

NUREG/CR-6333
CNWRA 94-020

BREATH Version 1.1 – Coupled Flow and Energy Transport in Porous Media

Simulator Description and User Guide

Prepared by
S. A. Stothoff

Southwest Research Institute
Center for Nuclear Waste Regulatory Analyses

Prepared for
U.S. Nuclear Regulatory Commission

9509200049 950731
PDR NUREG
CR-6333 R PDR

AVAILABILITY NOTICE

Availability of Reference Materials Cited in NRC Publications

Most documents cited in NRC publications will be available from one of the following sources:

1. The NRC Public Document Room, 2120 L Street, NW., Lower Level, Washington, DC 20555-0001
2. The Superintendent of Documents, U. S. Government Printing Office, P. O. Box 37082, Washington, DC 20402-9328
3. The National Technical Information Service, Springfield, VA 22161-0002

Although the listing that follows represents the majority of documents cited in NRC publications, it is not intended to be exhaustive.

Referenced documents available for inspection and copying for a fee from the NRC Public Document Room include NRC correspondence and internal NRC memoranda, NRC bulletins, circulars, information notices, inspection and investigation notices; licensee event reports; vendor reports and correspondence; Commission papers; and applicant and licensee documents and correspondence.

The following documents in the NUREG series are available for purchase from the Government Printing Office: formal NRC staff and contractor reports, NRC-sponsored conference proceedings, international agreement reports, grantee reports, and NRC booklets and brochures. Also available are regulatory guides, NRC regulations in the *Code of Federal Regulations*, and *Nuclear Regulatory Commission Issuances*.

Documents available from the National Technical Information Service include NUREG-series reports and technical reports prepared by other Federal agencies and reports prepared by the Atomic Energy Commission, forerunner agency to the Nuclear Regulatory Commission.

Documents available from public and special technical libraries include all open literature items, such as books, journal articles, and translations. *Federal Register* notices, Federal and State legislation, and congressional reports can usually be obtained from these libraries.

Documents such as theses, dissertations, foreign reports and translations, and non-NRC conference proceedings are available for purchase from the organization sponsoring the publication cited.

Single copies of NRC draft reports are available free, to the extent of supply, upon written request to the Office of Administration, Distribution and Mail Services Section, U. S. Nuclear Regulatory Commission, Washington, DC 20555-0001.

Copies of industry codes and standards used in a substantive manner in the NRC regulatory process are maintained at the NRC Library, Two White Flint North, 11545 Rockville Pike, Rockville, MD 20852-2738, for use by the public. Codes and standards are usually copyrighted and may be purchased from the originating organization or, if they are American National Standards, from the American National Standards Institute, 1430 Broadway, New York, NY 10018-3308.

DISCLAIMER NOTICE

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, expressed or implied, or assumes any legal liability or responsibility for any third party's use, or the results of such use, of any information, apparatus, product, or process disclosed in this report, or represents that its use by such third party would not infringe privately owned rights.

BREATH Version 1.1 – Coupled Flow and Energy Transport in Porous Media

Simulator Description and User Guide

Manuscript Completed: May 1995
Date Published: July 1995

Prepared by
S. A. Stothoff

Southwest Research Institute
Center for Nuclear Waste Regulatory Analyses
6220 Culebra Road
San Antonio, TX 78228-0510

Prepared for
Division of Regulatory Applications
Office of Nuclear Regulatory Research
U.S. Nuclear Regulatory Commission
Washington, DC 20555-0001
NRC Job Code B6666

ABSTRACT

This document describes the BREATH computer 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 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 U.S. 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	vii
TABLES	viii
NOMENCLATURE	ix
ACKNOWLEDGMENTS	xi
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 GENERAL LOCAL BALANCE LAW	3-1
3.2 CONSERVATION OF MASS	3-2
3.3 CONSERVATION OF ENERGY	3-6
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	4-4
4.3 BOUNDARY CONDITIONS	4-4
4.3.1 Liquid Phase Boundary Conditions	4-5
4.3.2 Vapor Species Boundary Conditions	4-6
4.3.3 Energy Equation Boundary Conditions	4-6
4.3.4 Meteorological Boundary Conditions	4-7
4.3.5 Boundary Layer Conductances	4-8
4.4 CONSTITUTIVE RELATIONSHIPS	4-10
4.4.1 Liquid/Soil Properties	4-10
4.4.1.1 Moisture Content/Capillary Pressure	4-11
4.4.1.2 Moisture Content/Mobility	4-11
4.4.1.3 Single-Phase Flow	4-12
4.4.2 Thermal Properties	4-13
4.4.3 Fluid Properties	4-14

CONTENTS (Cont'd)

Section		Page
4.5	TIME STEPPING AND ACTIVE REGION CONTROL	4-15
4.5.1	Constant and Courant-Based Time Stepping	4-15
4.5.2	Heuristic Convergence-Based Time Stepping	4-15
4.5.3	Active Region Control	4-16
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	VARIABLE NAMES	
C	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
3-1	Assignment of physical meaning to symbols in the general local balance law [based on Table 1-1 of Allen et al. (1988)]	3-3
4-1	Liquid-phase boundary conditions	4-5
4-2	Vapor-species boundary conditions	4-6
4-3	Energy equation boundary conditions	4-7
4-4	Typical thermal properties of soil constituents (after Hillel, 1980, Tables 9.1 and 9.2)	4-13
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

NOMENCLATURE

Units

J	Energy [ML ² T ⁻²]
L	Length
M	Mass
K	Temperature
T	Time

Greek Symbols

∇	= Gradient operator
Δ	= Difference operator
∂	= Partial derivative operator
α	= van Genuchten shape parameter [L ⁻¹]
α_s	= Ground surface albedo [-]
κ	= von Karman constant [-]
λ	= Mobility (relative permeability divided by viscosity) [LTM ⁻¹]
ρ	= Density [ML ⁻³]
ψ	= Matric head [L]
μ	= Dynamic viscosity [ML ⁻¹ T ⁻¹]
Φ	= Difference between actual and reference temperature [K]
θ	= Volumetric fraction of a phase [L ³ L ⁻³]
θ_e	= Effective saturation [-]
θ_{wr}	= Residual moisture content [L ³ L ⁻³]
θ_{ws}	= Saturated moisture content [L ³ L ⁻³]
ϵ	= Emissivity [-]
σ	= Stefan-Boltzmann constant [JT ⁻¹ L ⁻² K ⁻⁴]
τ	= Tortuosity factor in vapor equations [-]

Numeric Variables (Normal type: scalars. Bold type: vectors or tensors)

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 [-]
c_p	= Specific heat at constant pressure [JM ⁻¹ K ⁻¹]
c_v	= Specific heat at constant volume [JM ⁻¹ K ⁻¹]
\mathbf{D}_a	= Diffusivity of vapor in air [L ² T ⁻¹]
E	= Specific energy [JM ⁻¹]
E_{ext}	= External sources of energy
H	= Enthalpy [JM ⁻¹]
H_l	= Latent heat [JM ⁻¹]
\mathbf{K}_e	= Bulk thermal conductivity [ML ⁻¹ T ⁻¹]
\mathbf{K}_{sat}	= Saturated hydraulic conductivity [LT ⁻¹]
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

NOMENCLATURE (Cont'd)

Numeric Variables (Cont'd)

- M_{ext} = External sources of mass
 P = Pressure [$ML^{-1}T^{-2}$]
 P_0 = van Genuchten shape factor [$ML^{-1}T^{-2}$]
 P_c = Capillary pressure [$ML^{-1}T^{-2}$]
 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 [JM^{-1}]
 U_l = Change in internal energy due to phase change [JM^{-1}]
 V = Velocity [LT^{-1}]
 g = Acceleration due to gravity [LT^{-2}]
 h = Suction head [L]
 \mathbf{j} = Dispersive flux
 \mathbf{j}_e = Dispersive flux of energy
 \mathbf{k} = Intrinsic permeability [L^2]
 \mathbf{k}_e = Thermal conductance [$JL^2M^{-1}T^{-1}K^{-1}$]
 k_h = Boundary layer conductance for heat [L^2T^{-1}]
 k_v = Boundary layer conductance for vapor [L^2T^{-1}]
 k_r = Relative permeability [-]
 m = van Genuchten shape parameter [-]
 n = van Genuchten shape parameter [-]
 \mathbf{q} = Darcy flux [$L^3T^{-1}L^{-2}$]
 \mathbf{q}_e = Energy flux [$JT^{-1}L^{-2}$]
 t = Time [T]
 \mathbf{v} = Velocity [LT^{-1}]
 z = Elevation [L]

Subscripts/Superscripts

- α 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, A.B. Gureghian, H.D. Nguyen, J.D. Randall, B. Sagar, and Publications Services of Southwest Research Institute 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 of the requirements 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 (YM) that may be different from those in 40 CFR Part 191. A revised 40 CFR Part 191, applicable to repositories other than YM, 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 yr 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)—Controlled 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 by the U.S. Department of Energy (DOE) to design the site characterization program, and by the NRC to judge the adequacy of the site characterization program. 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 YM, in southwest Nevada, with reference to all six performance measures. Infiltration was treated as a spatially uniform, temporally constant parameter in each simulation in the analyses (Nuclear Regulatory Commission, 1992, 1995). 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 parameters to which estimated performance is most sensitive in corresponding TPA analyses (Sandia National Laboratories, 1992, 1994), as have risk-based approaches to PA [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 YM, for extended periods of time, with properties representative of various areas at YM. 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 comprises 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 exact solutions and other simulators. In Chapter 6, possible modifications are proposed. References are found in Chapter 7. A user guide to the code is presented in Appendix A, and a list of all variable names in BREATH is found in Appendix B. Finally, input files and resultant output files representative of the examples in Chapter 5 are shown in Appendix C.

2 SIMULATOR OVERVIEW

2.1 INTRODUCTION

Literature on isothermal liquid transport dates back to the contributions of Henri 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 classic 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 an attempt has been 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 YM 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 and Jones, 1990) was deemed the most suitable for the desired purposes, with most of the necessary features, but the user interface was 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 (Personal communication with M. Fayer at Pacific Northwest Laboratory).

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 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, since 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, these files 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 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, 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, which may completely fill the pore space, must always be present throughout the medium. 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; temperature serves as the dependent variable in the governing equations. Energy may redistribute by 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 equations 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; 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 boundary condition variation. 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, multispecies flow and energy transport presented in Sections 3.1 through 3.3 of this chapter is based on the development of Allen et al. (1988), except where otherwise noted. In Section 3.1, a general local balance law is developed for generic quantities. In Sections 3.2 and 3.3, the generic balance law is specialized to provide a balance law for moisture and for energy, which provide the basis for the BREATH simulator. The specific numerical implementation of the mathematical model, as embodied in the BREATH simulator, is presented in Chapter 4.

3.1 GENERAL LOCAL BALANCE LAW

A general local balance law for a generic quantity A in a continuum is

$$\frac{\partial}{\partial t}(\rho A) + \nabla \cdot (\rho A \mathbf{v}) + \nabla \cdot \boldsymbol{\tau} - \rho Q = 0, \quad (3-1)$$

where \mathbf{v} is velocity [LT^{-1}], ρ is mass density [ML^{-3}], $\boldsymbol{\tau}$ is flux of ρA across mathematical surfaces in space, Q represents the external supply of A , and t represents time. Depending on the physical quantity represented, A may be a scalar, vector, or tensor quantity, and $\boldsymbol{\tau}$ and Q must preserve consistency among the terms in Eq. (3-1).

Mixture theory, introduced by Eringen and Ingram (1965), provides a convenient methodology for describing multiphase, multispecies flow, and energy transport. Equation (3-1) describes the mixture as a whole. Additional variables may be defined describing the components of the mixture, where an α subscript denotes a phase-averaged variable and adding an i superscript specifies a species within the α phase. In general, mixture theory allows several solid and liquid phases as well as a gas phase; the BREATH simulator assumes that there is a single solid and a single liquid phase. Phase and species variables include:

- θ_α = the volume fraction occupied by phase α , or volume of phase α per unit volume of mixture
- ρ_α = the mass density of phase α (the mass of α per unit volume of mixture)
- \mathbf{v}_α = the bulk velocity of phase α
- w_α^i = the mass fraction of species i in phase α (the mass of i per unit mass of phase α)
- \mathbf{v}_α^i = the velocity of species i in phase α

These variables obey the following restrictions:

$$\sum_{\alpha} \theta_{\alpha} = 1, \quad (3-2)$$

$$\sum_i w_{\alpha}^i = 1, \quad (3-3)$$

$$\rho = \sum_{\alpha} \sum_i \theta_{\alpha} \rho_{\alpha} w_{\alpha}^i, \quad (3-4)$$

$$\mathbf{v} = \frac{1}{\rho} \sum_{\alpha} \sum_i \theta_{\alpha} \rho_{\alpha} w_{\alpha}^i \mathbf{v}_{\alpha}^i, \quad (3-5)$$

and the phase average, B_{α} , of a generic species variable, B_{α}^i , is defined by

$$B_{\alpha} = \sum_i w_{\alpha}^i B_{\alpha}^i. \quad (3-6)$$

Related to Eq. (3-2), the volumetric fractions of the liquid phases must sum to the void space volumetric fraction, or porosity.

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. For a multiphase mixture of N phases and M species, the general local constituent balance law for a quantity A in phase α and species i is

$$\begin{aligned} \frac{\partial}{\partial t} (\theta_{\alpha} \rho_{\alpha} w_{\alpha}^i A_{\alpha}^i) + \nabla \cdot (\theta_{\alpha} \rho_{\alpha} w_{\alpha}^i A_{\alpha}^i \mathbf{v}_{\alpha}^i) + \nabla \cdot \tau_{\alpha}^i - \theta_{\alpha} \rho_{\alpha} w_{\alpha}^i Q_{\alpha}^i = e_{\alpha}^i, \\ \alpha = 1, \dots, N; i = 1, \dots, M, \end{aligned} \quad (3-7)$$

where τ_{α}^i and Q_{α}^i are specializations of τ and Q , and e_{α}^i represents the exchange of A into species i in phase α from other phases and other species. A similar equation holds for each species and phase.

In order that A is conserved across all species and all phases, it is necessary that

$$\sum_{\alpha, i} e_{\alpha}^i = 0. \quad (3-8)$$

The general local constituent balance law is appropriate for mass, momentum, angular momentum, and energy. For generality, the assignment of physical meaning to A , τ , and Q is shown in Table 3-1 for each of these conserved quantities; however, only the mass and energy balance law will be considered in detail.

3.2 CONSERVATION OF MASS

For the conservation of the mass of a species i in phase α , the quantities in Eq. 3-7 are replaced with the corresponding quantities defined in Table 3-1, resulting in

$$\begin{aligned} \frac{\partial}{\partial t} (\theta_{\alpha} \rho_{\alpha} w_{\alpha}^i) + \nabla \cdot (\theta_{\alpha} \rho_{\alpha} w_{\alpha}^i \mathbf{v}_{\alpha}^i) - \theta_{\alpha} \rho_{\alpha} w_{\alpha}^i S_{m\alpha}^i = e_{\alpha}^i, \\ \alpha = 1, \dots, N; i = 1, \dots, M. \end{aligned} \quad (3-9)$$

Table 3-1: Assignment of physical meaning to symbols in the general local balance law [based on Table 1-1 of Allen et al. (1988)]

Balance Law	A	τ	Q
Mass	1	0	S_m
Momentum	\mathbf{v}	\mathbf{t}	\mathbf{b}
Angular momentum	$\mathbf{x} \times \mathbf{x}$	$\mathbf{x} \times \mathbf{t}$	$\mathbf{x} \times \mathbf{b}$
Energy	$U + \frac{1}{2} \mathbf{v} \cdot \mathbf{v}$	$\mathbf{q}_e + \mathbf{t} \cdot \mathbf{v}$	$S_e + \mathbf{b} \cdot \mathbf{v}$

Interpretations ($J=ML^2 T^{-2}$):
 \mathbf{t} = stress tensor [$ML^{-1}T^{-2}$]
 \mathbf{b} = body force vector per unit mass [LT^{-2}]
 \mathbf{x} = spatial coordinate vector [L]
 U = internal energy [JM^{-1}]
 \mathbf{q}_e = heat flux vector [$JT^{-1}L^{-2}$]
 S_m = external mass source rate per unit mass [T^{-1}]
 S_e = external energy source [L^2T^{-3}]

This may be rewritten in the form

$$\frac{\partial}{\partial t} (\theta_\alpha \rho_\alpha w_\alpha^i) + \nabla \cdot (\theta_\alpha \rho_\alpha w_\alpha^i \mathbf{v}_\alpha) + \nabla \cdot \mathbf{j}_\alpha^i - \theta_\alpha \rho_\alpha w_\alpha^i S_{m\alpha}^i = e_\alpha^i, \quad (3-10)$$

$$\alpha = 1, \dots, N; i = 1, \dots, M,$$

where \mathbf{j}_α^i is the diffusive flux of species i in phase α , defined by

$$\mathbf{j}_\alpha^i = \theta_\alpha \rho_\alpha w_\alpha^i (\mathbf{v}_\alpha^i - \mathbf{v}_\alpha). \quad (3-11)$$

Of immediate interest is movement of water in the near-surface soil column, both as a liquid phase and as a vapor species. Only a restricted system is considered, where the phases of interest are a solid (soil) phase; a wetting liquid (water) phase; and a gas phase. These are signified by s , w , and g , respectively. The primary species of interest are the water species in the liquid phase and the water species in the gas phase (signified by w). For this system, common assumptions are:

- The gas phase pressure throughout 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 ($\theta_g \rho_g w_g^w \mathbf{v}_g = 0$). This assumption disregards gas-phase trapping (which tends to slow down invading liquid pulses) and atmospheric pumping.
- The solid matrix is immobile, insoluble, and essentially incompressible, thus the mass balance equation for the solid phase can be neglected.
- Redistribution of the liquid phase is not affected by thermal gradients.
- 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.
- The liquid phase is entirely composed of the water species ($w_w^w = 1$); accordingly, there is no net dispersive flux of the water species in the liquid phase ($\mathbf{j}_w^w = 0$).

