Char[d],Real[e], Real[f], Char[g],
Elseif: Permeability Function Option = { Touma and Vauclin }
    α Parameter[c], β Parameter[d],
    Format: Char[a], Char[b], Real[c], Real[d],
Elseif: Permeability Function Option = { Free Corey }
    Endpoint Aqueous Relative Permeability[c], Exponent Aqueous Relative Permeability[d],
    Residual Aqueous Saturation[e], Residual Gas Saturation[f],
    Format: Char[a], Char[b], Real[c], Real[d], Real[e], Real[f],
Elseif: Permeability Function Option = { Tabular Water Content | [ Linear | Spline ] }
    Number of Table Entries[c],
    Format: Char[a], Char[b], Integer[c],
    For: Number of Table Entries
        Aqueous Moisture Content[a], Aqueous Relative Permeability[b],
        Format: Real[a], Real[b],
    Endfor: Number of Table Entries
ElseIf: Permeability Function Option = { Tabular Head [ Log ] }
        Number of Table Entries[c],
        Format: Char[a], Char[b], Integer[c],
        For: Number of Table Entries
            If: Permeabilty Function Option = {{ Tabluar Head Log }}
                Log Capillary Head[a], Units[b], Aqueous Relative Permeability[c],
                Format: Real[a], Char[b], Real[c],
            Else:
                Capillary Head[a], Units[b], Aqueous Relative Permeability[c],
                Format: Real[a], Char[b], Real[c],
            Endif:
        EndFor:
    Elseif: Permeability Function Option = { Tabular [ Saturation ] [ Linear | Spline ] }
    Number of Table Entries[c],
    Format: Char[a], Char[b], Integer[c],
    For: Number of Table Entries
        Aqueous Saturation[a], Aqueous Relative Permeability[b],
        Format: Real[a], Real[b],
    Endfor: Number of Table Entries
Endif:

Endfor: Number of Rock/Soil Types

Endcard: Aqueous Relative Permeability Card
```

B.12.1 Aqueous Relative Permeability Card Example

```
~Aqueous Relative Permeability Card
Aquifer,Mualem Irreducible,0.457,0.30,
```

B.12.2 Aqueous Relative Permeability Card Example

```
~Aqueous Relative Permeability Card
IJK Indexing,Mualem,file:data_m.dat,
```

B.12.3 Aqueous Relative Permeability Card Example

```
~Aqueous Relative Permeability Card
SM-ML1,Burdine,,
SW1,Burdine,,
SP3,Burdine,,
SM-SP1,Burdine,,
SP2,Burdine,,
SP1,Burdine,,
US,Touma and Vauclin,1.0,2.0,
```

B.12.4 Aqueous Relative Permeability Card Example

```
~Aqueous Relative Permeability Card
Soil,Tabular,33,
3.5026E-09,3.1593E-07,
0.01225919,6.7082E-07,
0.02977233,1.7099E-06,
0.04728546,3.8458E-06,
0.0647986,7.8611E-06,
0.08231173,1.4901E-05,
0.117338,4.5067E-05,
0.15236427,0.00011488,
0.18739054,0.00025837,
0.22241681,0.00052812,
0.25744308,0.00100111,
0.29246935,0.0017854,
0.32749562,0.00302769,
0.36252189,0.00492173,
0.39754816,0.00771754,
0.43257443,0.01173155,
0.4676007,0.01735761,
0.50262697,0.02507879,
0.53765324,0.03548017,
0.57267951,0.04926245,
0.60770578,0.06725641,
0.64273205,0.09043837,
0.67775832,0.11994642,
0.71278459,0.15709761,
0.74781086,0.20340604,
0.78283713,0.26060184,
0.8178634,0.330651,
0.85288967,0.41577624,
0.88791594,0.51847864,
0.92294221,0.64156028,
0.95796848,0.78814774,
0.99299475,0.96171657,
1,1,
```

B.13 Gas Relative Permeability Card

Card Title[a] { ~Gas Rel [ative Permeability Card] }
Format: Char[a]

```
If: Rock/Soil Name = { IJK | JKI | KIJ } Indexing
    Note: A parameter value input can be replaced with an external file using the
    following formatting for ASCII files:

        file: filename

    or the following formattings for binary files:

        binary file: filename

    where; the external file will contain unique parameter values for each node
    (active or inactive) arranged according to the indexing scheme (i.e., IJK, JKI, or KIJ).
    Applicable units will be applied to all parameter values in the external file. An example
    input card is included in section 4.3.12.1.

Elseif:

For: Number of Rock/Soil Types

Rock/Soil Name[a],
Permeability Function Option[b],
    { Constant | Mualem | Burdine | Fatt and Klikoff | Stone | Sandia | Corey |
    Free Corey |  Exponential | Tabular [ Linear | Spline ] [ Water Content | Saturation ]    }
If: Permeability Function Option = { Constant }
    If: Rock/Soil Name contains {{ Fractured }} {{ DP }}
        Matrix Gas Relative Permeability[c],
        Fracture Gas Relative Permeability[d],
        Format: Char[a], Char[b], Real[c], Real[d],
        If: Klinkenberg Gas Relative Permeability Extension is Considered
            Permeability Extension[e], { Klinkenberg }
            Scaling Parameter (C1)[f], Exponential Parameter[g], Pressure Units[h],
            Format: Char[a], Char[b], Real[c], Real[d], Char[e], Real[f], Real[g], Char[h],
        Endif:
    Else:
        Gas Relative Permeability[c]
        Format: Char[a], Char[b], Real[c],
    Endif:
Elseif: Permeability Function Option = { Mualem } { Burdine }
    If: Saturation Function Option = {{ van Genuchten }}
        and Rock/Soil Name contains {{ Fractured }} {{ DP }}
        Matrix van Genuchten m parameter[c],
        Fracture van Genuchten m parameter[d],
        Format: Char[a], Char[b], Real[c], Real[d],
        If: Klinkenberg Gas Relative Permeability Extension is Considered
            Permeability Extension[e], { Klinkenberg }
```

```
      Scaling Parameter (C1)^f, Exponential Parameter^g, Pressure Units^h,
      Format: Char^a, Char^b, Real^c, Real^d, Char^e, Real^f, Real^g, Char^h,
    Endif:
  Elseif: Saturation Function Option = {{ Brooks and Corey }}
    and Rock/Soil Name contains {{ Fractured }} {{ DP }}
    Matrix Brooks and Corey λ parameter^c,
    Fracture Brooks and Corey λ parameter^d,
    Format: Char^a, Char^b, Real^c, Real^d,
    If: Klinkenberg Gas Relative Permeability Extension is Considered
      Permeability Extension^e, { Klinkenberg }
      Scaling Parameter (C1)^f, Exponential Parameter^g, Pressure Units^h,
      Format: Char^a, Char^b, Real^c, Real^d, Char^e, Real^f, Real^g, Char^h,
    Endif:
  Elseif: Saturation Function Option = {{ van Genuchten }}
    van Genuchten m parameter^c,
    Format: Char^a, Char^b, Real^c,
    If: Klinkenberg Gas Relative Permeability Extension is Considered
      Permeability Extension^d, { Klinkenberg }
      Scaling Parameter (C1)^e, Exponential Parameter^f, Pressure Units^g,
      Format: Char^a, Char^b, Real^c, Char^d, Real^e, Real^f, Char^g,
    Endif:
  Elseif: Saturation Function Option = {{ Brooks and Corey }}
    Brooks and Corey λ parameter^c,
    Format: Char^a, Char^b, Real^c,
    If: Klinkenberg Gas Relative Permeability Extension is Considered
      Permeability Extension^d, { Klinkenberg }
      Scaling Parameter (C1)^e, Exponential Parameter^f, Pressure Units^g,
      Format: Char^a, Char^b, Real^c, Char^d, Real^e, Real^f, Char^g,
    Endif:
  Endif:
Elseif: Permeability Function Option = { Corey }
  Irreducible Gas Saturation^c, Irreducible Aqueous Saturation^d,
  Format: Char^a, Char^b, Real^c, Real^d,
  If: Klinkenberg Gas Relative Permeability Extension is Considered
    Permeability Extension^e, { Klinkenberg }
    Scaling Parameter (C1)^f, Exponential Parameter^g, Pressure Units^h,
    Format: Char^a, Char^b, Real^c, Real^d, Char^e, Real^f, Real^g, Char^h,
  Endif:
Elseif: Permeability Function Option = { Free Corey }
  Endpoint Gas Permeability^c, Exponent Gas Relative Permeability^d,
  Residual Aqueous Saturation^e, Residual Gas Saturation^f,
  Format: Char^a, Char^b, Real^c, Real^d, Real^e, Real^f,
  If: Klinkenberg Gas Relative Permeability Extension is Considered
    Permeability Extension^g, { Klinkenberg }
    Scaling Parameter (C1)^h, Exponential Parameter^i, Pressure Units^j,
  Format: Char^a, Char^b, Real^c, Real^d, Real^e, Real^f, Char^g, Real^h, Real^i, Char^j,
Elseif: Permeability Function Option = { Fatt and Klikoff }
  Format: Char^a, Char^b,
  If: Klinkenberg Gas Relative Permeability Extension is Considered
    Permeability Extension^c, { Klinkenberg }
    Scaling Parameter (C1)^d, Exponential Parameter^e, Pressure Units^f,
    Format: Char^a, Char^b, Char^c, Real^d, Real^e, Char^f,
    Endif:
Elseif: Permeability Function Option = { Sandia }
    Format: Char$^a$, Char$^b$,
    If: Klinkenberg Gas Relative Permeability Extension is Considered
        Permeability Extension$^c$, { Klinkenberg }
        Scaling Parameter (C1)$^d$, Exponential Parameter$^e$, Pressure Units$^f$,
        Format: Char$^a$, Char$^b$, Char$^c$, Real$^d$, Real$^e$, Char$^f$,
    Endif:
Elseif: Permeability Function Option = { Stone }
    Stone (Slr)$^c$, Stone (Sgr)$^d$, n Parameter$^e$,
    Format: Char$^a$, Char$^b$, Real$^c$, Real$^d$, Real$^e$,
    If: Klinkenberg Gas Relative Permeability Extension is Considered
        Permeability Extension$^f$, { Klinkenberg }
        Scaling Parameter (C1)$^g$, Exponential Parameter$^h$, Pressure Units$^i$,
        Format: Char$^a$, Char$^b$, Real$^c$, Real$^d$, Real$^e$, Char$^f$, Real$^g$, Real$^h$, Char$^i$,
    Endif:
Elseif: Permeability Function Option = { Exponential }
    Gas Relative Permeability Function Exponent$^c$,
    Format: Char$^a$, Char$^b$, Real$^c$,
    If: Klinkenberg Gas Relative Permeability Extension is Considered
        Permeability Extension$^d$, { Klinkenberg }
        Scaling Parameter (C1)$^e$, Exponential Parameter$^f$, Pressure Units$^g$,
        Format: Char$^a$, Char$^b$, Real$^c$, Char$^d$, Real$^e$, Real$^f$, Char$^g$,
    Endif:
Elseif: Permeability Function Option = { Tabular Water Content [ Linear | Spline ] }
    Number of Table Entries$^c$,
    Format: Char$^a$,Char$^b$,Integer$^c$,
    For: Number of Table Entries
        Water Content$^a$, Gas Relative Permeability$^b$,
        Format: Real$^a$, Real$^b$,
    Endfor:
Elseif: Permeability Function Option = { Tabular [ Saturation ] [ Linear | Spline ] }
    Number of Table Entries$^c$,
    Format: Char$^a$,Char$^b$,Integer$^c$,
    For: Number of Table Entries
        Saturation$^a$, Gas Relative Permeability$^b$,
        Format: Real$^a$, Real$^b$,
    Endfor:
Endif:

    Endfor: Number of Rock/Soil Types

Endif:

Endcard: Gas Relative Permeability Card
```

B.13.1 Gas Relative Permeability Card Example

```
~Gas Relative Permeability Card
Aquifer,Mualem,,
```

B.13.2 Gas Relative Permeability Card Example

```
~Gas Relative Permeability Card
Aquifer,Corey,0.05,0.30,
```

B.13.3 Gas Relative Permeability Card Example

```
~Gas Relative Permeability Card
Sands,van Genuchten,0.4,0.05,
Shale,van Genuchten,0.4,0.05,
```

B.13.4 Gas Relative Permeability Card Example

```
~Gas Relative Permeability Card
aquifer,tabular,2,
0.0,0.0,
1.0,1.0,
leaky well,tabular,2,
0.0,0.0,
1.0,1.0,
```

B.13.5 Gas Relative Permeability Card Example

```
~Gas Relative Permeability Card
IJK Indexing,Constant,file:rel_g_x.dat,
```

B.13.6 Gas Relative Permeability Card Example

```
~Gas Relative Permeability Card
Soil,Tabular,33,
1,1,
0.98774081,0.99999139,
0.97022767,0.99994918,
0.95271454,0.99987151,
0.9352014,0.99975797,
0.91768827,0.99960795,
0.882662,0.99919528,
0.84763573,0.99862545,
0.81260946,0.99788751,
0.77758319,0.99696733,
0.74255692,0.99584706,
0.70753065,0.99450437,
0.67250438,0.99291148,
0.63747811,0.99103385,
0.60245184,0.98882835,
0.56742557,0.98624081,
0.5323993,0.98320258,
0.49737303,0.97962579,
0.46234676,0.9753964,
0.42732049,0.97036419,
0.39229422,0.96432763,
0.35726795,0.95701052,
0.32224168,0.94802483,
0.28721541,0.93680881,
0.25218914,0.92251973,
0.21716287,0.90383668,
0.1821366,0.87857318,
0.14711033,0.84284783,
0.11208406,0.78910152,
0.07705779,0.70059708,
0.04203152,0.53266493,
0.00700525,0.13244732,
0,0,
```

B.14 Solute/Fluid Interaction Card

Card Title^{a} { ~Solute/Fluid [Interactions Card] }
Format: Char^{a}

Number of Solutes^{a}
Format: Integer^{a},

For: Number of Solutes

