

**BREATH VERSION 1.0—COUPLED FLOW AND  
ENERGY TRANSPORT IN POROUS MEDIA  
SIMULATOR DESCRIPTION AND USER GUIDE**

*Prepared for*

**Nuclear Regulatory Commission  
Contract NRC-02-93-005**

*Prepared by*

**Center for Nuclear Waste Regulatory Analyses  
San Antonio, Texas**

**September 1994**



**BREATH VERSION 1.0 — COUPLED FLOW AND  
ENERGY TRANSPORT IN POROUS MEDIA  
SIMULATOR DESCRIPTION AND USER GUIDE**

*Prepared for*

**Nuclear Regulatory Commission  
Contract NRC-02-93-005**

*Prepared by*

**S. Stothoff**

**Center for Nuclear Waste Regulatory Analyses  
San Antonio, Texas**

**September 1994**

**RECEIVED**  
SEP 26 1994  
CNWRA-V.

## ABSTRACT

This document describes the BREATH code, including the mathematical and numerical formulation for the simulator, usage description, and sample input files with corresponding output files. The BREATH computer code is designed to simulate one-dimensional flow of a liquid phase and dispersive transport of the corresponding vapor species, coupled with energy transfer, in a heterogeneous porous medium.

The BREATH simulator has been developed for use in auxiliary analyses which are a part of the Nuclear Regulatory Commission's Iterative Performance Assessment program. The simulator was developed in response to the observation from Total System Performance Assessments by both the Nuclear Regulatory Commission and the Department of Energy that total-system performance at the Yucca Mountain site in Nevada is highly sensitive to the infiltration rate. Accordingly, this first version of the code is primarily intended to simulate processes important to infiltration and evaporation in climatic and hydrologic near-surface environments representative of the Yucca Mountain site.

The simulation model assumes that there is an immobile solid phase, a mobile liquid phase, and an optional infinitely mobile gas phase. The liquid may have an associated vapor species, assumed to be in equilibrium with the liquid phase. The vapor phase may only move via diffusion within the gas phase. Energy may be transported in the form of enthalpy, thermal conduction, and latent heat. The temperature range is assumed to be between 0 and 100 °C.

Available boundary conditions include six liquid-phase conditions, four vapor-species conditions, and three energy conditions, all of which may be applied independently to either end of the domain. Meteorological conditions may also be input, thereby providing additional control over boundary fluxes. Boundary conditions may be updated as often as desired. The code is designed to act as a filter between input-generation programs and output-interpretation programs, thus flexibility in input and output is emphasized; although there are defaults for each variable, every parameter can be altered from the input file. Using a simple command language, several options are provided to input variables, and every variable can be output in a user-specified format. In addition, dependent variables and fluxes can be output at periodic intervals for selected nodes and elements, and for all nodes or elements simultaneously.

The BREATH code is organized into a flow equation simulator and an energy equation simulator, which can be coupled or used independently. Both simulators use linear finite element basis functions. A modified Picard iteration scheme is used to solve the nonlinear sets of equations. Heuristic algorithms are available to control time stepping and the active solution domain.

# CONTENTS

Section	Page
FIGURES .....	v
TABLES .....	vi
NOMENCLATURE .....	vii
ACKNOWLEDGMENTS .....	ix
1 BACKGROUND .....	1-1
1.1 REGULATORY AND TECHNICAL ASPECTS .....	1-1
1.2 PURPOSE OF SOFTWARE .....	1-2
1.3 REPORT CONTENT .....	1-2
2 SIMULATOR OVERVIEW .....	2-1
2.1 INTRODUCTION .....	2-1
2.2 MODEL ASSUMPTIONS .....	2-2
2.3 BOUNDARY CONDITIONS .....	2-3
2.4 SOLUTION PROCEDURE .....	2-3
3 MATHEMATICAL MODEL .....	3-1
3.1 CONSERVATION OF MASS .....	3-1
3.2 CONSERVATION OF ENERGY .....	3-3
3.3 CONSTITUTIVE RELATIONSHIPS .....	3-6
3.3.1 Liquid/Soil Properties .....	3-7
3.3.1.1 Moisture Content/Capillary Pressure .....	3-7
3.3.1.2 Moisture Content/Mobility .....	3-8
3.3.1.3 Single-Phase Flow .....	3-8
3.3.2 Thermal Properties .....	3-9
3.3.3 Fluid Properties .....	3-9
4 NUMERICAL MODEL .....	4-1
4.1 FINITE ELEMENT FORMULATION .....	4-1
4.2 CONSERVATION EQUATIONS .....	4-2
4.2.1 Conservation of Mass .....	4-2
4.2.2 Conservation of Energy .....	4-3
4.3 TIME STEPPING AND ACTIVE REGION CONTROL .....	4-4
4.3.1 Constant and Courant-Based Time Stepping .....	4-4
4.3.2 Heuristic Convergence-Based Time Stepping .....	4-4
4.3.3 Active Region Control .....	4-5
4.4 BOUNDARY CONDITIONS .....	4-5
4.4.1 Liquid Phase Boundary Conditions .....	4-6
4.4.2 Vapor Species Boundary Conditions .....	4-7
4.4.3 Energy Equation Boundary Conditions .....	4-7
4.4.4 Meteorological Boundary Conditions .....	4-8

## CONTENTS (CONT'D)

Section	Page
5	VERIFICATION AND BENCHMARKING..... 5-1
5.1	SOIL MOISTURE REDISTRIBUTION..... 5-1
5.2	SINUSOIDAL SOIL TEMPERATURE REDISTRIBUTION..... 5-1
5.3	ADVANCING TEMPERATURE FRONT..... 5-3
5.4	COUPLED SOIL MOISTURE/SOIL TEMPERATURE..... 5-5
6	CURRENT STATUS OF THE BREATH CODE ..... 6-1
7	REFERENCES..... 7-1
APPENDICES	
A	DATA INPUT
B	EXAMPLE INPUT AND OUTPUT LISTINGS

## FIGURES

Figure		Page
5-1	Pressure distribution at selected time instants, in the redistribution example . . . . .	5-2
5-2	Moisture content distribution at selected time instants, in the redistribution example . . . . .	5-2
5-3	Temperature trace at selected depths, in the sinusoidal-forcing example . . . . .	5-4
5-4	Temperature distribution at selected times, in the advancing-front example. . . . .	5-5
5-5	Response of moisture content to a 1 hr rainfall event, in the coupled example. . . . .	5-8
5-6	Temperature profile at selected times in the sixth day, in the coupled example . . . . .	5-8

## TABLES

Table		Page
4-1	Liquid-phase boundary conditions .....	4-6
4-2	Vapor-species boundary conditions .....	4-7
4-3	Energy equation boundary conditions .....	4-7
5-1	Uniform phase properties in the coupled example .....	5-6
5-2	Uniform layer properties in the coupled example .....	5-6
5-3	Miscellaneous constants in the coupled example .....	5-6
A-1	List of BREATH commands .....	A-1
A-2	The <b>set</b> command .....	A-2
A-3	The <b>file</b> command .....	A-2
A-4	Example input file segment .....	A-3
A-5	The <b>met</b> command .....	A-4
A-6	The <b>calc</b> command .....	A-5
A-7	The <b>echo</b> command .....	A-5
A-8	The <b>trace</b> and <b>snap</b> commands .....	A-6
A-9	The <b>mass</b> command .....	A-7
A-10	Numeric variable names and descriptions .....	A-8
A-11	Character variable names .....	A-12

# NOMENCLATURE

## Units

J	Energy [ML <sup>2</sup> T <sup>-2</sup> ]
L	Length
M	Mass
K	Temperature
T	Time

## Greek symbols

$\nabla$	= Gradient operator
$\Delta$	= Difference operator
$\partial$	= Partial derivative operator
$\alpha$	= van Genuchten shape parameter [L <sup>-1</sup> ]
$\alpha_s$	= Ground surface albedo [-]
$\lambda$	= Mobility (relative permeability divided by viscosity) [LTM <sup>-1</sup> ]
$\rho$	= Density [ML <sup>-3</sup> ]
$\psi$	= Matric head [L]
$\mu$	= Dynamic viscosity [ML <sup>-1</sup> T <sup>-1</sup> ]
$\theta$	= Volumetric fraction of a phase [L <sup>3</sup> L <sup>-3</sup> ]
$\theta_e$	= Effective saturation [-]
$\theta_{wr}$	= Residual moisture content [L <sup>3</sup> L <sup>-3</sup> ]
$\theta_{ws}$	= Saturated moisture content [L <sup>3</sup> L <sup>-3</sup> ]
$\varepsilon$	= Emissivity [-]
$\sigma$	= Stefan-Boltzmann constant [JT <sup>-1</sup> L <sup>-2</sup> K <sup>-4</sup> ]
$\tau$	= Tortuosity factor in vapor equations [-]

## Numeric variables

$A$	= Generic quantity
$B$	= Boundary condition value
$C_1$	= Coefficient multiplying $T - T_0$ in time term of the energy equation
$C_2$	= Coefficient independent of $T - T_0$ in time term of the energy equation
$Co$	= Courant number [-]
$D_a$	= Diffusivity of vapor in air [L <sup>2</sup> T <sup>-1</sup> ]
$E$	= Energy [L <sup>2</sup> T <sup>-1</sup> ]
$E_{ext}$	= External sources of energy
$H$	= Enthalpy [J]
$H_l$	= Latent heat [J]
$J(\bullet)$	= Dispersive flux of quantity ( $\bullet$ )
$J_e(\bullet)$	= Dispersive flux of energy
$K_e$	= Bulk thermal conductivity [ML <sup>-1</sup> T <sup>-1</sup> ]
$K_{sat}$	= Saturated hydraulic conductivity [LT <sup>-1</sup> ]
$M_1$	= Coefficient multiplying $T - T_0$ in space term of the energy equation
$M_2$	= Coefficient independent of $T - T_0$ in space term of the energy equation
$M_{ext}$	= External sources of mass
$P$	= Pressure [ML <sup>-1</sup> T <sup>-2</sup> ]
$P_0$	= van Genuchten shape factor [ML <sup>-1</sup> T <sup>-2</sup> ]
$P_c$	= Capillary pressure [ML <sup>-1</sup> T <sup>-2</sup> ]



$R$  = Ideal gas constant [ $L^2M \text{ mole}^{-1} K^{-1}T^{-2}$ ]  
 $S_s$  = Specific storage coefficient [ $L^{-1}$ ]  
 $S_t$  = Shortwave radiation [ $JT^{-1}L^{-2}$ ]  
 $T$  = Temperature [K]  
 $T_0$  = Reference temperature [K]  
 $U$  = Internal energy [J]  
 $U_l$  = Change in internal energy due to phase change [J]  
 $V$  = Velocity [ $LT^{-1}$ ]  
 $g$  = Acceleration due to gravity [ $LT^{-2}$ ]  
 $h$  = Suction head [L]  
 $k$  = von Karman constant [-]  
 $\mathbf{k}$  = Intrinsic permeability [ $L^2$ ]  
 $\mathbf{k}_e$  = Thermal conductivity [ $L^2T^{-1}$ ]  
 $k_h$  = Boundary layer conductivity for heat [ $L^2T^{-1}$ ]  
 $k_v$  = Boundary layer conductivity for vapor [ $L^2T^{-1}$ ]  
 $k_r$  = Relative permeability [-]  
 $m$  = van Genuchten shape parameter [-]  
 $n$  = van Genuchten shape parameter [-]  
 $\mathbf{q}$  = Darcy flux [ $LT^{-1}$ ]  
 $t$  = Time [T]  
 $\mathbf{v}$  = Velocity [ $LT^{-1}$ ]  
 $z$  = Elevation [L]

#### Subscripts/superscripts

$\alpha$  Phase  
 $a$  Atmosphere  
 $g$  Gas  
 $i$  Species  
 $s$  Solid or surface, in context  
 $w$  Water  
 $v$  Vapor

## **ACKNOWLEDGMENTS**

This report was prepared to document work performed by the Center for Nuclear Waste Regulatory Analyses (CNWRA) for the Nuclear Regulatory Commission (NRC) under Contract No. NRC-02-93-005. The activities reported here were performed on behalf of the NRC Office of Nuclear Regulatory Research, Division of Regulatory Applications. The report is an independent product of the CNWRA and does not necessarily reflect the views or regulatory position of the NRC.

The author wishes to thank R.G. Baca, A.C. Bagtzoglou, H.D. Nguyen and B. Sagar for reviewing this document and providing helpful comments.

## **QUALITY OF DATA, ANALYSES, AND CODE DEVELOPMENT**

**Data:** There is no CNWRA-generated original data contained in this report. Sources for other data should be consulted for determining the level of quality for those data.

**Analyses and Codes:** The configuration of the BREATH code is controlled under the CNWRA Technical Operating Procedure TOP-018. This User Guide satisfies one requirement of the Software Configuration Procedure.

# 1 BACKGROUND

## 1.1 REGULATORY AND TECHNICAL ASPECTS

The primary regulations applicable to the high-level waste geological repository were promulgated by the Nuclear Regulatory Commission (NRC) in 10 CFR Part 60—Disposal of High-Level Radioactive Wastes in Geologic Repositories. Two sections of 10 CFR Part 60 pertain specifically to post-closure performance. These sections include Part 60.112—Overall System Performance Objective for the Geologic Repository After Permanent Closure, and Part 60.113—Performance of Particular Barriers After Permanent Closure. Part 60.112 makes reference to satisfying the generally applicable environmental standards for radioactivity established by the Environmental Protection Agency (EPA). These environmental standards referred to in Part 60.112 were promulgated by the EPA in 40 CFR Part 191, in 1985. However, on litigation, certain provisions of these standards were remanded by a federal court. In late 1992, the U.S. Congress enacted a new law known as the Energy Policy Act according to which the EPA will develop standards applicable specifically to Yucca Mountain that may be different from those in 40 CFR Part 191. A revised 40 CFR Part 191, applicable to repositories other than Yucca Mountain, was issued in December, 1993.

Three different performance measures are used in Part 191. These measures are: (i) release of radioactivity over the entire accessible environment boundary (integrated over areal space), cumulative over a 10,000-yr period (integrated over time) after closure, must not exceed specific limits at specified probability levels (Part 191.13—Containment Requirements), where the preferred method of representing this performance measure is through a complementary cumulative (probability) distribution function; (ii) dose to humans in the first 10,000 years after repository closure must not exceed specified limits (Part 191.15—Individual Protection Requirements), with no probability attached to this requirement; and (iii) concentration of alpha-, beta-, and gamma-emitting radionuclides must not exceed specified limits (Part 191.16—Groundwater Protection Requirements), again with no probability attached to this requirement. While the first performance measure considers all credible future scenarios, the other two apply only to undisturbed performance.

In addition, three other performance measures are used in 10 CFR Part 60.113 to define performance of individual barriers, independently of total-system performance. These performance measures are: (i) life of the waste package must exceed specified limits [Part 60.113(a)(ii)(A)—Substantially Complete Containment Requirement]; (ii) release from engineered barriers must be less than specified limits [Part 60.113(a)(1)(ii)(B)—Groundwater Release Requirement]; and (iii) groundwater travel time must be greater than specified limits [Part 60.113(a)(2)—Groundwater Travel Time Requirement].

The conduct of performance assessment (PA) has several purposes, one of which is to aid in determining whether the geologic repository system satisfies regulatory requirements. This is done by comparing estimated values of regulatory performance measures, obtained during PA exercises, to the values of the same performance measures specified in the regulations. In addition to the regulatory function, PA will also be used to design the site characterization program, by the U.S. Department of Energy (DOE), and to judge the adequacy of the site characterization program, by the NRC. To meet these objectives, a Total-System Performance Assessment (TPA) code has been developed by the NRC and the Center for Nuclear Waste Regulatory Analyses (CNWRA) to provide computational algorithms for estimating values of various performance measures [see Sagar and Janetzke (1993) for the description of the TPA code].

In all, there are six distinct performance measures. In general, a TPA code must allow for estimation of the three measures related to 40 CFR Part 191 and preferably, but not necessarily, for the other three related to 10 CFR Part 60.113. In addition, PA modeling exercises using a TPA code can identify the model parameters that have the highest impact on estimated performance measures, thereby providing guidance on the processes and uncertain parameters that might be expected to be most critical to repository performance. The NRC TPA code has been used to examine the performance of the proposed geologic repository at Yucca Mountain, in southwest Nevada, with reference to all six performance measures. In both cases, infiltration was treated as a spatially uniform, temporally constant parameter in each simulation in the analyses (Nuclear Regulatory Commission, 1994). Nevertheless, performance measures in the exercises are found to be strongly affected by the infiltration parameter; infiltration is arguably the single most important parameter affecting system performance. The DOE has also found infiltration rate to be one of the most sensitive parameters in corresponding TPA analyses (Sandia National Laboratories, 1992; 1994), as have risk-based approaches to performance assessment [Electric Power Research Institute (EPRI), 1990; 1992].

## **1.2 PURPOSE OF SOFTWARE**

Because of its importance, the estimation of infiltration rate is being addressed in an auxiliary analysis in Phase 3 of the NRC Iterative Performance Assessment program. In the auxiliary analysis, a mechanistic model capable of simulating processes operating in the near-surface environment will be subjected to forcings representative of weather conditions at Yucca Mountain, for extended periods of time, with properties representative of various areas at Yucca Mountain. In addition, another project at the CNWRA, the Subregional Hydrogeologic Flow and Transport Processes Research Project, is also planning to examine infiltration processes using a mechanistic model. The BREATH code is intended for use in both of these projects.

## **1.3 REPORT CONTENT**

An overview of the BREATH simulator background and capabilities is provided in Chapter 2. A description of the mathematical models composing the simulator composes Chapter 3, with the numerical implementation of the models developed in Chapter 4. Representative examples of BREATH usage are presented in Chapter 5, including comparisons with analytical solutions and other simulators. In Chapter 6, additional modifications and comparisons are proposed. References are found in Chapter 7. A user's guide to the code is presented in Appendix A. Finally, input files and resultant output files representative of the examples in Chapter 5 are shown in Appendix B.

## 2 SIMULATOR OVERVIEW

### 2.1 INTRODUCTION

Literature on isothermal liquid transport dates back to the contributions of Henri Darcy in 1856 (Darcy, 1856). The development of theory describing the isothermal transport of moisture in unsaturated soils is reviewed in Swartzendruber (1969) and Philip (1957). Theory of isothermal moisture transport in unsaturated media is developed in the landmark paper by Richards (1931), while nonisothermal transport of moisture in porous media is considered in the landmark paper of Philip and de Vries (1957), with subsequent refinements of the theory in de Vries (1958). Representative predecessor work in the soil science literature includes the efforts of Gee (1966), Sasamori (1970), Jury (1973), Sophocleous (1978), Milly and Eagleson (1980), and Fayer et al. (1986).

Related work may be found in several disciplines, where the emphasis is on processes applicable away from the surface environment. The petroleum industry, for example, is concerned with the introduction of steam, hot water, and combustion fronts, in order to mobilize oil for more efficient extraction. Field application of thermal stimulation to oil reservoirs dates from at least 1865, and published simulations considering coupled fluid-flow and thermal effects date to the early 1960s (Prats, 1982; Jensen, 1992). A spate of geothermal simulation was performed in the 1970s, in response to the energy crisis, examining the possibility of extracting geothermal energy from the deep subsurface. More recently, the environmental restoration field has developed simulators capable of handling vapor stripping and steam extraction of non-aqueous phase liquids from the near-surface. Manteufel et al. (1993) review the current state of the art for coupled moisture and energy transport simulators. Simulators available to the DOE are analyzed in Reeves et al. (1994).

Although sophisticated simulators capable of handling strongly nonisothermal conditions are available, boundary conditions appropriate to the near-surface are generally not addressed well in such simulators. For example, only recently has an attempt being made to link such a simulator to atmospheric conditions using standard evapotranspiration models (Montazer et al., 1994). In addition, most deep subsurface simulations have forcings to the system which are much larger than are seen in near-surface environments, such as active wells, masking phenomena that the soil science community considers potentially important to the infiltration process.

In order to examine processes that might be important for investigating infiltration at the Yucca Mountain site, readily available simulators capable of detailed long-term (more than a century) one-dimensional simulations of the near-surface environment were investigated. Version 2.0 of the UNSAT-H simulator (Fayer et al., 1990) was deemed the most suitable for the desired purposes, with most of the necessary features, but the user interface was found to be awkward to use for simulations of greater than one year and the code was found to be difficult to modify and computationally slow. After examining the code closely, it was felt that it would be faster to develop a comparable simulator, based on the mathematical models in the UNSAT-H and more modern numerical models, than to retrofit the UNSAT-H code directly. Accordingly, this first version of BREATH was created, primarily based on the UNSAT-H documentation for the mathematical models and the UNSAT1D simulator (Celia et al., 1990; Celia, 1991) for the numerical approach and methods for accommodating saturated zones. The improved numerical algorithms in BREATH offered an improvement of nearly an order of magnitude in computational time for certain example problems, compared to UNSAT-H Version 2.0. It must be noted, however, that the UNSAT-H code is currently being converted to similar numerical algorithms, and may realize similar speedup in forthcoming versions.

BREATH is a numerical simulator designed to solve the partial differential equations governing the transient one-dimensional movement of fluid and energy within a porous medium, and was specifically created to examine processes affecting long-term infiltration of water into a soil or rock column in an arid environment. The equations used in the BREATH simulator are developed from a general multiphase perspective, specializing to equations comparable to those used in the soil science literature.

BREATH has several modes of operation. The most powerful mode is as a coupled system, with liquid movement, vapor diffusion, and energy transport all occurring simultaneously. It is also possible to simulate liquid movement or energy transport individually. A certain amount of care in problem discretization is in order when using BREATH for problems where advection dominates heat transfer; BREATH uses numerical methods appropriate for diffusive systems, as energy transport is dominated by diffusive behavior (heat conduction through the matrix) in a porous medium.

It is presumed that meteorological data files and output files will be rarely examined directly by human eyes; instead, they will be pre- and post-processed using other programs. Accordingly, BREATH is designed to be a filter between an arbitrary external boundary condition generator and an arbitrary external parameter analysis/graphics package. Significant flexibility in input and output is provided. Any variable can be input in several ways, and every variable can be output with a user-specified format. Dependent variables, fluxes, and mass balances can be output periodically during simulations as well.

## 2.2 MODEL ASSUMPTIONS

In the light of the intended use of BREATH, several important assumptions are made for this version of the code. It is assumed that the fluid of interest can exist as two species, liquid or vapor, and that there are no more than two fluid phases, comprising a liquid phase (the liquid species of the fluid of interest) and an infinitely mobile gas phase. It is assumed that the motion of the solid porous-matrix phase is negligible. It is further assumed that a continuous liquid phase must always be present throughout the medium, which may completely fill the pore space. The liquid phase may redistribute under the effects of gravity, capillary pressure gradients, and imposed pressure gradients; however, the vapor species may only redistribute through Fickian diffusion in the gas phase. Freezing and boiling are not considered. It is assumed that the liquid and vapor are always in equilibrium, except perhaps at the system boundary, due to the slow fluid velocities typically encountered in natural porous media.

The only forms of energy considered in BREATH are internal energy and vapor-species latent heat, with other forms assumed to be negligible for the slow-moving flow regimes characteristic of a porous medium and for the moderate temperature ranges considered. Internal energy is assumed to vary linearly with temperature, with temperature serving as the dependent variable in the governing equations. Energy may redistribute through conduction through the solid, liquid, and gas phases, and may be advected with the liquid phase and the vapor species. Latent heat may be advected with the vapor species.

It is tacitly assumed that the liquid of interest is water and the gas phase is air, insofar as the form of the constitutive equations describing the thermal behavior of the fluid properties is specialized into empirical relationships appropriate for these fluids. Two common forms of constitutive relationships, not restricted to water-air systems, are provided to describe the behavior of the fluid of interest within the particular porous medium, van Genuchten relationships and Brooks-Corey relationships. These constitutive properties describe the relationship between liquid-phase saturation and capillary pressure (the

difference in gas-phase and liquid-phase pressures), and the relationship between liquid-phase saturation and liquid-phase mobility. The parameters in the porous-medium-dependent constitutive relationships can be variable over the medium; however, it is assumed that the same type of constitutive relationship applies throughout the domain of interest. Hysteretic behavior is not considered.

## 2.3 BOUNDARY CONDITIONS

A variety of boundary conditions may be imposed, each of which may be specified at either boundary. The liquid phase may have a specified pressure, saturation, flux, pressure gradient, saturation gradient, or pressure-head gradient. Associated with the specified-flux boundary condition (used to specify precipitation), an option is available to simulate the ponding that occurs when the porous medium becomes saturated at the ground surface. The vapor phase may have a specified density, flux, density gradient, or atmospheric density (assuming diffusion across a boundary layer). The energy at the boundary may be treated with a specified temperature, flux, or generalized temperature gradient. The generalized temperature gradient accounts for common meteorologic energy sources and sinks, such as shortwave radiation, longwave radiation, conduction across a boundary layer, and evaporation.

## 2.4 SOLUTION PROCEDURE

BREATH is coded in FORTRAN 77, using standard numerical algorithms that were selected for accuracy, efficiency, and robustness. The solution procedure closely follows the algorithm outlined in Celia et al. (1990) and Celia (1991), who consider solution of the Richards equation and an associated contaminant transport equation. Celia et al. (1990) find that an inherently mass-conservative (within convergence and roundoff limits) finite-element approach to solve the mixed form of the Richards equation, posed with a fully implicit delta formulation and lumping the time term, and using a modified-Picard iteration scheme to update the constitutive properties, provides excellent results. Celia (1991) adds a transport equation, solving directly for concentration but again lumping the time term. Directly analogous approaches are employed herein, using the recommended approach for the Richards equation to solve a mass equation obtained by adding vapor-species diffusion to an equivalent of the Richards equation, and using the recommended approach for the contaminant transport equation to solve an energy equation. Analogous to the work of Celia and coworkers, the mass and energy equations are solved separately; however, since certain of the constitutive relationships governing fluid properties are temperature-dependent, additional iteration between the mass and energy equations is allowed. Discretization of both the mass and the energy equations results in a set of linearized algebraic equations; in both cases, the resultant set of tridiagonal equations is solved directly, using the extremely efficient Thomas algorithm (Lapidus and Pinder, 1982).