Several of these assumptions may need to be relaxed in order to simulate the behavior of a region that is strongly affected by heating from the proposed repository at Yucca Mountain.

As the water species, in both the liquid and gas phases, is the only species considered in the restricted system of interest, notational simplification is adopted in the remainder of this document blurring the distinction between a phase and a species. In general, a v refers to the vapor species in the gas phase; for example, ρ_v refers to $\rho_g w_g^v$. Also, as the water species is the only species in the liquid phase, w superscripts are dropped when referring to the liquid phase.

The mass balance equation for the water species in the liquid phase may be written in the form

$$\frac{\partial}{\partial t} (\theta_w \rho_w) + \nabla \cdot (\theta_w \rho_w \mathbf{v}_w) - \theta_w \rho_w S_{mw}^w = e_w^w, \quad (3-12)$$

and the corresponding equation for the water species in the gas phase may be written

$$\frac{\partial}{\partial t} (\theta_g \rho_v) + \nabla \cdot \mathbf{j}_g^w - \theta_g \rho_v S_{mg}^w = -e_w^w. \quad (3-13)$$

A water conservation equation can be derived by adding up the water species conservation equations over the phases, resulting in

$$\frac{\partial}{\partial t} (\theta_w \rho_w + \theta_g \rho_v) + \nabla \cdot (\theta_w \rho_w \mathbf{v}_w) + \nabla \cdot \mathbf{j}_g^w - M_{ext} = 0, \quad (3-14)$$

$$M_{ext} = \theta_w \rho_w S_{mw}^w + \theta_g \rho_v S_{mg}^w. \quad (3-15)$$

Assuming Darcy's law can be extended to situations where there is more than one phase occupying the pore space, which is the usual assumption for multiphase flow, results in the relationship

$$\mathbf{q}_w = \theta_w \mathbf{v}_w = -\mathbf{k}\lambda \cdot (\nabla P_w + \rho_w g \nabla z) , \quad (3-16)$$

where

- \mathbf{k} = intrinsic permeability tensor [L^2]
- λ = mobility [$LT\mu^{-1}$], or relative permeability divided by viscosity
- P_w = pressure in the liquid phase [$ML^{-1}T^{-2}$]
- g = gravitational acceleration constant [LT^{-2}]
- z = elevation above a reference datum [L]

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

$$\psi \equiv \int_{P_{w0}}^{P_w} \frac{1}{g\rho_w(P)} dP , \quad (3-17)$$

and where P_{w0} is a reference pressure (such as atmospheric pressure) [Bear, (1972)]. Strictly speaking, such formulations are only valid when ρ_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 in the form

$$\mathbf{q}_w = -\mathbf{K}_{sat} k_r(\psi) \cdot (\nabla \psi + \nabla z) , \quad (3-18)$$

where

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

Often \mathbf{K}_{sat} is reported rather than \mathbf{k} ; because viscosity is temperature-dependent, one should also be aware of the temperature at which the measurement is made. In BREATH, the use of matric head rather than liquid pressure is accommodated by defining intrinsic permeability in the input file using the numeric value for \mathbf{K}_{sat} , and setting ρ_w , g , and μ to one.

A common constitutive assumption for the flow of gases in a porous medium is that Fick's law holds. Assuming Fick's law holds for vapor transport, diffusion of water vapor in the gaseous phase is described by (Jury et al., 1991, based on Philip and de Vries, 1957)

$$\mathbf{j}_g^w = \rho_v \mathbf{q}_v = -\theta_g \tau \mathbf{D}_a \cdot \nabla \rho_v , \quad (3-19)$$

where vapor flux is defined by \mathbf{q}_v (for notational convenience in the following), τ is a dimensionless tortuosity factor accounting for tortuous flow paths (often set to 0.66), and \mathbf{D}_a is the diffusivity of vapor in air [L^2T^{-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 of primary importance in dry media, this phenomenon is not considered in BREATH.

Including Eqs. (3-16) and (3-19) into Eq. (3-14) results in the final form of the moisture conservation equation as solved by BREATH,

$$\frac{\partial}{\partial t} (\theta_w \rho_w + \theta_g \rho_v) + \nabla \cdot (\rho_w \mathbf{q}_w + \rho_v \mathbf{q}_v) - M_{ext} = 0, \quad (3-20)$$

The procedure for solving Eq. (3-20) is presented in Chapter 4.

3.3 CONSERVATION OF ENERGY

To obtain the conservation of energy balance law of the mass of a species i in phase α , the quantities in Eq. (3-7) are replaced with the corresponding quantities defined in Table 3-1. Introducing the total energy $E = U + \mathbf{v} \cdot \mathbf{v} / 2$, the resultant balance law is

$$\frac{\partial}{\partial t} (\theta_\alpha \rho_\alpha w_\alpha^i E_\alpha^i) + \nabla \cdot (\theta_\alpha \rho_\alpha w_\alpha^i E_\alpha^i \mathbf{v}_\alpha^i) + \nabla \cdot \theta_\alpha (\mathbf{q}_e + \mathbf{t} \cdot \mathbf{v})_\alpha^i - \theta_\alpha \rho_\alpha w_\alpha^i (\mathbf{b} \cdot \mathbf{v} + S_e)_\alpha^i = e_\alpha^i, \quad (3-21)$$

$\alpha = 1, \dots, N; i = 1, \dots, M.$

The BREATH simulator is primarily intended to address conditions appropriate to the relatively warm and semi-arid conditions in the near-surface porous media at Yucca Mountain. In the following development, it is assumed that:

- Kinetic and potential energy are both negligible compared to internal energy
- All phases are in thermal equilibrium
- A jump in energy may occur when a species changes phase
- Viscous dissipation of kinetic energy into heat is negligible
- 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

Some of these assumptions would need to be examined, and possibly relaxed, if simulations in the near-field region of a hot repository were desired.

In porous media flow, velocities are usually so small that the kinetic energy contribution to total energy may be neglected ($\mathbf{v} \cdot \mathbf{v} \ll U$); similarly, the work done by body forces, such as gravity, is negligible ($\mathbf{b} \cdot \mathbf{v} \approx 0$). The resultant balance law, omitting these terms, reduces to

$$\frac{\partial}{\partial t} (\theta_\alpha \rho_\alpha w_\alpha^i U_\alpha^i) + \nabla \cdot (\theta_\alpha \rho_\alpha w_\alpha^i U_\alpha^i \mathbf{v}_\alpha) + \nabla \cdot \theta_\alpha (\mathbf{q}_e + \mathbf{t} \cdot \mathbf{v})_\alpha^i - \theta_\alpha \rho_\alpha w_\alpha^i S_{e\alpha}^i = e_\alpha^i, \quad (3-22)$$

$$\alpha = 1, \dots, N; i = 1, \dots, M.$$

Summing over all species and all phases results in the total energy conservation equation

$$\sum_\alpha \left[\frac{\partial}{\partial t} (\theta_\alpha \rho_\alpha U_\alpha) + \nabla \cdot (\theta_\alpha \rho_\alpha U_\alpha \mathbf{v}_\alpha) + \nabla \cdot \theta_\alpha (\mathbf{q}_e + \mathbf{t} \cdot \mathbf{v})_\alpha - \theta_\alpha \rho_\alpha S_{e\alpha} \right] = 0. \quad (3-23)$$

The heat flux, or $\sum_\alpha \theta_\alpha \mathbf{q}_{e\alpha}$ is parametrized as a function of the multiphase mixture composition.

Assuming Fourier's law applies to the conduction of sensible heat,

$$\sum_\alpha \theta_\alpha \mathbf{q}_{e\alpha} = -\mathbf{K}_e \cdot \nabla T, \quad (3-24)$$

where \mathbf{K}_e is the thermal conductivity tensor [$\text{JL}^{-1}\text{T}^{-1}\text{K}^{-1}$], which is a function of the phase saturations and properties, and T is temperature.

Assuming that viscous dissipation of kinetic energy into heat is negligible, which is equivalent to assuming that momentum transfers attributable to shear stresses in the fluid are negligible compared to momentum transfers between the fluid and the rock, yields the relationship

$$\nabla \cdot [\theta_\alpha (\mathbf{t}_\alpha \cdot \mathbf{v}_\alpha)] = \nabla \cdot (\theta_\alpha P_\alpha \mathbf{v}_\alpha). \quad (3-25)$$

Incorporating the constitutive assumptions in Eqs. (3-24) and (3-25) into Eq. (3-23), defining enthalpy, $H = U + P/\rho$, and replacing subscripts,

$$\frac{\partial}{\partial t} (\rho U) + \nabla \cdot (\rho H \mathbf{v}) - \nabla \cdot (\mathbf{K}_e \cdot \nabla T) - E_{ext} = 0, \quad (3-26)$$

where

$$\rho U = \theta_s \rho_s U_s + \theta_w \rho_w U_w + \theta_g \rho_g U_g, \quad (3-27)$$

$$\rho H \mathbf{v} = \theta_s \rho_s H_s \mathbf{v}_s + \theta_w \rho_w H_w \mathbf{v}_w + \theta_g \rho_g H_g \mathbf{v}_g, \quad (3-28)$$

$$E_{ext} = \sum_\alpha \theta_\alpha \rho_\alpha S_{e\alpha}. \quad (3-29)$$

Due to the assumption that the porous medium is immobile, the first term on the right-hand side of Eq. (3-28) is zero, as $\mathbf{v}_s = 0$. To be completely consistent with the assumption that the gas phase instantaneously equilibrates throughout the medium would similarly require that $\mathbf{v}_g = 0$; however, the vapor species carries a disproportionate amount of energy compared to the air species, in the form of latent heat, thus flux of the vapor species is retained in the energy equation but not the flux of the air species. With these assumptions, Eq. (3-28) reduces to

$$\rho H \mathbf{v} = \rho_w H_w \mathbf{q}_w + \rho_v H_v \mathbf{q}_v \quad (3-30)$$

The vapor flux term is expected to be negligible throughout most of the domain, except in dry regions with strong temperature gradients, such as occur in the top few centimeters from the ground surface.

Linear variation of internal energy and enthalpy with temperature is considered a satisfactory approximation for liquids and solids, particularly in the range of temperatures likely to be encountered near the interface between a ground surface and the atmosphere; however, for gases, the effect of pressure variation may not be negligible in general (Green et al., 1993). As the gas phase is assumed to maintain atmospheric pressure, pressure dependence of gas phase internal energy and enthalpy is not considered in BREATH. Accounting for phase changes to and from the liquid phase, the linear approximation yields

$$U_\alpha = U_{l\alpha} + U_\alpha(T_0) + \left(\frac{dU_\alpha}{dT} \right)_V (T - T_0), \quad (3-31)$$

$$H_\alpha = H_{l\alpha} + H_\alpha(T_0) + \left(\frac{dH_\alpha}{dT} \right)_P (T - T_0), \quad (3-32)$$

where T_0 is the reference temperature at which all parameters are measured, and

- $U_{l\alpha}$ = change in internal energy from liquid to phase α , or $U_\alpha - U_l$ [JM^{-1}]
- $U_\alpha(T_0)$ = internal energy in phase α , evaluated at T_0 [JM^{-1}]
- $(dU_\alpha/dT)_V$ = rate of change of internal energy with respect to temperature, holding the volume fixed [$\text{J M}^{-1}\text{K}^{-1}$]
- $H_{l\alpha}$ = change in enthalpy from liquid to phase α , or $H_\alpha - H_l$ [JM^{-1}]
- $H_\alpha(T_0)$ = enthalpy in phase α , evaluated at T_0 [JM^{-1}]
- $(dH_\alpha/dT)_P$ = rate of change of enthalpy with respect to temperature, holding the pressure fixed [$\text{JM}^{-1}\text{K}^{-1}$]

The rate of change of internal energy with respect to temperature at constant volume, $(dU_\alpha/dT)_V$, is referred to as the specific heat at constant volume, c_v , and the rate of change of enthalpy with respect to temperature at constant pressure, $(dH_\alpha/dT)_P$, is referred to as the specific heat at constant pressure, c_p .

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)

$$H_{l\alpha} = U_{l\alpha} + P \left(\frac{1}{\rho_\alpha} - \frac{1}{\rho_l} \right), \quad (3-33)$$

where P is pressure and ρ_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.

Since internal energy at the reference temperature is a constant, and can freely be brought in and out of derivatives, Eq. (3-20) can be employed to result in

$$U_w(T_0) \left[\frac{\partial}{\partial t} (\theta_w \rho_w) + \nabla \cdot (\rho_w \mathbf{q}_w) \right] = -U_w(T_0) \left[\frac{\partial}{\partial t} (\theta_g \rho_v) + \nabla \cdot (\rho_v \mathbf{q}_v) - M_{ext} \right]. \quad (3-34)$$

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.

The final form of the energy balance law is obtained by substituting Eqs. (3-27), (3-29) through (3-32), and (3-34), into Eq. (3-26) and rearranging in terms of the temperature difference $\Phi = T - T_0$ to minimize numerical roundoff. Note that as ∇T_0 is identically zero, $\nabla \Phi = \nabla T$. The form of the energy balance law solved by BREATH is

$$\frac{\partial}{\partial t} [C_1 \Phi + C_2] + \nabla \cdot [M_1 \Phi + M_2 - \mathbf{K}_e \cdot \nabla \Phi] - E_{ext} = 0, \quad (3-35)$$

where

$$C_1 = \theta_s \rho_s c_{vs} + \theta_w \rho_w c_{vw} + \theta_g \rho_g c_{vg}, \quad (3-36)$$

$$C_2 = \theta_g \rho_g U_g(T_0) + \theta_g \rho_v [U_{lv} - U_w(T_0)], \quad (3-37)$$

$$M_1 = \rho_w \mathbf{q}_w c_{pw} + \rho_v \mathbf{q}_v c_{pv}, \quad (3-38)$$

$$M_2 = \rho_w \mathbf{q}_w [H_w(T_0) - U_w(T_0)] + \rho_v \mathbf{q}_v [H_v(T_0) + H_{lv} - U_w(T_0)]. \quad (3-39)$$

In Eq. (3-35), E_{ext} now incorporates the mass term arising from the Eq. (3-34) substitution. 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 time and space derivatives of the $\theta_s \rho_s U_s(T_0)$ and $\theta_s \rho_s U_{ls}$ terms are zero and thus are not included.

4 NUMERICAL MODEL

This chapter presents the specific numerical implementation of the mathematical model developed in Chapter 3. Generic issues in finite element procedures used to solve the mass and energy balance equations are briefly discussed in Section 4.1, a more specific treatment is developed in Section 4.2, with boundary conditions presented in Section 4.3, and constitutive equations are discussed in Section 4.4. Finally, issues related to effective control of simulation time stepping are introduced in Section 4.5.

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 second-order parabolic 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 .

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

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

satisfies the governing equation 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,

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

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 mixed-delta finite element 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,

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

where t is the time at the beginning of a time step, Δt is the size of the time step, here β is a time-weighting parameter ranging from 0 to 1, and the superscript refers to the time at which the term is evaluated. When β is 0, the resulting scheme is termed fully explicit; when β 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, and implicit schemes require additional computational effort for each time step. 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 found. 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.

The mixed-delta formulation with modified Picard iteration is discussed in Celia et al. (1990), where the procedure is used to solve for the change in the dependent variables during a nonlinear iteration within a time step. Using a simple model equation as an example,

$$\frac{\partial A}{\partial t} + \nabla \cdot \mathbf{q} + M = 0, \quad (4-6)$$

the fully implicit finite element assembly process results in an equation in the form

$$\frac{1}{\Delta t} (A^{m+1, n+1} - A^n) + \nabla \cdot \mathbf{q}^{m+1, n+1} + M^{m+1, n+1} = 0, \quad (4-7)$$

where n refers to the time-step number, m refers to the nonlinear iteration number within the time step, and other variables are generic quantities. As the nonlinear iteration only occurs for variables at the $n+1$ time-step level, for notational brevity $n+1$ will be implicitly assumed in the following whenever an m or $m+1$ is present. The modified Picard iteration process starts by transforming Eq. (4-7) into the form

$$\frac{1}{\Delta t} (A^{m+1} + (A^m - A^m) - A^n) + \nabla \cdot [\mathbf{q}^{m+1} + (\mathbf{q}^m - \mathbf{q}^m)] + M^{m+1} + (M^m - M^m) = 0, \quad (4-8)$$

which can be rearranged into

$$\frac{1}{\Delta t} (A^{m+1} - A^m) + \nabla \cdot (\mathbf{q}^{m+1} - \mathbf{q}^m) + M^{m+1} - M^m + RES^m = 0, \quad (4-9)$$

$$RES^m = \frac{1}{\Delta t} (A^m - A^n) + \nabla \cdot \mathbf{q}^m + M^m. \quad (4-10)$$

Defining a generic dependent variable, χ , denoting $\chi^{m+1} - \chi^m$ by δ , and assuming that $\mathbf{q} = \mathbf{K} \cdot \nabla \beta$, where β is another generic dependent variable, the terms in Eq. (4-9) can be expanded into

$$\frac{1}{\Delta t} (A^{m+1} - A^m) = \frac{1}{\Delta t} \left(\frac{dA}{d\chi} \right)^m (\chi^{m+1} - \chi^m) = \frac{1}{\Delta t} \left(\frac{dA}{d\chi} \right)^m \delta, \quad (4-11)$$

$$\nabla \cdot (\mathbf{q}^{m+1} - \mathbf{q}^m) = \nabla \cdot \left[\mathbf{K}^m \left(\frac{d\beta}{d\chi} \right)^m \nabla (\chi^{m+1} - \chi^m) \right] = \nabla \cdot \left[\mathbf{K}^m \left(\frac{d\beta}{d\chi} \right)^m \nabla \delta \right], \quad (4-12)$$

$$M^{m+1} - M^m = \left(\frac{dM}{d\chi} \right)^m (\chi^{m+1} - \chi^m) = \left(\frac{dM}{d\chi} \right)^m \delta. \quad (4-13)$$

With this approach in hand, the mass conservation equation is recast as

$$\frac{1}{\Delta t} \left[\frac{d(\theta_w \rho_w + \theta_g \rho_v)}{d\chi} \right] \delta - \nabla \cdot \left[\left(\rho_w \mathbf{k} \lambda \frac{dP_w}{d\chi} + \theta_g \tau \mathbf{D}_a \frac{d\rho_v}{d\chi} \right) \cdot \nabla \delta \right] - \frac{dM_{ext}}{d\chi} \delta + RES^m = 0, \quad (4-14)$$

$$RES^m = \frac{1}{\Delta t} [(\theta_w \rho_w + \theta_g \rho_v)^m - (\theta_w \rho_w + \theta_g \rho_v)^n] + [\nabla \cdot (\rho_w \mathbf{q}_w + \rho_v \mathbf{q}_v) - M_{ext}]^m. \quad (4-15)$$

Only terms explicitly dependent on the primary dependent variables P_w , ρ_v , or θ_w are expanded. Any coefficients depending on the primary dependent variables are evaluated using the most current value of the variable. Using head or pressure for χ is a relatively standard approach, and this approach is followed in the code so that $\delta = P_w^{m+1} - P_w^m$. Using a different dependent variable for χ is also possible, such as saturation or vapor density, in cases where fully saturated systems are not present. As χ represents a dependent variable, derivatives with respect to χ can be evaluated using constitutive relationships between the dependent variables.

