1/2/00

SOFTWARE RELEASE NOTICE

01. SRN Number: GHGC-SRN-038	
02. Project Title: Breath - Coupled Flow and Energy Transport in Porous Media	Project No. 20-5704-192
03. SRN Title: BREATH Ver 1.1	
04. Originator/Requester: Thomas J. Ratchford	Date: 9/20/95

05. Summary of Actions

- ■ Release of new code admitted to CM System
- ☐ Release of modified code:
  - ☐ Enhancements made
  - ☐ Corrections made
- ☐ Change of access code
- ☐ Software retirement

06. Persons Authorized Access

Name	RO/RW	A/C/D

07. Element Manager Approval: [signature] Date: 9/21/95

08. Remarks:

A copy of the software package BREATH, Ver 1.1 was retained by the Principle Investigator for use in the CNWRA work center; therefore, a new release may not be necessary.

CNWRA Form TOP-6 (08/93)

1/1

SOFTWARE RELEASE NOTICE

1. SRN Number: GHGC-SRN-038

2. Project Title: Breath - Coupled Flow and Energy Transport in Porous Media	Project No. 20.05704.192
3. SRN Title: BREATH V.1.1	
4. Originator/Requestor: Thomas J. Ratchford	Date: 10/18/2001

5. Summary of Actions

- ☐ Release of new software
- ☐ Change of access software
- ☐ Release of modified software:
  - ☐ Enhancements made
  - ☐ Corrections made
- ■ Software Retirement

6. Validation Status

- ☐ Validated
- ☐ Limited Validation
- ☐ Not Validated Explain: ______________________

7. Persons Authorized Access

Name	Read Only/Read-Write	Addition/Change/Delete

8. Element Manager Approval: English Pearcy EP Date: 10/22/01

9. Remarks:

2/100

SOFTWARE SUMMARY FORM

01.Summary Date: Sept. 20, 1995	02. Summary prepared by(Name and Phone) T. J. Ratchford	03. Summary Action: NEW
04. Software Date: Sept. 20, 1995	05. Short Title: Breath Ver 1.1	

06. Software Title: Breath V1.1 - Coupled Flow and Energy Transport in Porous Media	07. Internal Software ID: NONE

08. Software Type: ☐ Automated Data System ■ Computer Program ☐ Subroutine/Module	09.Processing Mode: ☐ Interactive ■ Batch ☐ Combination	10. APPLICATION AREA A. General: ■ Scientific/Engineering ■ Auxiliary Analyses ☐ Total System PA ☐ Subsystem PA ☐ Other b. Specific:

11. Submitting Organization and Address: CNWRA, SwRI, San Antonio, Texas	12. Technical Contact(s) and Phone: S. Stothoff 210/522-5208

13. Narrative: Breath computer code is designed to simulate saturated/unsaturated flow of liquid, vapor, and energy in a porous medium with atmospheric interaction.

14. Computer Platform Generic	15. Computer Operating System: Generic	16. Programming Language(s): Fortran 77	17. Number of Source Program Statements: 8360 Lines of code
18. Computer Memory Requirements: As required	19. Tape Drives: NONE	20. Disk/Drum Units: N/A	21. Graphics: NONE

22. Other Operational Requirements

NONE

23. Software Availability: ■ Available ☐ Limited ☐ In-House ONLY	24. Documentation Availability: ■ Available ☐ Inadequate ☐ In-House ONLY

25. Submission Package Status:

Acceptance Criteria: Met ■ Not Met ☐ Software QA Assessment: Successful ■ Unsuccessful ☐

Code Custodian: T J Ratchford Date: 9/20/95

CNWRA Form TOP-4-1 (08/93)

Diagnostics for BREATH run on FORWARN

Local Diagnostics for Subroutine RICHASSM File BREATH.F

Line	Severity	Message
95	I	(17) Local variable SATE0 is assigned but never used

Local Diagnostics for Subroutine PONDUPDT File BREATH.F

Line	Severity	Message
253	I	(17) Local variable ME is assigned but never used
267	I	(17) Local variable PMAXPR is assigned but never used

Local Diagnostics for Subroutine RICHSTEP File BREATH.F

Line	Severity	Message
834	I	(34) Dummy argument ALPHA0 is never referenced
834	I	(34) Dummy argument ALPHA1 is never referenced
862	W	(18) Local variable IFITEX is used but never initialized

Local Diagnostics for Function FDSDPC2 File BREATH.F

Line	Severity	Message
1113	I	(34) Dummy argument MND is never referenced
1113	I	(34) Dummy argument SATEFF is never referenced