In general, efficiently simulating detailed long-term infiltration processes requires automatic time-step selection, as the time step during a rainfall event can be on the order of seconds and minutes, while in dry periods the time step is governed by the diurnal variation resolution. BREATH offers an automatic time-step selection option that regulates the time step size by the number of iterations required to meet a convergence criterion. In addition, BREATH offers an active-region option, such that unknown values are solved for during fewer iterations in regions where the unknowns are changing slowly. When much of the domain is far from rapidly fluctuating boundary conditions, this latter option can cut the total computational effort by several times.

### 3 MATHEMATICAL MODEL

The mathematical model for multiphase flow starts with a generic conservation equation for a quantity  $A$ ,

$$\frac{\partial}{\partial t}(\theta_{\alpha}A) + \nabla \cdot (\theta_{\alpha}A\mathbf{v}_{\alpha}) + \nabla \cdot [\theta_{\alpha}\mathbf{J}(A)] - Src_{\alpha} = 0, \quad (3-1)$$

where  $\alpha$  refers to phase  $\alpha$ ,  $\theta_{\alpha}$  is the volumetric fraction occupied by phase  $\alpha$ ,  $\mathbf{v}_{\alpha}$  is the velocity of phase  $\alpha$ ,  $\mathbf{J}(A)$  is a vector operator denoting the dispersive flux of phase  $\alpha$ ,  $Src_{\alpha}$  represents sources and sinks of phase  $\alpha$ , and  $t$  represents time. A similar equation applies for each phase. In the following, the advective term in Eq. (3-1) will often be written in a synonymous way, using the flux  $\mathbf{q}_{\alpha}$  of phase  $\alpha$ , defined by

$$\mathbf{q}_{\alpha} = \theta_{\alpha}\mathbf{v}_{\alpha}. \quad (3-2)$$

Note that throughout this document, boldface lower-case symbolizes a vector quantity and boldface upper-case symbolizes a tensor quantity unless otherwise defined.

A general mathematical model for a multiphase system in a porous medium may consider several fluid phases as well as the solid phase. Each phase may in turn consist of several species, or components. In the following, only a restricted system is considered, where the phases of interest are a solid, or soil, phase; a liquid, or water, phase; and a gas phase. These are signified by  $s$ ,  $w$ , and  $g$  subscripts respectively. The primary species of interest are the water species in the liquid phase and the water species in the gas phase (signified by a  $v$  subscript, for vapor). In general, however, any multiphase model requires that all available space in the porous medium is accounted for. This is accomplished by requiring that the sum of the volumetric fractions must be one, or

$$\sum_{\alpha} \theta_{\alpha} = 1. \quad (3-3)$$

#### 3.1 CONSERVATION OF MASS

For the conservation of the mass of a species  $i$  in phase  $\alpha$ ,  $A$  is replaced with  $\rho_{\alpha}^i$ , the density of species  $i$  in phase  $\alpha$ , so that

$$\frac{\partial}{\partial t}(\theta_{\alpha}\rho_{\alpha}^i) + \nabla \cdot (\theta_{\alpha}\rho_{\alpha}^i\mathbf{v}_{\alpha}) + \nabla \cdot [\theta_{\alpha}\mathbf{J}(\rho_{\alpha}^i)] - Src_{\alpha} = 0. \quad (3-4)$$

Of immediate interest is near-surface movement of water in the soil column, both as a liquid phase and as a vapor species. For this system, common assumptions are:

- The gas phase pressure in the porous medium reaches equilibrium with the atmosphere instantaneously, thus the effects of gas-phase compressibility on moisture transport may be neglected, as can the advective transport of the vapor species. This assumption disregards the effects of gas-phase trapping, which tend to slow down invading liquid pulses.



- The solid matrix is immobile, insoluble, and essentially incompressible, thus the mass balance equation for the solid phase can be neglected.
- The liquid phase is essentially incompressible, thus buoyancy effects may be neglected.
- The system always remains in the temperature range in which boiling and freezing effects can be neglected.
- Darcy's law applies to the flow of liquid in a porous medium.
- Fick's law applies to vapor diffusion in a porous medium.
- There is no net dispersive flux of the water species in the liquid phase, as the liquid phase is entirely composed of the water species.

With these assumptions, a water conservation equation can be derived by adding up the water species conservation equations over the phases, resulting in

$$\frac{\partial}{\partial t} (\theta_w \rho_w + \theta_g \rho_v) + \nabla \cdot (\rho_w \mathbf{q}_w) + \nabla \cdot [\theta_g \mathbf{J}(\rho_v)] - M_{ext} = 0. \quad (3-5)$$

In this,

- $\theta_w$  = volumetric liquid water content [ $L^3 L^{-3}$ ],
- $\theta_g$  = volumetric gaseous phase content [ $L^3 L^{-3}$ ],
- $\rho_w$  = water density in the liquid phase [ $ML^{-3}$ ],
- $\rho_v$  = water vapor density in the gaseous phase [ $ML^{-3}$ ],
- $\mathbf{q}_w$  = flux of the liquid water ( $\mathbf{q}_w = \mathbf{v}_w \theta_w$ ) [ $LT^{-1}$ ],
- $\mathbf{v}_w$  = velocity of the liquid water [ $LT^{-1}$ ], and
- $M_{ext}$  = external source terms.

Note that the net mass transfer between phases is zero for a species. Assuming Darcy's law can be extended to situations where there is more than one phase occupying the pore space, by using the concept of relative permeability,

$$\mathbf{q}_w = -\mathbf{k}\lambda \cdot (\nabla P_w + \rho_w g \nabla z), \quad (3-6)$$

where

- $\mathbf{k}$  = intrinsic permeability tensor [ $L^2$ ],
- $\lambda$  = mobility [ $TLM^{-1}$ ], or relative permeability divided by viscosity,
- $P_w$  = pressure in the liquid phase [ $ML^{-1}T^{-2}$ ]
- $g$  = gravitational acceleration constant [ $LT^{-2}$ ], and
- $z$  = elevation [ $L$ ].

Rather than working with water pressure directly, many formulations work with either matric head,  $\psi$ , or suction head,  $h$ , both having units of length, where  $h = -\psi$ ,

$$\psi \equiv \int_{P_{w0}}^{P_w} \frac{1}{g\rho_w(P)} dP, \quad (3-7)$$

and where  $P_{w0}$  is a reference pressure (such as atmospheric pressure) [Bear, (1972)]. Strictly speaking, such formulations are only valid when  $\rho_w$  is solely a function of pressure, so that the dependence of density on temperature and dissolved species is negligible. These assumptions are questionable in geothermal situations or when dealing with brines, for example. In most near-surface situations, water can be assumed incompressible and contaminants are dilute, so the head-based formulations are valid; however, the original formulation is followed in this work.

Note that in head-based formulations, Darcy's law is often written

$$\mathbf{q}_\alpha = -\mathbf{K}_{sat} k_r (\psi) \cdot (\nabla\psi + \nabla z), \quad (3-8)$$

where

$$\begin{aligned} \mathbf{K}_{sat} &= \text{saturated hydraulic conductivity, or } k\rho_w g / \mu \text{ [LT}^{-1}\text{]}, \\ k_r &= \text{relative permeability [-], and} \\ \mu &= \text{dynamic viscosity [ML}^{-1}\text{T}^{-1}\text{]}. \end{aligned}$$

Often  $\mathbf{K}_{sat}$  is reported rather than  $\mathbf{k}$ ; as viscosity is temperature-dependent, one should also be aware of the temperature at which the measurement is made.

Assuming Fick's law holds for vapor transport in porous media, diffusion of water vapor in the gaseous phase is described by (Jury et al., 1991)

$$\mathbf{J}(\rho_v) = -\tau \mathbf{D}_a \cdot \nabla \rho_v, \quad (3-9)$$

where  $\tau$  is a tortuosity factor accounting for tortuous flow paths (often set to 0.66) and  $\mathbf{D}_a$  is the diffusivity of vapor in air [ $\text{L}^2\text{T}^{-1}$ ]. Jury et al. (1991) indicate that the tortuosity factor may decrease with increasing moisture content, perhaps by orders of magnitude. However, as vapor transport is only important in dry media, this phenomenon is not considered in BREATH. For notational convenience in the following, vapor flux is defined by  $\mathbf{q}_v = -\theta_g \mathbf{J}(\rho_v) / \rho_v$

### 3.2 CONSERVATION OF ENERGY

To obtain the conservation of energy equation for phase  $\alpha$ ,  $A$  is replaced with  $\rho_\alpha E_\alpha$ , the energy in phase  $\alpha$ , so that

$$\frac{\partial}{\partial t} (\theta_\alpha \rho_\alpha E_\alpha) + \nabla \cdot (\theta_\alpha \rho_\alpha E_\alpha \mathbf{v}_\alpha) + \nabla \cdot [\theta_\alpha \rho_\alpha \mathbf{J}_e(E_\alpha)] - Src_{e\alpha} = 0, \quad (3-10)$$

where

- $E_\alpha$  = energy in phase  $\alpha$ ,
- $\rho_\alpha$  = density of phase  $\alpha$ ,
- $\mathbf{J}_e(E_\alpha)$  = dispersive flux of energy in phase  $\alpha$ , and
- $Src_{e\alpha}$  = sources/sinks of energy in phase  $\alpha$ , including transfers of energy from phase to phase.

In the following, it is assumed that:

- Kinetic and potential energy are both negligible.
- All phases are in thermal equilibrium.
- $E_\alpha$  is approximated by linear variation with temperature.
- A jump in energy may occur when a species changes phase.
- Fourier's law applies to the conduction of sensible heat.
- Boiling and freezing temperatures are not encountered.
- Radiation within the porous medium is negligible.
- Heat of wetting effects are negligible.

Landau and Lifshitz (1959) demonstrate that for a single pure fluid, having a single-phase conservation equation corresponding to Eq. (3-10) with  $\theta_\alpha = 1$ , the time term is written using internal energy ( $U$ ) and the advective term is written using enthalpy ( $H$ ). This difference follows from thermodynamic considerations requiring that the transport of energy due to fluid compressibility be accounted for. Extending this to the multiphase situation, the energy in phase  $\alpha$  is assumed to be

$$E_\alpha = U_{l\alpha} + U_\alpha(T_0) + \left( \frac{dU_\alpha}{dT} \right)_V (T - T_0), \quad (3-11)$$

except that the advective term is approximated by

$$\theta_\alpha \rho_\alpha E_\alpha \mathbf{v}_\alpha = \left[ H_{l\alpha} + H_\alpha(T_0) + \left( \frac{dH_\alpha}{dT} \right)_P (T - T_0) \right] \rho_\alpha \mathbf{q}_\alpha, \quad (3-12)$$

where  $\mathbf{q}_\alpha$  is substituted for  $\theta_\alpha \mathbf{v}_\alpha$  and

- $U_{l\alpha}$  = change in internal energy from liquid to phase  $\alpha$ , or  $U_\alpha - U_l$ ,
- $U_\alpha(T_0)$  = internal energy in phase  $\alpha$ , evaluated at  $T_0$ ,
- $(dU_\alpha/dT)_V$  = rate of change of internal energy with respect to temperature, holding the volume fixed,
- $H_{l\alpha}$  = change in enthalpy from liquid to phase  $\alpha$ , or  $H_\alpha - H_l$ ,
- $H_\alpha(T_0)$  = enthalpy in phase  $\alpha$ , evaluated at  $T_0$ ,
- $(dH_\alpha/dT)_V$  = rate of change of enthalpy with respect to temperature, holding the volume fixed, and
- $T_0$  = reference temperature at which all parameters are measured.

Linear variation of internal energy and enthalpy with temperature is considered a good approximation for liquids and solids; however, for gases, the effect of pressure variation may not be negligible in general (Green et al. (1992)). As the gas phase is assumed to maintain atmospheric pressure, pressure dependence on gas phase internal energy and enthalpy is not considered in BREATH.

The change in enthalpy due to phase change, or latent heat, is related to the change in internal energy due to phase change by (Van Wylen and Sonntag, 1978)

$$H_{l\alpha} = U_{l\alpha} + P\left(\frac{1}{\rho_\alpha} - \frac{1}{\rho_l}\right), \quad (3-13)$$

where  $P$  is pressure and  $\rho_l$  is the density of the liquid phase.

The nicety of discriminating between internal energy and enthalpy is often ignored in practice, as the compressibility effects in liquids and solids may be considered negligible compared to the change in internal energy (Green et al., 1993). For convenience, BREATH internally sets corresponding coefficients for enthalpy and internal energy to the same numerical value unless separate values are input, thus only one set of constitutive parameters need be provided.

The diffusive flux of energy is parametrized as a function of the multiphase mixture composition. Assuming Fourier's law applies to the conduction of sensible heat,

$$\sum_{\alpha} \theta_{\alpha} \rho_{\alpha} \mathbf{J}_e(E_{\alpha}) = -\mathbf{K}_e \nabla T, \quad (3-14)$$

where  $\mathbf{K}_e$  is the thermal conductivity tensor, which is a function of the phase saturations and properties.

The energy balance equations can be summed over the phases present in the near-surface environment, resulting in the total energy balance equation

$$\frac{\partial}{\partial t} (\theta_s \rho_s E_s + \theta_w \rho_w E_w + \theta_g \rho_g E_g) + \nabla \cdot \mathbf{q}_e - E_{ext} = 0, \quad (3-15)$$

where

$$E_{ext} = \sum_{\alpha} Src_{e\alpha}, \quad (3-16)$$

$$\mathbf{q}_e = \rho_w E_w \mathbf{q}_w + \rho_v E_v \mathbf{q}_v - \mathbf{K}_e \nabla T, \quad (3-17)$$

$$\rho_w E_w \mathbf{q}_w = \rho_w \mathbf{q}_w \left[ H_w(T_0) + \left( \frac{dH_w}{dT} \right)_P (T - T_0) \right], \quad (3-18)$$

$$\rho_v E_v \mathbf{q}_v = \rho_v \mathbf{q}_v \left[ H_{lv} + H_v(T_0) + \left( \frac{dH_v}{dT} \right)_P (T - T_0) \right], \quad (3-19)$$

$$\frac{\partial}{\partial t}(\theta_s \rho_s E_s) = \frac{\partial}{\partial t} \left[ \theta_s \rho_s \left( \frac{dU_s}{dT} \right)_V (T - T_0) \right], \quad (3-20)$$

$$\frac{\partial}{\partial t}(\theta_w \rho_w E_w) = \frac{\partial}{\partial t} \left( \theta_w \rho_w \left[ U_w(T_0) + \left( \frac{dU_w}{dT} \right)_V (T - T_0) \right] \right), \quad (3-21)$$

$$\frac{\partial}{\partial t}(\theta_g \rho_g E_g) = \frac{\partial}{\partial t} \left( \theta_g \rho_g \left[ U_g(T_0) + \left( \frac{dU_g}{dT} \right)_V (T - T_0) \right] + \theta_g \rho_v U_{lv} \right). \quad (3-22)$$

The net transfer of energy between phases is zero. As in the total mass conservation equation, it is assumed that there is no advective flux of the gaseous phase. Also, as it is assumed that the solid matrix is incompressible, undeforming, and does not consist of a species that changes phase (freezing is not considered), the  $\theta_s \rho_s U_s(T_0)$  and  $\theta_s \rho_s U_{ls}$  terms are not included in Eq. (3-20).

Since internal energy at the reference temperature is a constant, and can freely be brought in and out of derivatives, Eq. (3-5) can be employed to result in

$$U_w(T_0) \left[ \frac{\partial}{\partial t}(\theta_w \rho_w) + \nabla \cdot (\rho_w \mathbf{q}_w) \right] = -U_w(T_0) \left[ \frac{\partial}{\partial t}(\theta_g \rho_v) + \nabla \cdot (\rho_v \mathbf{q}_v) - M_{ext} \right]. \quad (3-23)$$

This substitution is advantageous for controlling computational roundoff error, as the terms on the right are usually numerically much smaller than the terms on the left. Rearranging Eq. (3-15) in terms of temperature yields

$$\frac{\partial}{\partial t} [C_1 (T - T_0) + C_2] + \nabla \cdot [M_1 (T - T_0) + M_2 - \mathbf{K}_e \cdot \nabla (T - T_0)] - E_{ext} = 0, \quad (3-24)$$

where

$$C_1 = \theta_s \rho_s (dU_s/dT)_V + \theta_w \rho_w (dU_w/dT)_V + \theta_g \rho_g (dU_g/dT)_V, \quad (3-25)$$

$$C_2 = \theta_g \rho_g U_g(T_0) + \theta_g \rho_v [U_{lv} - U_w(T_0)], \quad (3-26)$$

$$M_1 = \rho_w \mathbf{q}_w (dH_w/dT)_P + \rho_v \mathbf{q}_v (dH_v/dT)_P, \quad (3-27)$$

$$M_2 = \rho_w \mathbf{q}_w [H_w(T_0) - U_w(T_0)] + \rho_v \mathbf{q}_v [H_v(T_0) + H_{lv} - U_w(T_0)]. \quad (3-28)$$

Note that  $\nabla T_0$  is identically zero and  $E_{ext}$  now incorporates the mass term arising from the Eq. (3-23) substitution.

### 3.3 CONSTITUTIVE RELATIONSHIPS

There are numerous relationships describing the particular properties of the solid, liquid, and gaseous phases appropriate to a system under consideration. In general, hysteretic, or history-dependent, behavior may be observed, particularly in partially saturated systems. Hysteresis is most significant in relationships involving capillary pressure, and is less significant in moisture-content-based relationships

(Mualem, 1992). The extremes in behavior occur when using properties appropriate to uniformly wetting (increasing moisture content) and uniformly drying (decreasing moisture content) conditions, with infiltration pulses penetrating faster and deeper in simulations with properties appropriate to wetting conditions (Milly and Eagleson, 1980); incorporation of hysteresis would be expected to yield behavior intermediate to these two extremes. Hysteretic behavior tends to be quite difficult to characterize mathematically, and can pose severe numerical difficulties as well. Hysteretic effects are not considered in BREATH.

### 3.3.1 Liquid/Soil Properties

Several relationships have been proposed in the literature to describe the interrelationships between the volumetric fraction of a fluid phase, the pressure differences between fluid phases, and the relative ease of movement of each phase. Measured relationships between these quantities depend on the pore distributions in the porous medium, the phases present, and may often depend on the history of the phases. A capillary pressure term,  $P_c$ , is often used to denote the difference between the nonwetting phase pressure and the wetting phase pressure. In water/air systems, the wetting phase is usually water, thus  $P_c = P_g - P_w$ . Capillary pressure is not the same as suction pressure in unsaturated systems unless the air phase is at atmospheric pressure ( $P_g = 0$ ); the capillary pressure notation is introduced for generality.

#### 3.3.1.1 Moisture Content/Capillary Pressure

Van Genuchten (1980) derived a closed-form solution relating moisture content and capillary pressure which is widely used in the literature. In a head-based form, this relationship is stated

$$\theta_w(h) = \theta_{wr} + \frac{\theta_{ws} - \theta_{wr}}{[1 + (\alpha|h|)^n]^m} \quad (3-29)$$

Using the Mualem (1976) capillary tube model,  $m$  is set equal to  $1 - 1/n$ . Making this substitution and generalizing into a pressure-based form results in

$$P_c = P_0 \left( \theta_e^{-1/m} - 1 \right)^{1-m} \quad (3-30)$$

where

- $\theta_e$  = saturation  $[(\theta - \theta_{wr}) / (\theta_{ws} - \theta_{wr})]$ ,
- $\theta_w$  = moisture content  $[L^3L^{-3}]$ ,
- $\theta_{wr}$  = residual (minimum) moisture content  $[L^3L^{-3}]$ ,
- $\theta_{ws}$  = saturated (maximum) moisture content  $[L^3L^{-3}]$ ,
- $\alpha$  = scale factor  $[L^{-1}]$ ,
- $m, n$  = shape factors  $[-]$ , and
- $P_0$  = reference pressure equal to  $1 / \rho_w g \alpha$ .

A relationship proposed by Brooks and Corey (1966), using the same pressure-based notation, is

$$P_c = P_0 \theta_e^{-m} \quad \text{for } 0 < \theta_e < 1, \quad (3-31)$$

where  $P_c$  is set to a maximum and a minimum value when  $\theta_e$  is zero or one respectively. Note that when the Brooks-Corey and van Genuchten-Mualem relationships are applied to the same soil sample, in general  $P_0$  and  $m$  will have a different numeric value in the two relationships.

In these relationships, it is implicitly assumed that the relationship between moisture content and capillary pressure is not dependent on temperature. As surface tension is known to be dependent on temperature, and as capillary pressure is also dependent on surface tension, the relationship between moisture content and capillary pressure is likely to be dependent on temperature. This temperature dependence is neglected in BREATH.

### 3.3.1.2 Moisture Content/Mobility

Consistently applying their respective theoretical assumptions, van Genuchten (1980) and Brooks and Corey (1966) derived moisture content/relative permeability relationships to accompany their moisture content/capillary pressure relationships. The van Genuchten relationship, again in a head-based form but written as a mobility  $\lambda$ , is

$$\lambda(h) = \frac{1}{\mu} \left( \frac{\left( 1 - (\alpha|h|)^{n-1} [1 + (\alpha|h|)^n]^{-m} \right)^2}{[1 + (\alpha|h|)^n]^{m/2}} \right). \quad (3-32)$$

Using the pressure-based notation, again assuming  $m = 1 - 1/n$ , and using Eq. (3-30), results in

$$\lambda(\theta_e) = \frac{1}{\mu} \theta_e^{1/2} \left[ 1 - \left( 1 - \theta_e^{1/m} \right)^m \right]^2. \quad (3-33)$$

The corresponding Brooks-Corey relationship is

$$\lambda(\theta_e) = \frac{1}{\mu} \theta_e^{2m+3}, \quad (3-34)$$

where, as before,  $m$  has a different numerical value in the van Genuchten and Brooks-Corey relationships.

### 3.3.1.3 Single-Phase Flow

Strictly speaking, the above relationships are only defined when unsaturated conditions exist. There is a small compressibility associated with the matrix and liquid water that is not accounted for as written. With high positive pressures, it is generally accepted that there is a linear change in storage as pressure varies, of the form

$$\frac{\partial}{\partial t} (\theta_w \rho_w + \theta_g \rho_v) = \frac{S_s}{g} \frac{\partial P_w}{\partial t}, \quad (3-35)$$

where  $S_s$  is a specific storage coefficient. However, the nature of the storage term is not yet established near the transition between saturated and unsaturated conditions. For numerical convenience, in the code the effect of the specific storage term linearly decreases with decreasing moisture content, with the effect optionally vanishing at zero moisture content or at residual moisture content, and the effect is full at zero pressure. Also, when the medium is completely saturated, relative permeability is set to unity.

### 3.3.2 Thermal Properties

A straightforward method for describing the phase-dependent variation of thermal conductance is used in BREATH, based on Huyakorn and Pinder (1983). In this approach, the bulk, or mixture, thermal conductivity is simply a weighted average of the phase conductances, or

$$K_e = \sum_{\alpha} \theta_{\alpha} \rho_{\alpha} k_e^{\alpha}, \quad (3-36)$$

where  $k_e^{\alpha}$  is the thermal conductance for phase  $\alpha$ . Representative values for quartz are  $8.8 \text{ W m}^{-1} \text{ K}^{-1}$ , other minerals  $2.9 \text{ W m}^{-1} \text{ K}^{-1}$ , organic matter  $0.25 \text{ W m}^{-1} \text{ K}^{-1}$ , liquid water  $0.57 \text{ W m}^{-1} \text{ K}^{-1}$ , and air  $0.025 \text{ W m}^{-1} \text{ K}^{-1}$  (Hillel, 1980).

There is some evidence (Jury et al., 1991; Campbell, 1985; de Vries, 1963) that straightforward thermal conductance weighting may not properly describe the thermal behavior of a matrix, particularly a coarse, dry matrix. The discrepancy arises due to the continuum approximation for the various phases, which does not properly account for the microphysics. When the matrix is quite dry, thermal conduction in the solid is controlled by the intergrain contact points, thus the bulk thermal conductivity is strongly affected by the air phase. Similarly, when the matrix is near saturation, the liquid phase has a strong effect on the bulk conductivity. As air is a poor conductor relative to soil particles and liquid water, the effect is more pronounced in dry soils. In such dry regimes, water tends to collect at the intergrain contact points, thereby considerably increasing bulk thermal conductance. The addition of a small amount of water tends to have a relatively larger effect at low moisture contents. This behavior is most significant in coarse media. Only straightforward averaging of thermal conductances is accommodated in BREATH.

### 3.3.3 Fluid Properties

The vapor in the porous medium is assumed to be at equilibrium with the liquid phase, with the equilibrium described by (modified from Campbell, 1985):

$$\rho_v = \rho_{vs} \exp\left(\frac{P_w M_w}{\rho_w R T}\right) \quad \text{for } P_w < 0, \quad (3-37)$$

where  $\rho_v$  is the vapor density,  $\rho_{vs}$  is the saturated vapor density at temperature  $T$ ,  $M_w$  is the molecular weight of water [ $18 \text{ gm mole}^{-1}$ ],  $R$  is the ideal gas constant [ $8.3143 \text{ J mole}^{-1} \text{ K}^{-1}$ ], and an empirical relationship (Fayer and Jones, 1990; 1993) describes the saturated vapor density:



$$\rho_{vs} \approx a_1 \exp [a_2 - (a_3/T) - a_4 \ln (T) ] . \quad (3-38)$$

In the cgs system,  $a_1 = 1$ ,  $a_2 = 46.440973$ ,  $a_3 = 6790.4985$ ,  $a_4 = 6.02808$ ,  $\rho_{vs}$  is in  $\text{gm cm}^{-3}$ , and  $T$  is in degrees K.