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 4.3.2), including the vapor boundary condition as part of the $(dM_{ext} / d\chi) \delta$ term in Eq. (4-14) 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 Φ as the dependent variable, where $\Phi = T - T_0$. With this substitution, the governing equation is

$$\frac{\partial}{\partial t} [C_1 \Phi + C_2] + \nabla \cdot [M_1 \Phi + M_2 - \mathbf{K}_e \cdot \nabla \Phi] - E_{ext} = 0, \quad (4-16)$$

where the coefficients are defined in Section 3.3. Unlike the procedure for solving the mass conservation equation, a δ formulation is not used; rather, Φ is directly determined in each iteration within a time step. Equation (4-16) is discretized using exactly the same procedures used for Eq. (4-14), resulting in a fully implicit approximation. 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 4.3.3), a boundary condition dependent on T^4 is imposed, where T is in an absolute temperature scale. This condition is handled implicitly by splitting the term into a known value and a term linearly varying with Φ , using the identity

$$T^4 = (\Phi + T_0) T^3 \quad (4-17)$$

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 Φ , determining the Φ -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 BOUNDARY CONDITIONS

There is 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,

$$\gamma_1 u + \gamma_2 \frac{\partial u}{\partial n} = \gamma_3, \quad (4-18)$$

where u is the dependent variable of interest, γ_1 , γ_2 , and γ_3 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 ($\gamma_1=0$ and $\gamma_2=1$). First-type conditions, where the value of u is specified directly ($\gamma_1=1$ and $\gamma_2=0$), 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 specify the boundary condition for water, vapor, and energy, respectively. The local coordinate used in the domain is referred to by x , increasing from the first boundary to the second.

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

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.

4.3.2 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 4.3.4.

Table 4-2. Vapor-species boundary conditions

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

4.3.3 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. 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 4.3.4 and Appendix A.2.

Table 4-3. Energy equation boundary conditions

Input Keyword	Definition of B_e	Corresponding Boundary Condition
<i>temp</i>	$B_e = T$	$T = B_e$
<i>flux</i>	$B_e = q_e$	$q_{es} = B_e \frac{\partial x}{\partial n}$
<i>gradtemp</i>	$B_e = \partial T / \partial x$	$q_{es} = [M_1 (T - T_0) + M_2 - K_e B_e] (\partial x / \partial n) + Q_{ext}$ $Q_{ext} = \rho_g c_p k_h (T_a - T) + (1 - \alpha_s) S_t + \epsilon_a \sigma T_a^4 - \epsilon_s \sigma T^4$
Interpretations: k_h = boundary layer heat conductance α_s = ground surface albedo [0 to 1] S_t = net shortwave radiation (net solar radiation) ϵ_a = atmospheric emissivity [0 to 1] ϵ_s = ground surface emissivity [0 to 1] σ = the Stefan-Boltzmann constant T_a = atmospheric absolute temperature		

4.3.4 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 ρ_{va} , the net shortwave radiation S_t , either the atmospheric emissivity times the Stefan-Boltzmann coefficient $\epsilon_a \sigma$ or the net atmospheric longwave radiation ($\epsilon_a \sigma T_a^4$), the windspeed v_{wind} , and the boundary condition values B_e , B_v , and B_w .

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,

$$\alpha_s = \alpha_s^{dry} + \left(\frac{\theta_w}{\theta_{ws}} \right) (\alpha_s^{wet} - \alpha_s^{dry}), \quad (4-19)$$

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

$$\epsilon_s = \epsilon_s^{dry} + \frac{\theta_w}{\theta_{ws}} (\epsilon_s^{wet} - \epsilon_s^{dry}). \quad (4-20)$$

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).

4.3.5 Boundary Layer Conductances

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. Some relationships in the literature between boundary-layer conductance and windspeed 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 may be calculated assuming a logarithmic increase in windspeed with height. Following Brutsaert (1982) and Campbell (1985), conductances are parameterized by a dimensionless variable [Monin and Obukhov (1954)]

$$\zeta = \frac{z-d}{L}, \quad (4-21)$$

where z is a measurement height and L is Obukhov's (1946) (Businger and Yaglom, 1971) stability length, defined by

$$L = \frac{-U^3 \rho}{\kappa g \left[\left(\frac{H}{T_a c_p} \right) + 0.61 E \right]}, \quad (4-22)$$

$$U = \frac{\kappa v_{wind}}{\ln[(z^U - d + z_m)/z_m] - \Psi_m}, \quad (4-23)$$

$$H = \frac{\kappa U \rho c_p (T_s - T_a)}{\ln[(z^T - d + z_h)/z_h] - \Psi_h} = \rho c_p k_v (T_s - T_a), \quad (4-24)$$

$$E = \frac{\kappa U (\rho_s - \rho_a)}{\ln[(z^T - d + z_v)/z_v] - \Psi_v} = k_v (\rho_s - \rho_a), \quad (4-25)$$

where

- κ = von Karman's constant (generally assumed to be 0.4)
- U = friction velocity [LT^{-1}]
- z^T = height above surface at which temperature is measured [L]
- z^U = height above surface at which wind speed is measured [L]
- z_h = surface roughness length for heat [L]
- z_m = surface roughness length for momentum [L]
- z_v = surface roughness length for vapor [L]
- Ψ_h = stability correction factor for heat [-]
- Ψ_m = stability correction factor for momentum [-]
- Ψ_v = stability correction factor for vapor [-]

- d = zero plane displacement height [L]
 ρ = atmospheric density [ML³]
 c_p = specific atmospheric heat capacity [JM⁻¹K⁻¹]

Guidance on selecting these parameters is given in Brutsaert (1982) and Campbell (1985). Campbell (1985) presents a set of empirical correlations, based on typical crop surfaces, where $d = 0.77 h$, $z_m = 0.13 h$, and $z_h = 0.2 z_m$, and h is the crop height. Brutsaert (1982) reports a collection of experiments examining bluff and wavelike elements suggesting that (h / z_m) is in the range of 4.2 through 16.6, with other experimental results in the range of 7.35 to 7.6 for rough snow and vegetation; however, Brutsaert cautions that (h / z_m) is a complex function of surface characteristics. Brutsaert (1982) recommends using (d / h) as approximately 2/3 for most purposes.

The stability parameters are generally parameterized as a function of ζ . The resulting set of equations is circularly defined, with ζ depending on U , H and E , which depend on ψ_m , ψ_h and ψ_v , which in turn depend on ζ . An iterative scheme similar to that in Campbell (1985) is used to calculate the boundary layer conductances; calculating U , then H and E , then ζ , then ψ_h , ψ_v , and ψ_m , and cycling until convergence. ψ_h , ψ_v , and ψ_m are assumed zero to start the iteration sequence; ten iterations are used. The iteration sequence converges quickly when windspeeds are large enough relative to the measurement height.

Conductances may be calculated using relationships collected in Campbell (1985) by setting the *zblk* keyword to *campbell*. In the form that is implemented in the code, conductances are estimated using

$$k_v = k_h = \frac{\kappa U}{\ln[(z^T - d + z_h)/z_h] - \psi_h}, \quad (4-26)$$

$$U = \frac{\kappa v_{wind}}{\ln[(z^U - d + z_m)/z_m] - \psi_m}, \quad (4-27)$$

where

$$\psi_h = \begin{cases} -4.7\zeta & \text{if } T_s < T_a \\ 2 \ln \left[\frac{1}{2} (1 + x^2) \right] & \text{if } T_s > T_a \end{cases} \quad (4-28)$$

$$\psi_m = \begin{cases} \psi_h & \text{if } T_s < T_a \\ 0.6\psi_h & \text{if } T_s > T_a \end{cases}$$

$$x = (1 - 16\zeta)^{1/4}$$

Campbell (1985) tacitly assumes that E is negligible compared to H in the definition of L . In BREATH, it is assumed that k_v and k_h are the maximum of the value calculated in Eq. (4-26) and the diffusion coefficient of vapor in air divided by the measurement height.

By setting the *zblk* keyword to *brut82*, the full collection of Eqs. (4-21) through (4-25) are used directly, again requiring that the conductances are nonzero at low windspeeds by considering the diffusion coefficient for vapor in air. The stability relationships presented in Brutsaert (1982) are

$$\Psi_m = \begin{cases} -5 [1 + \ln(\zeta)] & \text{if } 1 < \zeta \\ -5\zeta & \text{if } 0 \leq \zeta \leq 1 \\ 2 \ln \left[\frac{1}{2}(1+x) \right] + \ln \left[\frac{1}{2}(1+x^2) \right] - 2 \operatorname{atan}(x) + \frac{\pi}{2} & \text{if } \zeta < 0 \end{cases} \quad (4-29)$$

$$\Psi_h = \begin{cases} \Psi_m & \text{if } \zeta \geq 0 \\ 2 \ln \left[\frac{1}{2}(1+x^2) \right] & \text{if } \zeta < 0 \end{cases}$$

As the iterative procedure does not converge well for low windspeeds, a further option is available to extrapolate the conductance values at low windspeeds from estimated values at higher windspeeds. Low-windspeed conductances are extrapolated linearly from windspeeds 10 and 11 times z^U , again requiring that the conductances are at least as large as D_a / z^U , by setting *zblk* to *camplin* or *brutlin*, respectively.

4.4 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.

4.4.1 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 and 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.

4.4.1.1 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

$$\theta_w(h) = \theta_{wr} + \frac{\theta_{ws} - \theta_{wr}}{[1 + (\alpha|h|)^n]^m} \quad (4-30)$$

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

$$P_c = P_0 (\theta_e^{-1/m} - 1)^{1-m} \quad (4-31)$$

where

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

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

$$P_c = P_0 \theta_e^{-m} \quad \text{for } 0 < \theta_e < 1, \quad (4-32)$$

where P_c is set to a maximum and a minimum value when θ_e is zero or one, respectively. Note that when the Brooks-Corey and van Genuchten-Mualem relationships are applied to the same soil sample, P_0 and m will have a different numeric value in the two relationships. Selection of the pressure-saturation relationship is controlled using the input keyword *zpssel*.

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.

4.4.1.2 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 λ , is

$$\lambda(h) = \frac{1}{\mu} \left(\frac{(1 - (\alpha|h|)^{n-1} [1 + (\alpha|h|)^n]^{-m})^2}{[1 + (\alpha|h|)^n]^{m/2}} \right). \quad (4-33)$$

Using the pressure-based notation, again assuming $m = 1 - 1/n$, and using Eq. (4-31), results in

$$\lambda(\theta_e) = \frac{1}{\mu} \theta_e^{1/2} \left[1 - (1 - \theta_e^{1/m})^m \right]^2. \quad (4-34)$$

The corresponding Brooks-Corey relationship is

$$\lambda(\theta_e) = \frac{1}{\mu} \theta_e^{2m+3}, \quad (4-35)$$

where, as before, m has a different numerical value in the van Genuchten and Brooks-Corey relationships. Selection of the mobility relationship is controlled using the input keyword *zmosel*.

4.4.1.3 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

$$\frac{\partial}{\partial t} (\theta_w \rho_w + \theta_g \rho_v) = \frac{S_s}{g} \frac{\partial P_w}{\partial t}, \quad (4-36)$$

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.

The default in BREATH is to follow the procedure outlined in Paniconi et al. (1991). The Paniconi et al. procedure uses the storage term resulting from S_s if P_w is greater than a cutoff in pressure, otherwise, the unsaturated storage term is used. The cutoff P_w is automatically selected so that the storage term is a continuous function of pressure. For the Brooks-Corey relationship, the transition between unsaturated and saturated storage is at $P_w = 0$. For the van Genuchten relationship, a preprocessing step is used to find the cutoff pressure, using a line search algorithm. The van Genuchten unsaturated storage term increases from zero at zero saturation to a maximum value, then monotonically decreases from the maximum value to zero at full saturation. The line search algorithm scans the range of saturations from the saturation with the maximum storage value to full saturation. The saturation at which the maximum value occurs is approximately

$$\theta_e |_{\text{max storage}} = a_1 m^4 + a_2 m^3 + a_3 m^2 + a_4 m + a_5, \quad (4-37)$$

Table 4-4. Typical thermal properties of soil constituents (after Hillel, 1980, Tables 9.1 and 9.2)

Constituent	Density ρ (gm cm^{-3})	Heat Capacity C_V ($\text{cal gm}^{-1} \text{K}^{-1}$)	Thermal Conductance k_e ($10^{-3} \text{cal cm}^2 \text{gm}^{-1} \text{sec}^{-1} \text{K}^{-1}$)
Quartz	2.66	0.18	21
Other minerals	2.65	0.18	7
Organic matter	1.3	0.46	0.6
Liquid water	1.0	1.0	1.37
Air	0.00125	2.4	0.06

where a_1 is -0.146148 , a_2 is 0.705306 , a_3 is -1.06878 , a_4 is 0.00999538 , and a_5 is 0.999604 and m refers to the van Genuchten parameter.

Two other options are available which are used in the literature, but which can strongly affect the moisture capacity in dry media. Both schemes weight the specific storage by a linearly decreasing function of moisture content, so that

$$\frac{\partial}{\partial t} (\theta_w \rho_w + \theta_g \rho_v) = \left[\frac{\partial (\theta_w \rho_w + \theta_g \rho_v)}{d\rho_w} + f(\theta_w) \frac{S_s}{g} \right] \frac{\partial P_w}{\partial t} \quad (4-38)$$

One scheme weights the specific storage term by effective saturation [$f(\theta_w) = (\theta_w - \theta_r) / (\theta_s - \theta_r)$]; the other weights the specific storage term by normalized moisture content [$f(\theta_w) = \theta_w / \theta_s$].

4.4.2 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

$$\mathbf{K}_e = \sum_{\alpha} \theta_{\alpha} \rho_{\alpha} \mathbf{k}_e^{\alpha} \quad (4-39)$$

where \mathbf{k}_e^{α} is the thermal conductance for phase α . Representative values for quartz are shown in Table 4-4.

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.

4.4.3 Fluid Properties

The vapor in the porous medium is assumed to be at equilibrium with the liquid phase (Edlefsen and Anderson, 1943), with the equilibrium described by (modified from Campbell, 1985):

$$\rho_v = \rho_{vs} \exp\left(\frac{P_w M_w}{\rho_w RT}\right) \quad \text{for } P_w < 0, \quad (4-40)$$

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

$$\rho_{vs} = a_1 \exp [a_2 - (a_3/T) - a_4 \ln(T)]. \quad (4-41)$$

In the cgs system, $a_1 = 1$, $a_2 = 46.440973$, $a_3 = 6790.4985$, $a_4 = 6.02808$, ρ_{vs} is in 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, controlled by the input keyword *zwusel*; a constant value, and an empirical power-law expression suitable for liquid water (CRC, 1982, p. F-40):

$$\log_{10}\left(\frac{\mu_T}{\mu_0}\right) = \begin{cases} \left[\frac{a_1}{a_2 + a_3(T - T_r) + a_4(T - T_r)^2} - \frac{a_1}{a_2} \right] & \text{for } 0 \leq T \leq 20 \\ \left[\frac{a_5(T_r - T) + a_6(T_r - T)^2}{T + a_7} \right] & \text{for } 20 < T \leq 100 \end{cases} \quad (4-42)$$

where T is in degrees Celsius, T_r is 20 °C, μ_T and μ_0 are in centipoise (100 cp = 1 gm cm⁻¹ sec⁻¹), and the viscosity at T_r , μ_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 μ_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.

4.5 TIME STEPPING AND ACTIVE REGION CONTROL

4.5.1 Constant and Courant-Based Time Stepping

As discussed in Section 4.2, 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,

$$C_o = \frac{\mathbf{v}\Delta t}{\Delta x}, \quad (4-43)$$

where \mathbf{v} is $\mathbf{q}_m c_{VW} + \mathbf{q}_v c_{Vv}$, Δt is the time step, and Δx 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.

4.5.2 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, Δt^{new} , is selected by

$$\Delta t^{new} = \Delta t^{old} \left(\frac{10 + \min(N_{max} + 1, 10)}{10} \right) \left(\frac{\ln(C_{nom}) - \ln(C) + N_{nom} - N}{N} \right), \quad (4-44)$$

where

- Δt^{old} = the previous time step
- 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
- 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

$$\Delta t^{new} = \Delta t^{old} \left(\frac{\ln(C_0) - \ln(C)}{\ln(C_0) - \ln(C_{nom})} \right) \left(\frac{N_{nom}}{N} \right), \quad (4-45)$$

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 two iterations per additional requested cycle, both for N_{nom} and for N_{max} .

4.5.3 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.

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 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 α at 0.0335 and n at 2, corresponding to values of P_0 at 29.85 and m at 0.5. 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 C.1 and the associated output file is shown in Appendix C.2.

BREATH and UNSAT1D profiles are compared at several output time steps in Figures 5-1 and 5-2. The two codes yield indistinguishable profiles.