```
    Solute Name$^{a}$,
    Aqueous-Phase Molecular Diffusion Coefficient @ 20 C$^{b}$, Units$^{c}$ (m^2/s),
    Gas-Phase Molecular Diffusion Coefficient @ 20 C$^{d}$, Units$^{e}$ (m^2/s),
    Gas-Aqueous Partition Function Option$^{f}$

    { Constant | Note:
```

$$\boxed{K_{gl} = \bar{K}_{gl}}$$

```
    Temperature Dependent }   Note:
```

$$\boxed{\ln\left(K_{gl}\right) = a + \frac{b}{T\left(C\right)} + c \ln T\left(C\right) + d\, T\left(C\right) + e\left[T\left(C\right)\right]^{2}}$$

```
    If: Gas-Aqueous Partition Function Option = { Constant }
      Gas-Aqueous Partition Coefficient$^{g}$, Units$^{h}$ (m^3/m^3),
      IfDef: Radioactive
          Half-Life$^{i}$, Units$^{j}$ (s),
          Format: Char$^{a}$, Real$^{b}$, Char$^{c}$, Real$^{d}$, Char$^{e}$, Char$^{f}$, Real$^{g}$, Char$^{h}$, Real$^{i}$, Char$^{j}$,
      ElseifDef: Reactive
          Number of Parent Reactions$^{i}$,
          For: Number of Reactions
              First-Order Reaction Rate Constant$^{j}$, Units$^{k}$ (s),
          Endfor: Number of Reactions
          Format: Char$^{a}$, Real$^{b}$, Char$^{c}$, Real$^{d}$, Char$^{e}$, Char$^{f}$, Real$^{g}$, Char$^{h}$, Integer$^{i}$,
              <Real$^{j}$, Char$^{k}$,>
      EndifDef:

    Elseif: Gas-Aqueous Partition Function Option = { Temperature Dependent }
      Constant a$^{g}$, Constant b$^{h}$, Constant c$^{i}$, Constant d$^{j}$, Constant e$^{k}$,
      IfDef: Radioactive
          Half-Life$^{l}$, Units$^{m}$ (s),
          Format: Char$^{a}$, Real$^{b}$, Char$^{c}$, Real$^{d}$, Char$^{e}$, Char$^{f}$, Real$^{g}$, Real$^{h}$, Real$^{i}$, Real$^{j}$, Real$^{k}$
              Real$^{l}$, Char$^{m}$,
      ElseifDef: Reactive
          Number of Parent Reactions$^{l}$,
          For: Number of Reactions
              First-Order Reaction Rate Constant$^{m}$, Units$^{n}$ (s),
          Endfor: Number of Reactions
          Format: Char$^{a}$, Real$^{b}$, Char$^{c}$, Real$^{d}$, Char$^{e}$, Char$^{f}$, Real$^{g}$, Real$^{h}$, Real$^{i}$, Real$^{j}$, Real$^{k}$,
                Integer[l],<Real[m], Char[n],>
        Endif:

Endfor: Number of Solutes

IfDef: Radioactive
    Number of Chain Decay Lines[a],
    Format: Integer[a],
    For: Number of Chain Decay Lines
        Parent Solute Name[a], Progeny Solute Name[b], Chain Decay Fraction[c],
        Format: Char[a], Char[b], Real[c],
    Endfor: Number of Chain Decay Lines
ElseifDef: Reactive
    Number of Reaction Lines[a],
    Format: Integer[a],
    For: Number of Reaction Lines
        Parent Solute Name[a], Progeny Solute Name[b], Reaction Number[c],
        Reaction Stoichiometry Fraction[d],
        Format: Char[a], Char[b], Real[c], Real[d],
    Endfor: Number of Reaction Lines
EndifDef

Endcard: Solute/Fluid Interactions Card
```

B.14.1 Solute/Fluid Interaction Card Example

```
~Solute/Fluid Interaction Card
1,
Tracer,2.5e-5,cm^2/s,2.5e-5,cm^2/s,Constant,0.5,cm^3/cm^3,
```

B.15 Solute/Porous Media Interaction Card

```
Card Title[a] { ~Solute/Porous [ Media Interactions Card ] }
Format: Char[a]

Rock/Soil Name[a],

If: Rock/Soil Name = { IJK | JKI | KIJ } Indexing
    Note: A parameter value input can be replaced with an external file using the
    following formatting for ASCII files:

        file: filename

    or the following formattings for binary files:

        binary file: filename

    where; the external file will contain unique parameter values for each node
    (active or inactive) arranged according to the indexing scheme (i.e., IJK, JKI, or KIJ).
    Applicable units will be applied to all parameter values in the external file. An example
    input card is included in section 4.3.14.1.

Elseif:

    For: Number of Rock/Soil Types

        Longitudinal Dispersivity[b], Units[c] (m),
        Transverse Dispersivity[d], Units[e] (m)
        Format: Char[a], Real[b], Char[c], Real[d], Char[e],

        For: Number of Solutes
            Solute Name[a], Solid-Aqueous Adsorption Function[b], { Linear Kd | Linear
        Isotherm | Freundlich Isotherm | Langmuir Isotherm }
            If: Solid-Aqueous Adsorption Function = { Linear Kd}
                Kd Parameter[c], Units[d] (m^3/kg),
                Format: Char[a], Char[b], Real[c], Char[d],
            ElseIf: Solid-Aqueous Adsorption Function = { Linear Isotherm}
                K Parameter[c],
                Format: Char[a], Char[b], Real[c],
            ElseIf: Solid-Aqueous Adsorption Function = { Freundlich Isotherm}
                K Parameter[c], n Parameter[d],
                Format: Char[a], Char[b], Real[c], Real[d],
            ElseIf: Solid-Aqueous Adsorption Function = { Langmuir Isotherm}
                a Parameter[c], b Parameter[d], Units[e] (m^3),
                Format: Char[a], Char[b], Real[c], Real[d], Char[e],
            Endif:

    Endfor: Number of Solutes

    Endfor: Number of Rock/Soil Types
```

Endif:

Endcard: Solute/Porous Media Interactions Card

B.15.1 Solute/Porous Media Interactions Card Example

```
~Solute/Porous Media Interactions Card
Aquifer,1,m,0.1,m,
Tracer,0.0,m^3/kg,
```

B.15.2 Solute/Porous Media Interactions Card Example

```
~Solute/Porous Media Interactions Card
Sands,2,m,0.2,m,
Tracer,0.0,m^3/kg,
Shale,1,m,0.1,m,
Tracer,0.0,m^3/kg,
```

B.15.3 Solute/Porous Media Interactions Card Example

```
~Solute/Porous Media Interactions Card
Sands,2,m,0.2,m,
solute,0.01,ml/g,
Shale,1,m,0.1,m,
solute,0.01,ml/g,
```

B.16 Salt Transport Card

Card Title[a] { ~Salt Transport [Card] }
Format: Char[a]

Rock/Soil Name[a],

If: Rock/Soil Name = { IJK | JKI | KIJ } Indexing
Note: A parameter value input can be replaced with an external file using the following formatting for ASCII files:

file: filename

or the following formattings for binary files:

binary file: filename

where; the external file will contain unique parameter values for each node (active or inactive) arranged according to the indexing scheme (i.e., IJK, JKI, or KIJ). Applicable units will be applied to all parameter values in the external file. An example input card is included in section 4.3.15.1.

Elseif:

For: Number of Rock/Soil Types
Longitudinal Dispersivity[b], Units[c] (m),
Transverse Dispersivity[d], Units[e] (m)
Format: Char[a], Real[b], Char[c], Real[d], Char[e],
Endfor: Number of Rock/Soil Types

Endif:

Endcard: Salt Transport Card

B.16.1 Salt Transport Card Examples

```
~Salt Transport Card
Eau Claire Carbonate,20.0,ft,5.0,ft,
Eau Claire Shale,20.0,ft,5.0,ft,
Lower Eau Claire,20.0,ft,5.0,ft,
Upper Mt. Simon,20.0,ft,5.0,ft,
Middle Mt. Simon,20.0,ft,5.0,ft,
Lower Mt. Simon,20.0,ft,5.0,ft,
```

B.16.2 Salt Transport Card Examples

```
~Salt Transport Card
Aquifer,0.0,m,0.0,m,
```

B.16.3 Salt Transport Card Examples

```
~Salt Transport Card
IJK Indexing,binary file:ldisp.bin,m,binary file:tdisp.bin,m,
```

B.17 Coupled Well Card

```
Card Title[a] { ~Coupled Well Card }
Format: Char[a]

Number of Coupled Wells[a]
Format: Integer[a]

For: Number of Coupled Wells
    Coupled Well Type[a],
    { CO2 Injection Well | Aqueous Injection Well |
    [ Withdrawl | Production ] Volumetric Well |
    [ Withdrawl | Production ] Mass Well }
    If: Coupled Well Type = { CO2 Injection Well }
        Water-Vapor Option[b],
        { Rel[ative Humidity] | Mass Frac[tion] | null }
        Format: Char[a],Char[b],
    Elseif: Coupled Well Type = { Aqueous Injection Well }
        Dissolved CO2 Option[b],
        { Rel[ative Saturation] | Mass Frac[tion] | null }
        Dissolved Salt Option[c],
        { Rel[ative Saturation] | Mass Frac[tion] | null }
        Format: Char[a], Char[b], Char[c],
    Else:
        Format: Char[a],
    Endif:
    X-direction Well Fraction Factor[a],
    Y-direction Well Fraction Factor[b],
    Z-direction Well Fraction Factor[c],
    If: Coupled Well Type = { CO2 Injection Well | Aqueous Injection Well }
        Total Mass Injection Limit[d], Units[e],
    Elseif: Coupled Well Type = { Production Volumetric Well }
        Total Volume Produced Limit[d],  Units[e],
    Elseif: Coupled Well Type = { Production Mass Well }
        Total Mass Produced Limit[d], Units[e],
    Endif:
    Format: Real[a], Real[b], Real[c], Real[d], Char[e],
Endfor:

Number of well intervals[a],
Format: Integer[a],

For: Number of well intervals
    1st X-Transition Point[a], Units[b],
    1st Y-Transition Point[c], Units[d],
    1st Z-Transition Point[e],  Units[f],
    2nd X-Transition Point[g], Units[h],
    2nd Y-Transition Point[i], Units[j],
    2nd Z-Transition Point[k], Units[l],
    Well Bore Radius[m], Units[n],
    Interval Skin Factor[o],
    Interval Screen Option[p], { Screened | null }
    Format: Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j], Real[k], Char[l],   Real[m], Char[n],
    Real[o], Char[p],
Endfor:

Number of Well Time Points[a],
Format: Integer[a],

For: Number of Well Time Points
     Well Time[a], Units[b],
    If: Coupled Well Type = { CO2 Injection Well | Aqueous Injection Well }
        Injection Mass Rate[c], Units[d],
        Maximum Well-Top Pressure[e], Units[f],
    Elseif: Coupled Well Type = { Production Volumetric Well }
        Production Volume Flow Rate[c], Units[d],
        Minimum Well-Bottom Pressure[e], Units[f],
    Elseif: Coupled Well Type = { Production Mass Well }
        Production Mass Flow Rate[c], Units[d],
        Minimum Well-Bottom Pressure[e], Units[f],
    Endif:
    Format: Real[a], Char[b],Real[c], Char[d], Real[e], Char[f],
    If: Coupled Well Type = { CO2 Injection Well }
        If: Water-Vapor Option = { Rel[ative Humidity]
            Water Relative Humidity[a],
        Elseif: Water-Vapor Option = { Mass Frac[tion] }
            Water Mass Fraction[a],
        Endif:
        If: Operational Mode Option = { STOMP-CO2e }
            Injection Temperature[b],
        Endif:
        Format: Real[a], Real[b],
     Elseif: Coupled Well Type = { Aqueous Injection Well }
        If: Dissolved CO2 Option = { Rel[ative Saturation] }
            Dissolved CO2 Relative Saturation[a],
        Elseif: Dissolved CO2 Option = { Mass Frac[tion] }
            Dissolved CO2 Mass Fraction[a],
        Endif:
        If: Operational Mode Option = { STOMP-CO2e }
            Injection Temperature[b],
        Endif:
        If: Dissolved Salt Option = { Rel[ative Saturation] }
            Dissolved Salt Relative Saturation[c],
        Elseif: Dissolved Salt Option = { Mass Frac[tion] }
            Dissolved Salt Mass Fraction[c],
        Endif:
        Format: Real[a], Real[b], Real[c],
    Endif:
Endfor:

Endcard: Coupled Well Card
```

B.17.1 Coupled Well Card Example

```
~Coupled Well Card
1,
CO2 Injection Well,Water Relative Saturation,0.25,0.25,1.0,50,MMT,
1,
0.5,m,1.25,m,-2195.0,m,0.5,m,1.25,m,-2145,m,4.8125,in,0.0,screened,
2,
0,yr,0.25,MMT/yr,4739,psi,0.0,
50,yr,0.25,MMT/yr,4739,psi,0.0,
```

B.17.2 Coupled Well Card Example