Both fluid density and fluid viscosity vary with the temperature of the fluid. It is assumed that the temperature range is small enough that the variation in density due to temperature variation is negligible. Two choices for the thermal dependence of viscosity are allowed; a constant value, and an empirical power-law expression suitable for liquid water (CRC, 1982, p. F-40):

$$\log_{10} \left( \frac{\mu_T}{\mu_0} \right) = \begin{cases} \left[ \frac{a_1}{a_2 + a_3 (T - T_r) + a_4 (T - T_r)^2} - \frac{a_1}{a_2} \right] & \text{for } 0 \leq T \leq 20 \\ \left[ \frac{a_5 (T_r - T) + a_6 (T_r - T)^2}{T + a_7} \right] & \text{for } 20 < T \leq 100 \end{cases} \quad (3-39)$$

where  $T$  is in degrees Celsius,  $T_r$  is 20 °C,  $\mu_T$  and  $\mu_0$  are in centipoise ( $100 \text{ cp} = 1 \text{ gm cm}^{-1} \text{ sec}^{-1}$ ), and the viscosity at  $T_r$ ,  $\mu_0$ , is 1.002 cp. In the cgs system,  $a_1 = 1301$ ,  $a_2 = 998.333$ ,  $a_3 = 8.1855$ ,  $a_4 = 0.00585$ ,  $a_5 = 1.3273$ ,  $a_6 = -0.001053$ , and  $a_7 = 105$ .

In order to use the above empirical relationships during simulations, it is necessary to convert units for vapor density, viscosity, and temperature. Vapor density units are implicitly determined by supplying the value of  $a_1$ , and viscosity units are implicitly determined by supplying the value of  $\mu_0$ . Internal to BREATH, all empirical relationships that depend on temperature are coded assuming that temperature is in degrees Kelvin. At the start of each **calc** request (see Appendix A.3), all temperatures and temperature-dependent coefficients are converted to degrees Kelvin, using the relationship  $T = tscale T_c + tshift$ , where  $T_c$  is temperature in the units used in the energy equation,  $T$  is temperature in °K, and  $tscale$  and  $tshift$  are user-supplied parameters. Before control returns to the command file, these values are returned to the user system.

## 4 NUMERICAL MODEL

### 4.1 FINITE ELEMENT FORMULATION

In this section, a brief discussion of the implementation in the numerical code BREATH is presented. Further details on finite element methods can be found in the vast literature on numerical methods.

A generic partial differential equation in one dimension can be written:

$$L(u) = a(u) \frac{\partial u}{\partial t} + \frac{\partial}{\partial x} \left[ b(u) + c(u) \frac{\partial u}{\partial x} \right] + d(u)u + f(u) = 0, \quad (4-1)$$

where  $u$  is the unknown value,  $L(u)$  is the operator on  $u$ , and  $a$ ,  $b$ ,  $c$ ,  $d$ , and  $f$  are coefficients, which in general may depend on  $u$ .

By setting a set of weighted integrations of Eq. (4-1) to zero over the domain of interest,  $[x_0, x_1]$ ,

$$\int_{x_0}^{x_1} L(u) w_j(x) dx = 0, \quad j = 1, 2, \dots, M, \quad (4-2)$$

the governing equation is satisfied in a weighted sense. In order to reduce the system to a set of algebraic equations, the unknown  $u$  is approximated by a set of coefficients and associated spatial functions,

$$u \approx \sum_{i=1}^N u_i \phi_i(x). \quad (4-3)$$

By selecting exactly the same number of weighting functions as unknown coefficients  $u_i$ , a compatible set of equations is obtained.

In the Galerkin finite element approach, the same functions are used to approximate  $u$  as are used for the weighting functions [ $\phi_i(x) = w_i(x)$ ]. The BREATH code follows the Galerkin approach, with standard linear basis functions and no upstream weighting, except that the time terms are lumped. In general, the finite element method allows dependent variables and parameters to be defined either by node or by element. For convenience, all quantities in BREATH are defined by node, except for fluxes, following Celia (1991). As a consequence, it is assumed that any material discontinuities occur *between* nodes, not *at* nodes. In cases where a quantity must be evaluated within an element, such as the  $c(u)$  term in Eq. (4-1) (e.g. conductivities, diffusivities), the quantity is evaluated using the geometric mean, or

$$c_{i+1/2} \approx (c_i \cdot c_{i+1})^{1/2}. \quad (4-4)$$

## 4.2 CONSERVATION EQUATIONS

### 4.2.1 Conservation of Mass

The mass conservation equation is solved using a fully implicit, finite-element, mixed-delta formulation with modified Picard iteration on the unknown coefficients. When vapor is not considered, the solution procedure corresponds to the formulation recommended by Celia et al. (1990), which is known to exhibit excellent mass balance properties for unsaturated flow problems.

The mass conservation equation can be rewritten in the following form,

$$\frac{\partial}{\partial t} (\theta_w \rho_w + \theta_g \rho_v) + \{ \nabla \cdot (\rho_v \mathbf{q}_w) + \nabla \cdot [\theta_g \mathbf{J}(\rho_v)] - M_{ext} \}^{t + \alpha \Delta t} = 0, \quad (4-5)$$

where  $t$  is the time at the beginning of a time step,  $\Delta t$  is the size of the time step, here  $\alpha$  is a time-weighting parameter ranging from 0 to 1, and the superscript refers to the time at which the term is evaluated. When  $\alpha$  is 0, the resulting scheme is termed fully explicit; when  $\alpha$  is 1, the resulting scheme is termed fully implicit. In general, as a time-stepping scheme becomes more implicit, the more robust to numerical instability it becomes, allowing larger time steps to be taken. However, larger time steps come at the cost of additional numerical dispersion. For the purposes of the numerical code, the robustness associated with the fully implicit scheme has been judged to outweigh numerical dispersion arising from large time steps, thus the fully implicit scheme was implemented.

With the fully implicit time-stepping scheme, all coefficients are evaluated at time  $t + \Delta t$ ; coefficients that are functions of the dependent variables are thus *a priori* unknown. Such coefficients can only be evaluated at a guessed value for the dependent variable. At the start of a time step, the best guess is the current value; plugging in the values for the coefficients, new values for the dependent variables can be solved for. Evaluating the coefficients at the newly estimated values of the dependent variables and re-solving, a new estimated value for the dependent variable can be obtained. When the time step is small enough, the change in the dependent variables approaches zero as iteration continues.

A main feature of the mixed-delta formulation is the introduction of the new variable  $\delta$ , which is the change in the estimated value of a dependent variable over an iteration. Usually, the selected dependent variable is head or pressure, although saturation could be used as well if fully saturated systems are not considered. The mass conservation equation is recast as

$$\frac{d(\theta_w \rho_w + \theta_g \rho_v)}{d\delta} \left( \frac{\partial \delta}{\partial t} \right) + \nabla \cdot \left( \frac{d[\rho_w \mathbf{q}_w + \theta_g \mathbf{J}(\rho_v)]}{d\delta} \delta \right) - \frac{dM_{ext}}{d\delta} \delta + RES^m = 0, \quad (4-6)$$

$$RES^m = \left[ \frac{\partial}{\partial t} (\theta_w \rho_w + \theta_g \rho_v) + \nabla \cdot (\rho_w \mathbf{q}_w) + \{ \nabla \cdot [\theta_g \mathbf{J}(\rho_v)] - M_{ext} \} \right]^m, \quad (4-7)$$

where  $m$  refers to evaluation at the current iteration, and  $RES$  is residual error in evaluating the equation.

Using a relatively standard approach, where  $\delta = P_w^{m+1} - P_w^m$ , the form of Eq. (4-6) solved in the numerical code is

$$\left[ (\rho_w - \rho_v) \frac{d\theta_w}{dP_w} + \theta_g \frac{d\rho_v}{dP_w} \right] \frac{\partial \delta}{\partial t} + \nabla \cdot \mathbf{q}_\delta - \frac{dM_{ext}}{dP_w} \delta + RES^m = 0, \quad (4-8)$$

$$\mathbf{q}_\delta = - \left( \rho_w \mathbf{k} \lambda + \theta_g \tau \mathbf{D}_a \frac{d\rho_v}{dP_w} \right) \cdot \nabla \delta - \left( \tau \mathbf{D}_a \frac{d\theta_g}{dP_w} \cdot \nabla \rho_w \right) \delta. \quad (4-9)$$

Note that only terms explicitly dependent on the primary dependent variables  $P_w$ ,  $\rho_v$ , or  $\theta_w$  are expanded. Any coefficients depending on the primary dependent variables are evaluated using the most current value of the variable. Many formulations neglect the vapor terms entirely; under very dry conditions, numerical experiments indicate that including the vapor terms can decrease the number of iterations up to 15 percent. When evaporation is allowed (the *densva* boundary condition presented in Section 4.4.2), including the vapor boundary condition as part of the  $(dM_{ext} / dP_w) \delta$  term in Eq. (4-8) can decrease the number of iterations by an order of magnitude or more.

## 4.2.2 Conservation of Energy

The energy conservation equation is formulated using  $\Phi$  as the dependent variable, where  $\Phi = T - T_0$ . With this substitution, the governing equation is

$$\frac{\partial}{\partial t} [C_1 \Phi + C_2] + \nabla \cdot [M_1 \Phi + M_2 - \mathbf{K}_e \cdot \nabla \Phi] - E_{ext} = 0, \quad (4-10)$$

where the coefficients are defined in Section 3.2. Unlike the procedure for solving the mass conservation equation, a  $\delta$  formulation is not used; rather,  $\Phi$  is determined directly each iteration within a time step. Equation (4-10) is discretized using exactly the same procedures as is used for Eq. (4-8), resulting in a fully implicit approximation. As might be expected, selecting  $T_0$  to be somewhere near the middle of the range of temperatures expected in the simulation provides the best accuracy, by minimizing roundoff error.

When longwave radiation from the ground surface is allowed (the *gradtemp* option in Section 4.4.3), a boundary condition dependent on  $T^4$  is imposed, where  $T$  is in an absolute temperature scale. This condition is handled by splitting the term into a known value and a term linearly varying with  $\Phi$ , using the identity

$$T^4 = (\Phi + T_0) T^3 \quad (4-11)$$

and approximating  $T^3$  using the most recent value for temperature. Other temperature-dependent boundary conditions are similarly split into portions having a known value and a term linearly varying with  $\Phi$ , determining the  $\Phi$ -dependent term implicitly. As with the mass conservation equation, any temperature dependence in a coefficient is handled by using the most recent temperature to evaluate the coefficient.

## 4.3 TIME STEPPING AND ACTIVE REGION CONTROL

### 4.3.1 Constant and Courant-Based Time Stepping

As discussed in the previous section, fully implicit time stepping is used in the code. This is found to be fast and robust. Several options for selecting a time step are available, however. For some purposes, a constant time step is appropriate. A constant time step may be either directly input, or in the case that only the energy equation is being simulated, a Courant-based time step criterion may be used. A Courant number may be evaluated for each element,

$$C_o = \frac{\mathbf{v}\Delta t}{\Delta x}, \quad (4-12)$$

where  $\mathbf{v}$  is  $\mathbf{q}_w (dU_w/dT)_V + \mathbf{q}_v (dU_v/dT)_V$ ,  $\Delta t$  is the time step, and  $\Delta x$  is the length of the element. The largest Courant number found across all elements is compared with the criterion. Typically, a criterion on the order of 0.1 and smaller will give good results.

### 4.3.2 Heuristic Convergence-Based Time Stepping

In cases where the time step is expected to be variable, a heuristic selection procedure is provided, based on attempting to satisfy a desired number of iterations per time step with exactly the desired convergence accuracy. The selection procedure arose from observations that, with a sufficiently small time step, the convergence criterion improves log-linearly with the number of iterations. The procedure has been found to be very useful when solving the coupled mass and energy equations. In all cases, a new time step is required to lie within a minimum and maximum value. In cases where convergence has occurred within a set number of iterations, the new time step is selected by

$$\Delta t \Leftarrow \Delta t \left( \frac{10 + \min(N_{max} + 1, 10)}{10} \right) \left( \frac{\ln(C_{nom}) - \ln(C) + N_{nom} - N}{N} \right), \quad (4-13)$$

where

- $N$  = actual number of iterations taken in the previous time step,
- $N_{max}$  = maximum number of iterations allowed,
- $N_{nom}$  = nominal, or desired, number of iterations,
- $C_{nom}$  = nominal convergence criterion, or accuracy desired, and
- $C$  = actual convergence achieved in the previous time step.

When the convergence criterion is not met within the set iterations, the new time step is calculated with

$$\Delta t \Leftarrow \Delta t \left( \frac{\ln(C_0) - \ln(C)}{\ln(C_0) - \ln(C_{nom})} \right) \left( \frac{N_{nom}}{N} \right), \quad (4-14)$$

where  $C_0$  is the convergence criterion after the first iteration. If desired, whenever the convergence criterion increases within a time step (indicating that a divergent situation may be arising), the time step may be immediately halved and the time step restarted.

The convergence criterion is defined as the largest change in a dependent variable at any node over an iteration. A reasonable convergence criterion might be a value such that only the fourth or fifth decimal place in the dependent variable is affected by an error with this magnitude. Based on numerical experiments, good results are obtained by setting  $N_{nom}$  to 5 and  $N_{max}$  to 10. When more than one cycle of mass/energy solution iteration is requested, a rule-of-thumb practice is to add an additional 2 iterations per additional requested cycle, both for  $N_{nom}$  and for  $N_{max}$ .

### 4.3.3 Active Region Control

It is often the case that a relatively small portion of the domain of interest is changing rapidly compared to other portions of the domain. In such cases, the time step required to accurately represent the rapidly changing region may allow an explicit time step or reduced numbers of iterations in the slowly changing region. With this in mind, an option is provided where only the regions that change more than a user-specified criterion are solved for in the next iteration. This is accomplished by finding nodes where the dependent variable changes less than the criterion within an iteration, and turning off a flag at those nodes. The first and the last active node determines the active region, with a few already converged nodes added as a transition zone; the user specifies the number of converged nodes to use. An active region can exist at either end of the domain or somewhere in the middle; only one active region is currently allowed.

The active region control can have two levels. In cases where the boundary conditions are changing at a higher frequency than the response time for the system, a good default time step for the system may be the length of time between boundary condition changes. In such situations, by starting with this default time step and finding an inactive region as above, unknowns in the inactive region need not be solved for again until the boundary conditions change, thereby cutting total computational effort. The dependent variable at the edge of the inactive region becomes a Dirichlet boundary condition, linearly interpolated over the time step. At a second, finer, level, the inactive region can be greedily expanded with each iteration, again reducing the number of unknowns; however, the number of padding nodes specified by the user are added to the active zone each iteration, after convergence is checked, helping to keep the unknowns in the active and inactive zones compatible. This simple combined procedure has been observed to reduce total computational effort by a factor of two to five with minimal impact on the solution.

## 4.4 BOUNDARY CONDITIONS

There are a variety of boundary conditions that can be imposed on a column of variably saturated soil. The categories of boundary conditions implemented in the code represent conditions for the liquid phase and the vapor species in the mass balance equation, and conditions on the energy equation. A generic boundary condition equation can be defined,

$$\alpha u + \beta \frac{\partial u}{\partial n} = \gamma, \quad (4-15)$$

where  $u$  is the dependent variable of interest;  $\alpha$ ,  $\beta$ , and  $\gamma$  are possibly time-varying constants, and  $\partial u / \partial n$  is the gradient of  $u$  in the direction normal to the boundary.

In BREATH, the default boundary condition for each equation is a no-flux boundary. First-type conditions, where the value of  $u$  is specified directly ( $\alpha = 1$  and  $\beta = 0$ ), are imposed by neglecting the matrix equation corresponding to the known value, and moving any terms containing the known value to the forcing vector in all other equations. Second-type conditions, where the value of  $\partial u/\partial n$  is known, are applied by defining an equivalent point source at the corresponding boundary node, denoted by  $q_{ws}$ ,  $q_{vs}$ , and  $q_{es}$  for water, vapor, and energy, respectively, in the following sections. Saturation-dependent material properties used in second-type boundary conditions are evaluated using the values at the boundary node, lagged one iteration in the same manner as interior nodes.

Terms used in subsequent sections correspond to the same terms in the respective governing equations, unless defined otherwise. The terms denoted  $B_w$ ,  $B_v$ , and  $B_e$  are values required to completely specify the boundary condition for water, vapor, and energy, respectively.

#### 4.4.1 Liquid Phase Boundary Conditions

For the flow equation, the easily implemented conditions of specified water pressure (*press*), water saturation (*satw*), and water flux (*flux*) are allowed. Also, derived conditions are allowed, so that the gradient of pressure (*gradpress*), the gradient of saturation (*gradsatw*), and the gradient of pressure head (*gradhead*) can be specified. These boundary conditions are defined in Table 4-1, where the value  $B_w$  is supplied by the user. The value  $\partial x/\partial n$  is 1 at the first boundary and  $-1$  at the last boundary, so a positive  $B_w$  for the *flux* code word will be treated as a source at the first boundary and will be treated as a sink at the last boundary.

Table 4-1. Liquid-phase boundary conditions

Input Keyword	Corresponding Boundary Condition
<i>press</i>	$P_w = B_w$
<i>satw</i>	$P_w = P_w(B_w)$
<i>flux</i>	$q_{ws} = \rho_w B_w \left( \frac{\partial x}{\partial n} \right)$
<i>gradpress</i>	$q_{ws} = -\rho_w \mathbf{k} \lambda \left( B_w + \rho_w g \frac{\partial z}{\partial x} \right) \frac{\partial x}{\partial n}$
<i>gradsatw</i>	$q_{ws} = -\rho_w \mathbf{k} \lambda \left( \frac{dP}{d\theta} \right) B_w + \rho_w g \left( \frac{\partial z}{\partial x} \right) \frac{\partial x}{\partial n}$
<i>gradhead</i>	$q_{ws} = \rho_w \mathbf{k} \lambda B_w \frac{\partial x}{\partial n}$

Associated with the *flux* boundary condition, control of ponding is allowed, in which water pressure rising to a specified value changes the boundary condition from specified flux to specified pressure. When the allowed ponding value is above atmospheric, ponding storage occurs up to a user-specified maximum allowed pond height at which excess rain is assumed to run off. Ponding storage rise is calculated as the difference between the specified flux and the actual flux entering the soil. Once the pond height drops below zero, flux conditions are resumed. When the maximum pond height is negative, switching between the flux and pressure conditions occurs in the same way but ponding is disallowed.

#### 4.4.2 Vapor Species Boundary Conditions

Associated with the flow equation are conditions on the vapor species. These include specified vapor density (*densv*), specified vapor flux (*flux*), specified vapor density gradient (*graddenv*), and specified atmospheric vapor density (*densva*). Each condition is implemented as an additional source term. The specified vapor density flux is based on the element adjacent to the boundary, where the  $\rho_v(x^+)$  refers to the vapor density one node into the domain and the  $B_v$  represents the vapor density at the boundary. Definitions for the boundary conditions are shown in Table 4-2. The boundary layer vapor conductance term  $k_v$ , representing a diffusion coefficient divided by a distance, can be specified as a constant value, or can be calculated from meteorological conditions. Calculation of  $k_v$  from meteorological conditions is further described in Section 4.4.4.

Table 4-2. Vapor-species boundary conditions

Input Keyword	Corresponding Boundary Condition
<i>densv</i>	$q_{vs} = -\tau D_a \theta_g \frac{\rho_v(x^+) - B_v}{\Delta x}$
<i>flux</i>	$q_{vs} = B_v \frac{\partial x}{\partial n}$
<i>graddenv</i>	$q_{vs} = -\tau D_a \theta_g B_v \frac{\partial x}{\partial n}$
<i>densva</i>	$q_{vs} = k_v (B_v - \rho_v)$

#### 4.4.3 Energy Equation Boundary Conditions

The energy equation can have a specified temperature (*temp*), specified energy flux (*flux*), or a specified gradient of temperature (*gradtemp*). In the last condition, energy is allowed to leave the system by advection of water and vapor past the boundary. The water and vapor fluxes used in this case are the flux in the element adjacent to the boundary. If water and vapor are entering the boundary, it is assumed that they have the external temperature and bring this energy into the system. When water and vapor fluxes are specified, these are the fluxes used; otherwise, the fluxes in the element adjacent to the boundary are used. Code words and associated boundary conditions are specified in Table 4-3.

Table 4-3. Energy equation boundary conditions

Input Keyword	Corresponding Boundary Condition
<i>temp</i>	$T = B_e$
<i>flux</i>	$q_{es} = B_e \frac{\partial x}{\partial n}$
<i>gradtemp</i>	$q_{es} = [M_1 (T - T_0) + M_2 - M_3 B_e] (\partial x / \partial n) + Q_{ext}$ $Q_{ext} = k_h (T_a - T) + (1 - \alpha_s) S_t + \epsilon_a \sigma T_a^4 - \epsilon_s \sigma T^4$



In Table 4-3, the terms incorporated in  $Q_{ext}$  are defined by

- $k_h$  = boundary layer heat conductance,
- $\alpha_s$  = ground surface albedo [0 to 1],
- $S_t$  = net shortwave radiation (net solar radiation),
- $\epsilon_a$  = atmospheric emissivity [0 to 1],
- $\epsilon_s$  = ground surface emissivity [0 to 1],
- $\sigma$  = the Stefan-Boltzmann constant, and
- $T_a$  = atmospheric absolute temperature.

Each of the  $Q_{ext}$  terms can be specified as a constant value, or can be calculated from time-varying meteorological values, as described in Section 4.4.4.

#### 4.4.4 Meteorological Boundary Conditions

A set of conditions associated with meteorological forcing can be applied. With this option, it is possible to input a series of time-varying meteorological data that are processed into boundary conditions. The data that are supplied include the atmospheric temperature  $T_a$ , the vapor density in the atmosphere  $\rho_{va}$ , the net shortwave radiation  $S_t$ , the atmospheric emissivity times the Stefan-Boltzmann coefficient  $\epsilon_a\sigma$ , the windspeed  $v_{wind}$ , and the energy boundary condition value  $B_e$ .

The ground surface albedo, or the fraction of solar radiation reflected from the ground surface, is assumed to be a linear function of surface moisture content,

$$\alpha_s = \alpha_s^{dry} + \left( \frac{\theta_w}{\theta_{ws}} \right) \left( \alpha_s^{wet} - \alpha_s^{dry} \right), \quad (4-16)$$

as is the ground surface emissivity, or the ratio of surface emittance to blackbody emittance at the same temperature,

$$\epsilon_s = \epsilon_s^{dry} + \frac{\theta_w}{\theta_{ws}} \left( \epsilon_s^{wet} - \epsilon_s^{dry} \right). \quad (4-17)$$

Albedo can be as high as 0.95 for fresh snow; dry sandy soils range from 0.25 to 0.45; dry clay soils range from 0.20 to 0.35; and most field crops are in the range of 0.20 to 0.30 (Rosenberg, 1974). Emissivity is on the order of 0.92 for land, 0.98 for vegetation, and 0.96 for water (Peixoto and Oort, 1992).

The vapor and sensible heat boundary layer conductances,  $k_v$  and  $k_h$ , can be calculated from meteorological conditions; often the literature refers to the reciprocal of this term as resistance. The conductances represent a diffusion coefficient divided by a distance. Various methods for calculating these coefficients are found in the literature, mainly in agricultural applications, and usually require that uniform surface conditions apply upwind of the measurement point so that the windspeed profile attains an asymptotic shape. Relationships in the literature between boundary-layer conductance and windspeed usually have the overall form of  $k = a + b v_{wind}$ , accounting for an increase in turbulent dispersion with increased windspeed. Often the windspeed profile is assumed to be logarithmically increasing with height, perhaps offset from the ground surface with a zero-plane displacement arising from vegetative resistance.

In BREATH, the boundary layer conductances may be input as constants; or, as in UNSATH [Fayer and Jones, 1990; 1993], may be calculated using relationships collected in Campbell (1985). In the form that is implemented in the code, conductances are estimated using

$$k_v = k_h = \frac{kU}{\ln[(z^T - d + z_h)/z_h] + \psi_h}, \quad (4-18)$$

$$U = \frac{kv_{wind}}{\ln[(z^U - d + z_m)/z_m] + \psi_m}, \quad (4-19)$$

where

- $k$  = von Karman's constant (generally assumed to be 0.4),
- $U$  = friction velocity,
- $z^T$  = height above surface at which temperature is measured,
- $z^U$  = height above surface at which wind speed is measured,
- $z_h$  = surface roughness length for heat,
- $z_m$  = surface roughness length for momentum,
- $\psi_h$  = stability correction factor for heat,
- $\psi_m$  = stability correction factor for momentum, and
- $d$  = zero plane displacement height.

Guidance on selecting these parameters is given in Campbell (1985). A set of empirical correlations is presented, based on typical crop surfaces, where  $d = 0.77 h$ ,  $z_m = 0.13 h$ , and  $z_h = 0.2 z_m$ , and where  $h$  is the crop height. The stability correction factors are reported to depend on the direction of sensible heat flux (Businger, 1975; Campbell, 1985). Stable conditions occur when the surface temperature is lower than the air temperature, and unstable conditions occur in the converse case. Businger (1975) suggests using a stability parameter  $\xi$ , where (in the form incorporated in BREATH)

$$\xi = - \frac{kz^T gK (T - T_a)}{T_a U^3}. \quad (4-20)$$

In stable conditions, Businger (1975) recommends the empirical relationship

$$\psi_m = \psi_h = 4.7\xi, \quad (4-21)$$

and in unstable conditions,

$$\psi_h = -2 \ln \left( \frac{1 + (1 - 16\xi)^{1/2}}{2} \right), \quad (4-22)$$

$$\psi_m = 0.6\psi_h. \quad (4-23)$$

This set of equations is circularly defined, with  $\xi$  depending on  $U$  and  $K$ , which depend on  $\psi_m$  and  $\psi_h$ , which in turn depend on  $\xi$ . An iterative scheme similar to that in Campbell (1985) is used to calculate the boundary layer conductances; calculating  $U$ , then  $K$ , then  $\xi$ , then  $\psi_h$  and  $\psi_m$ , and cycling until convergence.  $\psi_h$  and  $\psi_m$  are assumed zero to start the iteration sequence; ten iterations are used, to ensure convergence.

## 5 VERIFICATION AND BENCHMARKING

Four test problems are presented to verify and benchmark BREATH. BREATH allows the flow equation and the energy equation to be simulated independently or in a coupled mode. The energy equation portion of the code is tested against analytic solutions. When flow is simulated, BREATH is compared to other numerical simulators.