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

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

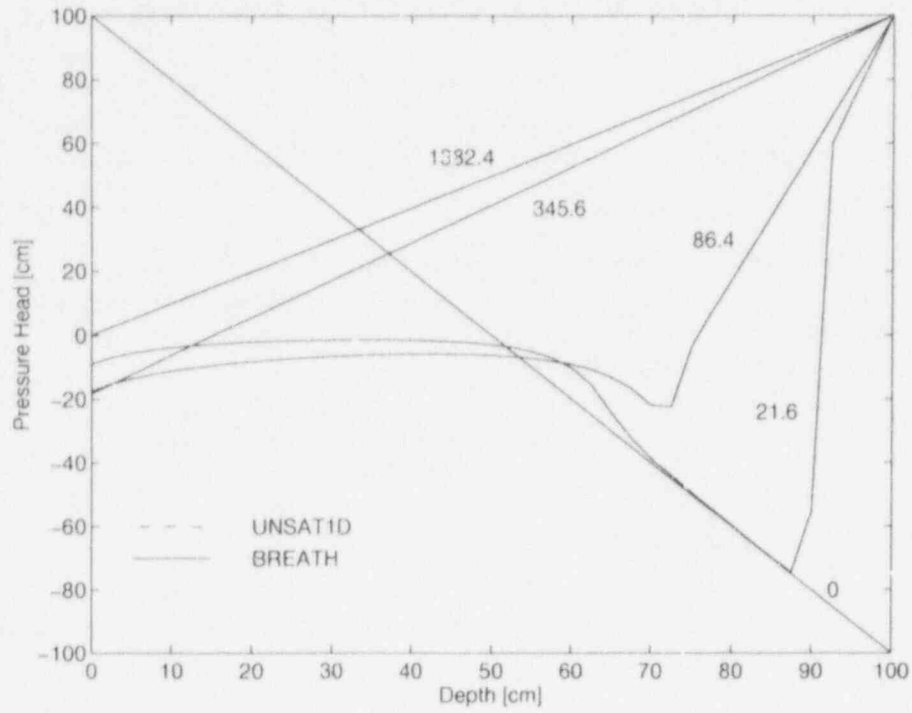


Figure 5-1. Pressure distribution at selected time instants in the redistribution example

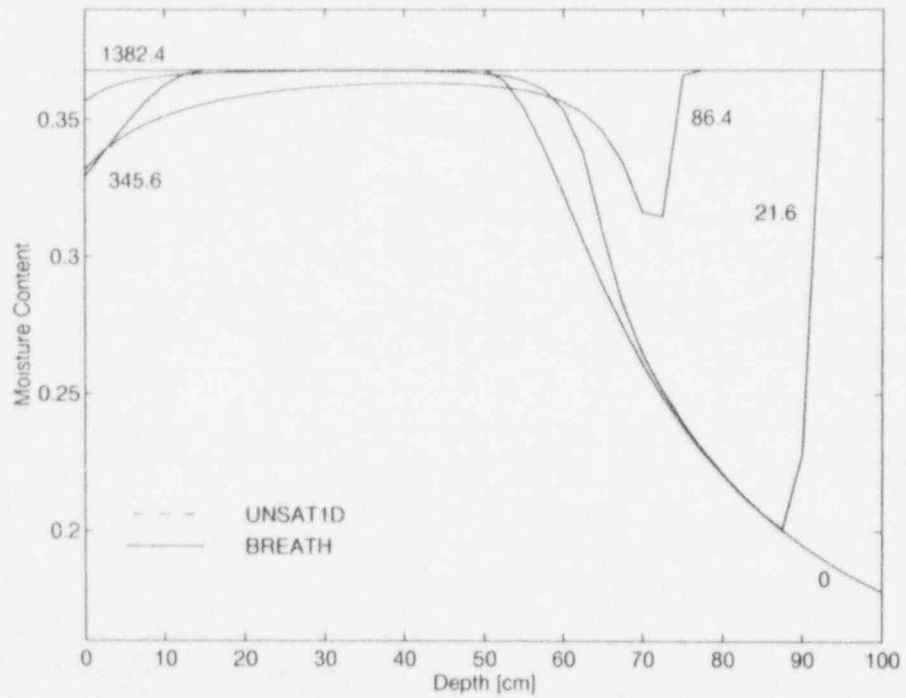


Figure 5-2. Moisture content distribution at selected time instants in the redistribution example

where

- T = temperature
- T_{mean} = the mean temperature
- T_{amp} = the temperature variation amplitude at the boundary
- x = distance from the boundary
- t = time
- ω = angular frequency of excitation
- 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, ω 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

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

allows either K_e 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 *dudts* parameter to be 1 results in a corresponding thermal conductivity of 8.181×10^{-2} .

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, boundary condition, and output files for this problem are shown in Appendices C.3, C.4, and C.5, 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)

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

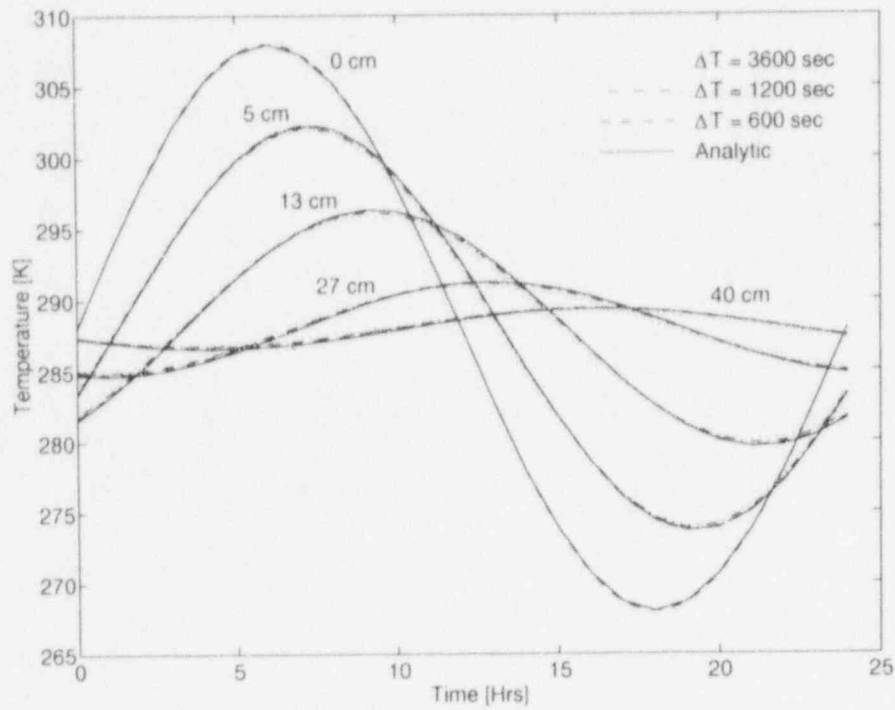


Figure 5-3. Temperature trace at selected depths in the sinusoidal-forcing example

where

T = temperature

x = distance from the boundary

t = time

V = velocity of the phase (Darcy flux divided by porosity)

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

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

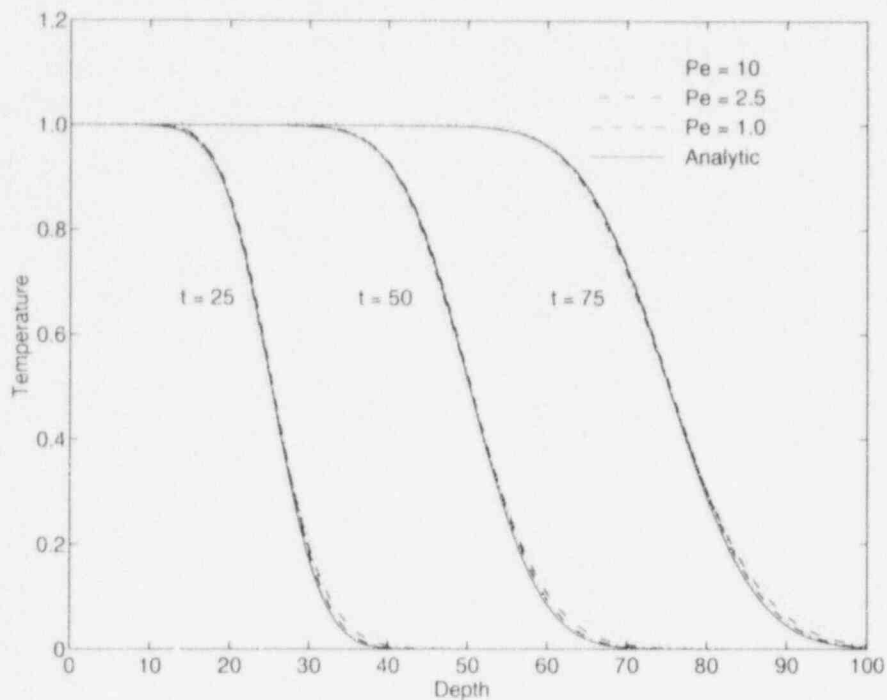


Figure 5-4. Temperature distribution at selected times in the advancing-front example

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 Δx and Δt 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 C.6, with associated output file and file for nodal time-snapshot information in Appendices C.7 and C.9.

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 hypothetical sequence of layers that similar to formations at YM: 30 cm of alluvium (alluv), 180 cm of Tiva Canyon columnar moderately welded tuff (tccmw), and 400 cm of Tiva Canyon shardy base (tcsbar). 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 C.10 and C.11 with the corresponding output file in Appendix C.12.

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

Phase α	ρ_α (gm cm ⁻²)	k_c^α (J cm ² s ⁻¹ gm ⁻¹ K ⁻¹)	$(dU_\alpha/dT)_V$ (J gm ⁻¹ K ⁻¹)	$(dH_\alpha/dT)_P$ (J gm ⁻¹ K ⁻¹)	$H_{f\alpha}$ (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 (cm ² s ⁻¹)	P_0 (Pa)	m	θ_{ws}	θ_{wr}	θ_s
alluv	1.15×10^{-6}	9.80×10^6	0.231	0.098	0.0001	0.098
tccmw	6.65×10^{-8}	1.14×10^6	0.231	0.240	0.0010	0.240
teshar	1.84×10^{-6}	1.14×10^6	0.231	0.420	0.0010	0.420

Table 5-3. Miscellaneous constants in the coupled example

Symbol	Value	Units	Symbol	Value	Units
R	8.3143×10^7	cm ² gm mole ⁻¹ K ⁻¹ s ⁻²	M	18	gm mole ⁻¹
μ_w	11.24	gm cm ⁻¹ s ⁻¹	D_v	0.1584	cm ² s ⁻¹
g	981	cm s ⁻²	k	0.4	
z^I	300	cm	z^{II}	300	cm
z_h	0.03	cm	z_m	0.03	cm
d	0	cm	σ	5.67×10^{-12}	J s ⁻¹ cm ⁻² K ⁻⁴
α_s^{dry}	0.25		α_s^{wet}	0.25	
ϵ_s^{dry}	0.9		ϵ_s^{wet}	0.918	

The column starts with uniform initial pressure of -10^8 Pascal and initial temperature of 286 °K. A 2.46 mm pulse of water, roughly equivalent to 1/3 of the average monthly precipitation for March at YM, is introduced over the first hour. 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 1-hr intervals in the met file (see Appendix A for detailed instructions on met file usage). Atmospheric vapor density, ρ_{va} , is held constant at 3.42×10^{-6} gm cm⁻³, emissivity of air, ϵ_a , is held constant at 0.812, and wind speed is held constant at 406.8 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 J s⁻¹cm⁻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, ϕ is latitude (37 °) and δ 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 response of the moisture content profile to 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 slightly slower evaporation rates 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 essentially 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 finely 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.

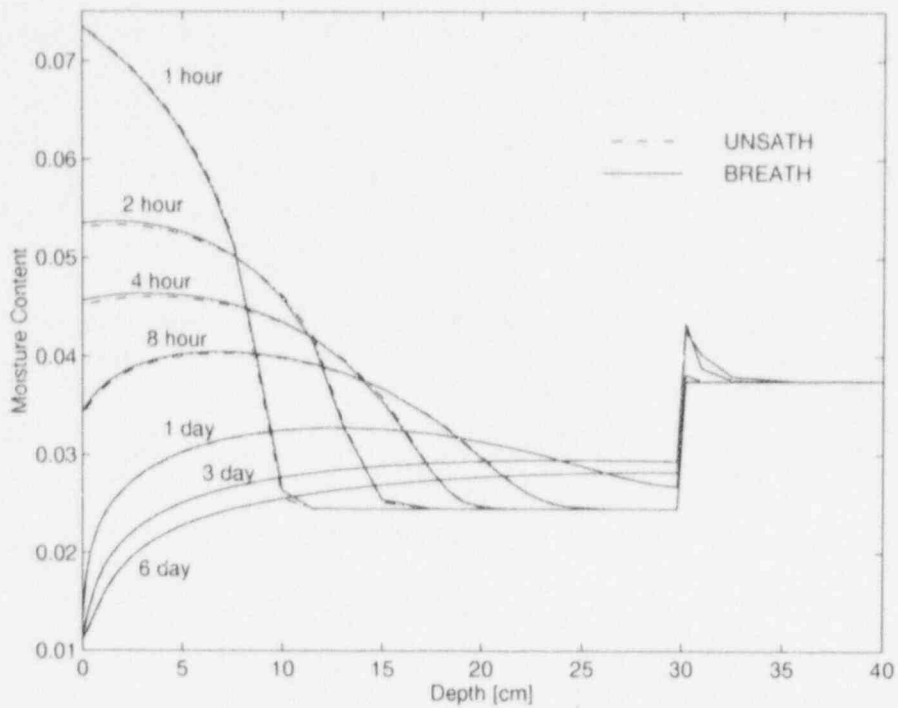


Figure 5-5. Response of moisture content to a 1-hr rainfall event in the coupled example

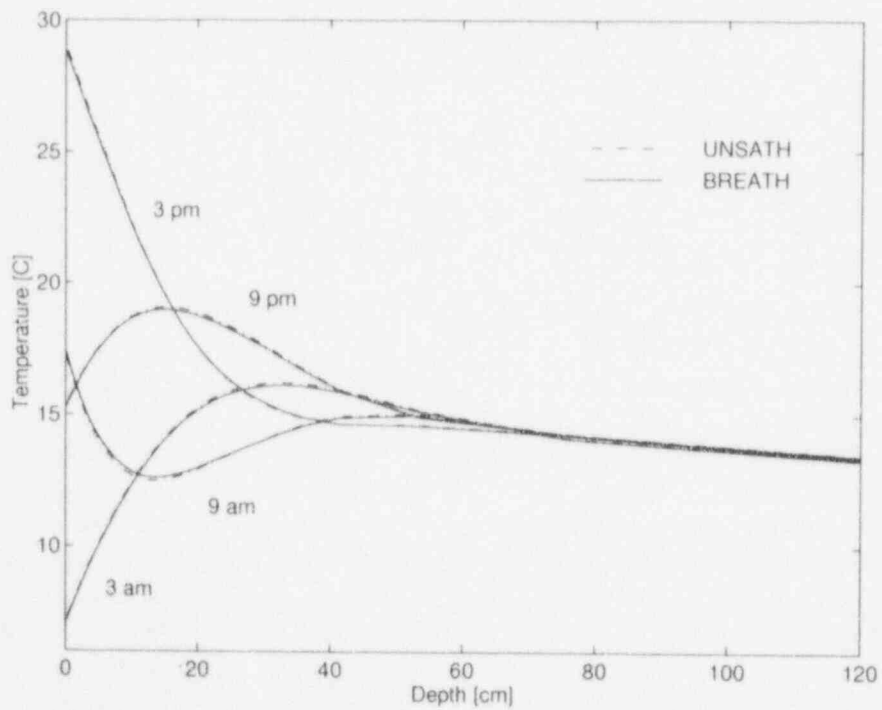


Figure 5-6. Temperature profile at selected times in the sixth day in the coupled example

Solutions to this non-linear problem are quite sensitive to the media properties and boundary condition representation, and creating exactly comparable input files for the two codes was found to be a significant challenge. Relatively minor changes in input yielded significantly different solutions. For example, changing the albedo by a few hundredths changed the surface temperature by more than 2 °K. 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 parameters. 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.

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 BREATH's predictions and both exact 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 has 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 YM 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. Because many of the layers at YM are sloping, this restriction may 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 YM, 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 boundary-layer evaporation model is based on the assumption that the terrain is uniform for a representative fetch upwind. This assumption is questionable in areas of YM, 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

- Allen, M.B., I. Herrera, and G.F. Pinder. 1988. *Numerical Modeling in Science and Engineering*. New York, NY: John Wiley & Sons.
- Bear, i. 1972. *Dynamics of Fluids in Porous Media*. New York, NY: Dover Publications, Inc.
- Beatley, J.C. 1974. Phenological events and their environmental triggers in Mojave Desert ecosystems. *Ecology* 55: 856-863.
- 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.
- Brutsaert, W. 1982. *Evaporation into the Atmosphere: Theory, History, and Applications*. Dordrecht, The Netherlands: Kluwer Academic Publishers.
- Businger, J.A., and A.M. Yaglom. 1971. Introduction to Obukhov's paper on "Turbulence in an atmosphere with a non-uniform temperature." *Boundary-Layer Meteorology* 2: 153-176.
- Campbell, G.S. 1985. *Soil Physics with BASIC*. Amsterdam, The Netherlands: 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*. 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, France.
- 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.
- Edlefsen, N.E., and A.B.C. Anderson. 1943. Thermodynamics of Soil Moisture. *Hilgardia* 15(2).
- 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.