```
~Coupled Well Card
1,
CO2 Injection Well,Water Relative Saturation,1.0,0.5,1.0,0.383184,MMT,
1,
-100.0,m,0.075,m,30.0,m,-100.0,m,0.075,m,0.0,m,0.15,m,0.0,screened,
1,
0.0,hr,4.435,kg/s,45,MPa,0.0,
```

B.18 Initial Conditions Card

Card Title[a] { ~Initial [Conditions Card] }
Format: Char[a]

If: Initial Saturation Option[a], Initial Saturation Option[b],
{ Gas Pressure, Aqueous Pressure |
Gas Pressure, Aqueous Saturation |
Aqueous Pressure, Aqueous Saturation }
Format: Char[a], Char[b],

Number of Initial Conditions Domains[a],
Format: Integer[a],

For: Number of Initial Conditions Domains

Note: The [Overwrite] option is used in conjunction with Restart simulations.

Variable Name Option[a],
{ File [Binary | ASCII] | Rock | Zonation |
Temperature [Overwrite] | Aqueous Pressure [Overwrite] |
Gas Pressure [Overwrite] |
Aqueous Saturation [Overwrite] | [Relative] Trapped Gas Saturation [Overwrite] |
CO2 Partial Pressure [Overwrite] |
Dissolved CO2 Relative Saturation [Overwrite] |
Dissolved CO2 Mass Fraction [Overwrite] |
Dissolved CO2 Aqueous Conc. [Overwrite] |
Salt Mass Fraction [Overwrite] |
Salt Relative Saturation [Overwrite] |
Salt Aqueous Conc. [Overwrite] |
Salt Volumetric Conc. [Overwrite] |
Solute [Volumetric Conc.] [Overwrite], Solute Name |
Solute Aqueous [Volumetric Conc.] [Overwrite], Solute Name |
Solute Gas [Volumetric Conc.] [Overwrite], Solute Name
Species [Overwrite]}

If: Variable Name Option = {{ Pressure }}
Pressure[b], Units[c] (Pa),
If: Variable Name Option = { File [Binary | ASCII] }
File Name[d], File Units[e] (Pa),
Note: File contains pressure values for every node.
Format: Char[a], Real[b], Char[c], Char[d], Char[e],
Elseif: Variable Name Option = { Zonation | Rock }
Rock/Soil or Scaling Group Name[d],
Format: Char[a], Real[b], Char[c], Char[d],
Else:
X-Dir. Gradient[d], Units[e] (1/m),
Y-Dir. Gradient[f], Units[g] (1/m),
Z-Dir. Gradient[h], Units[i] (1/m),
I-Start Index[j], I-End Index[k],

```
    J-Start Index[l], J-End Index[m],
    K-Start Index[n], K-End Index[o],
    Format: Char[a], Real[b], Char[c], Real[d], Char[e], Real[f], Char[g], Real[h],
        Char[i], Integer[j], Integer[k], Integer[l], Integer[m], Integer[n], Integer[o],
    Endif:

Elseif: Variable Name Option = {{ Temperature }}
    Temperature[b], Units[c] (C),
    If: Variable Name Option = { File [ Binary | ASCII ] }
        File Name[d], File Units[e] (C),
        Note: File contains temperature values for every node.
        Format: Char[a], Real[b], Char[c], Char[d], Char[e],
    Elseif: Variable Name Option = { Zonation | Rock }
        Rock/Soil or Scaling Group Name[d],
        Format: Char[a], Real[b], Char[c], Char[d],
    Else:
        X-Dir. Gradient[d], Units[e] (1/m),
        Y-Dir. Gradient[f], Units[g] (1/m),
        Z-Dir. Gradient[h], Units[i] (1/m),
        I-Start Index[j], I-End Index[k],
        J-Start Index[l], J-End Index[m],
        K-Start Index[n], K-End Index[o],
        Format: Char[a], Real[b], Char[c], Real[d], Char[e], Real[f], Char[g], Real[h],
                Char[i], Integer[j], Integer[k], Integer[l], Integer[m], Integer[n], Integer[o],
    Endif:

Elseif: Variable Name Option = {{ Saturation }}
    Saturation[b], Null[c],
    If: Variable Name Option = { File [ Binary | ASCII ] }
        Note: File contains saturation values for every node.
        Filename[d],
        Format: Char[a], Real[b], Char[c], Char[d],
    Elseif: Variable Name Option = { Zonation | Rock }
        Rock/Soil or Scaling Group Name[d],
        Format: Char[a], Real[b], Char[c], Char[d],
    Else:
        X-Dir. Gradient[d], Units[e] (1/m),
        Y-Dir. Gradient[f], Units[g] (1/m),
        Z-Dir. Gradient[h], Units[i] (1/m),
        I-Start Index[j], I-End Index[k],
        J-Start Index[l], J-End Index[m],
        K-Start Index[n], K-End Index[o],
        Format: Char[a], Real[b], Char[c], Real[d], Char[e], Real[f], Char[g], Real[h],
                Char[i], Integer[j], Integer[k], Integer[l], Integer[m], Integer[n], Integer[o],
    Endif:
Elseif: Variable Name Option = {{ Fraction }}
    Mass Fraction[b], Null[c],
    If: Variable Name Option = { File [ Binary | ASCII ] }
        File Name[d], Null[e],
        Note: File contains fraction values for every node.
        Format: Char[a], Real[b], Char[c], Char[d], Char[e],
    Elseif: Variable Name Option = { Zonation | Rock }
```

```
        Rock/Soil or Scaling Group Name[d],
        Format: Char[a], Real[b], Char[c], Char[d],
    Else:
        X-Dir. Gradient[d], Units[e] (1/m),
        Y-Dir. Gradient[f], Units[g] (1/m),
        Z-Dir. Gradient[h], Units[i] (1/m),
        I-Start Index[j], I-End Index[k],
        J-Start Index[l], J-End Index[m],
        K-Start Index[n], K-End Index[o],
        Format: Char[a], Real[b], Char[c], Real[d], Char[e], Real[f], Char[g], Real[h],
                Char[i], Integer[j], Integer[k], Integer[l], Integer[m], Integer[n], Integer[o],
    Endif:

Elseif: Variable Name Option = {{ Salt Aqueous Conc. }}
    Aqueous Conc[b], Units[c] (kg/m^3),
    If: Variable Name Option = { File [ Binary ] }
        File Name[d], Units[e] (kg/m^3),
        Note: File contains salt Conc. values for every node.
        Format: Char[a], Real[b], Char[c], Char[d], Char[e],
    Elseif: Variable Name Option = { Zonation | Rock }
        Rock/Soil or Scaling Group Name[d],
        Format: Char[a], Real[b], Char[c], Char[d],
    Else:
        X-Dir. Gradient[d], Units[e] (1/m),
        Y-Dir. Gradient[f], Units[g] (1/m),
        Z-Dir. Gradient[h], Units[i] (1/m),
        I-Start Index[j], I-End Index[k],
        J-Start Index[l], J-End Index[m],
        K-Start Index[n], K-End Index[o],
        Format: Char[a], Real[b], Char[c], Real[d], Char[e], Real[f], Char[g], Real[h],
                Char[i], Integer[j], Integer[k], Integer[l], Integer[m], Integer[n], Integer[o],
    Endif:

Elseif: Variable Name Option = {{ Salt Volumetric Conc. }}
    Volumetric Conc[b], Units[c] (kg/m^3),
    If: Variable Name Option = { File [ Binary | ASCII ] }
        File Name[d], Units[e] (kg/m^3),
        Note: File contains salt Conc. values for every node.
        Format: Char[a], Real[b], Char[c], Char[d], Char[e],
    Elseif: Variable Name Option = { Zonation | Rock }
        Rock/Soil or Scaling Group Name[d],
        Format: Char[a], Real[b], Char[c], Char[d],
    Else:
        X-Dir. Gradient[d], Units[e] (1/m),
        Y-Dir. Gradient[f], Units[g] (1/m),
        Z-Dir. Gradient[h], Units[i] (1/m),
        I-Start Index[j], I-End Index[k],
        J-Start Index[l], J-End Index[m],
        K-Start Index[n], K-End Index[o],
        Format: Char[a], Real[b], Char[c], Real[d], Char[e], Real[f], Char[g], Real[h],
                Char[i], Integer[j], Integer[k], Integer[l], Integer[m], Integer[n], Integer[o],
    Endif:
        Elseif: Variable Name Option = {{ Solute }}
            Solute Name[b], Volumetric Conc[c], Units[d] (kg/m^3),
            If: Variable Name Option = { File [ Binary | ASCII ] }
                File Name[e], Units[f] (kg/m^3),
                Note: File contains solute Conc. values for every node.
                Format: Char[a], Char[b], Real[c], Char[d], Char[e], Char[f],
            Elseif: Variable Name Option = { Zonation | Rock }
                Rock/Soil or Scaling Group Name[e],
                Format: Char[a], Char[b], Real[c], Char[d], Char[e],
            Else:
                X-Dir. Gradient[e], Units[f] (1/m), Y-Dir. Gradient[g], Units[h] (1/m),
                Z-Dir. Gradient[i], Units[j] (1/m),
                I-Start Index[k], I-End Index[l],
                J-Start Index[m], J-End Index[n],
                K-Start Index[o], K-End Index[p],
                Format: Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i],
                          Char[j], Integer[k], Integer[l], Integer[m], Integer[n], Integer[o], Integer[p],
            Endif:

        Elseif: Variable Name Option = {{ Species }}
            Species Name[b], Volumetric Conc[c], Units[d] (mol/m^3),
            If: Variable Name Option = { File [ Binary | ASCII ] }
                File Name[e], Units[f] (kg/m^3),
                Note: File contains solute Conc. values for every node.
                Format: Char[a], Char[b], Real[c], Char[d], Char[e], Char[f],
            Elseif: Variable Name Option = { Zonation | Rock }
                Rock/Soil or Scaling Group Name[e],
                Format: Char[a], Char[b], Real[c], Char[d], Char[e],
            Else:
                X-Dir. Gradient[e], Units[f] (1/m), Y-Dir. Gradient[g], Units[h] (1/m),
                Z-Dir. Gradient[i], Units[j] (1/m),
                I-Start Index[k], I-End Index[l],
                J-Start Index[m], J-End Index[n],
                K-Start Index[o], K-End Index[p],
                Format: Char[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i],
                          Char[j], Integer[k], Integer[l], Integer[m], Integer[n], Integer[o], Integer[p],
            Endif:

        Endif:

     Endfor: Number of Initial Conditions Domains

Else:
    Initial Condition Option[a], { Hydrostatic},
    Pressure[b], Units[c] (Pa),  Z-Elevation[d], Units[e] (m),
    Temperature[f], Units[g] (C), Z-Elevation[h], Units[i] (m),
    Z-Dir. Temperature Gradient[j], Units[k]  (C/m),
    Salt Mass Fraction[l], Z-Elevation[m], Units[n] (m),
    Z-Dir. Salt Mass Fraction Gradient[o], Units[p] (1/m),
```

Format: Char[a], Real[b], Char[c], Real[d], Char[e], Real[f], Char[g], Real[h], Char[i], Real[j], Char[k], Real[l], Real[m], Char[n], Real[o], Char[p],

Endcard: Initial Conditions Card

B.18.1 Initial Conditions Card Example

```
~Initial Conditions Card
Gas Pressure,Aqueous Pressure,
3,
Gas Pressure,120.0,Bar,,,,,,,1,100,1,1,1,1,
Aqueous Pressure,120.0,Bar,,,,,,,1,100,1,1,1,1,
Temperature,45.0,C,,,,,,,1,100,1,1,1,1,
```

B.18.2 Initial Conditions Card Example

```
~Initial Conditions Card
Gas Pressure,Aqueous Pressure,
8,
Gas Pressure,260,bar,,,,,,,1,1,1,1,1,1,
Aqueous Pressure,148.80475,bar,,,,,,,1,1,1,1,1,1,
Temperature,54.0,C,,,,,,,1,1,1,1,1,1,
Salt Mass Fraction,0.1,,,,,,,,1,1,1,1,1,1,
Species Aqueous Volumetric,Cl-,1.00e+00,mol/liter,,,,,,,1,1,1,1,1,1,
Species Aqueous Volumetric,pH,7.0,,,,,,,,1,1,1,1,1,1,
Species Aqueous Volumetric,Na+,1.00e+00,mol/liter,,,,,,,1,1,1,1,1,1,
Species Aqueous Volumetric,O2(aq),1.00e-30,mol/liter,,,,,,,1,1,1,1,1,1,
```

B.18.3 Initial Conditions Card Example

```
~Initial Conditions Card
Gas Pressure,Aqueous Pressure,
4,
Gas Pressure,112.0525,Bar,,,,,-0.1001218,1/m,1,100,1,1,1,78,
Aqueous Pressure,112.0525,Bar,,,,,-0.1001218,1/m,1,100,1,1,1,78,
Temperature,37.0,C,,,,,,,1,100,1,1,1,78,
Salt Mass Fraction,0.032,,,,,,,,1,100,1,1,1,78,
```

B.18.4 Initial Conditions Card Example

```
~Initial Conditions Card
Gas Pressure,Aqueous Pressure,
4,
Gas Pressure File,2000,psi,fg_uic_pl.dat,psi,,
Aqueous Pressure File,2000,psi,fg_uic_pl.dat,psi,,
Temperature File,105,F,fg_uic_t.dat,F,,
Dissolved Salt Mass Fraction,0.04,,,,,,,,1,80,1,78,1,51,
```

B.18.5 Initial Conditions Card Example

```
~Initial Conditions Card
Hydrostatic,32.0,MPa,-3169.5,m,100.0,C,-3000.0,m,-0.03,C/m,0.1,-3000.0,m,
0.0,1/m,
```

B.19 Boundary Conditions Card

```
Card Title[a] {~Boundary [ Conditions Card ] }
Format: Char[a]

Number of Boundary Condition Domains[a],
Format: Integer[a],

For: Number of Boundary Condition Domains
    Boundary Surface Direction Option[a]
        { Bottom } { South } { West }{ East }{ North }{ Top }{ File }