The Richards equation solution procedure is coded using the methods outlined in the UNSAT1D documentation (Celia, 1991), thus a comparison problem comparing combined saturated-unsaturated soil moisture redistribution with UNSAT1D predictions is presented. A pair of tests of the energy equation simulator against analytic solutions is presented for the energy equation, checking heat conduction in the solid phase and checking the combined effects of heat advection and diffusion in the liquid phase. Finally, a three-layer, fully coupled infiltration problem is simulated using BREATH and UNSATH Version 2.1 (Fayer and Jones, 1993).

### 5.1 SOIL MOISTURE REDISTRIBUTION

A somewhat contrived example is chosen to demonstrate that the saturated/unsaturated flow portion of the BREATH code is working properly, with a comparison against the UNSAT1D code [presented in Celia (1991)]. The two codes are exactly comparable, as the BREATH code uses the same algorithm as the UNSAT1D code. Both codes use a self-consistent set of units. The one-dimensional example problem features the redistribution of water in a vertical 100-unit column. The initial pressure-head distribution is linearly decreasing from the top of the column, at 100-unit head, to the bottom of the column, at -100-unit head, representing a saturated region overlying an unsaturated region. In order to make the example a little more interesting, at the onset of the example the bottom boundary is attached to a source of water at 100 units of head, while the top boundary is assumed to be impermeable to water.

In this example, the column is assumed to be uniformly filled with a porous medium having a saturated conductivity of 0.0092, residual moisture content of 0.102, and saturated moisture content of 0.368. The moisture retention and relative permeability properties are assumed to be described with van Genuchten parameters, with  $\alpha$  at 0.0335 and  $n$  at 2. These parameters are equivalent to values of  $P_0$  at 29.85 and  $m$  at 0.5, respectively. The column is uniformly discretized with 40 elements and uniform time steps of 0.3 are taken. The specific storage coefficient is assumed to be 0.0001, with the effect of the specific storage coefficient linearly decreasing to zero at zero moisture content. An input file for this problem is displayed in Appendix B.1, and the associated output file is shown in Appendix B.2.

BREATH and UNSAT1D are compared at several output time steps in Figures 5-1 and 5-2. The two codes yield nearly identical results, as expected, with a small amount of roundoff error manifesting at early time near the boundary between the lower saturated and unsaturated zones.

### 5.2 SINUSOIDAL SOIL TEMPERATURE REDISTRIBUTION

In this example, the energy equation is solved for the case of a soil column with a boundary subject to a sinusoidally varying temperature. For a semi-infinite soil column, the cyclic steady-state solution for this problem (see Campbell, 1985) is

$$T(x, t) = T_{mean} + T_{amp} \exp\left(-\frac{x}{D}\right) \sin\left(\omega t - \frac{x}{D}\right), \quad (5-1)$$

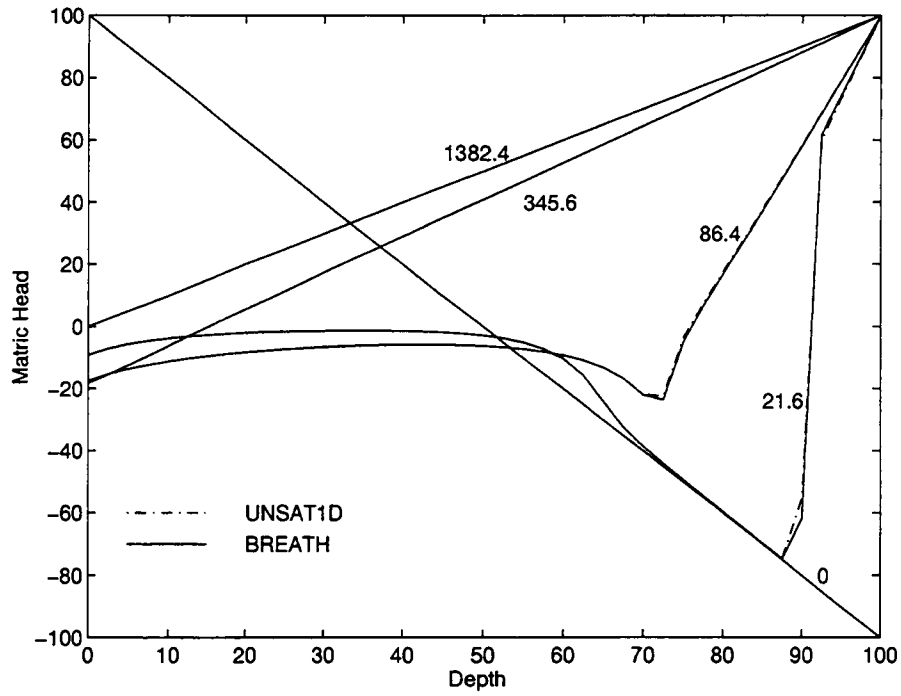


Figure 5-1. Pressure distribution at selected time instants, in the redistribution example

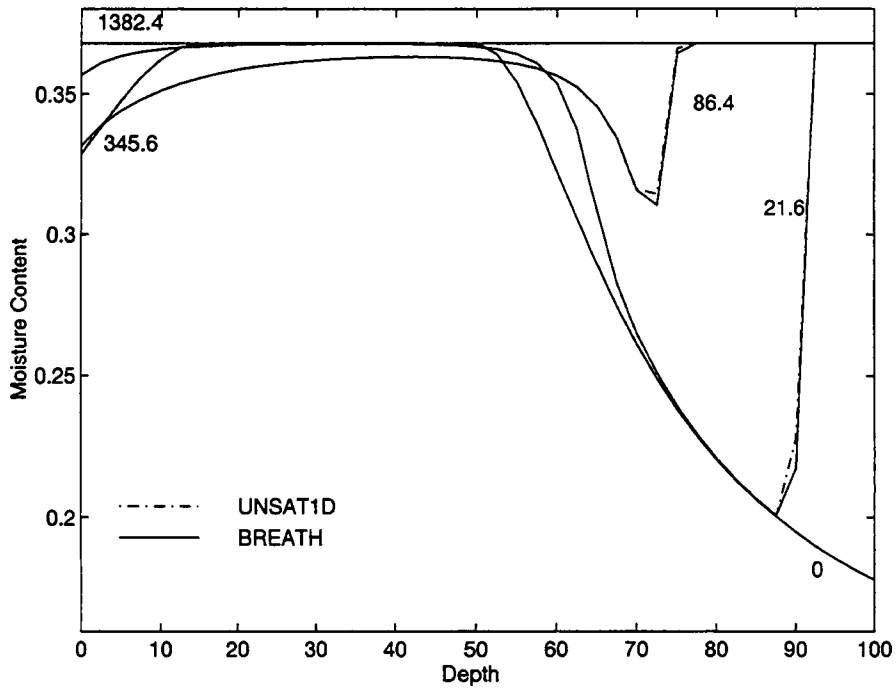


Figure 5-2. Moisture content distribution at selected time instants, in the redistribution example

where

- $T$  = temperature,
- $T_{mean}$  = the mean temperature,
- $T_{amp}$  = the temperature variation amplitude at the boundary,
- $x$  = distance from the boundary,
- $t$  = time,
- $\omega$  = angular frequency of excitation, and
- $D$  = damping distance (distance at which temperature is reduced to  $1/e$ ).

This case is simulated with a column of length 100, evenly discretized with 100 elements and with the second boundary subject to an adiabatic condition (no heat flux). Selecting the parameters  $T_{mean}$  to be 288,  $T_{amp}$  to be 20,  $\omega$  to be  $2\pi / (24 \times 3600)$ , and  $D$  to be 15 results in temperature variation at the second boundary of about 0.13 percent of  $T_{amp}$ . Neglecting advective transport and latent heat (setting  $C_2 = M_1 = M_2 = 0$ ), and using the relationship

$$D = \left( \frac{2M_3}{\omega C_1} \right)^{1/2}, \quad (5-2)$$

allows either  $M_3$  or  $C_1$  to be chosen arbitrarily. By making the thermal properties of two of the three phases (e.g., solid and gas phases) identically zero, thermal conductance in a single phase, the remaining phase, is simulated. Selecting the  $(dU / dT)_V$  parameter to be one results in a corresponding thermal conductivity of  $8.181 \times 10^{-3}$ .

In order to achieve cyclic steady state, the sinusoidal forcing is repeated ten times. Time histories reported in Figure 5-3 are the solutions at various nodes. Identical solutions result whether the single simulated phase is solid, liquid, or vapor; simulation results for one such phase are shown here. Phase and amplitude errors are evident, stemming from errors in representing the sinusoidal boundary condition; as the sinusoidal boundary condition is better resolved temporally, such errors vanish. Input and output files for this problem are shown in Appendices B.3 and B.4, respectively.

### 5.3 ADVANCING TEMPERATURE FRONT

The advancing front problem is a standard problem that simulates both advection and diffusion of a conservative tracer in a semi-infinite column. In this problem, uniform conditions initially exist throughout the domain; the temperature is instantaneously changed at the boundary at time zero. The governing equation for this problem is (Bear, 1972)

$$\frac{\partial T}{\partial t} = D \left( \frac{\partial^2 T}{\partial x^2} \right) - V \left( \frac{\partial T}{\partial x} \right), \quad (5-3)$$

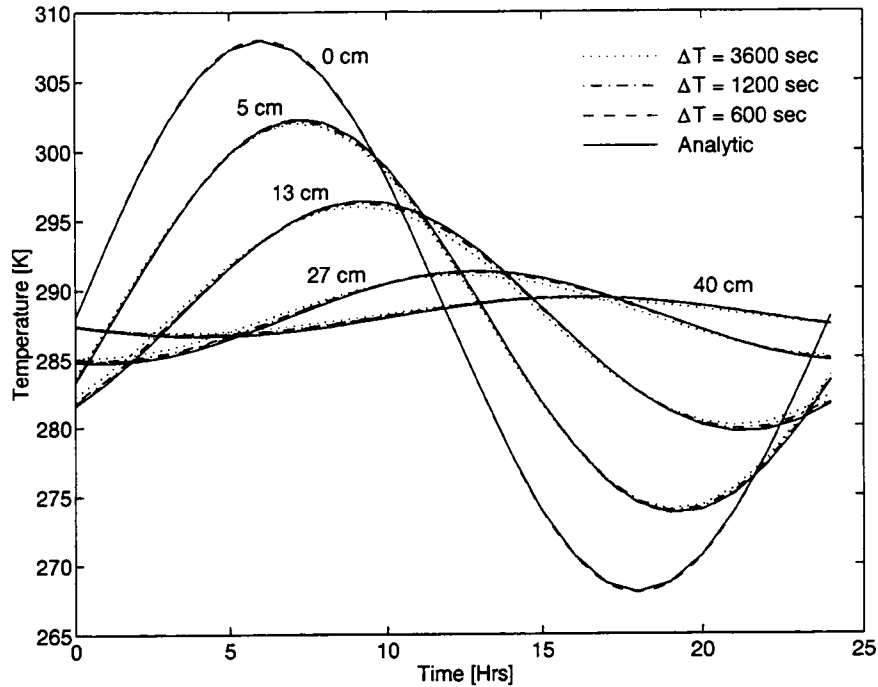


Figure 5-3. Temperature trace at selected depths, in the sinusoidal-forcing example

where

- $T$  = temperature,
- $x$  = distance from the boundary,
- $t$  = time,
- $V$  = velocity of the phase (Darcy flux divided by porosity), and
- $D$  = thermal conductivity of the medium.

Initial and boundary conditions are stated

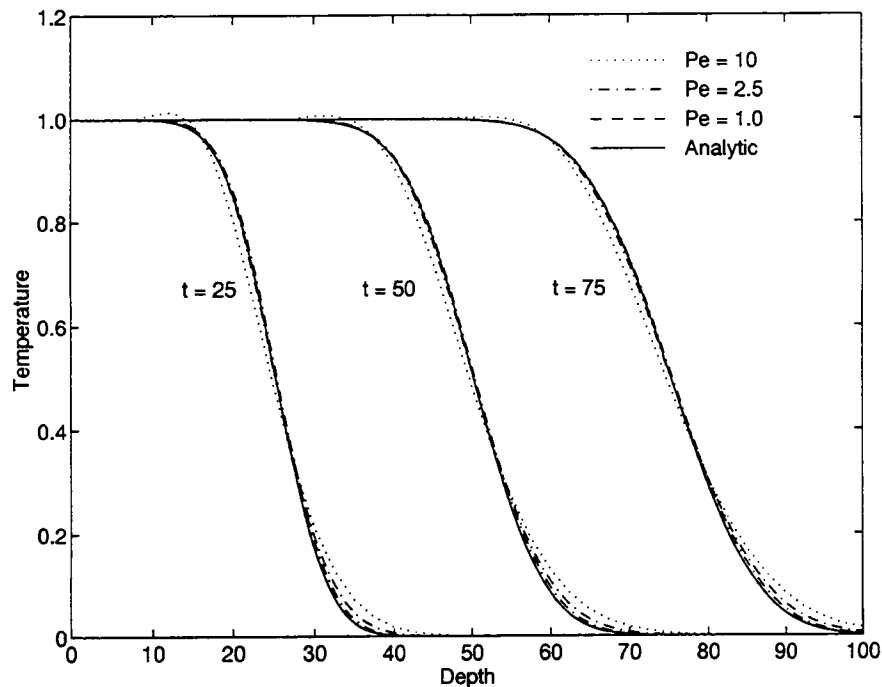
$$T(x, 0) = T_0 \quad (5-4)$$

$$T(0, t) = T_1 \quad (5-5)$$

$$\partial T / \partial x = 0 \quad \text{as } x \rightarrow \infty \quad (5-6)$$

The analytical solution to this problem is obtained by applying the Laplace transform to the problem, solving in Laplace space, and inverting, resulting in

$$\frac{T - T_0}{T_1 - T_0} = \frac{1}{2} \left[ \operatorname{erfc} \left( \frac{x - Vt}{2\sqrt{Dt}} \right) + \exp \left( \frac{xV}{D} \right) \operatorname{erfc} \left( \frac{x + Vt}{2\sqrt{Dt}} \right) \right]. \quad (5-7)$$



**Figure 5-4. Temperature distribution at selected times, in the advancing-front example**

As an example, an evenly discretized domain of length 100 is simulated, with  $V$  fixed at 1 and  $D$  fixed at 0.5. Defining a grid Peclet number,  $Pe = V \Delta x / D$ , and a grid Courant number,  $Co = V \Delta t / \Delta x$ , where  $\Delta x$  and  $\Delta t$  are the element length and the time step, respectively, the problem is simulated three times with varying Peclet numbers but the same Courant number of 0.1. The solutions are shown at three times in Figure 5-4. As expected, decreasing the grid Peclet number has the effect of decreasing numerical, or non-physical, dispersion arising from insufficient resolution of the solution. Rule-of-thumb guidelines suggest that a grid Courant number of 0.1 or less and a grid Peclet number of less than 2 will provide reasonably accurate results for this solution scheme, with accuracy improving as these numbers decrease. The input file for this problem is shown in Appendix B.5, with associated output file and file for nodal time-snapshot information in Appendices B.6 and B.7.

## 5.4 COUPLED SOIL MOISTURE/SOIL TEMPERATURE

A final example problem exercises the fully coupled mode of BREATH in a comparison against version 2.01 of UNSATH (Fayer and Jones, 1990; 1993). The problem of interest represents a purely hypothetical sequence of layers which are present at Yucca Mountain: 30 cm of alluvium (alluv), 180 cm of Tiva Canyon columnar moderately welded tuff (tccmw), and 400 cm of Tiva Canyon shardy base (tcshar). These layers were selected to represent a wide range of properties. Parameters for this case are shown in Tables 5-1 through 5-3. The column is assumed free of plants, but subject to solar radiation, atmospheric and ground-surface emission, and vapor diffusion across an atmospheric boundary layer at the top of the column. The default daily boundary condition modes in UNSATH were mimicked by creating hourly boundary conditions for BREATH based on the UNSATH documentation and source code. Input files for this example are shown in Appendices B.8 and B.9, with the corresponding output file in Appendix B.10.



**Table 5-1. Uniform phase properties in the coupled example**

Phase $\alpha$	$\rho_\alpha$ ( $\text{gm cm}^{-2}$ )	$k_e^\alpha$ ( $\text{J cm}^2 \text{s}^{-1} \text{gm}^{-1} \text{K}^{-1}$ )	$(dU_\alpha/dT)_V$ ( $\text{J gm}^{-1} \text{K}^{-1}$ )	$(dH_\alpha/dT)_P$ ( $\text{J gm}^{-1} \text{K}^{-1}$ )	$H_{l\alpha}$ ( $\text{J gm}^{-1}$ )
Solid	2.6	0.00806	0.84	0.84	0
Water	1.0	0.0058	4.2	4.2	0
Gas/Vapor	0.00125	0.0	1.0	1.0	2501

**Table 5-2. Uniform layer properties in the coupled example**

Layer	$k$ ( $\text{cm}^2 \text{s}^{-1}$ )	$P_0$ (Pa)	$m$	$\theta_{ws}$	$\theta_{wr}$	$\theta_s$
alluv	$1.15 \times 10^{-6}$	$9.80 \times 10^6$	0.231	0.098	0.0001	0.098
tccmw	$6.65 \times 10^{-8}$	$1.14 \times 10^6$	0.231	0.240	0.0010	0.240
tcschar	$1.84 \times 10^{-6}$	$1.14 \times 10^6$	0.231	0.420	0.0010	0.420

**Table 5-3. Miscellaneous constants in the coupled example**

Symbol	Value	Units	Symbol	Value	Units
$R$	$8.3143 \times 10^7$	$\text{cm}^2 \text{gm mole}^{-1} \text{K}^{-1} \text{s}^{-2}$	$M$	18	$\text{gm mole}^{-1}$
$\mu_w$	11.24	$\text{gm cm}^{-1} \text{s}^{-1}$	$D_v$	0.1584	$\text{cm}^2 \text{s}^{-1}$
$g$	981	$\text{cm s}^{-2}$	$k$	0.4	
$z^T$	300	cm	$z^U$	300	cm
$z_h$	0.03	cm	$z_m$	0.03	cm
$d$	0	cm	$\sigma$	$5.67 \times 10^{-12}$	$\text{J s}^{-1} \text{cm}^{-2} \text{K}^{-4}$
$\alpha_s^{dry}$	0.25		$\alpha_s^{wet}$	0.25	
$\epsilon_s^{dry}$	0.9		$\epsilon_s^{wet}$	0.918	

The column starts with uniform initial pressure of  $-10^8$  Pascal and initial temperature of 286 °K. A pulse of water is introduced over the first hour of 2.46 mm, roughly equivalent to 1/3 of the average monthly precipitation for March at Yucca Mountain. This pulse is allowed to redistribute under gravity and evaporate at the top of the column under diurnal forcing for the next six days. At the bottom boundary, water is assumed to undergo gravity drainage (the gradient of moisture content is zero) and is assumed to have negligible vapor flux. Similarly, a zero temperature gradient is assumed to exist at the bottom boundary, although in BREATH water draining from the bottom boundary is allowed to advect energy out of the domain.

Surface boundary conditions are created to mimic UNSATH conditions (Fayer and Jones, 1990), with all conditions sampled at one-hour intervals in the met file (see Appendix A for detailed instructions on met file usage). Atmospheric vapor density,  $\rho_{va}$ , is held constant at  $3.42 \times 10^{-6} \text{ gm cm}^{-3}$ , emissivity of air,  $\epsilon_a$ , is held constant at 0.812, and wind speed is held constant at  $406.8 \text{ cm s}^{-1}$  for the duration of the simulation. Air temperature,  $T_a$ , is allowed to vary sinusoidally according to

$$T_a = T_{mean} + T_{amp} \cos [2\pi (t_d - 15) / 24], \quad (5-8)$$

where  $T_{mean}$  is 280 °K,  $T_{amp}$  is 9.22 °K, and  $t_d$  is the time of day.

Solar radiation is modeled using

$$S_t = S_{ext} T_t \sin(e), \quad (5-9)$$

where the solar constant,  $S_{ext}$ , is  $0.1360 \text{ J s}^{-1} \text{ cm}^{-2}$ ; the transmission coefficient,  $T_t$ , is 0.5; and the solar elevation angle,  $e$ , is defined by

$$\sin(e) = \sin\phi \sin\delta + \cos\phi \cos\delta \cos [2\pi(t_d - 12)/24]. \quad (5-10)$$

In this relationship,  $\phi$  is latitude (37 °) and  $\delta$  is the solar declination angle, calculated using the formula

$$\sin\delta = 0.3985 \sin [4.869 + 0.172J + 0.03345 \sin (6.224 + 0.0172J)], \quad (5-11)$$

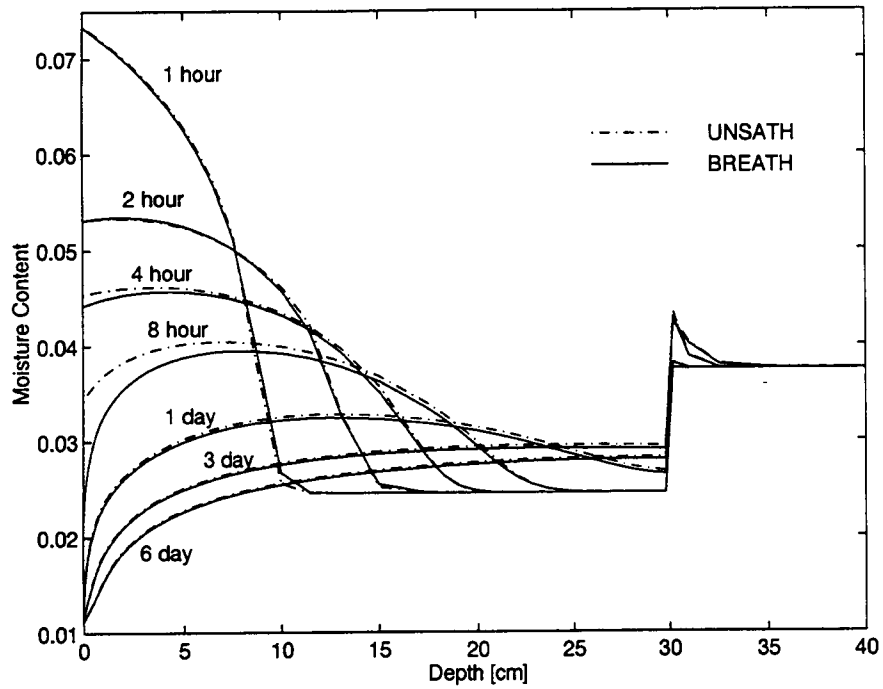
where  $J$  is the day of the year from 1 to 365. The simulation covered the period from days 96 through 101.

The history of the input water pulse is shown in Figure 5-5. The pattern of redistribution is similar for the two models; however, the models disagree somewhat regarding the evaporation occurring during the first day. Both models neglect evaporation for the first hour, during the input pulse; thereafter, BREATH predicts significantly faster evaporation, especially once the sun rises, until a dry zone has been established in the top few millimeters of the domain. Once the dry zone is established, thus limiting evaporation, the two simulators predict nearly identical water content behavior. Thermal profiles are shown in Figure 5-6 at four instants in the final simulated day. The two solutions match quite well, particularly in the well-discretized top 40 cm of the domain. A thermal pulse, akin to the water pulse, is already present more than a meter deep after six days, indicating that the soil is not starting in thermal equilibrium.

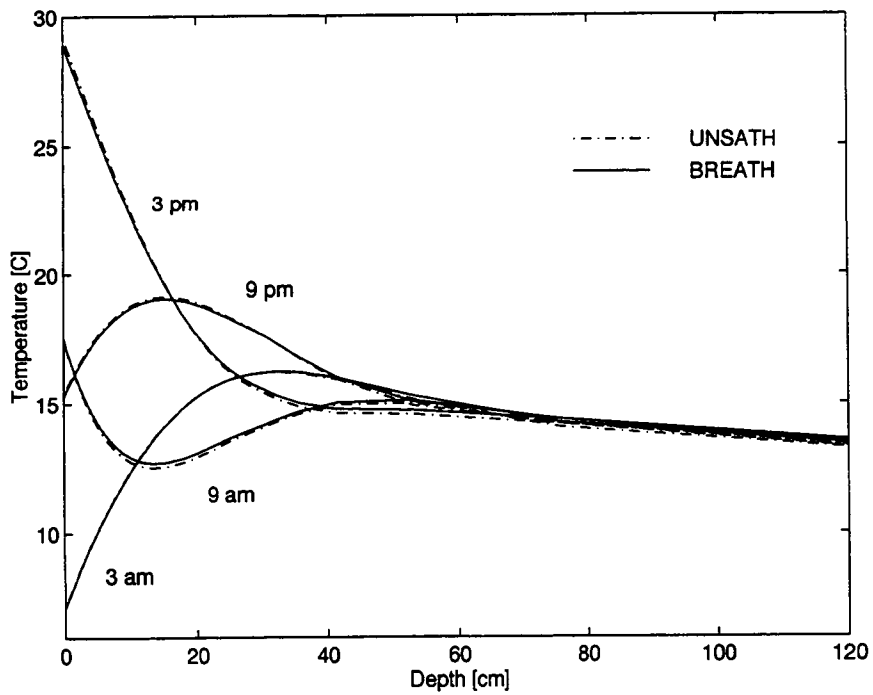
Solutions to this non-linear problem are quite sensitive to the media properties and boundary condition representation, and creating exactly comparable input decks for the two codes was found to be a significant challenge. Relatively minor changes in input yielded significantly different solutions. For example, increasing the albedo by a few hundredths changed the surface temperature by more than 2 °C. Accordingly, it is possible that some of the discrepancy between the early time evaporation-rate predictions may be due to small discrepancies between the input decks. Also, the evaporative flux predictor algorithm in UNSATH is only a rough approximation, perhaps explaining some of the discrepancy during stages of evaporation when the moisture content is rapidly changing.

In attempting to get UNSATH Ver. 2.01 and BREATH to simulate exactly the same problem, UNSATH was modified in several places: in subroutine B\_L\_R, increasing the boundary layer resistance calculation to 10 iterations from 3 iterations; in program DATAINH, correcting a data statement for variable T0; and in program UNSATH, allowing condensation to occur and correcting the statement adding evaporative flux to the right-hand side of the flow equation to match the documentation. Each of the UNSATH modifications aided in bringing the predictions from the two simulators closer.