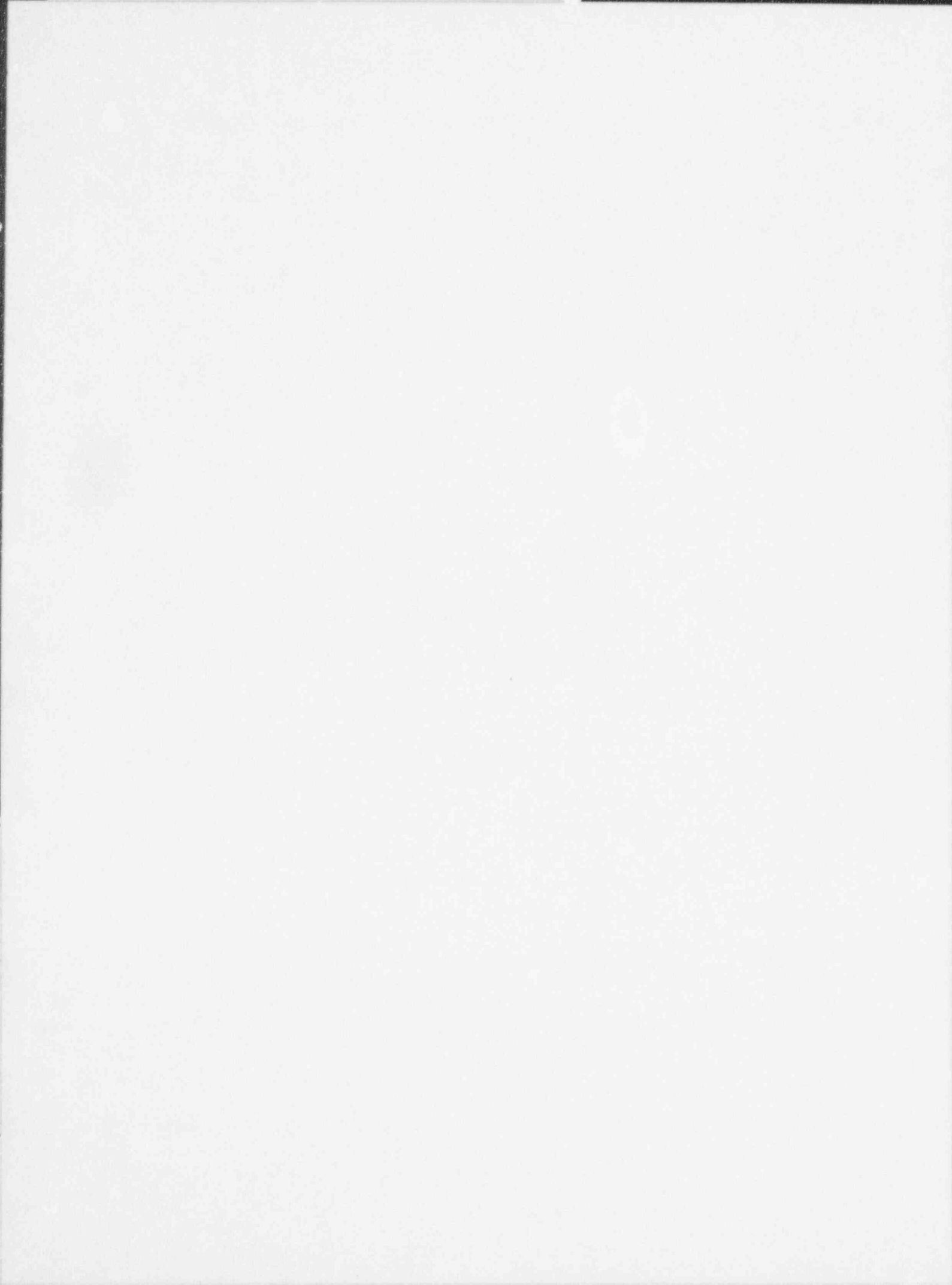
- Eringen, A.C., and J.D. Ingram. 1965. A continuum theory of chemically reacting media—I. *International Journal of Engineering Science* 3: 197–212.
- Fayer, M.J., and T.L. Jones. 1990. *UNSAT-H Version 2.0: Unsaturated Soil Water and Heat Flow Model*. PNL-6779, UC-702. Richland, WA: Pacific Northwest Laboratory.
- Fayer, M.J., and T.L. Jones. 1993. *UNSAT-H Version 2.1*. Unpublished code update. Richland, WA: Pacific Northwest Laboratory.
- Fayer, M.J., G.W. Gee, and T.L. Jones. 1986. *UNSAT-H Version 1.0: Unsaturated Flow Code Documentation and Application for the Hanford Site*. PNL-5899. Richland, WA: Pacific Northwest Laboratory.
- Gee, G.W. 1966. *Water Movement in Soils as Influenced by Temperature Gradients*. Ph.D. 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*. Ph.D. 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.
- Lane, L.J., E.M. Romney, and T.E. Hakonson. 1984. Water balance calculations and net production of perennial vegetation in the northern Mojave Desert. *Journal of Range Management* 37(1): 12–18.
- Lapidus, L., and G.F. Pinder. 1982. *Numerical Solution of Partial Differential Equations in Science and Engineering*. New York, NY: John Wiley & Sons.
- Leary, K.D. 1990. *Analysis of Techniques for Estimating Potential Recharge and Shallow Unsaturated Zone Water Balance Near Yucca Mountain, Nevada*. Ph.D. Dissertation. Reno, NV: University of Nevada.

- 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.
- Monin, A.S., and A.M. Obukhov. 1954. Basic laws of turbulent mixing in the ground layer of the atmosphere. *Th. Geofiz. Instit. Akad. Nauk.* 24 (151): 163-187. (German translation: 1954. *Sammelband zur Statistischen Theorie der Turbulenz*, ed. H. Goering. Akademie Verlag: Berlin).
- 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. 1995. *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.
- Obukhov, A.M. 1946. Turbulence in an atmosphere with non-uniform temperature. *Trudy Instit. Teoret. Geofiz: AN-S.S.S.R.* 1 (English translation: 1971. *Boundary Layer Meteorology* 2: 7-29).
- Paniconi, C., A.A. Aldama, and E.F. Wood. 1991. Numerical evaluation of iterative and noniterative methods for the solution of the nonlinear Richards equation. *Water Resources Research* 27(6): 1,147-1,163
- 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.
- Sasamori, T. 1970. A numerical study of atmospheric and soil boundary layers. *Journal of Atmospheric Science* 27(8): 1,122-1,137.
- Schlesinger, W.H., P.J. Fonteyn, and G.M. Marion. 1987. Soil moisture content and plant transpiration in the Chihuahuan Desert of New Mexico. *Journal of Arid Environments* 12: 119-126.
- Sophocleous, M.A. 1978. *Analysis of Heat and Water Transport in Unsaturated-Saturated Porous Media*. Ph.D. 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 ends 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. Arguments and modifier words in italics must be replaced by other input (character strings, names of BREATH variables, or numbers); nonitalicized arguments and modifier words are examples of expected input. 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 beginning above the heavy line for each command are optional, and as many of these words as desired can be strung together; modifier words below the heavy 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 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; this is demonstrated in the shorthand for setting the number of nodes on the first line (*nnd 11*). A complete list of variable names is presented in Appendix B.

Table A-4. Example input file segment

```
nnd 11
set          # keyword
  interp     # modifier
    xcoord 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
```

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.

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 the "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.

Meteorological 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.

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> richrun thmrun tmrun	Type of run Moisture only Energy only 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 ($\epsilon_a \sigma$)
lwrad*		Net longwave radiation ($\epsilon_a \sigma T_a^4$)
swrad		Net shortwave radiation (S_I)
windsp		Wind speed
set		End of met input

*No more than one of *esiga* and *lwrad* should be provided

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 runtype tmrun dummy bce set
calc metrun
```

opens a file called "metdat.tst" (consisting of two or more columns of numbers), determines that the second column is the energy boundary condition for the first side of the domain, and runs a coupled moisture/thermal simulation 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.

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 calc 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 met command

Table A-7. The echo command

Modifier Word	Argument(s)	Function
unit	<i>IOECH</i>	Unit number of output file
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

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-7 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

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
nperskip	<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
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
asatw		Flag output of time-averaged moisture content
adensv		Flag output of time-averaged vapor density
atemp		Flag output of time-averaged temperature
apers		Flag output of time-averaged moisture content perturbation
apert		Flag output of time-averaged temperature perturbation
amaxs		Flag output of maximum observed moisture content
amins		Flag output of minimum observed moisture content
amaxt		Flag output of maximum observed temperature
amint		Flag output of minimum observed temperature
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
amaxqw		Flag output of maximum observed water flux
amaxqv		Flag output of maximum observed vapor flux
amaxqe		Flag output of maximum observed energy flux
aminqw		Flag output of minimum observed water flux
aminqv		Flag output of minimum observed vapor flux
aminqe		Flag output of minimum observed energy flux
on		Turn on output of flagged variables
off		Turn off output of flagged variables

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 *idtrrel* 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.

The **echo** command is friendlier than the other output commands, with output that might be 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. All variables that can be output with **snap** or **trace** can also be output with **echo** and reset with **set**.

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

Modifier Word	Argument	Function
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
bcsrev		Flag instantaneous vapor boundary flux
bcsrw		Flag instantaneous liquid boundary flux
cumde		Flag cumulative energy mass balance
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

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

Modifier Word	Argument	Function
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

APPENDIX B
VARIABLE NAMES

B.1 VARIABLE NAMES

Table B-1 presents numeric variables that can be accessed using the **set** or **echo** commands, and Table B-2 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 B-1. Numeric variable names and descriptions

Variable Name	Size	Units	Default	Function
adensv	nnd	ML^{-3}	0	Time-averaged vapor density
albedd0,1	1	—	1	Dry albedo
albedw0,1	1	—	1	Wet albedo
aleft	nnd	—	—	Calculated left diagonal of system matrix
amaxqe	nel	JT^{-1}	-10^{32}	Flag output of maximum observed energy flux
amaxqv	nel	$ML^{-2}T^{-1}$	-10^{32}	Flag output of maximum observed vapor flux
amaxqw	nel	$ML^{-2}T^{-1}$	-10^{32}	Flag output of maximum observed water flux
amaxs	nnd	L^3L^{-3}	-10^{32}	Maximum observed moisture content
amaxt	nnd	K	-10^{32}	Maximum observed temperature
amid	nnd	—	—	Calculated middle diagonal of system matrix
aminqe	nel	JT^{-1}	10^{32}	Flag output of minimum observed energy flux
aminqv	nel	$ML^{-2}T^{-1}$	10^{32}	Flag output of minimum observed vapor flux
aminqw	nel	$ML^{-2}T^{-1}$	10^{32}	Flag output of minimum observed water flux
amins	nnd	L^3L^{-3}	10^{32}	Minimum observed moisture content
amint	nnd	K	10^{32}	Minimum observed temperature
apers	nnd	L^3L^{-3}	0	Time-averaged moisture content perturbation
apert	nnd	K	0	Time-averaged temperature perturbation
arigh	nnd	—	—	Calculated right diagonal of system matrix
asatw	nnd	L^3L^{-3}	0	Time-averaged moisture content
atemp	nnd	K	0	Time-averaged temperature
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	LT^{-1}	0	Boundary layer heat conductance

Table B-1. Numeric variable names and descriptions (cont'd)

Variable Name	Size	Units	Default	Function
blkv0,l	l	$L T^{-1}$	0	Boundary layer vapor density conductance
blzoff0,l	l	L	0	Boundary layer offset
blzh0,l	l	L	1	Boundary layer heat roughness height
blzm0,l	l	L	1	Boundary layer momentum roughness height
blzt0,l	l	L	10	Air temperature measurement height
blzu0,l	l	L	10	Wind speed measurement height
cmdecho	l	—	0	1/0 to turn on/turn off command-name echoing
cumqe	nel	$J T^{-1}$	0	Flag output of cumulative energy flux
cumqv	nel	$M L^{-2} T^{-1}$	0	Flag output of cumulative vapor flux
cumqw	nel	$M L^{-2} T^{-1}$	0	Flag output of cumulative water flux
coeflh	l	$J M^{-1}$	0	Coefficient of latent heat
conve	l	—	10^{32}	Convergence criterion for energy equation
convf	l	—	10^{32}	Convergence criterion for flow equation
coumax	l	—	1	Maximum courant number ($ztssel = courant$)
dcoefv	nnd	$L^2 T^{-1}$	0	Vapor diffusion coefficient (times tortuosity)
deltim	l	T	1	Time step size
delxnd	nnd	L	—	Length corresponding to a node
densg	nnd	$M L^{-3}$	0	Density of the gas phase
denss	nnd	$M L^{-3}$	0	Density of the solid matrix
densv	nnd	$M L^{-3}$	0	Vapor density in porous medium
densw	l	$M L^{-3}$	0	Water density
dhdtg	l	$J M^{-1} K^{-1}$	—	$(dH/dT)_p$ for the gas phase (c_{p_g})
dhdts	nnd	$J M^{-1} K^{-1}$	—	$(dH/dT)_p$ for the solid phase (c_{p_s})
dhdtv	l	$J M^{-1} K^{-1}$	—	$(dH/dT)_p$ for the vapor (c_{p_v})
dhdtw	l	$J M^{-1} K^{-1}$	—	$(dH/dT)_p$ for the liquid phase (c_{p_w})
dtmax	l	T	10^{32}	Maximum time step size
dtmin	l	T	0	Minimum time step size
dudtg	l	$J M^{-1} K^{-1}$	—	$(dU/dT)_v$ for the gas phase (c_{v_g})
dudts	nnd	$J M^{-1} K^{-1}$	—	$(dU/dT)_v$ for the solid phase (c_{v_s})
dudtv	l	$J M^{-1} K^{-1}$	—	$(dU/dT)_v$ for the vapor (c_{v_v})
dudtw	l	$J M^{-1} K^{-1}$	—	$(dU/dT)_v$ for the liquid phase (c_{v_w})
dzdx	l	$L L^{-1}$	0	Spatial gradient of elevation

Table B-1. Numeric variable names and descriptions (cont'd)

Variable Name	Size	Units	Default	Function
eintv0	1	JM^{-1}	—	Internal energy of vapor at reference temperature
eintw0	1	JM^{-1}	—	Internal energy of water at reference temperature
ekg	1	$JL^2M^{-1}T^{-1}K^{-1}$	0	Thermal conductivity of gas phase
eks	nnd	$JL^2M^{-1}T^{-1}K^{-1}$	0	Thermal conductivity of solid phase
ekw	1	$JL^2M^{-1}T^{-1}K^{-1}$	0	Thermal conductivity of water phase
eltrace	mnr	—	0	List of elements to trace (end with 0)
enthv0	1	JM^{-1}	—	Enthalpy of vapor at reference temperature
enthw0	1	JM^{-1}	—	Enthalpy of water at reference temperature
esiga0,1	1	$JT^{-1}L^{-2}K^{-4}$	0	Atmospheric emissivity times Stefan-Boltzmann constant
esigsd0,1	1	$JT^{-1}L^{-2}K^{-4}$	0	Dry soil emissivity times Stefan-Boltzmann constant
esigsw0,1	1	$JT^{-1}L^{-2}K^{-4}$	0	Wet soil emissivity times Stefan-Boltzmann constant
etot	nnd	J	—	Total internal plus latent energy
fluxe	nel	$JL^{-2}T^{-1}$	0	Flux of energy
fluxv	nel	$ML^{-2}T^{-1}$	0	Mass flux of vapor
fluxw	nel	$ML^{-2}T^{-1}$	0	Mass flux of water
frceve	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	LT^{-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
krnevap0,1	1	—	1	1/0 to allow evaporation during rain events
lwrad0,1	1	$JT^{-1}L^{-2}$	0	Incident longwave radiation
maxite	1	—	1	Maximum number of iterations in energy time step
maxitf	1	—	10	Maximum number of iterations in flow time step
ndtrace	mnr	—	0	List of nodes to trace (end with 0)
neitex	1	—	3	Nodes to expand for energy subdomain iterations

Table B-1. Numeric variable names and descriptions (cont'd)

Variable Name	Size	Units	Default	Function
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
ntmcyc	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
poros	nnd	$L^3 L^{-3}$	1	Porosity
pressw	nnd	$M L^{-1} T^{-2}$	0	Water pressure
prsmax	1	$M L^{-1} T^{-2}$	0	Maximum allowed pressure from saturation
prsmn	1	$M L^{-1} T^{-2}$	-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
timmax	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

Table B-1. Numeric variable names and descriptions (cont'd)

Variable Name	Size	Units	Default	Function
vonk	1	—	0.4	von Karman coefficient [0.4]
viscw	1	M T ⁻¹ L ⁻¹	nnd	Viscosity of water
viscw0	1	M T ⁻¹ L ⁻¹	1	Reference viscosity of water
windsp0,1	1	L T ⁻¹	0	Wind speed
wmolwt	1	M mole ⁻¹	18	Molecular weight of water [18 g/mole]
xcoor	nnd	L	0	Nodal coordinate

Table B-2. 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
	brut82	Boundary layer conductances from Brutsaert scheme
	camplin	Linearized Campbell boundary layer conductances
	brutlin	Linearized Brutsaert boundary layer conductances
zmose1	VANGEN	Type of mobility calculation scheme van Genuchten mobility
	brooks	Brooks-Corey mobility

B-2. Character variable names (cont'd)

Variable Name	Flag	Function
:l	VANGEN	Type of pressure/saturation relationship van Genuchten
	brooks	Brooks-Corey
el	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 (θ / θ_s)
	SMOOTH	Use the Paniconi et al. (1991) scheme
sel	CONSTANT	Type of time stepping scheme Time step <i>deltim</i> remains constant
	courant	Time step calculated from <i>coumax</i> (energy only)
	iterbased	Time step calculated from <i>nomitef</i>
dsel	EQUILIB	Type of vapor transport Equilibrium between liquid and vapor
	none	Vapor transport not considered
vwsel	CONSTANT	Type of wetting-phase viscosity calculation scheme Viscosity <i>viscw</i> remains constant (<i>viscw0</i>)
	crctemp	Viscosity <i>viscw</i> calculated using CRC scheme

APPENDIX C

EXAMPLE INPUT AND OUTPUT LISTINGS

C.1 INPUT FILE FOR UNSATID COMPARISON

```
* this breath input file is intended to generate solutions
* comparable to unsatid predictions [Ceiba, (1994)]

cmdecho 0
nnd 41
zssel moisture
zvsel nose
maxitf 50
convf 1e-04
dzdx -1
tmaxax 1000000