    If: Operational Mode Option = { STOMP-CO2}
        Aqueous-Phase Boundary Type Option[b],
        { Dirichlet | Neumann | Zero Flux |
        Saturated | Unit Gradient | Hydraulic Gradient | Initial Condition }
        Gas-Phase Boundary Type Option[c],
        { Dirichlet | Neumann | Zero Flux |
        Hydraulic Gradient | Initial Condition }
        Salt Boundary Type Option[d],
        { Volumetric Conc. | Aqueous Conc. | Zero Flux |
        Outflow | Initial Condition |
        Inflow Volumetric Conc. | Inflow Aqueous Conc. |
        Inflow Mass Fraction | Inflow Relative Saturation |
        Aqueous Relative Saturation | Aqueous Mass Fraction }
        For: Number of Solutes
            Solute Transport Boundary Type Option[e],
            { Volumetric Conc. | Aqueous Conc. | Gas Conc. |
            Zero Flux | Outflow | Initial Condition |  Inflow-Outflow Aqueous |
            Inflow-Outflow Gas | Inflow-Outflow | Inflow Aqueous | Inflow Gas |
            Inflow | Outflow }
        Endfor: Number of Solutes
        Format: Char[a], Char[b], Char[c], Char[d], <Char[e],>

    Elseif: Operational Mode Option = { STOMP-CO2e }
        If: Operational Mode Option is not isothermal
            Energy Boundary Type Option[b],
            { Dirichlet | Neumann | Zero Flux |
            Outflow | Initial Condition }
        Endif:
        Aqueous-Phase Boundary Type Option[c],
        { Dirichlet | Neumann | Zero Flux |
        Saturated | Unit Gradient | Hydraulic Gradient | Initial Condition }
        Gas-Phase Boundary Type Option[d],
        { Dirichlet | Neumann | Zero Flux |
        Hydraulic Gradient | Initial Condition }
        Salt Boundary Type Option[e],
        { Volumetric Conc. | Aqueous Conc. | Zero Flux |
        Outflow | Initial Condition |
        Inflow Volumetric Conc. | Inflow Aqueous Conc. |
        Inflow Mass Fraction | Inflow Relative Saturation |
        Aqueous Relative Saturation | Aqueous Mass Fraction }
        For: Number of Solutes
            Solute Transport Boundary Type Option$^f$,
            { Volumetric Conc. | Aqueous Conc. | Gas Conc. |
            Zero Flux | Outflow | Initial Condition |  Inflow-Outflow Aqueous |
            Inflow-Outflow Gas | Inflow-Outflow | Inflow Aqueous | Inflow Gas |
            Inflow | Outflow }
        Endfor: Number of Solutes
        Format: Char$^a$, [ Char$^b$ ], Char$^c$, Char$^d$, Char$^e$, <Char$^f$,>
    Endif:

    If: Boundary Surface Direction Option=
        { Bottom } { South } { West } { East } { North } { Top }
        I-Start Index$^a$, I-End Index$^b$, J-Start Index$^c$, J-End Index$^d$,
        K-Start Index$^e$, K-End Index$^f$, Number of Boundary Times$^g$,
        Format: Integer$^a$, Integer$^b$, Integer$^c$, Integer$^d$, Integer$^e$, Integer$^f$, Integer$^g$,
    Elseif: Boundary Surface Direction Option={ File }
        Number of Boundary Times$^a$,
        Format: Integer$^a$,
    Endif:

    If: Operational Mode Option = { STOMP-CO2}
        For: Number of Boundary Times
            Boundary Time$^a$, Units$^b$ (s),
            If: Aqueous-Phase Boundary Type Option = { Dirichlet }
                Aqueous Pressure$^c$, Units$^d$ (Pa),
            Elseif: Aqueous-Phase Boundary Type Option = { Neumann }
                Aqueous Volumetric Flux$^c$, Units$^d$ (m/s),
            Elseif: Aqueous-Phase Boundary Type Option = { Hydraulic Gradient }
                Base Aqueous Pressure$^c$, Units$^d$ (Pa),
            Elseif: Aqueous-Phase Boundary Type Option = {Initial Cond[ition]} { Zero Flux }
                        Null$^c$, Null$^d$(Pa),
            Endif:
                Aqueous Dissolved- CO$_2$-Relative Saturation$^e$,
                If: Gas-Phase Boundary Type Option = { Dirichlet }
                        Gas Pressure$^f$, Units$^g$ (Pa),
                Elseif: Gas-Phase Boundary Type Option = { Neumann }
                        Gas Volumetric Flux$^f$, Units$^g$ (m/s),
                Elseif: Gas-Phase Boundary Type Option = { Hydraulic Gradient }
                        Base Gas Pressure$^f$, Units$^g$ (Pa),
                Elseif: Gas-Phase Boundary Type Option = { Initial Condition} { Zero Flux }
                        Null$^f$, Null$^g$,
                Endif:
                Water-Vapor Relative Humidity$^h$,
                If: Salt Boundary Type Option = {  Inflow  Molality }
                        Salt Aqueous-Phase Conc[entration]$^i$, Units$^j$ (mol/kg),
                Elseif: Salt Boundary Type Option = { Inflow  Aqueous Conc[entration] }
                        Salt Aqueous-Phase Volumetric Conc$^i$, Units$^j$ (1/m^3),
                Elseif: Salt Boundary Type Option =
                        { Inflow Mass Fraction } { Mass Fraction }
                        Salt Mass Fraction$^i$, Null$^j$,
                Elseif: Salt Boundary Type Option =
                        { Inflow Relative Saturation } { Relative Saturation}
                        Salt Relative Saturation, Null$^{j}$,
                Elseif:  Salt Boundary Type Option =
                  {Zero Flux}{Outflow}{Initial Cond[ition]}
                  Null$^{i}$, Null$^{j}$,
                Endif:
                For: Number of Solutes
                  If: Solute Transport Boundary Type Option = {Volumetric
                        Conc[entration]}
                            Solute Volumetric Conc$^{k}$, Units$^{l}$ (1/m^3),
                  Elseif: Solute Transport Boundary Type Option = {Aqueous
                        Conc[entration]} {Inflow Aqu[eous]} {Inflow –Outflow
                         Aqu[eous]} {Inflow-Outflow Gas}
                        Solute Aqueous-Phase Conc$^{k}$, Units$^{l}$ (1/m^3),
                  Elseif:   Solute Transport Boundary Type Option =
                        {Gas Conc[entration]} {Inflow Gas}
                        Solute Gas-Phase Conc$^{k}$, Units$^{l}$ (1/m^3),
                  Else:
                            Null$^{k}$, Null$^{l}$,
                  Endif:
                Endfor: Number of Solutes
                Format: Real$^{a}$, Char$^{b}$, Real$^{c}$, Char$^{d}$, Real$^{e}$, Real$^{f}$, Char$^{g}$, Real$^{h}$, Real$^{i}$, Char$^{j}$,
                            < Real$^{k}$, Char$^{l}$, >

                For:  Number of Reactive Species
                        If:  Reactive Species Boundary Type Option = {Aqueous
                        Conc[entration]} {Inflow Aqu[eous]} {Inflow –Outflow
                         Aqu[eous]}
                        Aqueous-Phase Conc$^{k}$, Units$^{l}$ (1/m^3),
                  Elseif:   Reactive Species Boundary Type Option =
                        {Gas Conc[entration]} {Inflow Gas} {Inflow-Outflow Gas}
                        Gas-Phase Conc$^{k}$, Units$^{l}$ (1/m^3),
                  Else:
                            Null$^{k}$, Null$^{l}$,
                  Endif:
                Endfor:  Number of Reactive Species
                Format: Real$^{a}$, Char$^{b}$, Real$^{c}$, Char$^{d}$, Real$^{e}$, Real$^{f}$, Char$^{g}$, Real$^{h}$, Real$^{i}$, Char$^{j}$,
                            < Real$^{k}$, Char$^{l}$, >
            Endfor: Number of Boundary Times

        Elseif: Operational Mode Option = { STOMP-CO2e }
        For: Number of Boundary Times
            Boundary Time$^{a}$, Units$^{b}$ (s),
            If: Energy Boundary Type Option = { Dirichlet }
                Temperature$^{c}$, Units$^{d}$ (C),
            Elseif: Energy Boundary Type Option = { Neumann }
                Energy Flux$^{c}$, Units$^{d}$,
            Elseif: Energy Boundary Type Option = { Zero-Flux } { Outflow }
                { Initial Condition }
                Null$^{c}$, Null$^{d}$,
            Endif:
            If: Aqueous-Phase Boundary Type Option = { Dirichlet } { Zero Flux }
                Aqueous Pressure[e], Units[f] (Pa),
            Elseif: Aqueous-Phase Boundary Type Option = { Neumann }
                Aqueous Volumetric Flux[e], Units[f] (m/s),
            Elseif: Aqueous-Phase Boundary Type Option = { Hydraulic Gradient }
                        Base Aqueous Pressure[e], Units[f] (Pa),
            Else:
                Null[e], Null[f],
            Endif:
            Aqueous Dissolved-$CO_2$ Relative Saturation[g],
            If: Gas-Phase Boundary Type Option = { Dirichlet } { Zero Flux }
                Gas Pressure[h], Units[i] (Pa),
            Elseif: Gas-Phase Boundary Type Option = { Neumann }
                Gas Volumetric Flux[h], Units[i] (m/s),
            Elseif: Gas-Phase Boundary Type Option = { Hydraulic Gradient }
                Base Gas Pressure[h], Units[i] (Pa),
            Else:
                Null[h], Null[i],
            Endif:
            Water-Vapor Relative Humidity[j],
            If: Salt Boundary Type Option = { [ Inflow ] Volumetric Conc[entration] }
                Salt Volumetric Conc[entration][k], Units[l] (kg/m^3),
            Elseif: Salt Boundary Type Option = { [ Inflow ] Aqueous Conc[entration] }
                Salt Aqueous-Phase Volumetric Conc[entration][k], Units[l] (kg/m^3),
            Elseif: Salt Boundary Type Option =
                { Inflow Aq[ueous] Saturation } { Inflow Relative Saturation }
                Salt Relative Saturation[k], Null[l],
            Elseif: Salt Boundary Type Option =
                { Inflow Mass Fraction } { Aqueous Mass Fraction }
                Salt Mass Fraction[k], Null[l],
            Else:
                Null[k], Null[l],
            Endif:
            For: Number of Solutes
                If: Solute Transport Boundary Type Option = {Volumetric
                  Conc[entration]}
                            Solute Volumetric Conc[m], Units[n](1/m^3),
                Elseif: Solute Transport Boundary Type Option = {Aqueous
                  Conc[entration]} {Inflow Aqu[eous]} {Inflow –Outflow
                   Aqu[eous]} {Inflow-Outflow Gas}
                        Solute Aqueous-Phase Conc[m], Units[n] (1/m^3),
                Elseif:   Solute Transport Boundary Type Option =
                  {Gas Conc[entration]} {Inflow Gas}
                  Solute Gas-Phase Conc[m], Units[n] (1/m^3),
                Else:
                        Null[m], Null[n],
                Endif:
            Endfor: Number of Solutes
            Format: [ Real[a], Char[b], ] Real[c], Char[d], Real[e], Char[f], Real[g], Real[h],
                        Char[i], Real[j], Real[k], Char[l], < Real[m], Char[n], >
            For:  Number of Reactive Species
                If:  Reactive Species Boundary Type Option = {Aqueous
                  Conc[entration]} {Inflow Aqu[eous]} {Inflow –Outflow
```

```
                Aqu[eous]}  {Inflow-Outflow Gas}
                    Aqueous-Phase Conc$^m$, Units$^n$ (1/m^3),
            Elseif:   Solute Transport Boundary Type Option =
                    {Gas Conc[entration]}  {Inflow Gas}
                     Gas-Phase Conc$^k$, Units$^l$ (1/m^3),
            Else:
                              Null$^m$, Null$^n$,
        Endif:
        Endfor:  Number of Reactive Species
        Format: [ Real$^a$, Char$^b$, ] Real$^c$, Char$^d$, Real$^e$, Char$^f$, Real$^g$, Real$^h$,
                    Char$^i$, Real$^j$, Real$^k$, Char$^l$, < Real$^m$, Char$^n$, >

    Endfor: Number of Boundary Times

Endif:

Endfor: Number of Boundary Condition Domains

Endcard: Boundary Conditions Card
```

B.19.1 Boundary Conditions Card Example

```
~Boundary Conditions Card
4,
North,Aqu. Hydraulic Gradient,Gas Zero Flux,Aqu. Mass Frac.,
1,66,66,66,1,20,1,
0,s,3265,psi,,,,,0.13,,
East,Aqu. Hydraulic Gradient,Gas Zero Flux,Aqu. Mass Frac.,
66,66,1,66,1,20,1,
0,s,3265,psi,,,,,0.13,,
South,Aqu. Zero Flux,Gas Zero Flux,Aqu. Mass Frac.,
1,66,1,1,1,20,1,
0,s,3265,psi,,,,,0.13,,
West,Aqu. Zero Flux,Gas Zero Flux,Aqu. Mass Frac.,
1,1,1,66,1,20,1,
0,s,3265,psi,,,,,0.13,,
```

B.19.2 Boundary Conditions Card Example*

```
~Boundary Conditions Card
4,
file,bows_prop_bc_east.dat,Aqu Initial Condition,Gas Zero Flux,Aqu. Mass
Frac.,
1,
0,s,3000,psi,,,,,0.13,,,,
file,bows_prop_bc_south.dat,Aqu Initial Condition,Gas Zero Flux,Aqu. Mass
Frac.,
1,
0,s,3000,psi,,,,,0.13,,,,
file,bows_prop_bc_west.dat,Aqu Initial Condition,Gas Zero Flux,Aqu. Mass
Frac.,
1,
0,s,3000,psi,,,,,0.13,,,,
file,bows_prop_bc_north.dat,Aqu Initial Condition,Gas Zero Flux,Aqu. Mass
Frac.,
1,
0,s,3000,psi,,,,,0.13,,,,
```

*file format: I-Index, J-Index, K-Index, Boundary Direction Index {Bottom = -3, South = -2, West = -1, East = 1, North = 2, Top = 3}

B.19.3 Boundary Conditions Card Example