In general, the overall close match between the two predictions is a reasonable assurance that the two codes are indeed simulating the same problem, particularly in light of the sensitive dependence of the predictions on the input data. The close match builds confidence that BREATH is yielding predictions comparable to a similar widely available simulator.



**Figure 5-5. Response of moisture content to a 1 hr rainfall event, in the coupled example**



**Figure 5-6. Temperature profile at selected times in the sixth day, in the coupled example**

## 6 CURRENT STATUS OF THE BREATH CODE

This first version of the BREATH code is suitable for simulating moisture and energy transport in a porous medium. The mathematical models are largely derived from the UNSAT-H code (Fayer and Jones, 1990; 1993), with the numerical algorithms derived from Celia et al. (1990) and Celia (1991). The presented examples provide assurance that the algorithms in the simulator are correctly coded, based on successful comparison between analytic solutions and other simulators. However, as such comparisons depend on matching simple cases, or two independent simulators, the capabilities of the BREATH simulator have not necessarily been tested with all possible combinations of input parameters.

A number of significant assumptions have been incorporated in this version of BREATH, any of which may be relaxed in future versions of the code. Some of the assumptions that may be relaxed, in no particular order, are discussed below.

- The gas phase does not affect the transport of the liquid phase and is a passive medium for vapor diffusion. These assumptions are known to be weak when considering infiltration fronts, as it is possible for the gas phase to be trapped against low-permeability zones, exert pressure against the invading liquid, and thereby retard the advancing front. Simulating the gas phase is known to improve overall simulation robustness, potentially more than compensating for the additional unknown variables to be determined.
- Vegetation is not considered. This is a strong assumption for arid and semi-arid natural systems. The amount of annual precipitation removed as transpiration is estimated to be as much as 72 percent in the Chihuahuan desert basin (Schlesinger et al., 1987), while studies in the Rock Valley area of the Nevada Test Site yield estimates of 27 percent (Lane et al., 1984). Preliminary studies in the northern part of Yucca Mountain suggest that roughly one-third of annual precipitation is lost by transpiration (Leary, 1990). Transpiration is primarily restricted to a four-to-five month period at the Nevada Test Site; however, desert vegetation may enter dormancy in a dry year, not transpiring, with few ill effects (Beatley, 1974). Simulation of realistic long-term transpiration is not an easy task.
- Two- and three-dimensional flow and transport is not considered. As many of the layers at Yucca Mountain are sloping, this restriction may quite significantly limit the applicability of the current model.
- Only a few constitutive relations are provided for the flow equation. Although the relations are adequate to describe horizontally isotropic systems, the descriptions are not capable of handling more general relationships, such as might arise from horizontal averaging of flow properties.
- The system is assumed to always remain in the temperature range in which phase-change phenomena, such as boiling and freezing, need not be considered. Under the arid near-surface conditions at Yucca Mountain, boiling is not an issue; however, in the winter, air temperatures consistently drop below the freezing point of water, and a limited amount of snowfall is observed. In glacial climatic episodes, freezing effects would presumably increase in importance.
- Heat-of-wetting effects are assumed negligible. This effect would presumably be most significant in a drying zone at the ground surface.

- Thermal conductivity is calculated as a mixture of the phase conductances. Deviations from the straightforward mixing law are not considered. Errors arising from this simplification would be expected to be most significant in a drying zone at the ground surface.
- Hysteresis is not considered in constitutive relationships. Hysteresis is most important when frequent flow reversals occur. In arid and semi-arid environments, the effects of hysteresis may not be particularly important.
- The capability of modeling fully saturated moisture transport is currently allowed; however, questionable assumptions are imposed regarding the behavior of the specific storage coefficient at low moisture content. Numerical experimentation has demonstrated that current options can drastically distort the physical system for some combinations of parameters.
- The boundary-layer evaporation model assumes that the terrain is uniform for a representative fetch upwind. This assumption is questionable in areas of Yucca Mountain, due to significant elevation relief in the washes.
- The current ponding boundary condition is not able to accommodate run-on, which may provide significant amounts of infiltrating water during simulations of wash bottoms or local depressions.
- Albedo is known to be dependent on the incidence angle of incoming solar radiation. Angle dependence is not accommodated with the current formulation.

## 7 REFERENCES

- Bear, J. 1972. *Dynamics of Fluids in Porous Media*. New York, NY: Dover Publications, Inc.
- Brooks, R.H., and A.T. Corey. 1966. Properties of porous media affecting fluid flow. *Journal of the Irrigation and Drainage Division*. ASCE 92 (IR2): 61-88.
- Businger, J.A. 1975. Aerodynamics of vegetated surfaces. *Heat and Mass Transfer in the Biosphere*. D.A. de Vries and N.H. Afgan, eds. New York, NY: John Wiley & Sons.
- Campbell, G.S. 1985. *Soil Physics with BASIC*. Amsterdam: Elsevier Science Publishers.
- Celia, M.A. 1991. *The One-Dimensional Princeton Unsaturated Code*. Princeton, NJ: Princeton University.
- Celia, M.A., E.T. Bouloutas, and R.L. Zarba. 1990. A general mass-conservative numerical solution for the unsaturated flow equation. *Water Resources Research* 26(7): 1,483-1,496.
- CRC. 1982. *CRC Handbook of Chemistry and Physics*. 1982. R.C. Weast and M.J. Astle, eds. Boca Raton, FL: CRC Press, Inc.
- Darcy, H. 1856. *Les Fontaines Publique de la Ville de Digon*. Paris.
- de Vries, D.A. 1958. Simultaneous transfer of heat and moisture in porous media. *Transactions of the American Geophysical Union* 39(5): 909-916.
- de Vries, D.A. 1963. Thermal properties of soils. *Physics of Plant Environment*. W.R. van Wijk, ed. Amsterdam, the Netherlands: North-Holland Publishing Co.
- Electric Power Reserch Institute. 1990. *Demonstration of a Risk-Based Approach to High-Level Waste Repository Evaluation*. EPRI NP-7057. Palo Alto, CA: Electric Power Research Institute.
- Electric Power Reserch Institute. 1992. *Demonstration of a Risk-Based Approach to High-Level Waste Repository Evaluation: Phase 2*. EPRI TR-100384. Palo Alto, CA: Electric Power Research Institute.
- Fayer, M.J., and T.L. Jones. 1990. *UNSAT-H Version 2.0: Unsaturated Soil Water and Heat Flow Model*. PNL-6779, UC-702. Richland, WA: Pacific Northwest Laboratory.
- Fayer, M.J., and T.L. Jones. 1993. *UNSAT-H Version 2.1*. Unpublished code update. Richland, WA: Pacific Northwest Laboratory.
- Gee, G.W. 1966. *Water Movement in Soils as Influenced by Temperature Gradients*. PhD Dissertation. Pullman, WA: Washington State University.
- Green, R.T., R.D. Manteufel, F.T. Dodge, and S.J. Svedeman. 1993. *Theoretical and Experimental Investigation of Thermohydrologic Processes in a Partially Saturated, Fractured Porous Medium*. NUREG/CR-6026, CNWRA 92-006. Washington, DC: Nuclear Regulatory Commission.

- Hillel, D. 1980. *Introduction to Soil Physics*. New York, NY: Academic Press.
- Huyakorn, P.S., and G.F. Pinder. 1983. *Computational Methods in Subsurface Flow*. New York, NY: Academic Press.
- Jensen, O.K. 1992. A review of reservoir simulation techniques and considerations on future developments. *The Mathematics of Oil Recovery*. P.R. King, ed. New York, NY: Oxford University Press.
- Jury, W.A. 1973. *Simultaneous Transport of Heat and Moisture Through a Medium Sand*. PhD Dissertation. Madison, WI: University of Wisconsin.
- Jury, W.A., W.R. Gardner, and W.H. Gardner. 1991. *Soil Physics*. New York, NY: John Wiley & Sons.
- Lapidus, L., and G.F. Pinder. 1982. *Numerical Solution of Partial Differential equations in Science and Engineering*. New York, NY: John Wiley & Sons.
- Landau, L.D., and E.M. Lifshitz. 1959. *Fluid Mechanics*. London, UK: Pergamon Press Ltd.
- Manteufel, R.D., M.P. Ahola, D.R. Turner, and A.H. Chowdhury. 1993. *A Literature Review of Coupled Thermal-Hydrologic-Mechanical-Chemical Processes Pertinent to the Proposed High-Level Nuclear Waste Repository at Yucca Mountain*. NUREG/CR-6021, CNWRA 92-011. Washington, DC: Nuclear Regulatory Commission.
- Milly, P.C.D., and P.S. Eagleson. 1980. *The Coupled Transport of Water and Heat in a Vertical Soil Column Under Atmospheric Excitation*. Ralph Parsons Laboratory Report No. 258. Cambridge, MA: Massachusetts Institute of Technology.
- Montazer, P., D. Hammermeister, and J. Ginanni. 1994. A-TOUGH: A multimedia fluid-flow/energy-transport model for fully-coupled atmospheric-subsurface interactions. *Proceedings of the Fifth Annual International Conference on High Level Radioactive Waste Management*. La Grange Park, IL: American Nuclear Society: 2,333-2,340.
- Mualem, Y. 1976. A new model for predicting the hydraulic conductivity of unsaturated porous media. *Water Resources Research* 12: 513-522.
- Mualem, Y. 1992. Modeling the hydraulic conductivity of unsaturated porous media. *Indirect Methods for Estimating the Hydraulic Properties of Unsaturated Soils*. M.Th. van Genuchten, F.J. Leij, and L.J. Lund, eds. Riverside, CA: University of California.
- Nuclear Regulatory Commission. 1992. *Initial Demonstration of the NRC's Capability to Conduct a Performance Assessment for a High-Level Waste Repository*. NUREG-1327. Washington, DC: Nuclear Regulatory Commission.
- Nuclear Regulatory Commission. 1994. *Phase 2 Demonstration of the NRC's Capability to Conduct a Performance Assessment for a High-Level Waste Repository*. NUREG-1464. Washington, DC: Nuclear Regulatory Commission. In press.
- Peixoto, J.P., and A.H. Oort. 1992. *Physics of Climate*. New York, NY: American Institute of Physics.

- Philip, J.R. 1957. The theory of infiltration: 1. The infiltration equation and its solution. *Soil Science* 83(5): 345-357.
- Philip, J.R., and D.A. de Vries. 1957. Moisture movement in porous materials under temperature gradients. *Transactions of the American Geophysical Union* 38(2): 222-232.
- Prats, M. 1982. *Thermal Recovery*. SPE Monograph 7. New York, NY: Society of Petroleum Engineers of the American Institute of Mining, Metallurgical, and Petroleum Engineers, Inc.
- Reeves, M., N.A. Baker, and J.O. Duguid. 1994. *Review and Selection of Unsaturated Flow Models*. Document No. B00000000-01425-2200-00001, Rev. 00. Las Vegas, NV: INTERA, Inc.
- Richards, L.A. 1931. Capillary conduction of liquids through porous mediums. *Physics* 1: 318-333.
- Rosenberg, N.J. 1974. *Microclimate: The Biological Environment*. New York, NY: John Wiley & Sons.
- Sagar, B., and R.W. Janetzke. 1993. *Total-System Performance Assessment (TPA) Computer Code: Description of Executive Module, Version 2.0*. CNWRA 93-017. San Antonio, TX: Center for Nuclear Waste Regulatory Analyses.
- Sandia National Laboratories. 1992. *TSPA 1991: An Initial Total-System Performance Assessment for Yucca Mountain*. SAND91-2795. Albuquerque, NM: Sandia National Laboratories.
- Sandia National Laboratories. 1994. *Total-System Performance Assessment for Yucca Mountain — SNL Second Iteration (TSPA-1993)*. SAND93-2675. Albuquerque, NM: Sandia National Laboratories.
- Sophocleous, M.A. 1978. *Analysis of Heat and Water Transport in Unsaturated-Saturated Porous Media*. PhD Dissertation. Edmonton, Canada: University of Alberta.
- Swartzendruber, D. 1969. The flow of water in unsaturated soils. *Flow Through Porous Media*. R.J.M. De Wiest, ed. New York, NY: Academic Press.
- van Genuchten, M.Th. 1980. A closed-form equation for predicting the hydraulic conductivity of unsaturated soils. *Soil Science Society of America Journal* 44: 892-898.
- Van Wylen, G.J., and R.E. Sonntag. 1978. *Fundamentals of Classical Thermodynamics*. New York, NY: John Wiley & Sons.



**APPENDIX A:**

**DATA INPUT**

## A.1 INPUT OVERVIEW

Data input to BREATH is done using a simple command language. There are few commands, but each of the commands has several options. This procedure allows the user to select the order in which variables are defined, query the variables as desired, and perform simple file manipulations. Each command has the form of a command word, perhaps followed by several optional modifiers, and ending with a terminal phrase. Commands and modifiers all must be lowercase. Comments, which are signaled using sharps (#), can be interspersed almost at will, with a few notable exceptions described below.

The command language is implemented in FORTRAN, thus some of the common FORTRAN input limitations exist. In order to minimize the limitations, each line of the input file is read as a 132-character string and subsequently interpreted. This practice allows comments to be interspersed freely within the input file. However, when the set command is issued with the "vector" or "indexed" options, standard FORTRAN read statements are used and comments must be compatible with this practice.

A complete list of valid BREATH commands is shown in Table A-1, and are discussed in detail in this chapter. In other tables throughout this chapter, arguments and modifier words will be associated with the commands. Argument names beginning with "z" represent character strings, arguments beginning with "i" or "n" are integers, and other arguments are floating point numbers. Modifier words above the partial line for each argument are optional, and as many of these words as desired can be strung together; modifier words below the partial line signal that the command is finished. Capitalized modifiers are the default, while boldface modifiers must be supplied, and must be the first modifier. Arguments in brackets may or may not be required, depending on on the modifiers supplied.

Table A-1: List of BREATH commands

Command Name	Function
calc	Perform a calculation
echo	Echo the value of a variable to the screen or a file
end	End the simulation and exit BREATH
file	Perform file manipulations (open, close, rewind)
mass	Perform mass balance calculations
met	Define the file containing meteorological (boundary condition) data
quit	End the simulation and exit BREATH
set	Set the value of a variable
snap	Specify the treatment of solution "snapshots"
stop	End the simulation and exit BREATH
trace	Specify the treatment of node/element solution traces

## A.2 INPUT CONTROL COMMANDS

The set command, as shown in Table A-2, is used to input the value of a variable. Each variable (named *ZNAME*) is assumed to be a vector (scalars are vectors of length one). The entire vector can be filled with one value (*global*), part of the vector can be filled (*range*), some limited interpolation can be done (*interp*, *ramp*), or each entry can be filled with different values (*indexed*, *vector*).

**Table A-2: The set command**

Modifier Word	Argument(s)	Function
unit	<i>IOSET</i>	Unit for input (vector, indexed)
format	<i>ZSETFMT</i>	FORTTRAN format (vector, indexed only)
GLOBAL		Set all entries
interp		Interpolate over all entries
ramp	<i>IBEG, IEND</i>	Interpolate over a subset of entries
range	<i>IBEG, IEND</i>	Set a range of entries
indexed		Enter an indexed vector of entries
vector		Enter a vector of entries
<i>ZNAME</i>	<i>VAL[0,VAL1]</i>	Value(s) entered

The **file** command, as shown in Table A-3, is used to manipulate files. This command provides access to standard FORTRAN operations. The *unit* modifier is required, while the *status* and *format* modifiers are optional.

**Table A-3: The file command**

Modifier Word	Argument(s)	Function
unit	<i>IOUNIT</i>	Unit to manipulate
status	<i>ZSTAT</i>	Status of file (old, new, unknown)
format	<i>ZFORM</i>	Status of file (formatted, unformatted)
open	<i>ZNAME</i>	Open the named file
close		Close the unit
rewind		Rewind the unit
backrec	<i>NREC</i>	Backspace <i>NREC</i> records
skiprec	<i>NREC</i>	Skip <i>NREC</i> records

The snippet of a data file in Table A-4 illustrates several facets of the set command. If a command word is not recognized, it is assumed that a “set global” is issued; thus, the shorthand for setting the number of nodes on the first line (*nnd 11*). A complete list of variable names is presented in Tables (A-10) and (A-11).

The next three lines are one command, split up over several lines, with comments interspersed (the sharp characters). The *interp* modifier linearly interpolates from the first value to the second value within the water pressure vector. The next line overwrites the third, fourth, and fifth values of the water pressure vector, setting these entries to  $-200$ . Next, the first three entries of the water pressure vector are ramped from  $-300$  to  $-200$ , linearly interpolating.

The *vector* modifier signals that a list-directed FORTRAN read will occur. This read can be from the current data file, in which case the data must start on a fresh line, or it can be from a file with a unit specified by the unit modifier. The slash (/) ending the line signals that no more data is expected. The *indexed* modifier is identical to the *vector* procedure, except that entry for the data vector is also supplied. Thus, the *temp* vector ends up with values of 290 for the first three nodes, 289 for the next three nodes, and 288 for the remainder.

**Table A-4: Example input file segment**

```
nnd 11
set          # keyword
  interp     # modifier
    xcoor 0 100 # variable name and first and last arguments
set pressw -100
set range 3 5 pressw -200
set ramp 1 3 pressw -300 -200
temp 288 set vector temp
290
290
290 /
set indexed temp
4 289 6 289 5 289 /
file unit 20 open params.dat
set unit 20 format '(i5,f10.0)' indexed satmin
file 20 rewind
set unit 20 format '(t15,i5,f10.0)' indexed satmax
file 20 close
```

Additional control over the read can be provided with a read format. This must be surrounded with quotes if there are any commas in the format. The final sequence opens a file named "params.dat" and refers to the file by unit number 20. The *satmin* vector is read from that file, the file is rewound to its top, and the *satmax* vector is also read before the file is closed. Note that "unit" modifier word is not really required with the file command, as it *must* be the first modifier. Further discussion of formats can be found in any FORTRAN manual.

Boundary conditions and related variables can be input in tabular form, using the *met* command, as shown in Table A-5. This command sets up input of boundary condition variables for when the "calc metdat" command sequence is found. The order of the columns is specified by the order in which the columns are named. When the name of a column is not recognized, it is ignored, allowing columns to be skipped by specifying dummy names. Only the columns that are named are looked for. When no more rows remain, reading from the command file resumes. If the *timmax* variable is negative, reading from the command file resumes as well.

Each row of the input file represents the beginning of stress period of length *timmax*, with the *met* variables interpolated linearly from the beginning of the period to the beginning of the next period and all other boundary condition variables constant. In order to have discontinuous *met* variables between periods, a zero-length period may be used by setting the *timmax* variable to zero (of course, the *timmax* variable must be one of the expected *met* variables in this case). There is often one more row in the *met* file than the number of stress periods, with the final row having a *timmax* of -1.

An example use of the *met* command is shown below. This segment of input-file code,

```
file 31 open metdat.tst
met unit 31 side 0 runtime tmrn dummy bce set
calc metrnr
```

**Table A-5: The met command**

Modifier Word	Argument(s)	Function
unit	<i>IOMET</i>	Unit for input
format	<i>ZMEFMT</i>	FORTTRAN format for input
runtype	<i>ZMERUN</i>	Type of run
	richrun	Moisture only
	thrmrun	Energy only
	tmrun	Coupled energy/moisture
side	<i>MMET</i>	0/1 for side 0,1
[ <i>ZNAME</i> ]		Name of input columns in order
	timmax	Length of stress period
	bcw	Water boundary condition ( $B_w$ )
	bcv	Vapor boundary condition ( $B_v$ )
	bce	Energy boundary condition ( $B_e$ )
	tempa	Atmosphere temperature ( $T_a$ )
	esiga	Air emissivity times Stefan-Boltzmann constant ( $\epsilon_a \sigma$ )
	swrad	Net shortwave radiation ( $S_i$ )
	windsp	Wind speed
set		End of met input

second column is the energy boundary condition for the first side of the domain, and a coupled moisture/thermal simulation is run with successive values read from "metdat.tst" until all rows have been read. It is assumed that any other boundary conditions remain constant throughout the simulation, at least until the end of the file is reached.

### A.3 SIMULATION CONTROL COMMANDS

The command responsible for starting simulations and calculating derived variable values is introduced in Table A-6. The **calc** command calculates some quantity or runs a simulation. When running more than one simulation from the same input file, before each new simulation a number of internal arrays must be initialized. This initialization is done using the *init* modifier to the **calc** command. As a shorthand option, initialization can be performed with just the *init* modifier, omitting the **calc** command word.

### A.4 OUTPUT CONTROL COMMANDS

Three output commands are presented in Tables A-8 and A-8. Immediate output of any variable is allowed with the **echo** command. Periodic output at selected nodes and elements is allowed with the **trace** command, and output for all nodes and all elements is allowed with the **snap** command. The **snap** and **trace** commands have essentially identical specification structure. By default, output is at the end of each *timmax* period; however, each time step can be output as well by selecting the *allstep* modifier word. The nodes and elements are selected by specifying the *idtrnd* and *idtrcl* variables. In order to cut down on output, the *nperskip* option allows several *timmax* periods to be skipped; for example, if hourly periods are calculated, daily output is accomplished by specifying *nperskip* to be 23.

**Table A-6: The calc command**

Modifier Word	Function
densv	Calculate current vapor density
fluxw	Calculate mass fluxes of liquid and vapor
fluxe	Calculate flux of energy
etot	Calculate current total energy
init	Initialize internal variables (this <i>must</i> be the first <b>calc</b> call)
pressw	Calculate pressure from current saturation
satw	Calculate saturation from current pressure
richiter	Take a single Richards iteration
richstep	Take a single Richards step
richrun	Run a Richards-only period
thrmrun	Run a energy-only period
thrmstat	Calculate energy Peclet and Courant numbers
tmrun	Run a coupled Richards-energy period
metrun	Run periods as set up by the <b>met</b> command

**Table A-7: The echo command**

Modifier Word	Argument(s)	Function
unit	<i>IOECH</i>	Unit to echo to
format	<i>ZECHFMT</i>	FORTTRAN format
index	<i>ISON</i>	1/0 to echo/not echo indices
name	<i>ISON</i>	1/0 to echo/not echo name
string	<i>ZECHSTR</i>	Echo character string (annotation)
<i>ZNAME</i>		Name of variable to echo

The **echo** command is friendlier than the other output commands, with output that is intended to be possibly examined by human eyes, thus additional annotation capability is provided using the *string* modifier. Nevertheless, **BREATH** is primarily intended to be a filter between an input-generating program and output-interpreting programs, so minimal attempt has been made to provide attractive output files.

Output for the **snap** and **trace** commands have possibly different units for nodal quantities and elemental quantities, for a total of four possible output files. All output files are designed to be interpreted by other programs, so a FORTRAN format character string, *ZFMT*, is provided for the write statement used for each variable. For both commands, nodal information is processed before elemental information.

**Table A-8: The trace and snap commands**

Modifier Word	Argument(s)	Function
nodeunit	<i>IOTRN</i>	Unit for nodal quantity output
elemunit	<i>IOTRE</i>	Unit for elemental quantity output
format	<i>ZFMT</i>	FORTTRAN format for output numbers
npperskip	<i>NSKIP</i>	Periods to skip between outputs 0 for no skipping
allstep		Output each time step
all		Flag all outputs
nodes		Flag all nodal outputs
elems		Flag all elemental outputs
moisture		Flag all moisture outputs
energy		Flag all energy outputs
tottim		Flag output of total time
pressw		Flag output of water pressure
satw		Flag output of water saturation
densv		Flag output of vapor density
temp		Flag output of temperature
etot		Flag output of total energy
fluxw		Flag output of water flux
fluxv		Flag output of vapor flux
fluxe		Flag output of energy flux
cumqw		Flag output of cumulative water flux
cumqv		Flag output of cumulative vapor flux
cumqe		Flag output of cumulative energy flux
on		Turn on output of flagged variables
off		Turn off output of flagged variables

A **snap** output file has the following structure:

- (i) A line with the number of output variables.
- (ii) A line with the number of nodes or elements.
- (iii) A list of the output variable names, one name per line.
- (iv) The nodal/elemental coordinates, output format specified using *ZFMT*.
- (v) For each output event, the output variables, with output format specified using *ZFMT*, in the order specified by (iii). If the *tottim* variable is enabled, this is output for each event.

A **trace** output file has the following structure:

- (i) A line with the number of output variables.
- (ii) A line with the number of nodes or elements that are being traced.
- (iii) A list of the output variable names, one name per line.

- (iv) The traced nodal/elemental coordinates, each output as a node/element number and coordinate pair on separate lines, with coordinate output format specified using *ZFMT*.
- (v) For each output event, the list of nodes or elements is cycled through, with all variables output at that location before the next location is processed. The output format is specified using *ZFMT*. If the *tottim* variable is enabled, this is output at each location.

## A.5 MASS BALANCE OUTPUT

Mass balance information may be extensively reported in BREATH. It is recognized that overall mass balance information is interesting but may not be sufficient for all needs, thus tracking of boundary fluxes and cumulative fluxes is offered. Like the other output options, the intent is to allow the user to create output suitable for interpretation by an external program, not to make output that is intended for direct user inspection. All mass balance output is strung into one output ASCII file, and the file information is specified with the command. The valid mass balance options are reported in Table A-9.