# don't echo commands during input
# number of nodes
# use storage coefficient multiplier from unsatid
# neglect vapor density to mimic unsatid
# maximum number of flow iterations
# convergence criterion for flow
# elevation decreases with length of domain
# control maximum time with nts, not tmaxax

set vector xccor # vector input node coordinates
0 2.5 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
52.5 65 67.5 70 72.5 75 77.5 80
82.5 85 87.5 90 92.5 95 97.5 100

set vector press # 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
dcsolv 0
densw 1
visrw 1
grav 1
satmax 0.368
satmin 0.102
pcp0 29.8507
vgm 0.5
perm 0.0092
spstor 0.0001
poros 0.368
zbcw0 flux
zbcw0 -0
zbcw1 press
zbcw1 100

# document output
echo string ' Example: comparison with UNSATID'
echo nnd
echo zssel
echo zmosel
echo zpsel
echo zvsel
echo maxitf
echo convf
echo dzdx
echo satmin
echo satmax

# global input temperature
# global input vapor diffusion coefficient
# water density
# water viscosity
# gravitational constant
# global input maximum moisture content
# global input minimum moisture content
# global input reference capillary pressure
# global input van genuchten m
# global input intrinsic permeability
# global input specific storage coefficient
# global input porosity
# top boundary condition - set to flux
# top boundary condition - flux value
# bottom boundary condition - set to pressure
# bottom boundary condition - pressure value

cmdecho 0
nnd 41
zssel moisture
zvsel nose
maxitf 50
convf 1e-04
dzdx -1
tmaxax 1000000

# don't echo commands during input
# number of nodes
# use storage coefficient multiplier from unsatid
# neglect vapor density to mimic unsatid
# maximum number of flow iterations
# convergence criterion for flow
# elevation decreases with length of domain
# control maximum time with nts, not tmaxax

set vector xccor # vector input node coordinates
0 2.5 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
52.5 65 67.5 70 72.5 75 77.5 80
82.5 85 87.5 90 92.5 95 97.5 100

set vector press # 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
dcsolv 0
densw 1
visrw 1
grav 1
satmax 0.368
satmin 0.102
pcp0 29.8507
vgm 0.5
perm 0.0092
spstor 0.0001
poros 0.368
zbcw0 flux
zbcw0 -0
zbcw1 press
zbcw1 100

# document output
echo string ' Example: comparison with UNSATID'
echo nnd
echo zssel
echo zmosel
echo zpsel
echo zvsel
echo maxitf
echo convf
echo dzdx
echo satmin
echo satmax

* this breath input file is intended to generate solutions
* comparable to unsatid predictions [Ceiba, (1994)]

cmdecho 0
nnd 41
zssel moisture
zvsel nose
maxitf 50
convf 1e-04
dzdx -1
tmaxax 1000000

# don't echo commands during input
# number of nodes
# use storage coefficient multiplier from unsatid
# neglect vapor density to mimic unsatid
# maximum number of flow iterations
# convergence criterion for flow
# elevation decreases with length of domain
# control maximum time with nts, not tmaxax

set vector xccor # vector input node coordinates
0 2.5 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
52.5 65 67.5 70 72.5 75 77.5 80
82.5 85 87.5 90 92.5 95 97.5 100

set vector press # 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
dcsolv 0
densw 1
visrw 1
grav 1
satmax 0.368
satmin 0.102
pcp0 29.8507
vgm 0.5
perm 0.0092
spstor 0.0001
poros 0.368
zbcw0 flux
zbcw0 -0
zbcw1 press
zbcw1 100

# document output
echo string ' Example: comparison with UNSATID'
echo nnd
echo zssel
echo zmosel
echo zpsel
echo zvsel
echo maxitf
echo convf
echo dzdx
echo satmin
echo satmax

# prepare graphics output
file unit 20 open fl_br_40.grf
echo unit 20 name 0 index 0 format (lpe12.3) pressw
echo unit 20 same 0 index 0 format (lpe12.3) satw

# initialize setup and output initial conditions
init
calc satw
echo unit 20 name 0 index 0 format (lpe12.3) pressw
echo unit 20 same 0 index 0 format (lpe12.3) satw

# run output period
nts 72
deltim 0.3
calc richrun
echo tottim
echo pressw
echo satw
echo unit 20 name 0 index 0 format (lpe12.3) pressw
echo unit 20 same 0 index 0 format (lpe12.3) satw

# run to next output
deltim 0.3
nts 216
calc richrun
echo tottim
echo pressw
echo satw
echo unit 20 name 0 index 0 format (lpe12.3) pressw
echo unit 20 same 0 index 0 format (lpe12.3) satw

# run to next output
deltim 0.3
nts 864
calc richrun
echo tottim
echo pressw
echo satw
echo unit 20 name 0 index 0 format (lpe12.3) pressw
echo unit 20 same 0 index 0 format (lpe12.3) satw

# run to final output
deltim 0.3
```

```

nta 3456
calc richrun
echo tottim          # output total time
echo pressw
echo satw
echo unit 20 name 0 index 0 format {lpel2.3} pressw
echo unit 20 name 0 index 0 format {lpel2.3} satw
file unit 20 close

```

C.2 OUTPUT FILE FOR UNSATID COMPARISON

Example: comparison with UNSATID

```

nnd      = 41
zsssel   = moisture
zmosel   = vangen
zpsel    = vangen
zvdssel  = none
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] = 9.2000000000000D-03
poros [ 1: 41] = 0.36800000000000
spstor [ 1: 41] = 1.000000000000D-04
zbcw0    = flux
zbcw1    = 0
zbcw2    = press
bcw1     = 100.000000000000
tottim   = 21.600000000000
pressw :
  1 -9.00259E+00   2 -6.98940E+00   3 -5.58368E+00   4 -4.55387E+00
  5 -3.77397E+00   6 -3.16897E+00   7 -2.69143E+00   8 -2.31009E+00
  9 -2.00384E+00  10 -1.75836E+00  11 -1.56414E+00  12 -1.41534E+00
 13 -1.30918E+00  14 -1.24565E+00  15 -1.22764E+00  16 -1.26128E+00
 17 -1.35678E+00  18 -1.52989E+00  19 -1.80422E+00  20 -2.21541E+00
 21 -2.81825E+00  22 -3.70002E+00  23 -5.00666E+00  24 -6.99939E+00
 25 -1.01890E+01  26 -1.56437E+01  27 -2.41068E+01  28 -3.22568E+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 -6.97883E+01  36 -7.46490E+01
 37 -5.55053E+01  38  6.00301E+01  39  7.34683E+01  40  8.67655E+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 2.00765E-01
 37 2.27990E-01  38 3.68000E-01  39 3.68000E-01  40 3.68000E-01
 41 3.68000E-01
tottim   = 86.400000000000
pressw :
  1 -1.74471E+01   2 -1.52844E+01   3 -1.36025E+01   4 -1.22485E+01
  5 -1.11316E+01   6 -1.01937E+01   7 -9.39569E+00   8 -8.71054E+00
  9 -8.11881E+00  10 -7.60650E+00  11 -7.16355E+00  12 -6.78294E+00
 13 -6.46011E+00  14 -6.19268E+00  15 -5.98031E+00  16 -5.82473E+00
 17 -5.72993E+00  18 -5.70257E+00  19 -5.75257E+00  20 -5.89416E+00
 21 -6.14737E+00  22 -6.54055E+00  23 -7.11437E+00  24 -7.92886E+00
 25 -9.07609E+00  26 -1.07048E+01  27 -1.30710E+01  28 -1.66350E+01
 29 -2.19097E+01  30 -2.24231E+01  31 -3.60229E+00  32  7.39432E+00
 33  1.74543E+01  34  2.75971E+01  35  3.78114E+01  36  4.80860E+01

```

C.3 INPUT FILE FOR SINUSOIDAL EXAMPLE

```

37 5.84103E-01 38 6.87737E-01 39 7.91664E-01 40 8.95784E-01 4 3.48089E-01
41 1.00000E+02
szw
1 3.31851E-01 2 3.38768E-01 3 3.44054E-01 4 3.48089E-01
5 3.51244E-01 6 3.57272E-01 7 3.57282E-01 8 3.57352E-01
9 3.58676E-01 10 3.59763E-01 11 3.60658E-01 12 3.61388E-01
13 3.61982E-01 14 3.6284E-01 15 3.63817E-01 16 3.65078E-01
17 3.67232E-01 18 3.68752E-01 19 3.69194E-01 20 3.6951E-01
21 3.6933E-01 22 3.61836E-01 23 3.60753E-01 24 3.59085E-01
25 3.5649E-01 26 3.52387E-01 27 3.45664E-01 28 3.3455E-01
29 3.18438E-01 30 3.14680E-01 31 3.68084E-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
totlim = 345.600000000001
pressw
1 -1.81848E-01 2 -1.54356E-01 3 -1.24029E-01 4 -9.27132E-00
5 -6.15650E+00 6 -3.10810E+00 7 -1.41259E-01 8 3.78013E-00
9 5.70198E+00 10 8.62802E+00 11 1.15523E-01 12 1.44806E-01
13 1.74110E-01 14 2.03433E-01 15 2.3275E-01 16 2.62136E-01
17 2.91814E-01 18 3.20908E-01 19 3.50319E-01 20 3.79745E-01
21 4.09165E-01 22 4.3841E-01 23 4.68109E-01 24 4.97590E-01
25 5.27082E-01 26 5.56586E-01 27 5.86100E-01 28 6.15623E-01
29 6.45155E-01 30 6.7424E-01 31 7.02446E-01 32 7.30802E-01
33 7.69364E-01 34 7.92321E-01 35 8.24503E-01 36 8.52808E-01
37 8.8160E-01 38 9.1142E-01 39 9.40827E-01 40 9.70413E-01
41 1.00000E+02
szw
1 3.29167E-01 2 3.38267E-01 3 3.47640E-01 4 3.56029E-01
5 3.6217E-01 6 3.6570E-01 7 3.67997E-01 8 3.6900E-01
9 3.69000E-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
totlim = 1382.39999999999
pressw
1 9.4970E-08 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
szw ( 1: 41) * 0.368000000000000
# Compare to Campbell analytical equation
# energy simulation only
cmdecho 0 # don't echo commands on entry
nnd 101 # use 101 nodes
set inerp xcoor 0 100 # evenly distribute nodes
# linear problem, so force one iteration and ignore convergence
maxite 1 # maximum number of energy iterations
nomite 1 # nominal number of energy iterations
conve 1e-4 # convergence criteria, artificially large
stssel constant # constant time step
# artificially consider water only
grav 980 # gravity (cm/sec^2)
densw 1 # gm / cm^3
dudtw 1 # 1/3m K (dudtw = Ch/densw, Ch = 1)
ekw 2.161E-3 # 3/s cm K
# Initial and boundary conditions
temp 288 # initial conditions
tempr 298 # reference temp
tempex 0 # temp of external sources
poros 5 # porosity
satw 5 # initial moisture content
zbcwl gradtemp # zero-gradient for left side temperature
bcel 0
zbcw0 temp # sinusoidal BC for left side temperature
file 31 open sin_t.met # use sinusoid defined in external file
met unit 31 side 0 rntype thrmun timax bce set
nts 10000 # use timax in external file to control time period
deltim 1800 # specified time step
file 20 open s1_bri00.out
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 zbcw0
echo unit 20 zbcwl
echo unit 20 boel
# initialize setup

```

C.4 MET FILE FOR SINUSOIDAL EXAMPLE

```

init
# Days 1 through 9, cycle met file
met unit 31 nmetcyc 8 side 0 rntype thrrum timmax bce set
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 (ipe11.4) nperskip 0
#
#snap nodeunit 24 elemunit 25 temp format (ipe11.4) nperskip 0 on
tottim 0 # reset time to zero
file 31 rewind
met nmetcyc 0 timmax bce set # reset cycles - note BCs must be input
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

end

```

```

3600 268
3600 268.681
3600 270.679
3600 273.858
3600 278
3600 282.824
3600 288
3600 293.176
3600 298
3600 302.142
3600 305.321
3600 307.319
3600 308
3600 307.319
3600 305.321
3600 302.142
3600 298
3600 293.176
3600 288
3600 282.824
3600 278
3600 273.858
3600 270.679
3600 268.681
.1 268

```

C.5 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     = 6.1810000000000E-03
poros   [ 1: 101] = 0.500000000000000
satw    [ 1: 101] = 0.500000000000000
rbce0   = temp
rbce1   = gradtemp
bce1    = 0.
temp

```

```

1 2.68000E+02 2 2.69403E+02 3 2.70775E+02 4 2.72108E+02
5 2.73396E+02 6 2.74635E+02 7 2.75820E+02 8 2.76950E+02
9 2.78022E+02 10 2.79035E+02 11 2.79988E+02 12 2.80881E+02
13 2.81718E+02 14 2.82492E+02 15 2.83211E+02 16 2.83875E+02
17 2.84486E+02 18 2.85045E+02 19 2.85594E+02 20 2.86017E+02
21 2.86435E+02 22 2.86811E+02 23 2.87147E+02 24 2.87445E+02
25 2.87710E+02 26 2.87942E+02 27 2.88144E+02 28 2.88318E+02
29 2.88467E+02 30 2.88592E+02 31 2.88696E+02 32 2.88780E+02
33 2.88847E+02 34 2.88899E+02 35 2.88935E+02 36 2.88961E+02
37 2.88975E+02 38 2.88979E+02 39 2.88974E+02 40 2.88963E+02
41 2.88945E+02 42 2.88921E+02 43 2.88894E+02 44 2.88862E+02
45 2.88828E+02 46 2.88792E+02 47 2.88754E+02 48 2.88715E+02
49 2.88676E+02 50 2.88636E+02 51 2.88597E+02 52 2.88558E+02
53 2.88519E+02 54 2.88482E+02 55 2.88446E+02 56 2.88411E+02
57 2.88377E+02 58 2.88345E+02 59 2.88315E+02 60 2.88286E+02
61 2.88258E+02 62 2.88233E+02 63 2.88209E+02 64 2.88187E+02
65 2.88166E+02 66 2.88147E+02 67 2.88129E+02 68 2.88113E+02
69 2.88098E+02 70 2.88084E+02 71 2.88072E+02 72 2.88061E+02
73 2.88051E+02 74 2.88042E+02 75 2.88034E+02 76 2.88027E+02
77 2.88021E+02 78 2.88015E+02 79 2.88011E+02 80 2.88007E+02
81 2.88003E+02 82 2.88000E+02 83 2.87998E+02 84 2.87996E+02
85 2.87994E+02 86 2.87993E+02 87 2.87992E+02 88 2.87991E+02
89 2.87990E+02 90 2.87990E+02 91 2.87990E+02 92 2.87990E+02
93 2.87990E+02 94 2.87989E+02 95 2.87990E+02 96 2.87990E+02
97 2.87990E+02 98 2.87990E+02 99 2.87990E+02 100 2.87990E+02

```

```

fluxe
1 -5.74008E-03 2 -5.61059E-03 3 -5.45188E-03 4 -5.26910E-03
5 -5.06696E-03 6 -4.84967E-03 7 -4.62103E-03 8 -4.38444E-03
9 -4.14288E-03 10 -3.89902E-03 11 -3.65517E-03 12 -3.41334E-03
13 -3.17527E-03 14 -2.94242E-03 15 -2.71603E-03 16 -2.49712E-03
17 -2.28649E-03 18 -2.08481E-03 19 -1.89255E-03 20 -1.71006E-03
21 -1.53756E-03 22 -1.37514E-03 23 -1.22283E-03 24 -1.08054E-03
25 -9.48125E-04 26 -8.25375E-04 27 -7.12028E-04 28 -6.07777E-04
29 -5.12283E-04 30 -4.25178E-04 31 -3.46073E-04 32 -2.74565E-04
33 -2.10243E-04 34 -1.52687E-04 35 -1.01480E-04 36 -5.62059E-05
37 -1.64556E-05 38 1.81723E-05 39 4.80683E-05 40 7.36105E-05
41 9.51629E-05 42 1.13075E-04 43 1.27679E-04 44 1.39291E-04
45 1.48212E-04 46 1.54722E-04 47 1.59087E-04 48 1.61554E-04
49 1.62354E-04 50 1.61701E-04 51 1.59792E-04 52 1.56810E-04
53 1.52920E-04 54 1.48277E-04 55 1.43017E-04 56 1.37267E-04
57 1.31140E-04 58 1.24737E-04 59 1.18147E-04 60 1.11451E-04

```

```

61 1.04719E-04 62 9.80131E-05 63 9.13855E-05 64 8.48821E-05
65 7.85414E-05 66 7.23957E-05 67 6.64716E-05 68 6.07901E-05
69 5.53678E-05 70 5.02159E-05 71 4.53461E-05 72 4.07603E-05
73 3.64617E-05 74 3.24500E-05 75 2.87226E-05 76 2.52747E-05
77 2.21001E-05 78 1.91913E-05 79 1.65393E-05 80 1.41344E-05
81 1.19650E-05 82 1.00231E-05 83 8.29398E-06 84 6.76678E-06
85 5.42934E-06 86 4.26942E-06 87 3.27469E-06 88 2.43288E-06
89 1.73177E-06 90 1.15926E-06 91 7.03392E-07 92 3.52362E-07
93 9.45376E-08 94 -8.15417E-08 95 -1.87168E-07 96 -2.33477E-07
97 -2.31452E-07 98 -1.91595E-07 99 -1.25848E-07 100 -4.37230E-08

```

C5

C.6 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
file 20 open cdfront.out * set default result echo unit to output file
lores 20
echo string ' Advancing front problem for advection-diffusion equation'
echo nnd
echo densw
echo fluxw
echo ekw
echo poros
echo satw
echo zbce0
echo bce0
echo abce1
echo bce1

calc fluxe
calc etot
calc thrmstat

echo format '(a/5(14,1pe11.4))' xcoord
echo format '(a/5(14,1pe11.4))' temp
echo format '(a/5(14,1pe11.4))' etot
echo format '(a/5(14,1pe11.4))' fluxe

end

nnd 51 * number of nodes
set interp xcoord 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
zssel 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 - TD
ekw 0.10 * diffusion coefficient
temp 295 * initial condition
tempr 285 * minimize roundoff by setting Tref = IC
zbce0 temp
bce0 286 * first boundary condition
abce1 gradtemp * unit increase over background
bce1 0 * second boundary condition
* zero gradient