```
~Boundary Conditions Card
1,
East,Energy Dirichlet, Aqu. Dirichlet,Gas Dirichlet,Aqu. Mass Frac.,
100,100,1,1,1,1,1,
0,s,85,F,120.0,bar,0.0,120.0,bar,1.0,0.0,,
```

B.19.4 Boundary Conditions Card Example

```
~Boundary Conditions Card
2,
Top,Aqueous Dirichlet,Gas Dirichlet,Aqueous Mass Fraction,
1,1,1,1,100,100,1,
0,s,100.0,bar,0.0,100.0,bar,1.0,0.0,,
Bottom,Aqueous Zero Flux,Gas Dirichlet,Aqueous Mass Fraction,
1,1,1,1,1,1,1,
0,s,,,0.0,240.0,bar,0.0,0.0,,
```

B.19.5 Boundary Conditions Card Example

```
~Boundary Conditions Card
1,
East,Aqueous Initial Condition,Gas Initial Condition,Aqueous Initial
Condition,
100,100,1,1,1,193,1,
0,s,,,,,,,,,
```

B.19.6 Boundary Conditions Card Example

```
~Boundary Conditions Card
3,
West,Aqu. Initial Condition,Gas Initial Condition,Aqu. Mass Frac.,
1,1,1,32,1,44,1,
0,s,,,,,,,0.0,,,,
East,Aqu. Initial Condition,Gas Initial Condition,Aqu. Mass Frac.,
73,73,1,32,1,44,1,
0,s,,,,,,,0.0,,,,
North,Aqu. Initial Condition,Gas Initial Condition,Aqu. Mass Frac.,
1,73,32,32,1,44,1,
0,s,,,,,,,0.0,,,,
```

B.19.7 Boundary Conditions Card Example

```
~Boundary Conditions Card
1,
Top,Aqueous Dirichlet,Gas Dirichlet,Aqueous Mass Fraction,Aqu. Species Zero
Flux,Gas Species Zero Flux,
0,
1,1,1,1,1,1,1,
0,s,148.80475,bar,1,260,bar,1,0.1,,
```

B.20 Source Card

```
Card Title[a] { ~Source [ Card ] }
Format: Char[a]

Number of Source Domains[a],
Format: Integer[a],

For: Number of Source Domains

    Source Type Option[a]
        {
            If: Operational Mode Option = { STOMP-CO2e }
                Power [ Density ]
            Endif:
            Aqueous Volumetric | Aqueous Mass |
            Gas Volumetric | Gas Mass |
            Salt [ Density ] | Solute [ Density ], Solute Name
        }
    If: Source Type Option = { Aqueous Volumetric }
        Dissolved Salt Source Option[b],
        { Dissolved Salt Aqueous Concentration | Dissolved Salt Mass Fraction |
        Dissolved Salt Relative Saturation | Null }
        Dissolved CO2 Source Option[c],
        { Dissolved CO2 Aqueous Concentration | Dissolved CO2 Mass Fraction |
        Dissolved CO2 Relative Saturation | Null }
        I-Start Index[d], I-End Index[e], J-Start Index[f], J-End Index[g],
        K-Start Index[h], K-End Index[i], Number of Source Times[j],
        Format: Char[a], Char[b], Char[c], Integer[d], Integer[e], Integer[f], Integer[g], Integer[h], Integer[i],
        Integer[j],
    Elseif: Source Type Option = { Aqueous Mass }
        Dissolved Salt Source Option[b],
        { Dissolved Salt Aqueous Concentration | Dissolved Salt Mass Fraction |
        Dissolved Salt Relative Saturation | Null }
        Dissolved CO2 Source Option[c],
        { Dissolved CO2 Aqueous Concentration | Dissolved CO2 Mass Fraction |
        Dissolved CO2 Relative Saturation | Null }
        I-Start Index[d], I-End Index[e], J-Start Index[f], J-End Index[g],
        K-Start Index[h], K-End Index[i], Number of Source Times[j],
        Format: Char[a], Char[b], Char[c], Integer[d], Integer[e], Integer[f], Integer[g], Integer[h], Integer[i],
        Integer[j],
    Elseif: Source Type Option = { Gas Volumetric }
        Water Vapor Source Option[b],
        { Water Vapor Gas Relative Humidity | Water Vapor Gas Mass Fraction | Null }
        I-Start Index[c], I-End Index[d], J-Start Index[e], J-End Index[f],
        K-Start Index[g], K-End Index[h], Number of Source Times[i] ,
        Format: Char[a], Char[b], Integer[c], Integer[d], Integer[e], Integer[f], Integer[g], Integer[h],
        Integer[i],
    Elseif: Source Type Option = { Gas Mass }
        Water Vapor Source Option[b],
        { Water Vapor Gas Relative Humidity | Water Vapor Gas Mass Fraction | Null }
        I-Start Index$^{c}$, I-End Index$^{d}$, J-Start Index$^{e}$, J-End Index$^{f}$,
        K-Start Index$^{g}$, K-End Index$^{h}$, Number of Source Times$^{i}$ ,
        Format: Char$^{a}$, Char$^{b}$, Integer$^{c}$, Integer$^{d}$, Integer$^{e}$, Integer$^{f}$, Integer$^{g}$, Integer$^{h}$,
        Integer$^{i}$,
    Else:
        I-Start Index$^{b}$. I-End Index$^{c}$, J-Start Index$^{d}$, J-End Index$^{e}$,
        K-Start Index$^{f}$, K-End Index$^{g}$, Number of Source Times$^{h}$,
        Format: Char$^{a}$. Integer$^{b}$, Integer$^{c}$, Integer$^{d}$, Integer$^{e}$, Integer$^{f}$, Integer$^{g}$, Integer$^{h}$,
    Endif:

    If: Operational Mode Option = { STOMP-CO2 }
        For: Number of Source Times

        Source Time$^{a}$, Units$^{b}$ (s),
        If: Source Type Option = { Aqu[eous] Vol[umetric] }
            Pressure$^{c}$, Units$^{d}$ (Pa), Aqueous Volumetric Rate$^{e}$, Units$^{f}$ (m^3/s),
            If: Aqueous Salt Source Option = { Aqu[eous] Conc[entration] }
                Aqueous Salt Aqueous Concentration$^{g}$, Units$^{h}$ (kg/m^3),
                If: Aqueous CO2 Source Option = { Aqu[eous] Conc[entration] }
                    Aqueous CO2 Aqueous Concentration$^{i}$, Units$^{j}$ (kg/m^3),
                    Format: Real$^{a}$, Char$^{b}$, Real$^{c}$, Char$^{d}$, Real$^{e}$, Char$^{f}$, Real$^{g}$, Char$^{h}$, Real$^{i}$, Char$^{j}$,
                Elseif:  Aqueous CO2 Source Option = { Mass Frac[tion] }
                    Aqueous CO2 Mass Fraction$^{i}$,
                    Format: Real$^{a}$, Char$^{b}$, Real$^{c}$, Char$^{d}$, Real$^{e}$, Char$^{f}$, Real$^{g}$, Char$^{h}$, Real$^{i}$,
                Elseif:  Aqueous CO2 Source Option = { Rel[ative] Sat[uration] }
                    Aqueous CO2 Relative Saturation$^{i}$,
                    Format: Real$^{a}$, Char$^{b}$, Real$^{c}$, Char$^{d}$, Real$^{e}$, Char$^{f}$, Real$^{g}$, Char$^{h}$, Real$^{i}$,
                Endif:
            Elseif:  Aqueous Salt Source Option = { Mass Frac[tion] }
                Aqueous Salt Mass Fraction$^{g}$,
                If:  Aqueous CO2 Source Option = { Aqu[eous] Conc[entration] }
                    Aqueous CO2 Aqueous Concentration$^{h}$, Units$^{i}$ (kg/m^3),
                    Format: Real$^{a}$, Char$^{b}$, Real$^{c}$, Char$^{d}$, Real$^{e}$, Char$^{f}$, Real$^{g}$, Real$^{h}$, Char$^{i}$,
                 Elseif: Aqueous CO2 Source Option = { Mass Frac[tion] }
                    Aqueous CO2 Mass Fraction$^{h}$,
                    Format: Real$^{a}$, Char$^{b}$, Real$^{c}$, Char$^{d}$, Real$^{e}$, Char$^{f}$, Real$^{g}$, Real$^{h}$,
                Elseif:  Aqueous CO2 Source Option = { Rel[ative] Sat[uration] }
                    Aqueous CO2 Relative Saturation$^{h}$,
                     Format: Real$^{a}$, Char$^{b}$, Real$^{c}$, Char$^{d}$, Real$^{e}$, Char$^{f}$, Real$^{g}$, Real$^{h}$,
                Endif:
            Elseif:  Aqueous Salt Source Option = { Rel[ative] Sat[uration] }
                Aqueous Relative Saturation$^{g}$,
                If:  Aqueous CO2 Source Option = { Aqu[eous] Conc[entration] }
                    Aqueous CO2 Aqueous Concentration$^{h}$, Units$^{i}$ (kg/m^3),
                    Format: Real$^{a}$, Char$^{b}$, Real$^{c}$, Char$^{d}$, Real$^{e}$, Char$^{f}$, Real$^{g}$, Real$^{h}$, Char$^{i}$,
                Elseif:  Aqueous CO2 Source Option = { Mass Frac[tion] }
                    Aqueous CO2 Mass Fraction$^{h}$,
                    Format: Real$^{a}$, Char$^{b}$, Real$^{c}$, Char$^{d}$, Real$^{e}$, Char$^{f}$, Real$^{g}$, Real$^{h}$,
                Elseif:  Aqueous CO2 Source Option = { Rel[ative] Sat[uration] }
                    Aqueous CO2 Relative Saturation$^{h}$,
                    Format: Real$^{a}$, Char$^{b}$, Real$^{c}$, Char$^{d}$, Real$^{e}$, Char$^{f}$, Real$^{g}$, Real$^{h}$,
                 Else:
                    Format: Real[a], Char[b], Real[c], Char[d], Real[e], Char[f],
                Endif:
            Endif:
        Elseif: Source Type Option = { Aqu[eous] Mass }
            Pressure[c], Units[d] (Pa), Aqueous Mass Rate[e], Units[f] (kg/s),
             If:  Aqueous Salt Source Option = { Aqu[eous] Conc[entration] }
                Aqueous Salt Aqueous Concentration[g], Units[h] (kg/m^3),
                If:  Aqueous CO2 Source Option = { Aqu[eous] Conc[entration] }
                    Aqueous CO2 Aqueous Concentration[i], Units[j] (kg/m^3),
                    Format: Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j],
                Elseif:  Aqueous CO2 Source Option = { Mass Frac[tion] }
                    Aqueous CO2 Mass Fraction[i],
                    Format: Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i],
                Elseif:  Aqueous CO2 Source Option = { Rel[ative] Sat[uration] }
                    Aqueous CO2 Relative Saturation[i],
                    Format: Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i],
                Endif:
             Elseif:  Aqueous Salt Source Option = { Mass Frac[tion] }
                Aqueous Salt Mass Fraction[g],
                If: Aqueous CO2 Source Option = { Aqu[eous] Conc[entration] }
                    Aqueous CO2 Aqueous Concentration[h], Units[i] (kg/m^3),
                    Format: Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Real[h], Char[i],
                Elseif: Aqueous CO2 Source Option = { Mass Frac[tion] }
                    Aqueous CO2 Mass Fraction[h],
                    Format: Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Real[h],
                Elseif: Aqueous CO2 Source Option = { Rel[ative] Sat[uration] }
                    Aqueous CO2 Relative Saturation[h],
                    Format: Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Real[h],
                Endif:
            Elseif: Aqueous Salt Source Option = { Rel[ative] Sat[uration] }
                Aqueous Relative Saturation[g],
                If: Aqueous CO2 Source Option = { Aqu[eous] Conc[entration] }
                    Aqueous CO2 Aqueous Concentration[h], Units[i] (kg/m^3),
                    Format: Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Real[h], Char[i],
                Elseif: Aqueous CO2 Source Option = { Mass Frac[tion] }
                    Aqueous CO2 Mass Fraction[h],
                    Format: Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Real[h],
                Elseif: Aqueous CO2 Source Option = { Rel[ative] Sat[uration] }
                    Aqueous CO2 Relative Saturation[h],
                    Format: Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Real[h],
            Else:
                Format: Real[a], Char[b], Real[c], Char[d], Real[e], Char[f],
            Endif:
        Elseif: Source Type Option = { Gas Mass }
            Pressure[c], Units[d] (Pa), Gas Mass Rate[e], Units[f] (kg/s),
            If: Water Vapor Source Option = { Mass Frac[tion] }
                Water Vapor Mass Fraction[g],
                Format: Real[a], Char[b], Real[c], Char[d], Real[e], Char[f],Real[g],
             Elseif: Water Vapor Source Option = { Rel[ative] Sat[uration] }
                Water Vapor Relative Humidity[g],
                Format: Real[a], Char[b], Real[c], Char[d], Real[e], Char[f],Real[g],
            Else:
                Null$^g$,
                Format: Real$^a$, Char$^b$, Real$^c$, Char$^d$, Real$^e$, Char$^f$,Real$^g$,
            Endif:
        Elseif: Source Type Option = { Gas Volumetric }
            Pressure$^c$, Units$^d$ (Pa), Gas Volumetric Rate$^e$, Units$^f$ (m^3/s),
            If: Water Vapor Source Option = { Mass Frac[tion] }
                Water Vapor Mass Fraction$^g$,
                Format: Real$^a$, Char$^b$, Real$^c$, Char$^d$, Real$^e$, Char$^f$,Real$^g$,
            Elseif: Water Vapor Source Option = { Rel[ative] Sat[uration] }
                Water Vapor Relative Humidity$^g$,
                Format: Real$^a$, Char$^b$, Real$^c$, Char$^d$, Real$^e$, Char$^f$,Real$^g$,
            Else:
                Null$^g$,
                Format: Real$^a$, Char$^b$, Real$^c$, Char$^d$, Real$^e$, Char$^f$,Real$^g$,
        Endif:
        Elseif: Source Type Option = { Salt }
            Salt Mass Rate$^c$, Units$^d$ (kg/s)
            Format: Real$^a$, Char$^b$, Real$^c$, Char$^d$,
        Elseif: Source Type Option = { Salt Density }
            Salt Mass Density Rate$^c$, Units$^d$ (kg/m^3 s)
            Format: Real$^a$, Char$^b$, Real$^c$, Char$^d$,
        Elseif: Source Type Option = { Solute }
            Solute Rate$^c$, Units$^d$ (1/s),
            Format: Real$^a$, Char$^b$, Real$^c$, Char$^d$,
        Elseif: Source Type Option = { Solute Density }
             Solute Density Rate$^c$, Units$^d$ (1/s m^3),
            Format: Real$^a$, Char$^b$, Real$^c$, Char$^d$,
        Endif:

        Endfor: Number of Source Times

    Elseif: Operational Mode Option = { STOMP-CO2e }
        For: Number of Source Times

        Source Time$^a$, Units$^b$ (s),
        If: Source Type Option = { Power }
            Power$^c$, Units$^d$ (W),
            Format: Real$^a$, Char$^b$, Real$^c$, Char$^d$,
        Elseif: Source Type Option = { Power Density }
            Power Density$^c$, Units$^d$ (W/m^3),
            Format: Real$^a$, Char$^b$, Real$^c$, Char$^d$,
        Endif:
        If: Source Type Option = { Aqu[eous] Vol[umetric] }
            If: Operational Mode Option is not isothermal
                Temperature$^c$, Units$^d$ (C),
            Else:
                null$^c$, null$^d$,
            Endif:
            Pressure$^e$, Units$^f$ (Pa), Aqueous Volumetric Rate$^g$, Units$^h$ (m^3/s),
            If: Aqueous Salt Source Option = { Aqu[eous] Conc[entration] }
                Aqueous Salt Aqueous Concentration$^i$, Units$^j$ (kg/m^3),
                If: Aqueous CO2 Source Option = { Aqu[eous] Conc[entration] }
                    Aqueous CO2 Aqueous Concentration[k], Units[l] (kg/m^3),
                    Format: Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j],
                    Real[k], Char[l],
                Elseif: Aqueous CO2 Source Option = { Mass Frac[tion] }
                    Aqueous CO2 Mass Fraction[k],
                    Format: Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j],
                        Real[k],
                Elseif: Aqueous CO2 Source Option = { Rel[ative] Sat[uration]}
                    Aqueous CO2 Relative Saturation[k],
                    Format: Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j],
                        Real[k],
                Endif:
            Elseif: Aqueous Salt Source Option = { Mass Frac[tion] }
                Aqueous Salt Mass Fraction[i],
                If: Aqueous CO2 Source Option = { Aqu[eous] Conc[entration] }
                    Aqueous CO2 Aqueous Concentration[j], Units[k] (kg/m^3),
                    Format: Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j],
                        Real[k],
                Elseif: Aqueous CO2 Source Option = { Mass Frac[tion] }
                    Aqueous CO2 Mass Fraction[j],
                    Format: Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Char[j],
                Elseif: Aqueous CO2 Source Option = { Rel[ative] Sat[uration] }
                    Aqueous CO2 Relative Saturation[j],
                    Format: Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h],Real[i], Real[j],
                Endif:
            Elseif: Aqueous Salt Source Option = { Rel[ative] Sat[uration] }
                Aqueous Relative Saturation[i],
                If: Aqueous CO2 Source Option = { Aqu[eous] Conc[entration] }
                    Aqueous CO2 Aqueous Concentration[j], Units[k] (kg/m^3),
                    Format: Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Real[j],
                        Char[k],
                Elseif: Aqueous CO2 Source Option = { Mass Frac[tion] }
                    Aqueous CO2 Mass Fraction[j],
                    Format: Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Real[j],
                 Elseif: Aqueous CO2 Source Option = { Rel[ative] Sat[uration] }
                    Aqueous CO2 Relative Saturation[j],
                    Format: Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i], Real[j],
                Else:
                    Format: Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h], Real[i],
                Endif:
            Endif:
        Elseif: Source Type Option = { Aqu[eous] Mass }
            If: Operational Mode Option is not isothermal
                Temperature[c], Units[d] (C),
            Else:
                null[c], null[d],
            Endif:
            Pressure[e], Units[f] (Pa), Aqueous Volumetric Rate[g], Units[h] (m^3/s),
            If: Aqueous Salt Source Option = { Aqu[eous] Conc[entration] }
                Aqueous Salt Aqueous Concentration[i], Units[j] (kg/m^3),
                If: Aqueous CO2 Source Option = { Aqu[eous] Conc[entration] }
```

```
            Aqueous CO2 Aqueous Concentration$^{k}$, Units$^{l}$ (kg/m^3),
            Format: Real$^{a}$, Char$^{b}$, Real$^{c}$, Char$^{d}$, Real$^{e}$, Char$^{f}$, Real$^{g}$, Char$^{h}$, Real$^{i}$, Char$^{j}$,
            Real$^{k}$, Char$^{l}$,
        Elseif: Aqueous CO2 Source Option = { Mass Frac[tion] }
            Aqueous CO2 Mass Fraction$^{k}$,
            Format: Real$^{a}$, Char$^{b}$, Real$^{c}$, Char$^{d}$, Real$^{e}$, Char$^{f}$, Real$^{g}$, Char$^{h}$, Real$^{i}$, Char$^{j}$,
                Real$^{k}$,
        Elseif: Aqueous CO2 Source Option = { Rel[ative] Sat[uration] }
            Aqueous CO2 Relative Saturation$^{k}$,
            Format: Real$^{a}$, Char$^{b}$, Real$^{c}$, Char$^{d}$, Real$^{e}$, Char$^{f}$, Real$^{g}$, Char$^{h}$, Real$^{i}$, Char$^{j}$,
                Real$^{k}$,
        Endif:
    Elseif: Aqueous Salt Source Option = { Mass Frac[tion] }
        Aqueous Salt Mass Fraction$^{i}$,
        If: Aqueous CO2 Source Option = { Aqu[eous] Conc[entration] }
            Aqueous CO2 Aqueous Concentration$^{j}$, Units$^{k}$ (kg/m^3),
            Format: Real$^{a}$, Char$^{b}$, Real$^{c}$, Char$^{d}$, Real$^{e}$, Char$^{f}$, Real$^{g}$, Char$^{h}$, Real$^{i}$, Real$^{j}$,
                Real$^{k}$,
        Elseif: Aqueous CO2 Source Option = { Mass Frac[tion] }
            Aqueous CO2 Mass Fraction$^{j}$,
            Format: Real$^{a}$, Char$^{b}$, Real$^{c}$, Char$^{d}$, Real$^{e}$, Char$^{f}$, Real$^{g}$, Char$^{h}$, Real$^{i}$, Real$^{j}$,
        Elseif: Aqueous CO2 Source Option = { Rel[ative] Sat[uration] }
            Aqueous CO2 Relative Saturation$^{j}$,
            Format: Real$^{a}$, Char$^{b}$, Real$^{c}$, Char$^{d}$, Real$^{e}$, Char$^{f}$, Real$^{g}$, Char$^{h}$, Real$^{i}$, Real$^{j}$,
    Elseif: Aqueous Salt Source Option = { Rel[ative] Sat[uration] }
        Aqueous Relative Saturation$^{i}$,
        If: Aqueous CO2 Source Option = { Aqu[eous] Conc[entration] }
            Aqueous CO2 Aqueous Concentration$^{j}$, Units$^{k}$ (kg/m^3),
            Format: Real$^{a}$, Char$^{b}$, Real$^{c}$, Char$^{d}$, Real$^{e}$, Char$^{f}$, Real$^{g}$, Char$^{h}$, Real$^{i}$, Real$^{j}$,
                Real$^{k}$,
        Elseif: Aqueous CO2 Source Option = { Mass Frac[tion] }
            Aqueous CO2 Mass Fraction$^{j}$,
            Format: Real$^{a}$, Char$^{b}$, Real$^{c}$, Char$^{d}$, Real$^{e}$, Char$^{f}$, Real$^{g}$, Char$^{h}$, Real$^{i}$, Real$^{j}$,
        Elseif: Aqueous CO2 Source Option = { Rel[ative] Sat[uration] }
            Aqueous CO2 Relative Saturation$^{j}$,
            Format: Real$^{a}$, Char$^{b}$, Real$^{c}$, Char$^{d}$, Real$^{e}$, Char$^{f}$, Real$^{g}$, Char$^{h}$, Real$^{i}$, Real$^{j}$,
        Endif:
    Else:
        Format: Real$^{a}$, Char$^{b}$, Real$^{c}$, Char$^{d}$, Real$^{e}$, Char$^{f}$, Real$^{g}$, Char$^{h}$, Real$^{i}$,,
    Endif:
Elseif: Source Type Option = { Gas Mass }
    If: Operational Mode Option is not isothermal
        Temperature$^{c}$, Units$^{d}$ (C),
    Else:
        null$^{c}$, null$^{d}$,
    Endif:
    Pressure$^{e}$, Units$^{f}$ (Pa), Gas Mass Rate$^{g}$, Units$^{h}$ (kg/s),
    If: Water Vapor Source Option = { Mass Frac[tion] }
        Water Vapor Mass Fraction$^{i}$,
        Format: Real$^{a}$, Char$^{b}$, Real$^{c}$, Char$^{d}$, Real$^{e}$, Char$^{f}$, Real$^{g}$, Char$^{h}$, Real$^{i}$,
    Elseif: Water Vapor Source Option = { Rel[ative] Sat[uration] }
        Water Vapor Relative Humidity$^{i}$,
                Format: Real$^a$, Char$^b$, Real$^c$, Char$^d$, Real$^e$, Char$^f$, Real$^g$, Char$^h$, Real$^i$,
            Else:
                Null$^i$,
                Format: Real$^a$, Char$^b$, Real$^c$, Char$^d$, Real$^e$, Char$^f$, Real$^g$, Char$^h$, Real$^i$,
            Endif:
        Elseif: Source Type Option = { Gas Vol[umetric] }
            If: Operational Mode Optional is not isothermal
                Temperature$^c$, Units$^d$ (C),
            Else:
                null$^c$, null$^d$,
            Endif:
            Pressure$^e$, Units$^f$ (Pa), Gas Volumetric Rate$^g$, Units$^h$ (m^3/s),
            If: Water Vapor Source Option = { Mass Frac[tion] }
                Water Vapor Mass Fraction$^i$,
                Format: Real$^a$, Char$^b$, Real$^c$, Char$^d$, Real$^e$, Char$^f$, Real$^g$, Char$^h$, Real$^i$,
            Elseif: Water Vapor Source Option = { Rel[ative] Sat[uration] }
                Water Vapor Relative Humidity$^i$,
                Format: Real$^a$, Char$^b$, Real$^c$, Char$^d$, Real$^e$, Char$^f$, Real$^g$, Char$^h$, Real$^i$,
            Else:
                Null$^i$,
                Format: Real$^a$, Char$^b$, Real$^c$, Char$^d$, Real$^e$, Char$^f$, Real$^g$, Char$^h$, Real$^i$,
            Endif:
        Elseif: Source Type Option = { Salt }
            Salt Mass Rate$^c$, Units$^d$ (kg/s)
            Format: Real$^a$, Char$^b$, Real$^c$, Char$^d$,
        Elseif: Source Type Option = { Salt Density }
            Salt Mass Density Rate$^c$, Units$^d$ (kg/m^3 s)
            Format: Real$^a$, Char$^b$, Real$^c$, Char$^d$,
        Elseif: Source Type Option = { Solute }
            Solute Rate$^c$, Units$^d$ (1/s),
            Format: Real$^a$, Char$^b$, Real$^c$, Char$^d$,
        Elseif: Source Type Option = { Solute Density }
            Solute Density Rate$^c$, Units$^d$ (1/s m^3),
            Format: Real$^a$, Char$^b$, Real$^c$, Char$^d$,
        Endif:

    Endfor: Number of Source Times

    Endif:

Endfor: Number of Source Domains

Endcard: Source Card
```

B.20.1 Source Card Example

```
~Source Card
1,
Gas Mass Rate,Water-Vapor Mass Fraction,1,1,1,1,1,9,2,
0,day,,,0.352413,kg/s,0.0,
7300,day,,,0.352413,kg/s,0.0,
```

B.20.2 Source Card Example

```
~Source Card
5,
Aqueous Volumetric,null,null,12,12,12,12,1,5,2,
0,yr,3265,psi,-1.8E-03,m^3/s,
10,yr,3265,psi,-1.8E-03,m^3/s,
Aqueous Volumetric,null,null,16,16,16,16,1,5,2,
10,yr,3265,psi,-1.8E-03,m^3/s,
20,yr,3265,psi,-1.8E-03,m^3/s,
Aqueous Volumetric,null,null,18,18,18,18,1,5,2,
20,yr,3265,psi,-1.8E-03,m^3/s,
30,yr,3265,psi,-1.8E-03,m^3/s,
Aqueous Volumetric,null,null,19,19,19,19,1,5,2,
30,yr,3265,psi,-1.8E-03,m^3/s,
40,yr,3265,psi,-1.8E-03,m^3/s,
Aqueous Volumetric,null,null,21,21,21,21,1,5,2,
40,yr,3265,psi,-1.8E-03,m^3/s,
50,yr,3265,psi,-1.8E-03,m^3/s,
```

B.20.3 Source Card Example