**Table A-9: The mass command**

<b>Modifier Word</b>	<b>Argument</b>	<b>Function</b>
unit	<i>IOMBAL</i>	Unit number for mass balance file
format	<i>ZMBFMT</i>	FORTTRAN format for output
nperskip	<i>NSKIP</i>	Periods to skip between outputs
allstep		Output each time step (not just each period)
all		Flag all variables
enercum		Flag all cumulative energy variables
enerrate		Flag all energy flux/rate variables
energy		Flag all energy variables
moiscum		Flag all cumulative moisture variables
moisrate		Flag all moisture flux/rate variables
moisture		Flag all moisture variables
bccumv		Flag cumulative vapor boundary flux
bccumw		Flag cumulative liquid boundary flux
bces3		Flag instantaneous third-type boundary flux
bcesb		Flag instantaneous total boundary flux
bcesd		Flag instantaneous energy change minus boundary flux
bcesg		Flag instantaneous gradient (conductive) boundary flux
bcesl		Flag instantaneous longwave boundary flux
bcesn		Flag instantaneous specified boundary flux
bcess		Flag instantaneous shortwave boundary flux
bcesv		Flag instantaneous advective-vapor boundary flux
bcesw		Flag instantaneous advective-liquid boundary flux
bcsrv		Flag instantaneous vapor boundary flux
bcsrw		Flag instantaneous liquid boundary flux
cumde		Flag cumulative energy mass balance



**Table A-9: The mass command (cont'd)**

Modifier Word	Argument	Function
cumdm		Flag cumulative moisture mass balance
cumdv		Flag cumulative liquid mass balance
cumdw		Flag cumulative vapor mass balance
cumes3		Flag cumulative third-type boundary flux
cumesb		Flag cumulative total boundary flux
cumesd		Flag cumulative energy change minus boundary flux
cumesg		Flag cumulative gradient (conductive) boundary flux
cumesl		Flag cumulative longwave boundary flux
cumesn		Flag cumulative specified boundary flux
cumess		Flag cumulative shortwave boundary flux
cumesv		Flag cumulative advective-vapor boundary flux
cumesw		Flag cumulative advective-liquid boundary flux
on		Turn flagged variables on for output
off		Turn flagged variables off for output

## A.6 VARIABLE NAMES

Table A-10 presents numeric variables that can be accessed using the **set** or **echo** commands, and Table A-11 presents character variables that also are accessed using these commands. Names that end in "0,1" represent boundary condition values, where the zero refers to one end and the one refers to the other. These represent separate commands. Variables are either a constant, vectors with values at each node or each element, vectors with the maximum number of trace locations, or character strings. Units are assumed to be of consistent dimensions, with L representing length, M representing mass, T representing time, J representing energy, and K representing temperature. Temperatures are assumed to be in an absolute scale. Capitalized modifiers are the default.

**Table A-10: Numeric variable names and descriptions**

Variable Name	Size	Units	Default	Function
albedd0,1	1	—	1	Dry albedo
albedw0,1	1	—	1	Wet albedo
aleft	nnd	—	—	Calculated left diagonal of system matrix
amidd	nnd	—	—	Calculated middle diagonal of system matrix
arigh	nnd	—	—	Calculated right diagonal of system matrix
bce0,1	1		0	Energy boundary condition value
bcv0,1	1		0	Vapor boundary condition value
bcw0,1	1		0	Water boundary condition value
blkh0,1	1	L T <sup>-1</sup>	0	Boundary layer heat conductance
blkv0,1	1	L T <sup>-1</sup>	0	Boundary layer vapor density conductance
blzoff0,1	1	L	0	Boundary layer offset

**Table A-10: Numeric variable names and descriptions (cont'd)**

Variable Name	Size	Units	Default	Function
blzh0,1	1	L	1	Boundary layer heat roughness height
blzm0,1	1	L	1	Boundary layer momentum roughness height
blzt0,1	1	L	10	Air temperature measurement height
blzu0,1	1	L	10	Wind speed measurement height
cmdecho	1	—	0	1/0 to turn on/turn off command-name echoing
coeflh	1	$J M^{-1}$	0	Coefficient of latent heat
conve	1	—	$10^{32}$	Convergence criterion for energy equation
convf	1	—	$10^{32}$	Convergence criterion for flow equation
coumax	1	—	1	Maximum courant number ( $ztsel = courant$ )
dcoefv	nnd	$L^2 T^{-1}$	0	Vapor diffusion coefficient (times tortuosity)
deltim	1	T	1	Time step size
delxnd	nnd	L	—	Length corresponding to a node
densg	nnd	$ML^{-3}$	0	Density of the gas phase
denss	nnd	$ML^{-3}$	0	Density of the solid matrix
densv	nnd	$ML^{-3}$	0	Vapor density in porous medium
densw	1	$ML^{-3}$	0	Water density
dhdtg	1	$JK^{-1}$	—	$(dH / dT)_P$ for the gas phase
dhdts	nnd	$JK^{-1}$	—	$(dH / dT)_P$ for the solid phase
dhdtv	1	$JK^{-1}$	—	$(dH / dT)_P$ for the vapor
dhdtw	1	$JK^{-1}$	—	$(dH / dT)_P$ for the liquid phase
dtmax	1	T	$10^{32}$	Maximum time step size
dtmin	1	T	0	Minimum time step size
dudtg	1	$JK^{-1}$	—	$(dU / dT)_V$ for the gas phase
dudts	nnd	$JK^{-1}$	—	$(dU / dT)_V$ for the solid phase
dudtv	1	$JK^{-1}$	—	$(dU / dT)_V$ for the vapor
dudtw	1	$JK^{-1}$	—	$(dU / dT)_V$ for the liquid phase
dzdx	1	$LL^{-1}$	0	Spatial gradient of elevation
eintv0	1	J	—	Internal energy of vapor at reference temperature
eintw0	1	J	—	Internal energy of water at reference temperature
ekg	1	$JL^2 T^{-1} K^{-1}$	0	Thermal conductivity of gas phase
eks	nnd	$JL^2 T^{-1} K^{-1}$	0	Thermal conductivity of solid phase
ekw	1	$JL^2 T^{-1} K^{-1}$	0	Thermal conductivity of water phase
eltrace	mnr	—	0	List of elements to trace (end with 0)
enthv0	1	J	—	Enthalpy of vapor at reference temperature

Table A-10: Numeric variable names and descriptions (cont'd)

Variable Name	Size	Units	Default	Function
enthw0	1	J	—	Enthalpy of water at reference temperature
esiga0,1	1	$J T^{-1} L^{-2} K^{-4}$	0	Atmospheric emissivity times Stefan-Boltzmann constant
esigsd0,1	1	$J T^{-1} L^{-2} K^{-4}$	0	Dry soil emissivity times Stefan-Boltzmann constant
esigsw0,1	1	$J T^{-1} L^{-2} K^{-4}$	0	Wet soil emissivity times Stefan-Boltzmann constant
etot	nnd	J	—	Total internal plus latent energy
fluxe	nel	$J T^{-1}$	0	Flux of energy
fluxv	nel	$M L^{-2} T^{-1}$	0	Mass flux of vapor
fluxw	nel	$M L^{-2} T^{-1}$	0	Mass flux of water
frcve	1	—	$10^{-3}$	Fraction of <i>conve</i> determining energy subdomain
frcvf	1	—	$10^{-3}$	Fraction of <i>convf</i> determining flow subdomain
gascon	1	$J \text{ mole}^{-1} K^{-1}$	8.31434	Ideal gas constant R
grav	1	$L T^{-2}$	1	Acceleration due to gravity
iabortup	1	—	0	1/0 to abort immediately with increased convergence
idbmat	1	—	0	1/0 to output matrix information (debug only)
iecall	1	—	0	1/0 to echo each time step
iores	1	—	6	Unit number for result output
ipond0,1	1	—	0	1/0 to allow ponding
kgreedts	1	—	0	1/0 to try a greedy time step (largest possible) to start each period
kitsub	1	—	0	1/0 to use subdomains for iteration
knevap0,1	1	—	1	1/0 to allow evaporation during rain events
maxite	1	—	1	Maximum number of iterations in energy time step
maxitf	1	—	10	Maximum number of iterations in flow time step
ndtrace	mntr	—	0	List of nodes to trace (end with 0)
neitex	1	—	3	Nodes to expand for energy subdomain iterations
nel	1	—	0	Number of elements in the problem
nfitex	1	—	3	Nodes to expand for flow subdomain iterations
nnd	1	—	0	Number of nodes in the problem
nomite	1	—	1	Nominal number of iterations for energy
nomitf	1	—	4	Nominal number of iterations for flow
ntmcyt	1	—	2	Number of flow/energy solution cycles per time step
nts	1	—	0	Maximum number of time steps
pcap0	nnd	$M L^{-1} T^{-2}$	1	van Genuchten scaling pressure (negative)
perm	nnd	$L^2$	0	Intrinsic permeability
pondh0,1	1	L	$10^{32}$	Maximum allowed height of ponding

**Table A-10: Numeric variable names and descriptions (cont'd)**

Variable Name	Size	Units	Default	Function
poros	nnd	$L^3 L^{-3}$	1	Porosity
pressw	nnd	$ML^{-1} T^{-2}$	0	Water pressure
prsmx	1	$ML^{-1} T^{-2}$	0	Maximum allowed pressure from saturation
prsmn	1	$ML^{-1} T^{-2}$	$-10^{32}$	Minimum allowed pressure
rhs	nnd	—	—	Calculated right hand side of matrix equation
rvs1	1		1	Vapor density coefficient 1
rvs2	1		46.440973	Vapor density coefficient 2
rvs3	1		6.02808	Vapor density coefficient 3
rvs4	1		6790.4985	Vapor density coefficient 4
satmax	nnd	$L^3 L^{-3}$	1	Moisture content at saturation
satmin	nnd	$L^3 L^{-3}$	0	Moisture content at residual
satw	nnd	$L^3 L^{-3}$	0	Moisture content
spstor	nnd	$L^{-1}$	0	Specific storage coefficient
srce	nnd	—	0	Calculated energy source terms
srcv	nnd	—	0	Calculated water source terms
srcw	nnd	—	0	Calculated vapor source terms
swrad0,1	1	$J T^{-1} L^{-2}$	0	Incident shortwave radiation
temp	nnd	K	0	Absolute temperature
tempa0,1	1	K	0	Atmospheric temperature
tempex	nnd	K	0	Temperature of external water sources
tempr	1	K	0	Reference temperature for internal energy
timax	1	T	0	Maximum length of time for a simulation
tottim	1	T	0	Length of time since starting simulations
unk	nnd	—	—	Calculated unknowns
vgm	nnd	—	0.5	van Genuchten <i>m</i> parameter
vonk	1	—	0.4	von Karman coefficient [0.4]
viscw	1	$MT^{-1} L^{-1}$	nnd	viscosity of water
viscw0	1	$MT^{-1} L^{-1}$	1	reference viscosity of water
windsp0,1	1	$L T^{-1}$	0	Wind speed
wmolwt	1	$M \text{ mole}^{-1}$	18	Molecular weight of water [18 g/mole]
xcoor	nnd	L	0	nodal coordinate

**Table A-11: Character variable names**

Variable Name	Flag	Function
zbce0,1	temp	energy boundary condition type for <i>bce0,1</i> temperature
	FLUX	flux
	gradtemp	gradient of temperature (conduction specified)
zbcv0,1	densv	vapor boundary condition type for <i>bcv0,1</i> specified vapor density
	FLUX	mass flux of vapor phase
	graddenv	specified gradient of vapor density
	densva	specified atmospheric vapor density (with <i>zblk0,1</i> )
zbcw0,1	press	water boundary condition type for <i>bcw0,1</i> pressure
	satw	saturation
	FLUX	flux
	gradpress	gradient of pressure
	gradsatw	gradient of saturation
	gradhead	$\partial(P_w + \rho_w g z) / \partial x$
zblk0,1	CONSTANT	type of boundary layer calculation for <i>zblk0,1</i> boundary layer conductances constant
	campbell	boundary layer conductances from Campbell scheme
zmosel	VANGEN	type of mobility calculation scheme van Genuchten mobility
	brooks	Brooks-Corey mobility
zpssel	VANGEN	type of pressure/saturation relationship van Genuchten
	brooks	Brooks-Corey
zsssel	EFFSAT	type of specific storage coefficient weighting scheme weight using effective saturation $[(\theta - \theta_r) / (\theta_s - \theta_r)]$
	moisture	weight using relative moisture content $(\theta / \theta_s)$
ztssel	CONSTANT	type of time stepping scheme time step <i>deltim</i> remains constant
	courant	(energy only) time step calculated from <i>coumax</i>
	iterbased	time step calculated from <i>nomitef</i>
zvwsel	CONSTANT	type of wetting-phase viscosity calculation scheme viscosity <i>viscw</i> remains constant
	crctemp	viscosity <i>viscw</i> calculated using CRC scheme

**APPENDIX B:**  
**EXAMPLE INPUT AND OUTPUT LISTINGS**

## B.1 INPUT FILE FOR UNSAT1D COMPARISON

```
# this breath input file is intended to generate solutions
# comparable to unsat1d predictions (Celia, (1991))

cmdecho 0          # don't echo commands during input
nnd 41            # number of nodes
zsssel moisture   # use storage coefficient multiplier from unsat1d
maxitf 50         # maximum number of flow iterations
convf 1e-08       # convergence criterion for flow
dzdx -1          # elevation decreases with length of domain
timmax 1000000    # control maximum time with nts, not timmax

set vector xcoor   # vector-input node coordinates
0 2.5 5 7.5 10 12.5 15 17.5 20
22.5 25 27.5 30 32.5 35 37.5 40
42.5 45 47.5 50 52.5 55 57.5 60
62.5 65 67.5 70 72.5 75 77.5 80
82.5 85 87.5 90 92.5 95 97.5 100

set vector pressw  # vector-input nodal pressure
100 95 90 85 80 75 70 65 60
55 50 45 40 35 30 25 20
15 10 5 0 -5 -10 -15 -20
-25 -30 -35 -40 -45 -50 -55 -60
-65 -70 -75 -80 -85 -90 -95 -100

temp 288          # global-input temperature
dcoefv 0          # global-input vapor diffusion coefficient
densw 1           # water density
viscw 1          # water viscosity
grav 1           # gravitational constant
satmax 0.368     # global-input maximum moisture content
satmin 0.102     # global-input minimum moisture content
pcap0 29.8507    # global-input reference capillary pressure
vgm 0.5          # global-input van genuchten m
perm 0.0092      # global-input intrinsic permeability
spstor 0.0001    # global-input specific storage coefficient
poros 0.368     # global-input porosity
zbcw0 flux       # top boundary condition - set to flux
bcw0 -0          # top boundary condition - flux value
zbcw1 press      # bottom boundary condition - set to pressure
bcw1 100        # bottom boundary condition - pressure value

# document output

echo string ' Example: comparison with UNSAT1D'
echo nnd
echo zsssel
echo zmosel
echo zpsssel
echo maxitf
echo convf
echo dzdx
echo satmin
echo satmax
echo pcap0
echo vgm
```

```
echo perm
echo poros
echo spstor
echo zbcw0
echo bcw0
echo zbcw1
echo bcw1

# prepare graphics output

file unit 20 open fl_brll_40.grf
echo unit 20 name 0 index 0 format (1pe12.3) xcoor

# initialize setup and output initial conditions

init
calc satw
echo unit 20 name 0 index 0 format (1pe12.3) pressw
echo unit 20 name 0 index 0 format (1pe12.3) satw

# run output period

nts 72           # number of time steps in the period
deltim 0.3       # time step
calc richrun     # run simulation
echo pressw      # output water pressure
echo satw        # output water moisture content
echo unit 20 name 0 index 0 format (1pe12.3) pressw
echo unit 20 name 0 index 0 format (1pe12.3) satw

# run to next output

deltim 0.3
nts 216
calc richrun
echo pressw
echo satw
echo unit 20 name 0 index 0 format (1pe12.3) pressw
echo unit 20 name 0 index 0 format (1pe12.3) satw

# run to next output

deltim 0.3
nts 864
calc richrun
echo pressw
echo satw
echo unit 20 name 0 index 0 format (1pe12.3) pressw
echo unit 20 name 0 index 0 format (1pe12.3) satw

# run to final output

deltim 0.3
nts 3456
calc richrun
echo pressw
echo satw
echo unit 20 name 0 index 0 format (1pe12.3) pressw
```

echo unit 20 name 0 index 0 format (lpe12.3) satw  
file unit 20 close

B-2

## B.2 OUTPUT FILE FOR UNSAT1D COMPARISON

```
Example: comparison with UNSAT1D
nnd      = 41
zsssel  = moisture
zmosel  = vangen
zpsael  = vangen
maxitf  = 50
convf   = 1.00000000000000D-08
dzdx    = -1.00000000000000
satmin  [ 1: 41] = 1.02000000000000D-01
satmax  [ 1: 41] = 0.36800000000000
pcap0   [ 1: 41] = 29.850700000000
vgm     [ 1: 41] = 0.50000000000000
perm    [ 1: 41] = 9.20000000000000D-03
poros   [ 1: 41] = 0.36800000000000
spstor  [ 1: 41] = 1.00000000000000D-04
zbcw0   = flux
bcw0    = 0.
zbcwl   = press
bcwl    = 100.000000000000
step 72: nit = 5: tot = 454 deltim = 3.00000E-01 tottim = 2.16000E+01
press1 --9.00262E+00
pressw :
  1 -9.00262E+00  2 -6.98943E+00  3 -5.58371E+00  4 -4.55390E+00
  5 -3.77399E+00  6 -3.16899E+00  7 -2.69145E+00  8 -2.31011E+00
  9 -2.00386E+00 10 -1.75838E+00 11 -1.56416E+00 12 -1.41536E+00
 13 -1.30919E+00 14 -1.24566E+00 15 -1.22764E+00 16 -1.26128E+00
 17 -1.35679E+00 18 -1.52989E+00 19 -1.80422E+00 20 -2.21541E+00
 21 -2.81825E+00 22 -3.70001E+00 23 -5.00664E+00 24 -6.99936E+00
 25 -1.01890E+01 26 -1.56435E+01 27 -2.41067E+01 28 -3.22565E+01
 29 -3.85763E+01 30 -4.40931E+01 31 -4.93643E+01 32 -5.45347E+01
 33 -5.96498E+01 34 -6.47306E+01 35 -6.97886E+01 36 -7.47335E+01
 37 -6.19823E+01 38 6.17094E+01 39 7.43103E+01 40 8.72436E+01
 41 1.00000E+02
satw :
  1 3.56670E-01  2 3.60995E-01  3 3.63465E-01  4 3.64958E-01
  5 3.65899E-01  6 3.66514E-01  7 3.66925E-01  8 3.67207E-01
  9 3.67403E-01 10 3.67540E-01 11 3.67636E-01 12 3.67702E-01
 13 3.67745E-01 14 3.67769E-01 15 3.67775E-01 16 3.67763E-01
 17 3.67726E-01 18 3.67651E-01 19 3.67515E-01 20 3.67270E-01
 21 3.66822E-01 22 3.65980E-01 23 3.64336E-01 24 3.60976E-01
 25 3.53739E-01 26 3.37607E-01 27 3.08944E-01 28 2.82669E-01
 29 2.64787E-01 30 2.51121E-01 31 2.39642E-01 32 2.29719E-01
 33 2.21041E-01 34 2.13393E-01 35 2.06609E-01 36 2.00668E-01
 37 2.17418E-01 38 3.68000E-01 39 3.68000E-01 40 3.68000E-01
 41 3.68000E-01
step 216: nit = 6: tot = 1430 deltim = 3.00000E-01 tottim = 8.64000E+01
press1 --1.74472E+01
pressw :
  1 -1.74472E+01  2 -1.52844E+01  3 -1.36026E+01  4 -1.22486E+01
  5 -1.11317E+01  6 -1.01937E+01  7 -9.39575E+00  8 -8.71059E+00
  9 -8.11886E+00 10 -7.60655E+00 11 -7.16360E+00 12 -6.78298E+00
 13 -6.46015E+00 14 -6.19272E+00 15 -5.98035E+00 16 -5.82476E+00
 17 -5.72996E+00 18 -5.70260E+00 19 -5.75260E+00 20 -5.89419E+00
 21 -6.14740E+00 22 -6.54056E+00 23 -7.11438E+00 24 -7.92889E+00
 25 -9.07617E+00 26 -1.07051E+01 27 -1.30729E+01 28 -1.66476E+01
 29 -2.20441E+01 30 -2.36761E+01 31 -4.98598E+01 32 6.60687E+00
```



```

33 1.67669E+01 34 2.70062E+01 35 3.73133E+01 36 4.76773E+01
37 5.80879E+01 38 6.85352E+01 39 7.90098E+01 40 8.95023E+01
41 1.00000E+02
satw :
1 3.31650E-01 2 3.38767E-01 3 3.44053E-01 4 3.48089E-01
5 3.51234E-01 6 3.53727E-01 7 3.55728E-01 8 3.57350E-01
9 3.58676E-01 10 3.59763E-01 11 3.60656E-01 12 3.61388E-01
13 3.61981E-01 14 3.62454E-01 15 3.62817E-01 16 3.63076E-01
17 3.63231E-01 18 3.63275E-01 19 3.63194E-01 20 3.62961E-01
21 3.62533E-01 22 3.61836E-01 23 3.60753E-01 24 3.59085E-01
25 3.56496E-01 26 3.52386E-01 27 3.45658E-01 28 3.34315E-01
29 3.15978E-01 30 3.10406E-01 31 3.64365E-01 32 3.68000E-01
33 3.68000E-01 34 3.68000E-01 35 3.68000E-01 36 3.68000E-01
37 3.68000E-01 38 3.68000E-01 39 3.68000E-01 40 3.68000E-01
41 3.68000E-01
step 864: nit = 4: tot = 3716 deltim = 3.00000E-01 tottim = 3.45600E+02
press1 --1.82610E+01
pressw :
1 -1.82610E+01 2 -1.55142E+01 3 -1.24734E+01 4 -9.33722E+00
5 -6.21798E+00 6 -3.16572E+00 7 -1.95590E-01 8 2.72829E+00
9 5.65171E+00 10 8.57739E+00 11 1.15052E+01 12 1.44352E+01
13 1.73672E+01 14 2.03011E+01 15 2.32369E+01 16 2.61746E+01
17 2.91140E+01 18 3.20551E+01 19 3.49977E+01 20 3.79419E+01
21 4.08876E+01 22 4.38346E+01 23 4.67830E+01 24 4.97326E+01
25 5.26835E+01 26 5.56354E+01 27 5.85883E+01 28 6.15422E+01
29 6.44971E+01 30 6.74527E+01 31 7.04092E+01 32 7.33663E+01
33 7.63240E+01 34 7.92823E+01 35 8.22411E+01 36 8.52003E+01
37 8.81598E+01 38 9.11196E+01 39 9.40796E+01 40 9.70398E+01
41 1.00000E+02
satw :
1 3.28909E-01 2 3.38026E-01 3 3.47434E-01 4 3.55870E-01
5 3.62410E-01 6 3.66517E-01 7 3.67994E-01 8 3.68000E-01
9 3.68000E-01 10 3.68000E-01 11 3.68000E-01 12 3.68000E-01
13 3.68000E-01 14 3.68000E-01 15 3.68000E-01 16 3.68000E-01
17 3.68000E-01 18 3.68000E-01 19 3.68000E-01 20 3.68000E-01
21 3.68000E-01 22 3.68000E-01 23 3.68000E-01 24 3.68000E-01
25 3.68000E-01 26 3.68000E-01 27 3.68000E-01 28 3.68000E-01
29 3.68000E-01 30 3.68000E-01 31 3.68000E-01 32 3.68000E-01
33 3.68000E-01 34 3.68000E-01 35 3.68000E-01 36 3.68000E-01
37 3.68000E-01 38 3.68000E-01 39 3.68000E-01 40 3.68000E-01
41 3.68000E-01
step 3456: nit = 1: tot = 7858 deltim = 3.00000E-01 tottim = 1.38240E+03
press1 --1.02422E-07
pressw :
1 -1.02422E-07 2 2.50000E+00 3 5.00000E+00 4 7.50000E+00
5 1.00000E+01 6 1.25000E+01 7 1.50000E+01 8 1.75000E+01
9 2.00000E+01 10 2.25000E+01 11 2.50000E+01 12 2.75000E+01
13 3.00000E+01 14 3.25000E+01 15 3.50000E+01 16 3.75000E+01
17 4.00000E+01 18 4.25000E+01 19 4.50000E+01 20 4.75000E+01
21 5.00000E+01 22 5.25000E+01 23 5.50000E+01 24 5.75000E+01
25 6.00000E+01 26 6.25000E+01 27 6.50000E+01 28 6.75000E+01
29 7.00000E+01 30 7.25000E+01 31 7.50000E+01 32 7.75000E+01
33 8.00000E+01 34 8.25000E+01 35 8.50000E+01 36 8.75000E+01
37 9.00000E+01 38 9.25000E+01 39 9.50000E+01 40 9.75000E+01
41 1.00000E+02
satw [ 1: 41] = 0.368000000000000

```

### B.3 INPUT FILE FOR SINUSOIDAL EXAMPLE

```

# Compare to Campbell analytical equation
# energy simulation only

cmdecho 0 # don't echo commands on entry
nnd 101 # use 101 nodes
set interp xcoor 0 100 # evenly distribute nodes

# linear problem, so force one iteration and ignore convergence

maxite 1 # maximum number of energy interations
nomite 1 # nominal number of energy interations
conve 1e+4 # convergence criteria, artificially large
ztsel constant # constant time step

# artificially consider water only

grav 980 # gravity (cm/sec^2)
densw 1 # gm / cm^3
dudtw 1 # J/gm K (dudtw = Ch/densw, Ch = 1)
ekw 8.181E-3 # J/s cm K

# Initial and boundary conditions

temp 288 # initial conditions
tempr 288 # reference temp
tempex 0 # temp of external sources
poros .5 # porosity
satw .5 # initial moisture content

zbcel gradtemp # zero-gradient for right side temperature
bcel 0

zbce0 temp # sinusoidal BC for left side temperature
file 31 open sin_t.out # use sinusoid defined in external file
met unit 31 side 0 runtime thrmrn timmax bce set

nts 10000 # use timmax in external file to control time period
deltim 1800 # specified time step

file 20 open si_br_100.output
echo unit 20 string 'Example: comparison with sinusoidal analytic solution'
echo unit 20 nnd
echo unit 20 deltim
echo unit 20 maxite
echo unit 20 conve
echo unit 20 densw
echo unit 20 dudtw
echo unit 20 ekw
echo unit 20 poros
echo unit 20 satw
echo unit 20 zbce0
echo unit 20 zbcel
echo unit 20 bcel

# initialize setup