Local Diagnostics for Function FVISCW1 File BREATH.F

Line	Severity	Message
1251	I	(34) Dummy argument TEMP is never referenced

Local Diagnostics for Subroutine TSSEL2 File BREATH.F

Line	Severity	Message
1375	I	(34) Dummy argument CONV is never referenced
1375	I	(34) Dummy argument CONV0 is never referenced
1375	I	(34) Dummy argument CONVN is never referenced
1375	I	(34) Dummy argument NIT is never referenced
1375	I	(34) Dummy argument NITNOM is never referenced
1375	I	(34) Dummy argument YCONV is never referenced

Local Diagnostics for Subroutine BRMTRACE File BTRACE.F

Line	Severity	Message
757	W	(18) Local variable IOMBN is used but never initialized

Local Diagnostics for Subroutine INPARS File COMMAND.F

Line	Severity	Message
154	W	(18) Local variable IOMES is used but never initialized

Local Diagnostics for Subroutine PARVCZ File COMMAND.F

Line	Severity	Message
452	I	(34) Dummy argument ZNAME is never referenced

Local Diagnostics for Subroutine VECHZ File COMMAND.F

Line	Severity	Message
728	I	(34) Dummy argument NVAR is never referenced

Local Diagnostics for Subroutine THRMSTEP File THERMAL.F

Line	Severity	Message
834	W	(18) Local variable IEITEX is used but never initialized

Global Diagnostics

Line	Routine	Severity	Message
501(1)	RICH1BC	E	(31) Argument #2 to PT2DENV has the wrong type
1004(1)	FMOBW2	I	(13) Routine FMOBW2 is never called
1012(1)	FMOBW3	I	(13) Routine FMOBW3 is never called
1112(1)	FDSDPC2	I	(13) Routine FDSDPC2 is never called
1836(1)	BRSELM	E	(30) Argument #1 to BRSELV must be an external routine name
1948(1)	BRCAL	E	(11) RICHSTEP is called with too few arguments (11, not 14)
1977(1)	BRCAL	E	(11) TMRUN is called with too few arguments (5, not 6)
795(2)	BRMBINIT	I	(13) Routine BRMBINIT is never called
581(3)	BRINP0	E	(11) BRMASS is called with too few arguments (0, not 1)
1030(3)	BRVACT	E	(36) Type of function FZI2Y at call differs from definition
1032(3)	BRVACT	E	(36) Type of function FZI2Y at call differs from definition
1038(3)	BRVACT	E	(36) Type of function FZI2Y at call differs from definition
1048(3)	BRVACT	E	(36) Type of function FZI2Y at call differs from definition
1050(3)	BRVACT	E	(36) Type of function FZI2Y at call differs from definition
1052(3)	BRVACT	E	(36) Type of function FZI2Y at call differs from definition
1054(3)	BRVACT	E	(36) Type of function FZI2Y at call differs from definition
1056(3)	BRVACT	E	(36) Type of function FZI2Y at call differs from definition
1058(3)	BRVACT	E	(36) Type of function FZI2Y at call differs from definition
1284(3)	OBRMETRD	I	(13) Routine OBRMETRD is never called
605(4)	OTHRMITER	I	(13) Routine OTHRMITER is never called
927(4)	OTMSTEP	I	(13) Routine OTMSTEP is never called
966(4)	OTMSTEP	E	(11) RICHITER is called with too few arguments (8, not 9)
992(4)	OTMSTEP	E	(11) THRMITER is called with too few arguments (3, not 6)

Summary

File Name	Lines	Local			Global		
		I	W	E	I	W	E
1. BREATH.F	2024	15	1	0	3	0	4
2. BTRACE.F	1116	0	1	0	1	0	0
3. COMMAND.F	1454	2	1	1	1	0	10
4. THERMAL.F	1370	0	1	0	2	0	2
5. BRECOM.H	337	0	0	0	0	0	0
6. BREFUN.H	24	0	0	0	0	0	0
Totals	6325	17	4	1	7	0	16

SOFTWARE REQUIREMENTS DESCRIPTION

1/18

BREATH VERSION 1.2

User's Guide

by

Stuart A. Stothoff

Center for Nuclear Waste Regulatory Analyses
Southwest Research Institute
San Antonio, Texas

June 12, 1997

2/18

3/18

CONTENTS

4/18

TABLES

5/18

6/18

1 INTRODUCTION

The one-dimensional (1D) finite-element code BREATH is designed to simulate the weakly nonisothermal transport of moisture and energy in a porous medium. The BREATH code was particularly designed to simulate infiltration and evaporation from a bare-soil porous medium. The theory underlying the code is presented in the BREATH Version 1.1 user's manual (Stothoff, 1995). The current document is an addendum to the BREATH Version 1.1 user's manual, so only the modifications unique to BREATH Version 1.2 are reported here.