```
~Source Card
1,
Gas Volumetric Rate,Water-Vapor Mass Fraction,3,3,1,1,3,3,2,
0,hr,25.0,C,138.0,bar,4.2908,cm^3/min,0.0,
0.2,hr,25.0,C,138.0,bar,4.2908,cm^3/min,0.0,
```

B.20.4 Source Card Example

```
~Source Card,
3,
Gas Mass,Water Relative Humidity,4,4,4,4,6,6,4,
0,yr,120.0,F,,psi,4.223981E-03,MMT/yr,0.0,
5,yr,120.0,F,,psi,4.223981E-03,MMT/yr,0.0,
5,yr,120.0,F,,psi,0,MMT/yr,0.0,
50,yr,120.0,F,,psi,0,MMT/yr,0.0,
Gas Mass,Water Relative Humidity,4,4,4,4,7,7,4,
0,yr,120.0,F,,psi,4.223981E-03,MMT/yr,0.0,
5,yr,120.0,F,,psi,4.223981E-03,MMT/yr,0.0,
5,yr,120.0,F,,psi,0,MMT/yr,0.0,
50,yr,120.0,F,,psi,0,MMT/yr,0.0,
Gas Mass,Water Relative Humidity,4,4,4,4,8,8,4,
0,yr,120.0,F,,psi,9.208764E-03,MMT/yr,0.0,
5,yr,120.0,F,,psi,9.208764E-03,MMT/yr,0.0,
5,yr,120.0,F,,psi,0,MMT/yr,0.0,
50,yr,120.0,F,,psi,0,MMT/yr,0.0,
```

B.21 Output Control Card

Card Title[a] { ~Output [Control Card] }
Format: Char[a]

Number of Reference Nodes[a],
Format: Integer[a],

For: Number of Reference Nodes
 I Index[a], J Index[b], K Index[c],
Endfor: Number of Reference Nodes
Format: Integer [a], Integer[b], Integer[c],

Reference Node Screen Output Frequency [a],
Reference Node Output File Frequency [b],
Output Time Units[c] (s),
Output Length Units[d] (m),
Screen Significant Digits[e],
Output File Significant Digits[f],
Plot File Significant Digits[g]
Format: Integer[a], Integer[b], Char[c], Char[d], Integer[e], Integer[f], Integer[g],

Number of Reference Node Variables [a],
Format: Integer[a],

For: Number of Reference Node Variables
 Reference Node Variable Option [a], Reference Node Variable Units [b],
 Format: Real[a], Char[b],
Endfor: Number of Reference Node Variables

Note: Refer to following pages for Reference Node Variable Options and Units.

Number of Plot File Times[a]
Format: Integer[a],

For: Number of Plot File Times
 Plot File Output Time [a], Units[b] (s)
 Format: Real[a], Char[b],
Endfor: Number of Plot File Times

Number of Plot File Variables[a]
Format: Integer[a],

For: Number of Plot File Variables
 Plot File Variable Option [a], Plot File Variable Units [b],
 Format: Char[a], Char[b],
Endfor: Number of Plot File Variables

Note: Refer to the following pages for Plot File Variable Options and Units.

Endcard: Output Control Card

Reference Node Variable and Plot File Variable Options

If: Operational Mode Option = { **STOMP-CO2** }

{ apparent aqueous saturation | aqueous courant [number] |
aqueous density | aqueous fracture |
aqueous gauge pressure | aqueous hydraulic head |
aqueous matrix | aqueous moisture cont | aqueous pressure |
aqueous relative perm[eability] | aqueous salt conc[entration] |
aqueous salt mass frac[tion] | aqueous saturation | aqueous viscosity |
CO2 Aqueous Diff[usion coefficient] CO2 aqueous conc[entration] |
CO2 aqueous mass frac[tion] | CO2 aqueous mole frac[tion] | CO2 gas conc[entration] |
CO2 gas mass frac[tion] | CO2 gas mole frac[tion] | CO2 mass source int[egral]* |
CO2 mass source rate | coupled well mass nodal CO2 rate |
coupled well mass nodal water rate | coupled well press[ure] |
coupled-well CO2 mass rate | coupled-well CO2 mass int[egral] |
coupled-well water mass rate | coupled-well water mass integral |
apparent aqueous saturation | diffusive porosity |
effective trapped gas saturation | final restart** | gas courant [number] |
gas density | gas fracture saturation | gas gauge pressure |
gas hydraulic head | gas matrix saturation | gas pressure |
gas relative perm[eability] | gas saturation | gas viscosity |
gas-aqueous scaling [factor] | integrated CO2 mass* |
integrated CO2 mass sour[ce] * | integrated aqueous CO2 [mass] * |
integrated aqueous water [mass] * | integrated gas CO2 [mass] * |
integrated gas water [mass] * | integrated water mass* |
integrated water mass sour[ce] * | integrated precipitated salt mass * |
mean eff[ective] stress | no restart** | node number | osmotic eff |
osmotic pressure | phase condition |
rock/soil type | salt aqueous conc[entration] | salt aqueous mass fraction |
salt aqueous mole frac[tion] | salt conc[entration] | salt mass source int[egral] |
salt mass source rate * | salt saturation | similitude variable |
[solute | species] aqueous conc[entration] | solute aqueous mole frac[tion] |
[solute | species] gas conc[entration] | solute [solute | species] integrated mass * | solute gas mole frac[tion] | [solute | species] source int[egral] |
[solute | species] volumetric conc[entration] | source-well pres[sure] | temperature | total CO2 mass |
total salt mass | total water mass | trapped gas sat[uration]|
vert[ical-]equil[ibrium gas-aqueous] inter[face] eleva[tion] |
vert[ical-]equil[ibrium] gas press[ure] | ver[tical-]equil[ibrium] aqu[eous] press[ure] |
vert[ical-]equil[ibrium] gas sat[uration] |
vert[ical-]equil[ibrium] trap[ped] gas sat[uration] |
vert[ical-]equil[ibrium] aqu[eous] sat[uration] |
vert[ical-]equil[ibrium] gas rel[ative] perm[eability] |
vert[ical-]equili[brium] aqu[eous] rel[ative] perm[eability] |
vert[ically-]int[egrated] aqu[eous] CO2 mass |
vert[ically-]int[egrated] aqu[eous] CO2 mass [per] area |
vert[ically-]int[egrated] CO2 mass | vert[ically-]int[egrated] CO2 mass [per] area |
verti[cally-]int[egrated] gas CO2 mass |
vert[ically-]int[egrated] gas CO2 mass [per] area |
water aqueous conc[entration] | water gas conc[entration] | water aqueous mass frac[tion] | water gas diff[usion coefficient] |
water gas mass frac[tion] | water gas mole frac[tion] | water mass source int[egral* | water mass source

rate | water vapor part[ial] press[ure] | well-node pres[sure] * |
x aqueous vol[umetric flux] | x displ[acement] | x gas vol[umetric flux] |
x intrinsic perm[eability] | x normal strain |
x salt flux | x solute flux | xnc aqueous vol[umetric flux (node centered)] |
xnc gas vol[umetric flux (node centered)] | xnc salt flux (node centered) |
y aqueous vol[umetric flux] | y displ[acement] | y gas vol[umetric flux] |
y intrinsic perm[eability] | y normal strain | y salt flux | y solute flux |
ync aqueous vol[umetric flux (node centered)] |
ync gas vol[umetric flux (node centered)] | ync salt flux (node centered) |
z aqueous vol[umetric flux] | z displ[acement] | z gas vol[umetric flux] |
z intrinsic perm[eability] | z normal strain | z salt flux |z solute flux |
znc aqueous vol[umetric flux (node centered)] |
znc gas vol[umetric flux (node centered)] | znc salt flux (node centered) }

* Reference Node Variable Only

Elseif: Operational Mode Option = { **STOMP-CO2e** }
{ apparent aqueous saturation | aqueous courant [number] | aqueous density | aqueous enthalpy |
aqueous fracture | aqueous gauge pressure |
aqueous hydraulic head | aqueous internal energy | aqueous matrix |
aqueous moisture cont | aqueous pressure | aqueous relative perm[eability] |
aqueous salt conc[entration] | aqueous salt mass frac[tion] | aqueous saturation | aqueous thermal conduc[tivity] | aqueous viscosity |
CO2 aqueous diff[usion coefficient] | CO2 aqueous conc[entration] |
CO2 aqueous mass frac[tion] | CO2 aqueous mole frac[tion] | CO2 gas conc[entration] |
CO2 gas mass frac[tion] | CO2 gas mole frac[tion] | CO2 mass source int[egral] |
CO2 mass source rate | coupled well mass nodal CO2 rate |
coupled well mass nodal water rate | coupled well press[ure] |
coupled-well CO2 mass rate | coupled-well CO2 mass int[egral] |
coupled-well water mass rate | coupled-well water mass integral | diffusive porosity | mean eff[ective] stress | effective trapped gas saturation | | final restart** |
gas courant [number] | gas density | gas enthalpy |
gas fracture saturation | gas gauge pressure | gas hydraulic head |
gas internal energy | gas matrix saturation | gas pressure | gas relative perm[eability] | gas saturation |
gas thermal conduc[tivity] | gas viscosity |
gas-aqueous scaling [factor] | integrated CO2 mass* | integrated aqueous CO2 [mass] * |
integrated aqueous water [mass] * | integrated CO2 mass sour[ce] * |
integrated gas CO2 [mass] * | integrated gas water [mass] * |
integrated water mass sour[ce] * | integrated water mass* |
integrated precipitated salt mass * | no restart** | node number | osmotic eff | osmotic pressure |
phase condition | saturated CO_2 aqueous mass fraction | rock/soil type | salt aqueous conc[entration] |
salt aqueous mass fraction | salt aqueous mole frac[tion] | salt conc[entration] | salt mass source int[egral] | salt mass source rate * |
salt saturation | similitude variable |
[solute | species] aqueous conc[entration] | solute aqueous mole frac[tion] |
[solute | species] gas conc[entration] | [solute | species] integrated mass * |
solute gas mole frac[tion] | [solute | species] source int[egral] |
[solute | species] volumetric conc[entration] | source-well pres[sure] | temperature | total CO2 mass |
total salt mass | total water mass | trapped gas sat[uration] |
vert[ical-]equil[ibrium gas-aqueous] inter[face] eleva[tion] |
vert[ical-]equil[ibrium] gas press[ure] | ver[tical-]equil[ibrium] aqu[eous] press[ure] |
vert[ical-]equil[ibrium] gas sat[uration] |

vert[ical-]equil[ibrium] trap[ped] gas sat[uration] |
vert[ical-]equil[ibrium] aqu[eous] sat[uration] |
vert[ical-]equil[ibrium] gas rel[ative] perm[eability] |
vert[ical-]equili[brium] aqu[eous] rel[ative] perm[eability] |
vert[ically-]int[egrated] aqu[eous] CO2 mass |
vert[ically-]int[egrated] aqu[eous] CO2 mass [per] area |
vert[ically-]int[egrated] CO2 mass | vert[ically-]int[egrated] CO2 mass [per] area |
verti[cally-]int[egrated] gas CO2 mass |
vert[ically-]int[egrated] gas CO2 mass [per] area |
water aqueous conc[entration] | water aqueous mass frac[tion] |
water gas conc[entration] | water gas diff[usion coefficient] | water gas mass frac[tion] |
water gas mole frac[tion] | water mass source int[egral | water mass source rate |
water vapor part[ial] press[ure] | well-node pres[sure] * | x aqueous vol[umetric flux] |
x displ[acement] | x gas vol[umetric flux] | x intrinsic perm[eability] | x normal strain |
x salt flux | x solute flux | x thermal cond[uctivity] |
xnc aqueous vol[umetric flux (node centered)] |
xnc gas vol[umetric flux (node centered)] |xnc salt flux (node centered) | y aqueous vol[umetric flux] | y displ[acement] |
y gas vol[umetric flux] | y intrinsic perm[eability] | y normal strain | y salt flux |
y solute flux | y thermal cond[uctivity] |
ync aqueous vol[umetric flux (node centered)] |
ync gas vol[umetric flux (node centered)] | ync salt flux (node centered) |
z aqueous vol[umetric flux] | z displ[acement] |
z gas vol[umetric flux] | z intrinsic perm[eability] | z normal strain | z salt flux |
z solute flux | z thermal cond[uctivity] | znc aqueous vol[umetric flux (node centered)] |
znc gas vol[umetric flux (node centered)] | znc salt flux (node centered) }

* Reference Node Variable Only
** Plot File Variable Only

B.21.1 Output Control Card Examples