```

```

init
end

#Day 1
  calc metrun

#Day 2
  file 31 rewind
  calc metrun

#Day 3
  file 31 rewind
  calc metrun

#Day 4
  file 31 rewind
  calc metrun

#Day 5
  file 31 rewind
  calc metrun

#Day 6
  file 31 rewind
  calc metrun

#Day 7
  file 31 rewind
  calc metrun

#Day 8
  file 31 rewind
  calc metrun

#Day 9
  file 31 rewind
  calc metrun

#Day 10 - turn on output reporting

# define trace and snap output files

#   file 22 open ndtrace.wo
#   file 23 open eltrace.wo
#   file 24 open ndsnap.wo
#   file 25 open elsnap.wo
#   set vector ndtrace
#   1 6 14 28 41 /
#   set vector eltrace
#   1 6 14 28 41 /

#trace nodeunit 22 elemunit 23 tottim temp fluxe format (1pe11.4) nperskip 0
on
#snap nodeunit 24 elemunit 25 temp format (1pe11.4) nperskip 0 on

tottim 0           # reset time to zero
file 31 rewind
calc metrun

echo unit 20 temp   # echo final solutions
echo unit 20 fluxe

file 20 close
#   file 22 close
#   file 23 close
#   file 24 close
#   file 25 close

```

B-4

## B.4 OUTPUT FILE FOR SINUSOIDAL EXAMPLE

Example: comparison with sinusoidal analytic solution

```
nnd      = 101
deltim   = 1800.0000000000
maxite    = 1
conve    = 10000.0000000000
densw    = 1.00000000000000
dudtw    = 1.00000000000000
ekw      = 8.18100000000000D-03
poros    [ 1: 101] = 0.500000000000000
satw     [ 1: 101] = 0.500000000000000
zbcce0   = temp
zbccl    = gradtemp
bcel     = 0.
```

```
temp
:
 1 2.68000E+02  2 2.69389E+02  3 2.70749E+02  4 2.72073E+02
 5 2.73357E+02  6 2.74595E+02  7 2.75785E+02  8 2.76921E+02
 9 2.78003E+02 10 2.79029E+02 11 2.79996E+02 12 2.80906E+02
13 2.81757E+02 14 2.82551E+02 15 2.83287E+02 16 2.83969E+02
17 2.84595E+02 18 2.85170E+02 19 2.85694E+02 20 2.86170E+02
21 2.86600E+02 22 2.86987E+02 23 2.87332E+02 24 2.87638E+02
25 2.87908E+02 26 2.88144E+02 27 2.88349E+02 28 2.88524E+02
29 2.88673E+02 30 2.88797E+02 31 2.88898E+02 32 2.88979E+02
33 2.89042E+02 34 2.89088E+02 35 2.89119E+02 36 2.89137E+02
37 2.89143E+02 38 2.89139E+02 39 2.89127E+02 40 2.89106E+02
41 2.89079E+02 42 2.89047E+02 43 2.89010E+02 44 2.88970E+02
45 2.88927E+02 46 2.88882E+02 47 2.88835E+02 48 2.88788E+02
49 2.88740E+02 50 2.88692E+02 51 2.88645E+02 52 2.88599E+02
53 2.88553E+02 54 2.88509E+02 55 2.88467E+02 56 2.88426E+02
57 2.88387E+02 58 2.88350E+02 59 2.88315E+02 60 2.88282E+02
61 2.88251E+02 62 2.88222E+02 63 2.88195E+02 64 2.88170E+02
65 2.88147E+02 66 2.88125E+02 67 2.88106E+02 68 2.88088E+02
69 2.88072E+02 70 2.88057E+02 71 2.88044E+02 72 2.88033E+02
73 2.88022E+02 74 2.88013E+02 75 2.88005E+02 76 2.87998E+02
77 2.87992E+02 78 2.87987E+02 79 2.87982E+02 80 2.87979E+02
81 2.87976E+02 82 2.87973E+02 83 2.87972E+02 84 2.87970E+02
85 2.87969E+02 86 2.87968E+02 87 2.87968E+02 88 2.87968E+02
89 2.87968E+02 90 2.87968E+02 91 2.87968E+02 92 2.87969E+02
93 2.87969E+02 94 2.87969E+02 95 2.87970E+02 96 2.87970E+02
97 2.87970E+02 98 2.87971E+02 99 2.87971E+02 100 2.87971E+02
101 2.87971E+02
```

```
fluxe
:
 1 -5.68223E-03  2 -5.56198E-03  3 -5.41751E-03  4 -5.25118E-03
 5 -5.06579E-03  6 -4.86440E-03  7 -4.65006E-03  8 -4.42576E-03
 9 -4.19432E-03 10 -3.95837E-03 11 -3.72028E-03 12 -3.48220E-03
13 -3.24602E-03 14 -3.01342E-03 15 -2.78581E-03 16 -2.56444E-03
17 -2.35030E-03 18 -2.14424E-03 19 -1.94691E-03 20 -1.75884E-03
21 -1.58037E-03 22 -1.41175E-03 23 -1.25312E-03 24 -1.10450E-03
25 -9.65843E-04 26 -8.37008E-04 27 -7.17805E-04 28 -6.07982E-04
29 -5.07244E-04 30 -4.15259E-04 31 -3.31663E-04 32 -2.56072E-04
33 -1.88082E-04 34 -1.27281E-04 35 -7.32482E-05 36 -2.55615E-05
37 1.61997E-05 38 5.24508E-05 39 8.35996E-05 40 1.10044E-04
41 1.32169E-04 42 1.50348E-04 43 1.64936E-04 44 1.76274E-04
45 1.84687E-04 46 1.90480E-04 47 1.93943E-04 48 1.95347E-04
49 1.94945E-04 50 1.92972E-04 51 1.89648E-04 52 1.85174E-04
53 1.79735E-04 54 1.73502E-04 55 1.66627E-04 56 1.59252E-04
57 1.51502E-04 58 1.43491E-04 59 1.35320E-04 60 1.27077E-04
```

```
61 1.18842E-04 62 1.10682E-04 63 1.02658E-04 64 9.48184E-05
65 8.72072E-05 66 7.98594E-05 67 7.28038E-05 68 6.60631E-05
69 5.96546E-05 70 5.35910E-05 71 4.78803E-05 72 4.25269E-05
73 3.75317E-05 74 3.28927E-05 75 2.86052E-05 76 2.46625E-05
77 2.10559E-05 78 1.77750E-05 79 1.48083E-05 80 1.21432E-05
81 9.76623E-06 82 7.66323E-06 83 5.81964E-06 84 4.22055E-06
85 2.85087E-06 86 1.69542E-06 87 7.39047E-07 88 -3.33409E-08
89 -6.36679E-07 90 -1.08572E-06 91 -1.39499E-06 92 -1.57879E-06
93 -1.65116E-06 94 -1.62589E-06 95 -1.51657E-06 96 -1.33651E-06
97 -1.09889E-06 98 -8.16655E-07 99 -5.02668E-07 100 -1.69667E-07
```

## B.5 INPUT FILE FOR ADVANCING-FRONT EXAMPLE

```
# Compare to advancing-front advection-diffusion analytical solution
# Use only the liquid phase in the energy equation to handle transport
```

```
nnd      81          # number of nodes
set interp xcoor 0 100 # evenly spaced nodes

# linear problem, so force one iteration
maxite 1          # maximum number of energy iterations
nomite 1          # nominal number of energy iterations
conve 1e+4        # artificially large convergence criterion
ztssel courant    # set the time step from a courant criterion
coumax 0.1        # maximum courant number for any element

poros 1          # ignore solid phase
satw 1           # ignore vapor species
densw 1          # unit density
fluxw 1          # unit flux
dudtw 1          # make etot = T - T0
ekw 0.10         # diffusion coefficient
temp 285         # initial condition
tempr 285        # minimize roundoff by setting Tref = IC
zbce0 temp       # first boundary condition
bce0 286         # unit increase over background
zbcel gradtemp   # second boundary condition
bcel 0           # zero-gradient
```

```
# output periodic values of selected variables
```

```
#file 22 open cdfont.ntr
#file 23 open cdfont.etr
file 24 open cdfont.nsn
file 25 open cdfont.esn
#set vector ndtrace
#1 10 21 54 78 114 161 0/
#set vector eltrace
#1 10 21 54 78 114 160 0/
#trace nodeunit 22 elemunit 23 tottim etot fluxe format (1pe11.4) allstep on
snap nodeunit 24 elemunit 25 etot fluxe format (6(1pe11.4)) nperskip 0 on
```

```
# time stepping
```

```
nts      10000     # control time period with timmax
timmax 25          # length of control period
```

```
calc init          # initialize
deltim 0.75        # set the time step
calc thmrun        # run a simulation period
```

```
deltim 0.75        # set the time step
calc thmrun        # run a simulation period
```

```
deltim 0.75        # set the time step
calc thmrun        # run a simulation period
```

```
# output results
```

```
file 20 open cdfont.out
iores 20          # set default result echo unit to output file
```

```
echo string ' Advancing front problem for advection-diffusion equation'
echo string ' '
echo nnd
echo densw
echo fluxw
echo ekw
echo poros
echo satw
echo zbce0
echo bce0
echo zbcel
echo bcel
```

```
calc fluxe
calc etot
calc thmstat
```

```
echo format '(a/5(i4,1pe11.4))' xcoor
echo format '(a/5(i4,1pe11.4))' temp
echo format '(a/5(i4,1pe11.4))' etot
echo format '(a/5(i4,1pe11.4))' fluxe
```

```
end
```

## B.6 OUTPUT FILE FOR ADVANCING-FRONT EXAMPLE

Advancing front problem for advection-diffusion equation

```

nnd      =      81
densw    =      1.000000000000000
fluxw    [ 1: 80] =      1.000000000000000
ekw      =      1.000000000000000D-01
poros    [ 1: 81] =      1.000000000000000
satw     [ 1: 81] =      1.000000000000000
zbce0    = temp
bce0     =      286.0000000000000
zbcel    = gradtemp
bcel     =      0.
minimum grid peclat =      12.5000000000000
maximum grid peclat =      12.5000000000000
minimum grid courant =      9.9999999999997D-02
maximum grid courant =      9.9999999999997D-02
xcoor
  1 0.0000E+00  2 1.2500E+00  3 2.5000E+00  4 3.7500E+00  5 5.0000E+00
  6 6.2500E+00  7 7.5000E+00  8 8.7500E+00  9 1.0000E+01 10 1.1250E+01
 11 1.2500E+01 12 1.3750E+01 13 1.5000E+01 14 1.6250E+01 15 1.7500E+01
 16 1.8750E+01 17 2.0000E+01 18 2.1250E+01 19 2.2500E+01 20 2.3750E+01
 21 2.5000E+01 22 2.6250E+01 23 2.7500E+01 24 2.8750E+01 25 3.0000E+01
 26 3.1250E+01 27 3.2500E+01 28 3.3750E+01 29 3.5000E+01 30 3.6250E+01
 31 3.7500E+01 32 3.8750E+01 33 4.0000E+01 34 4.1250E+01 35 4.2500E+01
 36 4.3750E+01 37 4.5000E+01 38 4.6250E+01 39 4.7500E+01 40 4.8750E+01
 41 5.0000E+01 42 5.1250E+01 43 5.2500E+01 44 5.3750E+01 45 5.5000E+01
 46 5.6250E+01 47 5.7500E+01 48 5.8750E+01 49 6.0000E+01 50 6.1250E+01
 51 6.2500E+01 52 6.3750E+01 53 6.5000E+01 54 6.6250E+01 55 6.7500E+01
 56 6.8750E+01 57 7.0000E+01 58 7.1250E+01 59 7.2500E+01 60 7.3750E+01
 61 7.5000E+01 62 7.6250E+01 63 7.7500E+01 64 7.8750E+01 65 8.0000E+01
 66 8.1250E+01 67 8.2500E+01 68 8.3750E+01 69 8.5000E+01 70 8.6250E+01
 71 8.7500E+01 72 8.8750E+01 73 9.0000E+01 74 9.1250E+01 75 9.2500E+01
 76 9.3750E+01 77 9.5000E+01 78 9.6250E+01 79 9.7500E+01 80 9.8750E+01
 81 1.0000E+02
temp
  1 2.8600E+02  2 2.8600E+02  3 2.8600E+02  4 2.8600E+02  5 2.8600E+02
  6 2.8600E+02  7 2.8600E+02  8 2.8600E+02  9 2.8600E+02 10 2.8600E+02
 11 2.8600E+02 12 2.8600E+02 13 2.8600E+02 14 2.8600E+02 15 2.8600E+02
 16 2.8600E+02 17 2.8600E+02 18 2.8600E+02 19 2.8600E+02 20 2.8600E+02
 21 2.8600E+02 22 2.8600E+02 23 2.8600E+02 24 2.8600E+02 25 2.8600E+02
 26 2.8600E+02 27 2.8600E+02 28 2.8600E+02 29 2.8600E+02 30 2.8600E+02
 31 2.8600E+02 32 2.8600E+02 33 2.8600E+02 34 2.8600E+02 35 2.8600E+02
 36 2.8600E+02 37 2.8600E+02 38 2.8600E+02 39 2.8600E+02 40 2.8600E+02
 41 2.8600E+02 42 2.8600E+02 43 2.8600E+02 44 2.8600E+02 45 2.8600E+02
 46 2.8600E+02 47 2.8600E+02 48 2.8600E+02 49 2.8601E+02 50 2.8602E+02
 51 2.8602E+02 52 2.8602E+02 53 2.8602E+02 54 2.8599E+02 55 2.8595E+02
 56 2.8589E+02 57 2.8582E+02 58 2.8574E+02 59 2.8564E+02 60 2.8555E+02
 61 2.8545E+02 62 2.8537E+02 63 2.8529E+02 64 2.8523E+02 65 2.8517E+02
 66 2.8513E+02 67 2.8509E+02 68 2.8506E+02 69 2.8504E+02 70 2.8503E+02
 71 2.8502E+02 72 2.8501E+02 73 2.8501E+02 74 2.8501E+02 75 2.8500E+02
 76 2.8500E+02 77 2.8500E+02 78 2.8500E+02 79 2.8500E+02 80 2.8500E+02
 81 2.8500E+02
etot
  1 1.0000E+00  2 1.0000E+00  3 1.0000E+00  4 1.0000E+00  5 1.0000E+00
  6 1.0000E+00  7 1.0000E+00  8 1.0000E+00  9 1.0000E+00 10 1.0000E+00
 11 1.0000E+00 12 1.0000E+00 13 1.0000E+00 14 1.0000E+00 15 1.0000E+00

```

B-7

```

16 1.0000E+00 17 1.0000E+00 18 1.0000E+00 19 1.0000E+00 20 1.0000E+00
21 1.0000E+00 22 1.0000E+00 23 1.0000E+00 24 1.0000E+00 25 9.9999E-01
26 9.9998E-01 27 1.0000E+00 28 1.0000E+00 29 1.0000E+00 30 1.0000E+00
31 9.9993E-01 32 9.9992E-01 33 1.0000E+00 34 1.0001E+00 35 1.0002E+00
36 1.0001E+00 37 9.9976E-01 38 9.9950E-01 39 9.9958E-01 40 1.0001E+00
41 1.0010E+00 42 1.0015E+00 43 1.0011E+00 44 9.9942E-01 45 9.9702E-01
46 9.9511E-01 47 9.9530E-01 48 9.9888E-01 49 1.0060E+00 50 1.0151E+00
51 1.0229E+00 52 1.0249E+00 53 1.0163E+00 54 9.9311E-01 55 9.5275E-01
56 8.9488E-01 57 8.2134E-01 58 7.3574E-01 59 6.4280E-01 60 5.4764E-01
61 4.5499E-01 62 3.6874E-01 63 2.9164E-01 64 2.2521E-01 65 1.6991E-01
66 1.2530E-01 67 9.0385E-02 68 6.3809E-02 69 4.4114E-02 70 2.9883E-02
71 1.9846E-02 72 1.2929E-02 73 8.2656E-03 74 5.1887E-03 75 3.1998E-03
76 1.9391E-03 77 1.1560E-03 78 6.7628E-04 79 3.9314E-04 80 2.1559E-04
81 1.3888E-04

```

```

fluxe
  1 1.0000E+00  2 1.0000E+00  3 1.0000E+00  4 1.0000E+00  5 1.0000E+00
  6 1.0000E+00  7 1.0000E+00  8 1.0000E+00  9 1.0000E+00 10 1.0000E+00
 11 1.0000E+00 12 1.0000E+00 13 1.0000E+00 14 1.0000E+00 15 1.0000E+00
 16 1.0000E+00 17 1.0000E+00 18 1.0000E+00 19 1.0000E+00 20 1.0000E+00
 21 1.0000E+00 22 1.0000E+00 23 1.0000E+00 24 1.0000E+00 25 9.9998E-01
 26 9.9999E-01 27 1.0000E+00 28 1.0000E+00 29 1.0000E+00 30 9.9997E-01
 31 9.9992E-01 32 9.9995E-01 33 1.0001E+00 34 1.0002E+00 35 1.0002E+00
 36 9.9994E-01 37 9.9965E-01 38 9.9953E-01 39 9.9981E-01 40 1.0005E+00
 41 1.0012E+00 42 1.0013E+00 43 1.0004E+00 44 9.9842E-01 45 9.9622E-01
 46 9.9519E-01 47 9.9681E-01 48 1.0019E+00 49 1.0098E+00 50 1.0183E+00
 51 1.0237E+00 52 1.0213E+00 53 1.0066E+00 54 9.7616E-01 55 9.2844E-01
 56 8.6400E-01 57 7.8539E-01 58 6.9671E-01 59 6.0283E-01 60 5.0873E-01
 61 4.1876E-01 62 3.3636E-01 63 2.6374E-01 64 2.0198E-01 65 1.5117E-01
 66 1.1064E-01 67 7.9223E-02 68 5.5537E-02 69 3.8137E-02 70 2.5668E-02
 71 1.6941E-02 72 1.0970E-02 73 6.9733E-03 74 4.3534E-03 75 2.6703E-03
 76 1.6102E-03 77 9.5450E-04 78 5.5737E-04 79 3.1857E-04 80 1.8337E-04

```

Element Snap File for Advancing Front

```

1
80
fluxel
  6.2500E-01 1.8750E+00 3.1250E+00 4.3750E+00 5.6250E+00 6.8750E+00
  8.1250E+00 9.3750E+00 1.0625E+01 1.1875E+01 1.3125E+01 1.4375E+01
  1.5625E+01 1.6875E+01 1.8125E+01 1.9375E+01 2.0625E+01 2.1875E+01
  2.3125E+01 2.4375E+01 2.5625E+01 2.6875E+01 2.8125E+01 2.9375E+01
  3.0625E+01 3.1875E+01 3.3125E+01 3.4375E+01 3.5625E+01 3.6875E+01
  3.8125E+01 3.9375E+01 4.0625E+01 4.1875E+01 4.3125E+01 4.4375E+01
  4.5625E+01 4.6875E+01 4.8125E+01 4.9375E+01 5.0625E+01 5.1875E+01
  5.3125E+01 5.4375E+01 5.5625E+01 5.6875E+01 5.8125E+01 5.9375E+01
  6.0625E+01 6.1875E+01 6.3125E+01 6.4375E+01 6.5625E+01 6.6875E+01
  6.8125E+01 6.9375E+01 7.0625E+01 7.1875E+01 7.3125E+01 7.4375E+01
  7.5625E+01 7.6875E+01 7.8125E+01 7.9375E+01 8.0625E+01 8.1875E+01
  8.3125E+01 8.4375E+01 8.5625E+01 8.6875E+01 8.8125E+01 8.9375E+01
  9.0625E+01 9.1875E+01 9.3125E+01 9.4375E+01 9.5625E+01 9.6875E+01
  9.8125E+01 9.9375E+01
  9.9994E-01 1.0000E+00 1.0004E+00 1.0003E+00 9.9905E-01 9.9862E-01
  1.0010E+00 1.0039E+00 1.0023E+00 9.9471E-01 9.8812E-01 9.9294E-01
  1.0126E+00 1.0370E+00 1.0455E+00 1.0175E+00 9.4261E-01 8.2503E-01
  6.8031E-01 5.2854E-01 3.8754E-01 2.6882E-01 1.7690E-01 1.1073E-01
  6.6107E-02 3.7738E-02 2.0647E-02 1.0850E-02 5.4871E-03 2.6756E-03
  1.2600E-03 5.7395E-04 2.5325E-04 1.0839E-04 4.5054E-05 1.8208E-05
  7.1619E-06 2.7446E-06 1.0257E-06 3.7415E-07 1.3331E-07 4.6437E-08
  1.5824E-08 5.2788E-09 1.7250E-09 5.5238E-10 1.7335E-10 5.3276E-11
  1.5996E-11 4.7157E-12 1.2460E-12 2.6375E-13 0.0000E+00 0.0000E+00

```

```

0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
0.0000E+00 0.0000E+00
1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00 9.9999E-01
1.0000E+00 1.0000E+00 1.0000E+00 9.9997E-01 9.9998E-01 1.0000E+00
1.0001E+00 9.9999E-01 9.9985E-01 9.9987E-01 1.0001E+00 1.0004E+00
1.0002E+00 9.9962E-01 9.9912E-01 9.9945E-01 1.0008E+00 1.0022E+00
1.0021E+00 9.9960E-01 9.9556E-01 9.9267E-01 9.9428E-01 1.0023E+00
1.0152E+00 1.0279E+00 1.0323E+00 1.0201E+00 9.8492E-01 9.2427E-01
8.4015E-01 7.3832E-01 6.2678E-01 5.1398E-01 4.0732E-01 3.1219E-01
2.3164E-01 1.6655E-01 1.1616E-01 7.8678E-02 5.1801E-02 3.3186E-02
2.0706E-02 1.2594E-02 7.4739E-03 4.3308E-03 2.4523E-03 1.3579E-03
7.3573E-04 3.9035E-04 2.0291E-04 1.0340E-04 5.1688E-05 2.5355E-05
1.2213E-05 5.7782E-06 2.6868E-06 1.2283E-06 5.5230E-07 2.4436E-07
1.0642E-07 4.5635E-08 1.9276E-08 8.0224E-09 3.2908E-09 1.3309E-09
5.3073E-10 2.0870E-10 8.0868E-11 3.0864E-11 1.1473E-11 4.1700E-12
1.3642E-12 4.2519E-13
1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00
1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00
1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00
1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00
9.9998E-01 9.9999E-01 1.0000E+00 1.0000E+00 1.0000E+00 9.9997E-01
9.9992E-01 9.9995E-01 1.0001E+00 1.0002E+00 1.0002E+00 9.9994E-01
9.9965E-01 9.9953E-01 9.9981E-01 1.0005E+00 1.0012E+00 1.0013E+00
1.0004E+00 9.9842E-01 9.9622E-01 9.9519E-01 9.9681E-01 1.0019E+00
1.0098E+00 1.0183E+00 1.0237E+00 1.0213E+00 1.0066E+00 9.7616E-01
9.2844E-01 8.6400E-01 7.8539E-01 6.9671E-01 6.0283E-01 5.0873E-01
4.1876E-01 3.3636E-01 2.6374E-01 2.0198E-01 1.5117E-01 1.1064E-01
7.9223E-02 5.5537E-02 3.8137E-02 2.5668E-02 1.6941E-02 1.0970E-02
6.9733E-03 4.3534E-03 2.6703E-03 1.6102E-03 9.5450E-04 5.5737E-04
3.1857E-04 1.8337E-04

```

## B.7 SNAP FILE FOR ADVANCING-FRONT EXAMPLE

```

1
81
etot
0.0000E+00 1.2500E+00 2.5000E+00 3.7500E+00 5.0000E+00 6.2500E+00
7.5000E+00 8.7500E+00 1.0000E+01 1.1250E+01 1.2500E+01 1.3750E+01
1.5000E+01 1.6250E+01 1.7500E+01 1.8750E+01 2.0000E+01 2.1250E+01
2.2500E+01 2.3750E+01 2.5000E+01 2.6250E+01 2.7500E+01 2.8750E+01
3.0000E+01 3.1250E+01 3.2500E+01 3.3750E+01 3.5000E+01 3.6250E+01
3.7500E+01 3.8750E+01 4.0000E+01 4.1250E+01 4.2500E+01 4.3750E+01
4.5000E+01 4.6250E+01 4.7500E+01 4.8750E+01 5.0000E+01 5.1250E+01
5.2500E+01 5.3750E+01 5.5000E+01 5.6250E+01 5.7500E+01 5.8750E+01
6.0000E+01 6.1250E+01 6.2500E+01 6.3750E+01 6.5000E+01 6.6250E+01
6.7500E+01 6.8750E+01 7.0000E+01 7.1250E+01 7.2500E+01 7.3750E+01
7.5000E+01 7.6250E+01 7.7500E+01 7.8750E+01 8.0000E+01 8.1250E+01
8.2500E+01 8.3750E+01 8.5000E+01 8.6250E+01 8.7500E+01 8.8750E+01
9.0000E+01 9.1250E+01 9.2500E+01 9.3750E+01 9.5000E+01 9.6250E+01
9.7500E+01 9.8750E+01 1.0000E+02
1.0000E+00 9.9986E-01 1.0002E+00 1.0007E+00 9.9971E-01 9.9814E-01
9.9928E-01 1.0033E+00 1.0048E+00 9.9882E-01 9.8903E-01 9.8688E-01
1.0013E+00 1.0282E+00 1.0491E+00 1.0405E+00 9.8572E-01 8.8308E-01
7.4487E-01 5.9116E-01 4.4207E-01 3.1223E-01 2.0888E-01 1.3273E-01
8.0343E-02 4.6448E-02 2.5710E-02 1.3656E-02 6.9750E-03 3.4325E-03
1.6303E-03 7.4858E-04 3.3278E-04 1.4343E-04 6.0009E-05 2.4402E-05
9.6542E-06 3.7201E-06 1.3975E-06 5.1228E-07 1.8339E-07 6.4163E-08
2.1957E-08 7.3543E-09 2.4125E-09 7.7551E-10 2.4431E-10 7.5431E-11
2.2681E-11 6.7644E-12 1.8190E-12 4.5475E-13 0.0000E+00 0.0000E+00
0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00 0.0000E+00
1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00 9.9999E-01
9.9999E-01 1.0000E+00 1.0000E+00 9.9998E-01 9.9995E-01 1.0000E+00
1.0001E+00 1.0001E+00 9.9989E-01 9.9979E-01 9.9998E-01 1.0003E+00
1.0004E+00 9.9993E-01 9.9919E-01 9.9902E-01 1.0000E+00 1.0018E+00
1.0027E+00 1.0012E+00 9.9734E-01 9.9310E-01 9.9208E-01 9.9732E-01
1.0091E+00 1.0237E+00 1.0337E+00 1.0304E+00 1.0059E+00 9.5592E-01
8.8056E-01 7.8434E-01 6.7476E-01 5.6051E-01 4.4972E-01 3.4877E-01
2.6168E-01 1.9015E-01 1.3396E-01 9.1592E-02 6.0845E-02 3.9311E-02
2.4726E-02 1.5155E-02 9.0587E-03 5.2854E-03 3.0125E-03 1.6786E-03
9.1496E-04 4.8822E-04 2.5518E-04 1.3073E-04 6.5674E-05 3.2373E-05
1.5665E-05 7.4449E-06 3.4767E-06 1.5960E-06 7.2051E-07 3.2001E-07
1.3989E-07 6.0203E-08 2.5518E-08 1.0656E-08 4.3854E-09 1.7792E-09
7.1179E-10 2.8075E-10 1.0914E-10 4.1894E-11 1.5632E-11 5.7980E-12
1.9895E-12 5.6843E-13 2.2737E-13
1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00
1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00
1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00
1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00
9.9999E-01 9.9998E-01 1.0000E+00 1.0000E+00 1.0000E+00 1.0000E+00
9.9993E-01 9.9992E-01 1.0000E+00 1.0001E+00 1.0002E+00 1.0001E+00
9.9976E-01 9.9950E-01 9.9958E-01 1.0001E+00 1.0010E+00 1.0015E+00
1.0011E+00 9.9942E-01 9.9702E-01 9.9511E-01 9.9530E-01 9.9888E-01
1.0060E+00 1.0151E+00 1.0229E+00 1.0249E+00 1.0163E+00 9.9311E-01
9.5275E-01 8.9488E-01 8.2134E-01 7.3574E-01 6.4280E-01 5.4764E-01
4.5499E-01 3.6874E-01 2.9164E-01 2.2521E-01 1.6991E-01 1.2530E-01