* output periodic values of selected variables
file 22 open cdfront.ctr
file 23 open cdfront.ctr
file 24 open cdfront.esn
file 25 open cdfront.esn
rset vector ndtrace
#1 10 21 54 78 114 161 0/
rset vector eltrace
#1 10 21 54 78 114 160 0/
trace nodeunit 22 element 23 tottim etot fluxe format (1pe11.4) al1step on
snap nodeunit 24 element 25 etot fluxe format (7(1pe10.3)) npe1skip 0 on

* time stepping
nts 10000 * control time period with timmax
timmax 25 * length of control period

calc init
de1tim 0.75 * initialize
calc thrmrun * set the time step
* run a simulation period

de1tim 0.75 * set the time step
calc thrmrun * run a simulation period

de1tim 0.75 * set the time step
calc thrmrun * run a simulation period

* output results

```

C.7 OUTPUT FILE FOR ADVANCING-FRONT EXAMPLE

Advancing front problem for advection-diffusion equation

```

end
densw = 81
flux ( 1: 80) = 1.00000000000000
ekw ( 1: 81) = 1.00000000000000
poros ( 1: 81) = 1.00000000000000
satw ( 1: 81) = 1.00000000000000
zbc0 = temp
bce0 = 286.000000000000
zbc1 = gradtemp
bce1 = 0.
minimum grid peclct = 12.500000000000
maximum grid peclct = 12.500000000000
minimum grid cournt = 9.9999999999997D-02
maximum grid cournt = 9.9999999999997D-02
xcoord
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 2.8600E+02 50 2.8600E+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 2.8552E+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

```


C.10 INPUT FILE FOR COUPLED EXAMPLE

* example file comparing breath with sweat.
* three-layer problem tracking input rain pulse
* consistent units are cm, sec, gm, Joule, Kelvin
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
0.0001 0.0001 0.0001 0.0001 0.0001 0.0001 0.0001 0.0001 0.0001 0.0001
0.0001

set vector pcap0 # scaling capillary pressure
980000 980000 980000 980000 980000 980000 980000 980000 980000 980000
980000 980000 980000 980000 980000 980000 980000 980000 980000 980000
980000 980000 980000 980000 980000 980000 980000 980000 980000 980000
980000 980000 980000 980000 980000 980000 980000 980000 980000 980000
980000

113953 113953 113953 113953 113953 113953 113953 113953 113953 113953
113953 113953 113953 113953 113953 113953 113953 113953 113953 113953
113953 113953 113953 113953 113953 113953 113953 113953 113953 113953
113953 113953 113953 113953 113953 113953 113953 113953 113953 113953
113953 113953 113953 113953 113953 113953 113953 113953 113953 113953

vgm 0.230789 # van genuchten m

set vector perm # intrinsic permeability
1.14694e-06 1.14694e-06 1.14694e-06 1.14694e-06 1.14694e-06 1.14694e-06
1.14694e-06 1.14694e-06 1.14694e-06 1.14694e-06 1.14694e-06 1.14694e-06
1.14694e-06 1.14694e-06 1.14694e-06 1.14694e-06 1.14694e-06 1.14694e-06
1.14694e-06 1.14694e-06 1.14694e-06 1.14694e-06 1.14694e-06 1.14694e-06
1.14694e-06 1.14694e-06 1.14694e-06 1.14694e-06 1.14694e-06 1.14694e-06
1.14694e-06 1.14694e-06 1.14694e-06 1.14694e-06 1.14694e-06 1.14694e-06
1.14694e-06 1.14694e-06 1.14694e-06 1.14694e-06 1.14694e-06 1.14694e-06
1.14694e-06 1.14694e-06 1.14694e-06 1.14694e-06 1.14694e-06 1.14694e-06
1.14694e-06 1.14694e-06 1.14694e-06 1.14694e-06 1.14694e-06 1.14694e-06
1.14694e-06

6.65224e-08 6.65224e-08 6.65224e-08 6.65224e-08 6.65224e-08 6.65224e-08
6.65224e-08 6.65224e-08 6.65224e-08 6.65224e-08 6.65224e-08 6.65224e-08
1.8351e-06 1.8351e-06 1.8351e-06 1.8351e-06 1.8351e-06 1.8351e-06
1.8351e-06 1.8351e-06 1.8351e-06 1.8351e-06 1.8351e-06 1.8351e-06
1.8351e-06 1.8351e-06 1.8351e-06 1.8351e-06 1.8351e-06 1.8351e-06
1.8351e-06 1.8351e-06 1.8351e-06 1.8351e-06 1.8351e-06 1.8351e-06
1.8351e-06 1.8351e-06 1.8351e-06 1.8351e-06 1.8351e-06 1.8351e-06
1.8351e-06 1.8351e-06 1.8351e-06 1.8351e-06 1.8351e-06 1.8351e-06
1.8351e-06 1.8351e-06 1.8351e-06 1.8351e-06 1.8351e-06 1.8351e-06
1.8351e-06 1.8351e-06 1.8351e-06 1.8351e-06 1.8351e-06 1.8351e-06
1.8351e-06 1.8351e-06 1.8351e-06 1.8351e-06 1.8351e-06 1.8351e-06

setstor 0 # specific storage coefficient

set vector pores # porosity
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

nel 96 # number of elements
nts 10000 # number of time steps per period
rtssel 1 # use iteration-based time step
maxitt 26 # maximum flow iterations
nomalf 9 # nominal flow iterations
cnvof 0.0001 # flow convergence criterion
maxite 10 # maximum energy iterations
nomite 4 # nominal energy iterations
cnvme 0.001 # energy convergence criterion
atlim 360 # antislip time step
tmax 3600 # length of a time period
dtmin 0.01 # minimum time step
dtmax 720 # maximum time step
nmaxc 7 # number of cycles per time step
krcvrad 0 # don't evaporate during rain
drdx 1 # vertical column

set vector boor # nodal coordinates
0 0 1 0.213622 0.342721 0.489405 0.556071 0.845439 1.0505 1.30508
1.58285 1.89845 2.25707 2.66452 3.12747 3.65349 4.24126 4.93024
5.70182 6.52651 7.57462 8.70642 9.99239 11.4535 13.1137 15
17.8998 18.3777 19.1218 19.9334 20.7137 21.4639 22.1953 22.8788
23.5455 24.1866 24.857 25.5656 26.3131 27.0398 27.5462
28.0331 28.5012 28.9357 29.384 29.8 30.2 30.9838 32.5197 35.5293
41.4287 52.9825 75.8274 120.164 178.017 198.573 204.471
207.48 209.036 209.8 210.2 210.66 211.188 211.796 212.494
213.237 214.219 215.28 216.499 217.9 219.51 221.361 223.485
225.935 228.746 231.977 235.652 239.961 244.869 250.51 256.993
264.446 273.013 282.86 294.379 307.186 322.143 339.333 350.091
381.802 407.907 437.916 472.404 512.049 557.62 610

press 1e+08 # initial pressure
temp 286 # reference temperature
tempal 286 # bottom-of-column air temperature
grav 980 # gravity
dcoef 0.24 # vapor diffusion coefficient
tort 0.66 # tortuosity
viscw 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
0.28 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

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
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
0.0001 0.0001

```
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
0.42 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
```

```
coeflh 2501      * coefficient of latent heat
densg  0.00125   * density of air
denss  2.6       * density of solid
densw  1         * density of water
dudtg  1         * du/dt|V for gas
dudts  0.84      * du/dt|V for solid
dudtw  4.2       * du/dt|V for water
ekg    0         * thermal conductivity for gas
eks    0.00807692 * thermal conductivity for solid
ekw    0.0058    * thermal conductivity for water
eintw0 0         * don't consider reference internal energy
eintv0 0         * don't consider reference internal energy
zbcw0  flux      * flux bc for water at upper boundary
bcw0   0         * value of lower water bc
zbcwl  gradsaw  * saturation gradient bc for water at lower boundary
bcwl   0         * value of upper water bc
pondh0 1000      * maximum pond height
```

```
* use campbell bi equation for upper bc
```

```
zbcv0 densva     * bc type
zbik0  campbell  * bi type
blzt0  300
blzh0  0.03
blzu0  300
blzm0  0.03
blzoff0 0
```

```
zbcv1 densva     * densva bc for lower vapor bc
bcv1   1.1271e-05 * lower external vapor density
zbik1  constant  * constant bi for lower vapor diffusion
blkv1  0.0001584 * diffusion coef. in lower vapor equation
bikh1  0         * diffusion coef. in lower heat equation
zbcce0 gradtemp  * gradtemp bc for upper boundary
zbccl  gradtemp  * gradtemp bc for lower boundary
bccl   0         * zero lower temperature gradient
esigs0 5.103e-12 * dry stefan-boltzmann const. times soil emissiv.
esigs0 5.20292e-12 * wet stefan-boltzmann const. times soil emissiv.
albed0 0.25      * dry albedo
albed0 0.25      * wet albedo
```

```
* set up input/output files
```

```
#file unit 21 open coupled.ntr
#file unit 22 open coupled.etr
#file unit 23 open coupled.nsn
#file unit 24 open coupled.esn
#file unit 25 open coupled.mtr
file unit 26 open coupled.met
#trace nodeunit 21 elemunit 22 pressw satw temp cumqw cumqv cumqe
*      format (lpel2.5) nperskip 0 on
```

```
#set vector ndtrace
#l 22 29 52 54 57 55 57 0/
#set vector eitrace
#l 21 29 51 53 57 94 96 0/
#snap nodeunit 23 elemunit 24 pressw satw temp cumqw cumqv cumqe
*      format (lpel2.5) nperskip 0 on
#mass unit 25 moiscum enercom format (lpel2.5) nperskip 5 on
```

```
* specify input columns
```

```
met unit 26 runtime tarun 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 ztsel
echo maxiff
echo ncmiff
echo convf
echo maxite
echo nomite
echo conve
echo dtmin
echo dtmax
echo ntmcyt
echo krnevap0
echo string '
echo string ' Domain and parameter setup... '
echo string '
echo drdx
echo xcoor
echo tempr
echo tempal
echo grav
echo dcoefv
echo tort
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 bisoff0

echo zbcv1
echo bcv1
echo zblk1
echo blkv1
echo blkhl
echo zbce0
echo zbccl
echo bccl
echo esigsd0
echo esigsw0
echo albedd0
echo albedw0

echo pressw
echo satw
echo densv
echo temp

file unit 27 close
end

```

C-11

C.11 MET FILE FOR COUPLED EXAMPLE

```

3600 5.525e-05 277.211 3.41644e-06 4.61063e-12 0 406.806
0 9.525e-05 275.736 3.41644e-06 4.61063e-12 0 406.806
0 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 00437845 406.806
3600 0 279.132 3.41644e-06 4.61063e-12 0 0183535 406.806
3600 0 281.37 3.41644e-06 4.61063e-12 0 0313762 406.806
3600 0 283.771 3.41644e-06 4.61063e-12 0 042559 406.806
3600 0 286.172 3.41644e-06 4.61063e-12 0 0511399 406.806
3600 0 288.41 3.41644e-06 4.61063e-12 0 0565341 406.806
3600 0 290.371 3.41644e-06 4.61063e-12 0 0583739 406.806
3600 0 291.806 3.41644e-06 4.61063e-12 0 0565341 406.806
3600 0 292.733 3.41644e-06 4.61063e-12 0 0511399 406.806
3600 0 293.049 3.41644e-06 4.61063e-12 0 042559 406.806
3600 0 292.733 3.41644e-06 4.61063e-12 0 0313762 406.806
3600 0 291.806 3.41644e-06 4.61063e-12 0 0183535 406.806
3600 0 290.331 3.41644e-06 4.61063e-12 0 00437845 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 00464793 406.806
3600 0 279.132 3.41644e-06 4.61063e-12 0 0186127 406.806
3600 0 281.37 3.41644e-06 4.61063e-12 0 0316258 406.806
3600 0 283.771 3.41644e-06 4.61063e-12 0 0428004 406.806
3600 0 286.172 3.41644e-06 4.61063e-12 0 051375 406.806
3600 0 288.41 3.41644e-06 4.61063e-12 0 0567652 406.806
3600 0 290.331 3.41644e-06 4.61063e-12 0 0586037 406.806
3600 0 291.806 3.41644e-06 4.61063e-12 0 0567652 406.806
3600 0 292.733 3.41644e-06 4.61063e-12 0 051375 406.806
3600 0 293.049 3.41644e-06 4.61063e-12 0 0428004 406.806
3600 0 292.733 3.41644e-06 4.61063e-12 0 0316258 406.806
3600 0 291.806 3.41644e-06 4.61063e-12 0 0186127 406.806
3600 0 290.331 3.41644e-06 4.61063e-12 0 00464793 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

```


C.12 OUTPUT FILE FOR COUPLED EXAMPLE

Example comparing BREATH with UNEATH

Simulation control...

```

mod - 97
ztsseel - iterbase
maxitf - 16
nomitf - 9
convf - 1.000000000000000D-04
maxite - 10
nomite - 4
conve - 1.000000000000000D-03
dtmin - 1.000000000000000D-02
dtmax - 720.0000000000000
stmsyc - 3
krnevp0 - 0

```

Domain and parameter setup...

```

dzdx - -1.000000000000000
xc00r -

```

```

1 0.0000E+00 2 1.0000E-01 3 2.1562E-01 4 3.4272E-01
5 4.8940E-01 6 6.5607E-01 7 8.4519E-01 8 1.0606E-00
9 1.3050E+00 10 1.5828E+00 11 1.8984E+00 12 2.2570E+00
13 2.6645E+00 14 3.1274E+00 15 3.6534E+00 16 4.2511E+00
17 4.9302E+00 18 5.7016E+00 19 6.5785E+00 20 7.5762E+00
21 8.7042E+00 22 9.9239E+00 23 1.1453E+01 24 1.3113E+01
25 1.5000E+01 26 1.7398E+01 27 1.8277E+01 28 1.9121E+01
29 1.9933E+01 30 2.0713E+01 31 2.1463E+01 32 2.2185E+01
33 2.2878E+01 34 2.3585E+01 35 2.4186E+01 36 2.4693E+01
37 2.5356E+01 38 2.5965E+01 39 2.6513E+01 40 2.7039E+01
41 2.7582E+01 42 2.8031E+01 43 2.8512E+01 44 2.8913E+01
45 2.9340E+01 46 2.9800E+01 47 2.9800E+01 48 2.9838E+01
49 3.0519E+01 50 3.0523E+01 51 3.0527E+01 52 3.0528E+01
53 3.0527E+01 54 3.0523E+01 55 3.0519E+01 56 3.0527E+01
57 3.0527E+01 58 3.0519E+01 59 3.0527E+01 60 3.0527E+01
61 3.0527E+01 62 3.0519E+01 63 3.0527E+01 64 3.0527E+01
65 3.0527E+01 66 3.0519E+01 67 3.0527E+01 68 3.0527E+01
69 3.0527E+01 70 3.0519E+01 71 3.0527E+01 72 3.0527E+01
73 3.0527E+01 74 3.0519E+01 75 3.0527E+01 76 3.0527E+01
77 3.0527E+01 78 3.0519E+01 79 3.0527E+01 80 3.0527E+01
81 3.0527E+01 82 3.0519E+01 83 3.0527E+01 84 3.0527E+01
85 3.0527E+01 86 3.0519E+01 87 3.0527E+01 88 3.0527E+01
89 3.0527E+01 90 3.0519E+01 91 3.0527E+01 92 4.0790E+02
93 4.3791E+02 94 4.7240E+02 95 5.1204E+02 96 5.5762E+02
97 6.1000E+02

```

tempf - 286.0000000000000

temp1 - 286.0000000000000

grav - 980.0000000000000

droefv - 0.2400000000000000

tort [1. 97] = 0.5600000000000000

viscw0 - 11.240000000000000

coefih - 2501.0000000000000

Setmax

```

1 9.8000E-02 2 9.8000E-02 3 9.8000E-02 4 9.8000E-02
5 9.8000E-02 6 9.8000E-02 7 9.8000E-02 8 9.8000E-02

```

```

9 9.8000E-02 10 9.8000E-02 11 9.8000E-02 12 9.8000E-02
13 9.8000E-02 14 9.8000E-02 15 9.8000E-02 16 9.8000E-02
17 9.8000E-02 18 9.8000E-02 19 9.8000E-02 20 9.8000E-02
21 9.8000E-02 22 9.8000E-02 23 9.8000E-02 24 9.8000E-02
25 9.8000E-02 26 9.8000E-02 27 9.8000E-02 28 9.8000E-02
29 9.8000E-02 30 9.8000E-02 31 9.8000E-02 32 9.8000E-02
33 9.8000E-02 34 9.8000E-02 35 9.8000E-02 36 9.8000E-02
37 9.8000E-02 38 9.8000E-02 39 9.8000E-02 40 9.8000E-02
41 9.8000E-02 42 9.8000E-02 43 9.8000E-02 44 9.8000E-02
45 9.8000E-02 46 9.8000E-02 47 9.8000E-02 48 9.8000E-01
49 2.8000E-01 50 2.8000E-01 51 2.8000E-01 52 2.8000E-01
53 2.8000E-01 54 2.8000E-01 55 2.8000E-01 56 4.2000E-01
57 4.2000E-01 58 4.2000E-01 59 4.2000E-01 60 4.2000E-01
61 4.2000E-01 62 4.2000E-01 63 4.2000E-01 64 4.2000E-01
65 4.2000E-01 66 4.2000E-01 67 4.2000E-01 68 4.2000E-01
69 4.2000E-01 70 4.2000E-01 71 4.2000E-01 72 4.2000E-01
73 4.2000E-01 74 4.2000E-01 75 4.2000E-01 76 4.2000E-01
77 4.2000E-01 78 4.2000E-01 79 4.2000E-01 80 4.2000E-01
81 4.2000E-01 82 4.2000E-01 83 4.2000E-01 84 4.2000E-01
85 4.2000E-01 86 4.2000E-01 87 4.2000E-01 88 4.2000E-01
89 4.2000E-01 90 4.2000E-01 91 4.2000E-01 92 4.2000E-01
93 4.2000E-01 94 4.2000E-01 95 4.2000E-01 96 4.2000E-01
97 4.2000E-01

```

setmin