The modifications made to BREATH Version 1.1 are relatively minor, primarily involved with enhancing the data flow and making the internal variables more accessible to the user. Significant modifications include:

- Providing additional options for calculating second derivatives (section 2)
- Reporting additional simulation statistics (section 3)
- Enhancing meteorological input options (section 4)
- Expanding input and output capability (section 5)

Errata in the BREATH Version 1.1 documentation are summarized in section 6. The modifications are summarized in section 7.

2 TREATMENT OF SECOND DERIVATIVES

In order to calculate material and energy fluxes across element boundaries, it is necessary to estimate the value of coefficients associated with the second derivative (i.e., conductivity, the vapor diffusion coefficient) at the boundary between each pair of nodes. In BREATH Version 1.1 only one option was available; the coefficient was estimated using geometric weighting $\left[c_{i+1/2} \approx \left(c_i \cdot c_{i+1}\right)^{1/2}\right]$ However, for simulations considering flow in fractures, the original formulation generated highly implausible results due to the strongly contrasting mobilities that could occur at adjacent nodes. In BREATH Version 1.2, additional weighting options are available.

The typically relatively well-behaved vapor diffusion coefficient may be calculated using arithmetic, geometric, or harmonic means between nodes i and $i + 1$,

$$c^a_{i+1/2} = \left(c_i + c_{i+1}\right)/2 \tag{2-1}$$

$$c^g_{i+1/2} = \left(c_i \cdot c_{i+1}\right)^{1/2} \tag{2-2}$$

$$c^h_{i+1/2} = 2\left(c_i \cdot c_{i+1}\right) / \left(c_i \cdot c_{i+1}\right) \tag{2-3}$$

where c is the diffusion coefficient and superscripts a, g, and h represent arithmetic, geometric, and harmonic means, respectively. The option is selected by setting *zavsel* to "arithmet", "geometri", or "harmonic", respectively.

The same set of options can be used to calculate the hydraulic conductivity parameter, by specifying *zawsel* as "arithmet", "geometri", or "harmonic". In addition, it is possible to upstream weight either the conductivity or the mobility. In upstream weighting, $c_{i+1/2} = c_i$ if flow is from node i to node $i + 1$ and $c_{i+1/2} = c_{i+1}$ otherwise, where c represents conductivity or mobility. If *zawsel* is specified as "upstream", the conductivity is upstream weighted. If *zawsel* is specified as "uparithm", "upgeomet", or "upharmon", the mobility is upstream weighted and the intrinsic permeability is averaged using arithmetic, geometric, or harmonic means, respectively.

The selection of upstream weighting or arithmetic-mean weighting is found to provide more consistent results in BREATH simulations, insofar as simulation predictions are less affected by grid refinement, particularly in sharp-front simulations.

3 REPORTING THE STATISTICAL BEHAVIOR OF SIMULATIONS

Several types of long-term average statistics are of interest in long-term infiltration simulations. In particular, the time-averaged fluxes and state variables (i.e., pressure, saturation, density, and temperature) can be quite useful, as well as time-averaged values for some of the derived quantities (e.g., relative permeability, grid Courant number, grid Peclet number). Minimum and maximum observed values for the state variables over the period of the simulation can also be relevant. In addition, average perturbation of the state variables about a mean value can be useful information. Some of these quantities were already tracked and could be reported in BREATH Version 1.1; all additional variables added in BREATH Version 1.2 are listed section 5. The mathematical basis for the new time-averaged, perturbation, and indicator variables are discussed in this section.

A time-averaged generic quantity, $\bar{u}$ is updated at the end of a time step from the previous value of $\bar{u}$ and the value of the generic quantity, u at the end of the time step by the formula

$$\bar{u}^{n+1} = \bar{u}^{n} \, T^{n} + u^{n+1} \frac{\left(T^{n+1} - T^{n}\right)}{T^{n+1}} \tag{3-1}$$

where n and $n + 1$ represent the old and new time levels, respectively, and T represents the cumulative time since the start of the simulation.