```

9.0385E-02 6.3809E-02 4.4114E-02 2.9883E-02 1.9846E-02 1.2929E-02  
8.2656E-03 5.1887E-03 3.1998E-03 1.9391E-03 1.1560E-03 6.7628E-04  
3.9314E-04 2.1559E-04 1.3888E-04

## B.8 INPUT FILE FOR COUPLED EXAMPLE

# example file comparing breath with unsath  
# three-layer problem tracking input rain pulse  
# consistent units are cm, sec, gm, joule, kelvin

```
nel      96          # number of elements
nts      10000       # number of time steps per period
ztsasel  iterbased  # use iteration-based time step
maxitf   16         # maximum flow iterations
nomitf   9          # nominal flow iterations
convf    0.0001     # flow convergence criterion
maxite   10         # maximum energy iterations
nomite   4          # nominal energy iterations
conve    0.001     # energy convergence criterion
deltim   360        # initial time step
timmax   3600       # length of a time period
dtmin    0.01       # minimum time step
dtmax    720        # maximum time step
ntmcyc   3          # number of cycles per time step
krnevap0 0          # don't evaporate during rain
dzdx     -1         # vertical column
set vector xcoor    # nodal coordinates
0 0.1 0.213622 0.342721 0.489405 0.656071 0.845439 1.0606 1.30508
1.58285 1.89846 2.25707 2.66452 3.12747 3.65349 4.25116 4.93024
5.70182 6.57851 7.57462 8.70642 9.99239 11.4535 13.1137 15
17.3998 18.2777 19.1218 19.9334 20.7137 21.4639 22.1853 22.8788
23.5455 24.1866 24.803 25.3956 25.9653 26.5131 27.0398 27.5462
28.0331 28.5012 28.9513 29.384 29.8 30.2 30.9838 32.5197 35.5293
41.4267 52.9829 75.6274 120 164.373 187.017 198.573 204.471
207.48 209.016 209.8 210.2 210.66 211.188 211.796 212.494
213.297 214.219 215.28 216.499 217.9 219.51 221.361 223.489
225.935 228.746 231.977 235.692 239.961 244.869 250.51 256.993
264.446 273.013 282.86 294.179 307.189 322.143 339.333 359.091
381.802 407.907 437.913 472.404 512.049 557.62 610
pressw -1e+08      # initial pressure
temp    286        # initial temperature
tempr   286        # reference temperature
tempal  286        # bottom-of-column air temperature
grav    980        # gravity
dcoefv  0.1584     # tortuosity times vapor diffusion coefficient
viscw0  11.24      # water viscosity

set vector satmax   # maximum moisture content
0.098 0.098 0.098 0.098 0.098 0.098 0.098 0.098 0.098 0.098 0.098 0.098
0.098 0.098 0.098 0.098 0.098 0.098 0.098 0.098 0.098 0.098 0.098 0.098
0.098 0.098 0.098 0.098 0.098 0.098 0.098 0.098 0.098 0.098 0.098 0.098
0.098 0.098 0.098 0.098 0.098 0.098 0.098 0.098 0.098 0.098 0.098 0.098
0.28 0.28 0.28 0.28 0.28 0.28 0.28 0.28 0.28 0.28 0.28
0.42 0.42 0.42 0.42 0.42 0.42 0.42 0.42 0.42 0.42 0.42 0.42 0.42 0.42
0.42 0.42 0.42 0.42 0.42 0.42 0.42 0.42 0.42 0.42 0.42 0.42 0.42 0.42
0.42 0.42 0.42 0.42 0.42 0.42 0.42 0.42 0.42 0.42 0.42 0.42 0.42 0.42
0.42 0.42 0.42

set vector satmin   # minimum moisture content
0.0001 0.0001 0.0001 0.0001 0.0001 0.0001 0.0001 0.0001 0.0001 0.0001
0.0001 0.0001 0.0001 0.0001 0.0001 0.0001 0.0001 0.0001 0.0001 0.0001
```





```

#1 22 29 52 54 57 95 97 0/
#set vector eltrace
#1 21 29 51 53 57 94 96 0/
#snap nodeunit 23 elemunit 24 pressw satw temp cumqw cumqv cumqe
# format (lpe12.5) nperskip 0 on
#mass unit 25 moiscum enercum format (lpe12.5) nperskip 5 on

# specify input columns

met unit 26 runtype tmrn timmax bcw tempa bcv esiga swrad windsp set

calc metrun
#file 26 rewind
#file unit 21 close
#file unit 22 close
#file unit 23 close
#file unit 24 close
#file unit 25 close
file unit 26 close

# set up output file information for final results

file unit 27 open coupled.out
iores 27
echo string ' Example comparing BREATH with UNSATH'
echo string ' '
echo string ' Simulation control...'
echo string ' '
echo nnd
echo ztssel
echo maxitf
echo nomitf
echo convf
echo maxite
echo nomite
echo conve
echo dtmin
echo dtmax
echo ntmcy
echo krnevap0
echo string ' '
echo string ' Domain and parameter setup...'
echo string ' '
echo dzdx
echo xcoor
echo tempr
echo tempal
echo grav
echo dcoefv
echo viscw0
echo coeflh
echo satmax
echo satmin
echo pcap0
echo vgm
echo perm
echo poros
echo densg

```

```

echo denss
echo densw
echo dudtg
echo dudts
echo dudtw
echo ekg
echo eks
echo ekw
echo eintw0
echo eintv0
echo string ' '
echo string ' Boundary conditions...'
echo string ' '
echo zbcw0
echo bcw0
echo zbcw1
echo bcw1
echo pondh0

echo zbcv0
echo zblk0
echo blzt0
echo blzh0
echo blzu0
echo blzm0
echo blzoff0

echo zbcv1
echo bcv1
echo zblk1
echo blkv1
echo blkhl
echo zbce0
echo zbcel
echo bcel
echo esigsd0
echo esigsw0
echo albedd0
echo albedw0

echo pressw
echo satw
echo densv
echo temp

file unit 27 close
end

```



```

3600 0 292.733 3.41644e-06 4.61063e-12 0.0323589 406.806
3600 0 291.806 3.41644e-06 4.61063e-12 0.0193776 406.806
3600 0 290.331 3.41644e-06 4.61063e-12 0.00544689 406.806
3600 0 288.41 3.41644e-06 4.61063e-12 0 406.806
3600 0 286.172 3.41644e-06 4.61063e-12 0 406.806
3600 0 283.771 3.41644e-06 4.61063e-12 0 406.806
3600 0 281.37 3.41644e-06 4.61063e-12 0 406.806
3600 0 279.132 3.41644e-06 4.61063e-12 0 406.806
3600 0 277.211 3.41644e-06 4.61063e-12 0 406.806
3600 0 275.736 3.41644e-06 4.61063e-12 0 406.806
3600 0 274.809 3.41644e-06 4.61063e-12 0 406.806
3600 0 274.493 3.41644e-06 4.61063e-12 0 406.806
3600 0 274.809 3.41644e-06 4.61063e-12 0 406.806
3600 0 275.736 3.41644e-06 4.61063e-12 0 406.806
3600 0 277.211 3.41644e-06 4.61063e-12 0.0057098 406.806
3600 0 279.132 3.41644e-06 4.61063e-12 0.0196281 406.806
3600 0 281.37 3.41644e-06 4.61063e-12 0.0325978 406.806
3600 0 283.771 3.41644e-06 4.61063e-12 0.0437352 406.806
3600 0 286.172 3.41644e-06 4.61063e-12 0.0522812 406.806
3600 0 288.41 3.41644e-06 4.61063e-12 0.0576534 406.806
3600 0 290.331 3.41644e-06 4.61063e-12 0.0594858 406.806
3600 0 291.806 3.41644e-06 4.61063e-12 0.0576534 406.806
3600 0 292.733 3.41644e-06 4.61063e-12 0.0522812 406.806
3600 0 293.049 3.41644e-06 4.61063e-12 0.0437352 406.806
3600 0 292.733 3.41644e-06 4.61063e-12 0.0325978 406.806
3600 0 291.806 3.41644e-06 4.61063e-12 0.0196281 406.806
3600 0 290.331 3.41644e-06 4.61063e-12 0.0057098 406.806
3600 0 288.41 3.41644e-06 4.61063e-12 0 406.806
3600 0 286.172 3.41644e-06 4.61063e-12 0 406.806
3600 0 283.771 3.41644e-06 4.61063e-12 0 406.806
3600 0 281.37 3.41644e-06 4.61063e-12 0 406.806
3600 0 279.132 3.41644e-06 4.61063e-12 0 406.806

```

## B.10 OUTPUT FILE FOR COUPLED EXAMPLE

Example comparing BREATH with UNSATH

Simulation control...

```

nnd      = 97
ztssel  = iterbase
maxitf  = 16
nomitf  = 9
convf   = 1.0000000000000D-04
maxite  = 10
nomite  = 4
conve   = 1.0000000000000D-03
dtmin   = 1.0000000000000D-02
dtmax   = 720.0000000000000
ntmcyc  = 3
krnevap0 = 0

```

Domain and parameter setup...

```

dzdx    = -1.0000000000000
xcoor   :
  1 0.00000E+00  2 1.00000E-01  3 2.13622E-01  4 3.42721E-01
  5 4.89405E-01  6 6.56071E-01  7 8.45439E-01  8 1.06060E+00
  9 1.30508E+00 10 1.58285E+00 11 1.89846E+00 12 2.25707E+00
13 2.66452E+00 14 3.12747E+00 15 3.65349E+00 16 4.25116E+00
17 4.93024E+00 18 5.70182E+00 19 6.57851E+00 20 7.57462E+00
21 8.70642E+00 22 9.99239E+00 23 1.14535E+01 24 1.31137E+01
25 1.50000E+01 26 1.73998E+01 27 1.82777E+01 28 1.91218E+01
29 1.99334E+01 30 2.07137E+01 31 2.14639E+01 32 2.21853E+01
33 2.28788E+01 34 2.35455E+01 35 2.41866E+01 36 2.48030E+01
37 2.53956E+01 38 2.59653E+01 39 2.65131E+01 40 2.70398E+01
41 2.75462E+01 42 2.80331E+01 43 2.85012E+01 44 2.89513E+01
45 2.93840E+01 46 2.98000E+01 47 3.02000E+01 48 3.09838E+01
49 3.25197E+01 50 3.55293E+01 51 4.14267E+01 52 5.29829E+01
53 7.56274E+01 54 1.20000E+02 55 1.64373E+02 56 1.87017E+02
57 1.98573E+02 58 2.04471E+02 59 2.07480E+02 60 2.09016E+02
61 2.09800E+02 62 2.10200E+02 63 2.10660E+02 64 2.11188E+02
65 2.11796E+02 66 2.12494E+02 67 2.13297E+02 68 2.14219E+02
69 2.15280E+02 70 2.16499E+02 71 2.17900E+02 72 2.19510E+02
73 2.21361E+02 74 2.23489E+02 75 2.25935E+02 76 2.28746E+02
77 2.31977E+02 78 2.35692E+02 79 2.39961E+02 80 2.44869E+02
81 2.50510E+02 82 2.56993E+02 83 2.64446E+02 84 2.73013E+02
85 2.82860E+02 86 2.94179E+02 87 3.07189E+02 88 3.22143E+02
89 3.39333E+02 90 3.59091E+02 91 3.81802E+02 92 4.07907E+02
93 4.37913E+02 94 4.72404E+02 95 5.12049E+02 96 5.57620E+02
97 6.10000E+02
tempr   = 286.00000000000
tempal  = 286.00000000000
grav    = 980.00000000000
dcoefv { 1: 97} = 0.158400000000000
viscw0  = 11.2400000000000
coeflh  = 2501.00000000000
satmax  :
  1 9.80000E-02  2 9.80000E-02  3 9.80000E-02  4 9.80000E-02
  5 9.80000E-02  6 9.80000E-02  7 9.80000E-02  8 9.80000E-02
  9 9.80000E-02 10 9.80000E-02 11 9.80000E-02 12 9.80000E-02

```



57 4.20000E-01 58 4.20000E-01 59 4.20000E-01 60 4.20000E-01  
 61 4.20000E-01 62 4.20000E-01 63 4.20000E-01 64 4.20000E-01  
 65 4.20000E-01 66 4.20000E-01 67 4.20000E-01 68 4.20000E-01  
 69 4.20000E-01 70 4.20000E-01 71 4.20000E-01 72 4.20000E-01  
 73 4.20000E-01 74 4.20000E-01 75 4.20000E-01 76 4.20000E-01  
 77 4.20000E-01 78 4.20000E-01 79 4.20000E-01 80 4.20000E-01  
 81 4.20000E-01 82 4.20000E-01 83 4.20000E-01 84 4.20000E-01  
 85 4.20000E-01 86 4.20000E-01 87 4.20000E-01 88 4.20000E-01  
 89 4.20000E-01 90 4.20000E-01 91 4.20000E-01 92 4.20000E-01  
 93 4.20000E-01 94 4.20000E-01 95 4.20000E-01 96 4.20000E-01  
 97 4.20000E-01

densg [ 1: 97] = 1.25000000000000D-03  
 denss [ 1: 97] = 2.60000000000000  
 densw = 1.00000000000000  
 dudtg = 1.00000000000000  
 dudts [ 1: 97] = 0.840000000000000  
 dudtw = 4.20000000000000  
 ekg = 0.  
 eks [ 1: 97] = 8.07692000000000D-03  
 ekw = 5.80000000000000D-03  
 eintw0 = 0.  
 eintv0 = 0.

Boundary conditions...

zbcw0 = flux  
 bcw0 = 0.  
 zbcw1 = gradswatw  
 bcw1 = 0.  
 pondh0 = 1000.000000000000  
 zbcv0 = densva  
 zblk0 = campbell  
 blzt0 = 300.000000000000  
 blzh0 = 3.00000000000000D-02  
 blzu0 = 300.000000000000  
 blzm0 = 3.00000000000000D-02  
 blzoff0 = 0.  
 zbcv1 = densva  
 bcv1 = 1.12710000000000D-05  
 zblk1 = constant  
 blkv1 = 1.58400000000000D-04  
 blkhl = 0.  
 zbce0 = gradtemp  
 zbce1 = gradtemp  
 bcel = 0.  
 esigsd0 = 5.10300000000000D-12  
 esigsw0 = 5.20292000000000D-12  
 albedd0 = 0.250000000000000  
 albedw0 = 0.250000000000000

pressw :  
 1 -1.39527E+09 2 -1.25682E+09 3 -1.11118E+09 4 -9.62697E+08  
 5 -8.16763E+08 6 -6.79999E+08 7 -5.59196E+08 8 -4.58823E+08  
 9 -3.79261E+08 10 -3.17680E+08 11 -2.70160E+08 12 -2.33138E+08  
 13 -2.03860E+08 14 -1.80325E+08 15 -1.61105E+08 16 -1.45180E+08  
 17 -1.31814E+08 18 -1.20469E+08 19 -1.10744E+08 20 -1.02343E+08  
 21 -9.50438E+07 22 -8.86818E+07 23 -8.31376E+07 24 -7.83266E+07  
 25 -7.41935E+07 26 -7.03598E+07 27 -6.92578E+07 28 -6.83238E+07  
 29 -6.75303E+07 30 -6.68555E+07 31 -6.62819E+07 32 -6.57951E+07

33 -6.53833E+07 34 -6.50367E+07 35 -6.47470E+07 36 -6.45070E+07  
 37 -6.43109E+07 38 -6.41534E+07 39 -6.40299E+07 40 -6.39366E+07  
 41 -6.38698E+07 42 -6.38265E+07 43 -6.38039E+07 44 -6.37996E+07  
 45 -6.38113E+07 46 -6.38369E+07 47 -6.60371E+07 48 -7.94578E+07  
 49 -9.59374E+07 50 -9.97011E+07 51 -9.98843E+07 52 -9.99291E+07  
 53 -9.99594E+07 54 -9.99822E+07 55 -9.99848E+07 56 -9.99900E+07  
 57 -9.99967E+07 58 -9.99972E+07 59 -9.99974E+07 60 -9.99975E+07  
 61 -9.99976E+07 62 -9.99976E+07 63 -9.99977E+07 64 -9.99977E+07  
 65 -9.99977E+07 66 -9.99978E+07 67 -9.99978E+07 68 -9.99979E+07  
 69 -9.99980E+07 70 -9.99980E+07 71 -9.99981E+07 72 -9.99982E+07  
 73 -9.99983E+07 74 -9.99984E+07 75 -9.99985E+07 76 -9.99986E+07  
 77 -9.99988E+07 78 -9.99989E+07 79 -9.99990E+07 80 -9.99992E+07  
 81 -9.99993E+07 82 -9.99995E+07 83 -9.99996E+07 84 -9.99997E+07  
 85 -9.99998E+07 86 -9.99999E+07 87 -9.99999E+07 88 -1.00000E+08  
 89 -1.00000E+08 90 -1.00000E+08 91 -1.00000E+08 92 -1.00000E+08  
 93 -1.00000E+08 94 -1.00000E+08 95 -1.00000E+08 96 -1.00000E+08  
 97 -9.99837E+07

satw :  
 1 1.11853E-02 2 1.15383E-02 3 1.19688E-02 4 1.24906E-02  
 5 1.31169E-02 6 1.38524E-02 7 1.46833E-02 8 1.55748E-02  
 9 1.64843E-02 10 1.73783E-02 11 1.82384E-02 12 1.90577E-02  
 13 1.98358E-02 14 2.05748E-02 15 2.12781E-02 16 2.19488E-02  
 17 2.25901E-02 18 2.32045E-02 19 2.37941E-02 20 2.43603E-02  
 21 2.49035E-02 22 2.54230E-02 23 2.59167E-02 24 2.63810E-02  
 25 2.68104E-02 26 2.72373E-02 27 2.73657E-02 28 2.74765E-02  
 29 2.75722E-02 30 2.76548E-02 31 2.77258E-02 32 2.77867E-02  
 33 2.78387E-02 34 2.78828E-02 35 2.79199E-02 36 2.79507E-02  
 37 2.79761E-02 38 2.79965E-02 39 2.80125E-02 40 2.80247E-02  
 41 2.80334E-02 42 2.80391E-02 43 2.80420E-02 44 2.80426E-02  
 45 2.80411E-02 46 2.80377E-02 47 4.23677E-02 48 4.01347E-02  
 49 3.79837E-02 50 3.75593E-02 51 3.75392E-02 52 3.75343E-02  
 53 3.75310E-02 54 3.75285E-02 55 3.75271E-02 56 5.58554E-02  
 57 5.58558E-02 58 5.58557E-02 59 5.58556E-02 60 5.58556E-02  
 61 5.58556E-02 62 5.58556E-02 63 5.58556E-02 64 5.58556E-02  
 65 5.58556E-02 66 5.58556E-02 67 5.58556E-02 68 5.58556E-02  
 69 5.58556E-02 70 5.58555E-02 71 5.58555E-02 72 5.58555E-02  
 73 5.58555E-02 74 5.58555E-02 75 5.58555E-02 76 5.58554E-02  
 77 5.58554E-02 78 5.58554E-02 79 5.58554E-02 80 5.58554E-02  
 81 5.58553E-02 82 5.58553E-02 83 5.58553E-02 84 5.58553E-02  
 85 5.58553E-02 86 5.58552E-02 87 5.58552E-02 88 5.58552E-02  
 89 5.58552E-02 90 5.58552E-02 91 5.58552E-02 92 5.58552E-02  
 93 5.58552E-02 94 5.58552E-02 95 5.58552E-02 96 5.58552E-02  
 97 5.58579E-02

densv :  
 1 3.49200E-06 2 3.89589E-06 3 4.37238E-06 4 4.92032E-06  
 5 5.52934E-06 6 6.17445E-06 7 6.81642E-06 8 7.41497E-06  
 9 7.94645E-06 10 8.40860E-06 11 8.81252E-06 12 9.17317E-06  
 13 9.50451E-06 14 9.81813E-06 15 1.01232E-05 16 1.04269E-05  
 17 1.07347E-05 18 1.10508E-05 19 1.13778E-05 20 1.17168E-05  
 21 1.20670E-05 22 1.24252E-05 23 1.27851E-05 24 1.31367E-05  
 25 1.34659E-05 26 1.37827E-05 27 1.38729E-05 28 1.39471E-05  
 29 1.40073E-05 30 1.40555E-05 31 1.40932E-05 32 1.41218E-05  
 33 1.41428E-05 34 1.41570E-05 35 1.41656E-05 36 1.41694E-05  
 37 1.41690E-05 38 1.41652E-05 39 1.41585E-05 40 1.41494E-05  
 41 1.41383E-05 42 1.41256E-05 43 1.41115E-05 44 1.40964E-05  
 45 1.40805E-05 46 1.40639E-05 47 1.40223E-05 48 1.38401E-05  
 49 1.35797E-05 50 1.33388E-05 51 1.29010E-05 52 1.21548E-05  
 53 1.14983E-05 54 1.09013E-05 55 1.06131E-05 56 1.05314E-05

57	1.04996E-05	58	1.04864E-05	59	1.04803E-05	60	1.04774E-05
61	1.04759E-05	62	1.04752E-05	63	1.04744E-05	64	1.04734E-05
65	1.04724E-05	66	1.04712E-05	67	1.04698E-05	68	1.04683E-05
69	1.04666E-05	70	1.04646E-05	71	1.04625E-05	72	1.04602E-05
73	1.04576E-05	74	1.04548E-05	75	1.04518E-05	76	1.04485E-05
77	1.04451E-05	78	1.04415E-05	79	1.04378E-05	80	1.04341E-05
81	1.04305E-05	82	1.04270E-05	83	1.04237E-05	84	1.04208E-05
85	1.04183E-05	86	1.04163E-05	87	1.04148E-05	88	1.04138E-05
89	1.04132E-05	90	1.04128E-05	91	1.04127E-05	92	1.04126E-05
93	1.04126E-05	94	1.04127E-05	95	1.04128E-05	96	1.04131E-05
97	1.04139E-05						

temp :

1	2.84304E+02	2	2.84365E+02	3	2.84434E+02	4	2.84511E+02
5	2.84598E+02	6	2.84696E+02	7	2.84806E+02	8	2.84930E+02
9	2.85068E+02	10	2.85222E+02	11	2.85394E+02	12	2.85586E+02
13	2.85798E+02	14	2.86032E+02	15	2.86290E+02	16	2.86572E+02
17	2.86879E+02	18	2.87210E+02	19	2.87565E+02	20	2.87941E+02
21	2.88333E+02	22	2.88735E+02	23	2.89137E+02	24	2.89526E+02
25	2.89885E+02	26	2.90224E+02	27	2.90319E+02	28	2.90396E+02
29	2.90458E+02	30	2.90507E+02	31	2.90545E+02	32	2.90573E+02
33	2.90592E+02	34	2.90605E+02	35	2.90612E+02	36	2.90613E+02
37	2.90610E+02	38	2.90604E+02	39	2.90594E+02	40	2.90582E+02
41	2.90568E+02	42	2.90553E+02	43	2.90536E+02	44	2.90518E+02
45	2.90499E+02	46	2.90480E+02	47	2.90458E+02	48	2.90406E+02
49	2.90295E+02	50	2.90044E+02	51	2.89493E+02	52	2.88512E+02
53	2.87604E+02	54	2.86739E+02	55	2.86307E+02	56	2.86183E+02
57	2.86134E+02	58	2.86114E+02	59	2.86104E+02	60	2.86100E+02
61	2.86098E+02	62	2.86096E+02	63	2.86095E+02	64	2.86094E+02
65	2.86092E+02	66	2.86090E+02	67	2.86088E+02	68	2.86086E+02
69	2.86083E+02	70	2.86080E+02	71	2.86077E+02	72	2.86073E+02
73	2.86069E+02	74	2.86065E+02	75	2.86060E+02	76	2.86055E+02
77	2.86050E+02	78	2.86045E+02	79	2.86039E+02	80	2.86033E+02
81	2.86028E+02	82	2.86022E+02	83	2.86017E+02	84	2.86013E+02
85	2.86009E+02	86	2.86006E+02	87	2.86004E+02	88	2.86002E+02
89	2.86001E+02	90	2.86000E+02	91	2.86000E+02	92	2.86000E+02
93	2.86000E+02	94	2.86000E+02	95	2.86000E+02	96	2.86001E+02
97	2.86002E+02						