```
~Output Options Card
6,
1,1,1,
1,1,10,
1,1,20,
20,21,1,
20,21,10,
20,21,20,
1,1,yr,m,6,6,6,
18,
Gas Saturation,,
Temperature,,
Integrated CO2 Mass,MMT,
Integrated CO2 Aqueous,MMT,
Integrated CO2 Gas,MMT,
Aqueous Relative Permeability,,
Gas Relative Permeability,,
Salt Aqueous Mass Fraction,,
Salt Saturation,,
CO2 Aqueous Mass Fraction,,
Gas Pressure,psi,
Gas Density,,
Aqueous Density,kg/m^3,
Aqueous Pressure,psi,
Diffusive Porosity,,
Well-Node Top Press,psi,
aqueous hydraulic head,,
Integrated Trapped CO2 Gas,kg,
11,
0,yr,
10,yr,
20,yr,
30,yr,
40,yr,
50,yr,
60,yr,
70,yr,
80,yr,
90,yr,
100,yr,
18,
Rock/Soil Type,,
Gas Saturation,,
Salt Saturation,,
Temperature,,
CO2 Aqueous Concentration,gm/cm^3,
Salt Aqueous Mass Fraction,,
CO2 Aqueous Mass Fraction,,
Gas Pressure,psi,
Aqueous Pressure,psi,
Diffusive Porosity,,
Gas Density,kg/m^3,
Aqueous Density,kg/m^3,
total CO2 mass,,
total water mass,,
CO2 gas mass frac,,
water aqueous mass frac,,
water gas mass frac,,
effective trapped gas,,
```

B.21.2 Output Control Card Examples

```
~Output Options Card
4,
3,1,3,
3,1,4,
4,1,3,
98,1,3,
1,1,hr,cm,6,6,6,
5,
Gas Saturation,,
Temperature,C,
Salt Aqueous Mass Fraction,,
CO2 Aqueous Mass Fraction,,
Gas Pressure,Bar,
7,
0.5,min,
1.0,min,
2.0,min,
5.0,min,
10.0,min,
20.0,min,
50.0,min,
6,
Gas Saturation,,
Temperature,C,
Salt Aqueous Mass Fraction,,
CO2 Aqueous Mass Fraction,,
Gas Pressure,Bar,
Aqueous Density,kg/m^3,
```

B.22 Surface Flux Card

Card Title^a { ~Surface [Flux Card] }
Format: Char^a

Number of Surface Flux Inputs^a
Format: Integer^a,

Note: The number of surface flux inputs may be written to one or more user specified output files. For <u>each</u> user specified file, the following input line has to be specified. An example has been included in the Surface Flux Card Examples section (second example). The sum of the "Number of Surface Flux Inputs in File" has to be equal to the "Number of Surface Flux Inputs".

Number of Surface Flux Inputs in File^a, Filename^b,
Format: Integer^a, Char^b,

```
For: Number of Surface Flux Inputs:

    If: Operation Mode Option = { STOMP-CO2 }
        Surface Flux Type Option$^a$,
            {
                Aqueous Volumetric | Aqueous Mass | Gas Volumetric |
                Gas Mass | Salt Mass | CO2-Mass |
                Aqueous-Phase CO2 Mass | Gas-Phase CO2 Mass |
                Solute, Solute Name |
                If: Operational Mode Option contains { { ECKEChem }}
                        Conservation Component, Species Name |
                        Kinetic Component, Species Name
                Endif:
            }

    Elseif: Operation Mode Option = { STOMP-CO2e }
        Surface Flux Type Option$^a$,
            { Aqueous Volumetric | Aqueous Mass | Gas Volumetric |
            Gas Mass | Salt Mass | CO2-Mass |
            Aqueous-Phase CO2 Mass | Gas-Phase CO2 Mass |
            Heat | Solute, Solute Name }
    Endif:

    If: Surface Flux Type Option = { Heat }
        Units$^b$ (W), Units$^c$ (J),
    ElseIf: Surface Flux Type Option = { Volumetric }
        Units$^b$ (m^3/s), Units$^c$ (m^3),
    Elseif: Surface Flux Type Option = { Mass }
        Units$^b$ (kg/s), Units$^c$ (kg),
    Elseif: Surface Flux Type Option = { Solute }
        Units$^b$ (sol/s), Units$^c$ (sol),
    ElseIf: Surface Flux Type Option = { Component }
        Units$^b$ (mol/s), Units$^c$ (mol),
    Endif:
```

Surface Flux Orientation Option[d]
{ West } { East } { South } { North } { Top } { Bottom }
I-Start Index[e], I-End Index[f],
J-Start Index[g], J-End Index[h],
K-Start Index[i], K-End Index[j],
Format: Char[a], Char[b], Char[c], Char[d], Integer[e], Integer[f], Integer[g], Integer[h], Integer[i], Integer[j],

Endfor: Number of Surface Flux Inputs

Endcard: Surface Flux Card

B.22.1 Surface Flux Card Examples

```
~Surface Flux Card
12,
Aqueous Volumetric Flux,gal/min,gal,east,21,21,21,21,1,20,
Aqueous Volumetric Flux,gal/min,gal,north,21,21,21,21,1,20,
Aqueous Volumetric Flux,gal/min,gal,west,21,21,21,21,1,20,
Aqueous Volumetric Flux,gal/min,gal,south,21,21,21,21,1,20,
Aqueous-Phase CO2 Mass,kg/s,kg,east,21,21,21,21,1,20,
Aqueous-Phase CO2 Mass,kg/s,kg,north,21,21,21,21,1,20,
Aqueous-Phase CO2 Mass,kg/s,kg,west,21,21,21,21,1,20,
Aqueous-Phase CO2 Mass,kg/s,kg,south,21,21,21,21,1,20,
CO2 Mass Flux,kg/s,kg,east,1,1,1,1,1,20,
CO2 Mass Flux,kg/s,kg,north,1,1,1,1,1,20,
CO2 Mass Flux,kg/s,kg,west,1,1,1,1,1,20,
CO2 Mass Flux,kg/s,kg,south,1,1,1,1,1,20,
```

B.22.2 Surface Flux Card Examples

```
~Surface Flux Card
15,
CO2 Mass Flux,MT/yr,MT,top,1,80,1,78,51,51,
CO2 Mass Flux,MT/yr,MT,west,1,1,1,78,1,51,
CO2 Mass Flux,MT/yr,MT,east,80,80,1,78,1,51,
CO2 Mass Flux,MT/yr,MT,south,1,80,1,1,1,51,
CO2 Mass Flux,MT/yr,MT,north,1,80,78,78,1,51,
Aqueous Mass Flux,MT/yr,MT,top,1,80,1,78,51,51,
Aqueous Mass Flux,MT/yr,MT,west,1,1,1,78,1,51,
Aqueous Mass Flux,MT/yr,MT,east,80,80,1,78,1,51,
Aqueous Mass Flux,MT/yr,MT,south,1,80,1,1,1,51,
Aqueous Mass Flux,MT/yr,MT,north,1,80,78,78,1,51,
Salt Mass Flux,MT/yr,MT,top,1,80,1,78,51,51,
Salt Mass Flux,MT/yr,MT,west,1,1,1,78,1,51,
Salt Mass Flux,MT/yr,MT,east,80,80,1,78,1,51,
Salt Mass Flux,MT/yr,MT,south,1,80,1,1,1,51,
Salt Mass Flux,MT/yr,MT,north,1,80,78,78,1,51,
```

B.23 Geomechanical Properties Card

Card Title[a] { ~Geomechanical Properties Card }
Format: Char[a]

Rock/Soil or Scaling Group Name[a],
Format: Char[a]

If: Rock/Soil or Scaling Group Name = { IJK | JKI | KIJ } Indexing

Note: Parameter input can be replaced with an external file using the following formatting for ASCII files:

file: filename

or the following formattings for binary files:

binary file: filename

where; the external file will contain unique parameter values for each node (active or inactive) arranged according to the indexing scheme (i.e., IJK, JKI, or KIJ). Applicable units will be applied to all parameter values in the external file.

Elseif:

For: Number of Rock/Soil or Scaling-Group Types
Young's Modulus (Drained Bulk)[a], Units[b],
Poisson's ratio (Drained Bulk)[c],
Biot Coefficient[d],
Thermal expansion Coefficient, Units[f],
Format: Real[a], Char[b], Real[c], Real[d], Real[e], Char[f]
Endfor:

Endif:

Endcard: Geomechanical Properties Card

B.23.1 Geomechanical Properties Card Example

```
~Geomechanical Properties Card
Aquifer,1.85e-06,Pa,0.35,0.43,2.9e-06,1/C,
Fault,3.77e-06,Pa,0.55,0.56,5.7e-06,1/C,
```

B.23.2 Geomechanical Properties Card Example

```
~Geomechanical Properties Card
IJK Indexing,file:mod.dat,Pa,file:poisR.dat,file:bCoef.dat,
file:thermExp.dat1/C,
```

B.24 Geomechanical Link Card

```
Card Title[a] { ~Geomechanical Properties Card }
Format: Char[a]

Rock/Soil or Scaling Group Name [a],
Format: Char[a]

If: Rock/Soil or Scaling Group Name = { IJK | JKI | KIJ } Indexing

   Note: Parameter input can be replaced with an external file using the
   following formatting for ASCII files:

      file: filename

   or the following formattings for binary files:

      binary file: filename

   where; the external file will contain unique parameter values for each node
   (active or inactive) arranged according to the indexing scheme (i.e., IJK, JKI, or KIJ).
   Applicable units will be applied to all parameter values in the external file.

Elseif:

   For: Number of Rock/Soil or Scaling-Group Types

      Porosity-Mean Stress Function Type[a],
      { Davis[-Davis] | [none | n/a | null | zero ] }
      Intrinsic Permeability-Porosity Function Type[b],
      { Davis[-Davis] | [none | n/a | null | zero ] }
      Capillary Presssure-Permeability/Porosity Function Type[c],
      { Leverett | [none | n/a | null | zero ] }

      If:  Porosity-Mean Stress Function Type = { [none | n/a | null | zero ]  }
       If:  Intrinsic Permeability-Porosity Function Type = { [none | n/a | null | zero ]  }
         Format: Char[a], Char[b], Char[c],
          Elseif:  : Intrinsic Permeability-Porosity Function Type = { Davis[-Davis] }
            Davis-Davis Intrinsic Permeability-Porosity Function Exponent[f],
            Format: Char[a], Char[b], Char[c],Real[f],
         Endif:
      Elseif:  Porosity-Mean Stress Function Type = { Davis[-Davis] }
         Davis-Davis Porosity-Mean Stress Function Exponent[d],
         Residual Porosity at High Stress[e],


        If:  Intrinsic Permeability-Porosity Function Type = { [none | n/a | null | zero ]  }
         Format: Char[a], Char[b], Char[c],Real[d],Real[e],
        Elseif:  Intrinsic Permeability-Porosity Function Type = { Davis[-Davis] }
            Davis-Davis Intrinsic Permeability-Porosity Function Exponent[f],
           Format: Char[a], Char[b], Char[c], Real[d],Real[e],Real[f],
            Endif:
        Endif:

    Endfor:

Endcard: Geomechanical Link Card
```

B.24.1 Geomechanical Link Card Example

```
~Geomechanical Link Card
Aquifer,,,,
Fault,,,,
```

B.24.2 Geomechanical Link Card Example

```
~Geomechanical Link Card
IJK Indexing,Davis-Davis,Davis-Davis,null,file:stressExp.dat,
file:residPor.dat,file:exp.dat,
```

B.24.3 Geomechanical Link Card Example

```
~Geomechanics Link Card
cap,Davis-Davis,Davis-Davis,zero,1.e-8,0.009,22.22,
aquifer,Davis-Davis,Davis-Davis,zero,1.e-8,0.09,22.22,
```

B.25 Geomechanical Boundary Conditions Card

```
Card Title[a] { ~Geomechanical Boundary Conditions Card }
Format: Char[a]

Number of Geomechanical Boundary Conditions[a],
Format: Integer[a],

For: Number of Geomechanical Boundary Conditions
   Geomechanical Boundary Condition Orientation[a],
   { West | East | South | North | Bottom | Top | File }
   If: Geomechanical Boundary Condition Orientation = { File }
      File Name[b],
      Format: Char[a], Char[b],
  Else:
      X-Direction Traction Boundary Condition Type[b],
      { [ Zero | No ] | [ Traction | Prescribed ] }
      Y-Direction Traction Boundary Condition Type[c],
      { [ Zero | No ] | [ Traction | Prescribed ] }
      Z-Direction Traction Boundary Condition Type[d],
      { [ Zero | No ] | [ Traction | Prescribed ] }
      Normal Displacement Boundary Condition Type[e],
      { [ Zero | No | Fixed ] | [ Free | Yes ] }
      Shear Displacement Boundary Condition Type[f],
      { [ Zero | No | Fixed ] | [ Free | Yes ] }
      Format: Char[a], Char[b], Char[c], Char[d], Char[e], Char[f],
   Endif:

   I-Index Start[a], I-Index End[b],
   J-Index Start[c], J-Index End[d],
   K-Index Start[e], K-Index End[f],
   Format: Integer[a], Integer[b], Integer[c], Integer[d], Integer[e], Integer[f],

   Number of Geomechanical Boundary Condition Times[a],
   Format: Integer[a],

   For: Number of Geomechanical Boundary Condition Times
      Start Time[a], Units[b],
      X-Direction Stress[c], Units[d],
      Y-Direction Stress[e], Units[f],
      Z-Direction Stress[g], Units[h],
      Format: Real[a], Char[b], Real[c], Char[d], Real[e], Char[f], Real[g], Char[h],
   Endfor:

Endfor:

Endcard: Geomechanical Boundary Conditions Card
```

B.25.1 Geomechanical Boundary Conditions Card Example

```
~Geomechanical Boundary Conditions Card
4,
Bottom,zero,zero,zero,zero,free,
1,1,1,1,1,1,1,
0,s,
East,zero,zero,zero,zero,free,
1,1,1,1,1,300,1,
0,s,
West,zero,zero,zero,zero,free,
1,1,1,1,1,300,1,
0,s,
Top,traction,zero,zero,free,free,
1,1,1,1,300,300,1,
0,s,-101325,Pa,
```

Pacific Northwest
NATIONAL LABORATORY

U.S. DEPARTMENT OF
ENERGY

902 Battelle Boulevard
P.O. Box 999
Richland, WA 99352
1-888-375-PNNL (7665)
www.pnl.gov