```

1 1.0000E-04 2 1.0000E-04 3 1.0000E-04 4 1.0000E-04
5 1.0000E-04 6 1.0000E-04 7 1.0000E-04 8 1.0000E-04
9 1.0000E-04 10 1.0000E-04 11 1.0000E-04 12 1.0000E-04
13 1.0000E-04 14 1.0000E-04 15 1.0000E-04 16 1.0000E-04
17 1.0000E-04 18 1.0000E-04 19 1.0000E-04 20 1.0000E-04
21 1.0000E-04 22 1.0000E-04 23 1.0000E-04 24 1.0000E-04
25 1.0000E-04 26 1.0000E-04 27 1.0000E-04 28 1.0000E-04
29 1.0000E-04 30 1.0000E-04 31 1.0000E-04 32 1.0000E-04
33 1.0000E-04 34 1.0000E-04 35 1.0000E-04 36 1.0000E-04
37 1.0000E-04 38 1.0000E-04 39 1.0000E-04 40 1.0000E-04
41 1.0000E-04 42 1.0000E-04 43 1.0000E-04 44 1.0000E-04
45 1.0000E-04 46 1.0000E-04 47 1.0000E-04 48 1.0000E-04
49 1.0000E-03 50 1.0000E-03 51 1.0000E-03 52 1.0000E-03
53 1.0000E-03 54 1.0000E-03 55 1.0000E-03 56 1.0000E-03
57 1.0000E-03 58 1.0000E-03 59 1.0000E-03 60 1.0000E-03
61 1.0000E-03 62 1.0000E-03 63 1.0000E-03 64 1.0000E-03
65 1.0000E-03 66 1.0000E-03 67 1.0000E-03 68 1.0000E-03
69 1.0000E-03 70 1.0000E-03 71 1.0000E-03 72 1.0000E-03
73 1.0000E-03 74 1.0000E-03 75 1.0000E-03 76 1.0000E-03
77 1.0000E-03 78 1.0000E-03 79 1.0000E-03 80 1.0000E-03
81 1.0000E-03 82 1.0000E-03 83 1.0000E-03 84 1.0000E-03
85 1.0000E-03 86 1.0000E-03 87 1.0000E-03 88 1.0000E-03
89 1.0000E-03 90 1.0000E-03 91 1.0000E-03 92 1.0000E-03
93 1.0000E-03 94 1.0000E-03 95 1.0000E-03 96 1.0000E-03
97 1.0000E-03

```

pscp0

```

1 9.8000E-05 2 9.8000E-05 3 9.8000E-05 4 9.8000E-05
5 9.8000E-05 6 9.8000E-05 7 9.8000E-05 8 9.8000E-05
9 9.8000E-05 10 9.8000E-05 11 9.8000E-05 12 9.8000E-05
13 9.8000E-05 14 9.8000E-05 15 9.8000E-05 16 9.8000E-05
17 9.8000E-05 18 9.8000E-05 19 9.8000E-05 20 9.8000E-05
21 9.8000E-05 22 9.8000E-05 23 9.8000E-05 24 9.8000E-05
25 9.8000E-05 26 9.8000E-05 27 9.8000E-05 28 9.8000E-05
29 9.8000E-05 30 9.8000E-05 31 9.8000E-05 32 9.8000E-05

```


C-14

```

85 1.13953E+05
89 1.13953E+05 90 1.13953E+05
93 1.13953E+05 94 1.13953E+05 95 1.13953E+05
97 1.13953E+05

vgm [ 1: 5/ ] = 0.230769000000000
perm
  1 1.14694E-06  2 1.14694E-06  3 1.14694E-06  4 1.14694E-06
  5 1.14694E-06  6 1.14694E-06  7 1.14694E-06  8 1.14694E-06
  9 1.14694E-06 10 1.14694E-06 11 1.14694E-06 12 1.14694E-06
 13 1.14694E-06 14 1.14694E-06 15 1.14694E-06 16 1.14694E-06
 17 1.14694E-06 18 1.14694E-06 19 1.14694E-06 20 1.14694E-06
 21 1.14694E-06 22 1.14694E-06 23 1.14694E-06 24 1.14694E-06
 25 1.14694E-06 26 1.14694E-06 27 1.14694E-06 28 1.14694E-06
 29 1.14694E-06 30 1.14694E-06 31 1.14694E-06 32 1.14694E-06
 33 1.14694E-06 34 1.14694E-06 35 1.14694E-06 36 1.14694E-06
 37 1.14694E-06 38 1.14694E-06 39 1.14694E-06 40 1.14694E-06
 41 1.14694E-06 42 1.14694E-06 43 1.14694E-06 44 1.14694E-06
 45 1.14694E-06 46 1.14694E-06 47 6.65224E-08 48 6.65224E-08
 49 6.65224E-08 50 6.65224E-08 51 6.65224E-08 52 6.65224E-08
 53 6.65224E-08 54 6.65224E-08 55 6.65224E-08 56 1.83510E-06
 57 1.83510E-06 58 1.83510E-06 59 1.83510E-06 60 1.83510E-06
 61 1.83510E-06 62 1.83510E-06 63 1.83510E-06 64 1.83510E-06
 65 1.83510E-06 66 1.83510E-06 67 1.83510E-06 68 1.83510E-06
 69 1.83510E-06 70 1.83510E-06 71 1.83510E-06 72 1.83510E-06
 73 1.83510E-06 74 1.83510E-06 75 1.83510E-06 76 1.83510E-06
 77 1.83510E-06 78 1.83510E-06 79 1.83510E-06 80 1.83510E-06
 81 1.83510E-06 82 1.83510E-06 83 1.83510E-06 84 1.83510E-06
 85 1.83510E-06 86 1.83510E-06 87 1.83510E-06 88 1.83510E-06
 89 1.83510E-06 90 1.83510E-06 91 1.83510E-06 92 1.83510E-06
 93 1.83510E-06 94 1.83510E-06 95 1.83510E-06 96 1.83510E-06
 97 1.83510E-06

poros
  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
 13 9.80000E-02 14 9.80000E-02 15 9.80000E-02 16 9.80000E-02
 17 9.80000E-02 18 9.80000E-02 19 9.80000E-02 20 9.80000E-02
 21 9.80000E-02 22 9.80000E-02 23 9.80000E-02 24 9.80000E-02
 25 9.80000E-02 26 9.80000E-02 27 9.80000E-02 28 9.80000E-02
 29 9.80000E-02 30 9.80000E-02 31 9.80000E-02 32 9.80000E-02
 33 9.80000E-02 34 9.80000E-02 35 9.80000E-02 36 9.80000E-02
 37 9.80000E-02 38 9.80000E-02 39 9.80000E-02 40 9.80000E-02
 41 9.80000E-02 42 9.80000E-02 43 9.80000E-02 44 9.80000E-02
 45 9.80000E-02 46 9.80000E-02 47 2.80000E-01 48 2.80000E-01
 49 2.80000E-01 50 2.80000E-01 51 2.80000E-01 52 2.80000E-01

```

```

dudts 1
dudtw = 4.200000000000000
ekg = 0.
eks [ 1: 97 ] = 8.076920000000000D-03
ekw = 5.800000000000000D-03
eintw0 = 0.
eintv0 = 0.

Boundary conditions...

zbcw0 = flux
bcw0 = 0.
zbcw1 = gradsat
bcw1 = 0.
pondh0 = 1000.0000000000000
zbcv0 = densva
zblk0 = campbell
blzt0 = 300.0000000000000
blzh0 = 3.000000000000000D-02
blzu0 = 300.0000000000000
blzm0 = 3.000000000000000D-02
blzoff0 = 0.
zbcv1 = densva
bcv1 = 1.127100000000000D-05
zblk1 = constant
blkv1 = 1.584000000000000D-04
blkh1 = 0.
zbccl = gradtemp
zbccl = gradtemp
bccl = 0.
esigsd0 = 5.103000000000000D-12
esigsw0 = 5.202920000000000D-12
albedd0 = 0.250000000000000
albedw0 = 0.250000000000000

pressw
  1 -1.39041E+09  2 -1.24377E+09  3 -1.09337E+09  4 -9.41794E+08
  5 -7.94687E+08  6 -6.58756E+08  7 -5.40423E+08  8 -4.43301E+08
  9 -3.66896E+08 10 -3.07923E+08 11 -2.62388E+08 12 -2.26834E+08
 13 -1.98637E+08 14 -1.75905E+08 15 -1.57290E+08 16 -1.41830E+08
 17 -1.28828E+08 18 -1.17772E+08 19 -1.08281E+08 20 -1.00072E+08
 21 -9.29312E+07 22 -8.67004E+07 23 -8.12643E+07 24 -7.65411E+07
 25 -7.24774E+07 26 -6.87007E+07 27 -6.76136E+07 28 -6.66914E+07

```

29	-6.59073E+07	30	-6.52359E+07	31	-6.46720E+07	32	-6.41895E+07	33	-6.37809E+07	34	-6.34366E+07	35	-6.31483E+07	36	-6.29091E+07	37	-6.27131E+07	38	-6.25552E+07	39	-6.24310E+07	40	-6.23365E+07	41	-6.22682E+07	42	-6.22233E+07	43	-6.21988E+07	44	-6.21924E+07	45	-6.22019E+07	46	-6.22253E+07	47	-6.43279E+07	48	-7.84257E+07	49	-9.57464E+07	50	-9.96984E+07	51	-9.98881E+07	52	-9.99330E+07	53	-9.99530E+07	54	-9.99848E+07	55	-9.99957E+07	56	-9.99990E+07	57	-9.99976E+07	58	-9.99980E+07	59	-9.99982E+07	60	-9.99983E+07	61	-9.99983E+07	62	-9.99983E+07	63	-9.99984E+07	64	-9.99984E+07	65	-9.99984E+07	66	-9.99985E+07	67	-9.99985E+07	68	-9.99985E+07	69	-9.99986E+07	70	-9.99986E+07	71	-9.99987E+07	72	-9.99988E+07	73	-9.99988E+07	74	-9.99989E+07	75	-9.99990E+07	76	-9.99991E+07	77	-9.99992E+07	78	-9.99993E+07	79	-9.99994E+07	80	-9.99995E+07	81	-9.99996E+07	82	-9.99997E+07	83	-9.99998E+07	84	-9.99998E+07	85	-9.99999E+07	86	-9.99999E+07	87	-1.00000E+08	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	-9.99836E+07
----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------	----	--------------

satw

1	1.11969E-02	2	1.15741E-02	3	1.20264E-02	4	1.25724E-02	5	1.32243E-02	6	1.39840E-02	7	1.48334E-02	8	1.57353E-02	9	1.66479E-02	10	1.75406E-02	11	1.83978E-02	12	1.92141E-02	13	1.99899E-02	14	2.07277E-02	15	2.14307E-02	16	2.21021E-02	17	2.27450E-02	18	2.33617E-02	19	2.39542E-02	20	2.45238E-02	21	2.50709E-02	22	2.55947E-02	23	2.60933E-02	24	2.65628E-02	25	2.69979E-02	26	2.74116E-02	27	2.75621E-02	28	2.76750E-02	29	2.77266E-02	30	2.78569E-02	31	2.79295E-02	32	2.79918E-02	33	2.80451E-02	34	2.80903E-02	35	2.81284E-02	36	2.81603E-02	37	2.81863E-02	38	2.82075E-02	39	2.82242E-02	40	2.82369E-02	41	2.82461E-02	42	2.82522E-02	43	2.82555E-02	44	2.82564E-02	45	2.82551E-02	46	2.82519E-02	47	4.26943E-02	48	4.02885E-02	49	3.80059E-02	50	3.75596E-02	51	3.75388E-02	52	3.75339E-02	53	3.75306E-02	54	3.75282E-02	55	3.75270E-02	56	5.58554E-02	57	5.58555E-02	58	5.58555E-02	59	5.58555E-02	60	5.58555E-02	61	5.58555E-02	62	5.58555E-02	63	5.58555E-02	64	5.58555E-02	65	5.58555E-02	66	5.58555E-02	67	5.58555E-02	68	5.58555E-02	69	5.58555E-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	5.58555E-02	77	5.58555E-02	78	5.58555E-02	79	5.58555E-02	80	5.58555E-02	81	5.58555E-02	82	5.58555E-02	83	5.58555E-02	84	5.58555E-02	85	5.58555E-02	86	5.58555E-02	87	5.58555E-02	88	5.58555E-02	89	5.58555E-02	90	5.58555E-02	91	5.58555E-02	92	5.58555E-02	93	5.58555E-02	94	5.58555E-02	95	5.58555E-02	96	5.58555E-02	97	5.58555E-02
---	-------------	---	-------------	---	-------------	---	-------------	---	-------------	---	-------------	---	-------------	---	-------------	---	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------

densv

1
5
9
13
17
21
25
29
33
37
41
45
49

temp

1	2.84267E+02	2	2.84297E+02	3	2.84365E+02	4	2.84443E+02	5	2.84530E+02	6	2.84628E+02	7	2.84738E+02	8	2.84861E+02	9	2.84999E+02	10	2.85154E+02	11	2.85326E+02	12	2.85517E+02	13	2.85729E+02	14	2.85964E+02	15	2.86221E+02	16	2.86503E+02	17	2.86810E+02	18	2.87141E+02	19	2.87495E+02	20	2.87870E+02	21	2.88261E+02	22	2.88662E+02	23	2.89063E+02	24	2.89451E+02	25	2.89808E+02	26	2.90145E+02	27	2.90399E+02	28	2.90315E+02	29	2.90377E+02	30	2.90425E+02	31	2.90461E+02	32	2.90488E+02	33	2.90507E+02	34	2.90519E+02	35	2.90525E+02	36	2.90525E+02	37	2.90522E+02	38	2.90514E+02	39	2.90504E+02	40	2.90492E+02	41	2.90477E+02	42	2.90461E+02	43	2.90443E+02	44	2.90425E+02	45	2.90406E+02	46	2.90386E+02	47	2.90363E+02	48	2.90311E+02	49	2.90197E+02	50	2.89941E+02	51	2.89382E+02	52	2.88389E+02	53	2.87473E+02	54	2.86635E+02	55	2.86244E+02	56	2.86138E+02	57	2.86098E+02	58	2.86082E+02	59	2.86074E+02	60	2.86071E+02	61	2.86069E+02	62	2.86068E+02	63	2.86067E+02	64	2.86066E+02	65	2.86065E+02	66	2.86063E+02	67	2.86062E+02	68	2.86060E+02	69	2.86058E+02	70	2.86055E+02	71	2.86053E+02	72	2.86050E+02	73	2.86047E+02	74	2.86044E+02	75	2.86040E+02	76	2.86037E+02	77	2.86033E+02	78	2.86029E+02	79	2.86025E+02	80	2.86021E+02	81	2.86017E+02	82	2.86013E+02	83	2.86010E+02	84	2.86007E+02	85	2.86004E+02	86	2.86003E+02	87	2.86001E+02	88	2.86001E+02	89	2.86000E+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	2.85999E+02
---	-------------	---	-------------	---	-------------	---	-------------	---	-------------	---	-------------	---	-------------	---	-------------	---	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------	----	-------------

BIBLIOGRAPHIC DATA SHEET

(See instructions on the reverse)

1. REPORT NUMBER
(Assigned by NRC, Add Vol., Supp., Rev.,
and Addendum Numbers, if any.)

NUREG/CR-6333
CNWRA 94-020

2. TITLE AND SUBTITLE

BREATH Version 1.1 - Coupled Flow and Energy Transport in
Porous Media

Simulator Description and User Guide

3. DATE REPORT PUBLISHED

MONTH	YEAR
July	1995

4. FIN OR GRANT NUMBER
B6666

5. AUTHOR(S)

S.A. Stothoff

6. TYPE OF REPORT

7. PERIOD COVERED (Inclusive Dates)

8. PERFORMING ORGANIZATION - NAME AND ADDRESS (If NRC, provide Division, Office or Region, U.S. Nuclear Regulatory Commission, and mailing address, if contractor, provide name and mailing address.)

Southwest Research Institute
Center for Nuclear Waste Regulatory Analyses
6220 Culebra Road
San Antonio, TX 78228-0510

9. SPONSORING ORGANIZATION - NAME AND ADDRESS (If NRC, type "Same as above"; if contractor, provide NRC Division, Office or Region, U.S. Nuclear Regulatory Commission, and mailing address.)

Division of Regulatory Applications
Office of Nuclear Regulatory Research
U.S. Nuclear Regulatory Commission
Washington, DC 20555-0001

10. SUPPLEMENTARY NOTES

11. ABSTRACT (200 words or less)

This document describes the BREATH computer 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 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.

The BREATH simulator has been developed for use in auxiliary analyses which are a part of the Nuclear Regulatory Commission 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 U.S. 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.

12. KEY WORDS/DESCRIPTORS (List words or phrases that will assist researchers in locating the report.)

infiltration analysis, heat transfer in soils, Richards' equation,
modified Picard, finite element code

13. AVAILABILITY STATEMENT

unlimited

14. SECURITY CLASSIFICATION

(This Page)

unclassified

(This Report)

unclassified

15. NUMBER OF PAGES

16. PRICE



Federal Recycling Program

UNITED STATES
NUCLEAR REGULATORY COMMISSION
WASHINGTON, D.C. 20555-0001

SPECIAL FOURTH CLASS RATE
POSTAGE AND FEES PAID
USNRC
PERMIT NO. G-67

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE, \$300

120555139531 1 1AN1RW
US NRC-OADM
DIV FOIA & PUBLICATIONS SVCS
TPS-PDR-NUREG
2WFN-6E7
WASHINGTON DC 20555