A perturbation variable is calculated at the end of a time step by the formula

$$(u')^2 = (u - \hat{u})^2 \tag{3-2}$$

where

u	is the generic variable
u'	is the perturbed variable
$\hat{u}$	is the average of the variable (input by the user)

A perturbation variable is related to the variance of the variable.

Courant and Peclet numbers are useful indicators for numerical stability and numerical dispersion problems in a transport simulation. The prototypical form of a transport equation used to calculate the indicators is

9/18

$$\frac{\partial c}{\partial t} + \frac{\partial}{\partial x}\left(Vc - D\frac{\partial c}{\partial x}\right) = 0 \tag{3-3}$$

where

- c is the quantity being transported
- V is the fluid velocity $[L\ T^{-1}]$, and
- D is a diffusion coefficient $[L^2\ T^{-1}]$

The grid Courant number is a measure of the fluid velocity relative to the length of the element and the time step, or

$$Co = \frac{V\Delta x}{\Delta t} \tag{3-4}$$

where

- Co is the grid Courant number
- Δx is the spacing between nodes, and
- Δt is the time step

The grid Peclet number, Pe, is a measure of the importance of velocity relative to diffusion within an element, or

$$Pe = \frac{V\Delta x}{D} \tag{3-5}$$

In general, stable BREATH simulations require grid Courant numbers less than 0.1. Grid Peclet numbers less than 2 are needed for accuracy (higher grid Peclet numbers cause greater numerical diffusion).

The Courant number that is used as a time stepping criterion, as discussed in the BREATH Version 1.1 documentation, is based on the transport of energy. For moisture flow problems, however, pressure-transport indicators are useful as well. Internally to BREATH pressure-transport indicators are calculated for the equation

$$\frac{\partial(\rho_w \theta_w)}{\partial t} + \nabla \cdot \left[-\rho w k \lambda \left(\frac{\partial P}{\partial x} + \rho w g \frac{\partial z}{\partial x}\right)\right] = 0 \tag{3-6}$$

by transforming it into the form

$$\frac{d(\rho_w \theta_w)}{dP}\frac{\partial P}{\partial t} - \nabla \cdot \left(k_1 \frac{\partial P}{\partial x} + K_2\right) = 0 \tag{3-7}$$

10/18

where

ρ_w	is the density of water $[M\ L^{-3}]$
θ_w	is the moisture content $[L^3\ L^{-3}]$
k	is the intrinsic permeability $[L^2]$
λ	is the mobility $[LTM^{-1}]$
g	is the acceleration due to gravity $[LT^{-2}]$, and
z	is the elevation above an arbitrary datum $[L]$

These quantities are described in detail in the BREATH Version 1.1 documentation. The variables K_1 and K_2 are obtained by inspection.

The prototypical transport Eq. (3-3) can be obtained by holding constant the coefficients in the moisture-balance equation and rearranging. The transport-equation form is recovered when

$$V = \frac{\rho_w k d\lambda / dP}{d(\rho_w \theta_w)/dP}\left(\rho_w g \frac{dz}{dx}\right) \qquad (3\text{-}8)$$

$$D = \frac{\rho_w k \lambda}{d(\rho_w \theta_w)/dP} \qquad (3\text{-}9)$$

These expressions are used to calculate the grid Courant and Peclet numbers whenever *kmobder* is not equal to zero. The default mode is to not undergo the overhead burden of calculating $d\lambda/dP$. The user must set *kmobder* not equal to zero in order to have $d\lambda/dP$ calculated and have nonzero grid Courant and Peclet reported.

4 MODIFYING THE METEOROLOGICAL BOUNDARY CONDITIONS

A set of simple augmentations were made to the **met** command, designed to allow greater flexibility in the specification of boundary conditions. The **met** command controls boundary-condition input from an external data file. The external data file can be quite large, particularly if years of hourly readings are provided. So that climatic influences can be examined without modifying the input file, each of the variables can now be scaled and shifted. In addition, each of the variables can now be linearly interpolated between readings or be assumed constant until the next reading (previously it was assumed that linear variation occurs). The full set of **met** variables are shown in table 4-1 which replaces table A-5 in the BREATH Version 1.1 documentation.

The formula for modifying **met** variables is:

$$u' = C_{shift} \left(U_{file} + C_{scale} \right) \tag{4-1}$$

where

- u' is the variable used for calculations
- u_{file} is the value in the input file
- C_{shift} is the scale parameter for u, and
- C_{scale} is the shift parameter for u.

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

```
file 31 open metdat.tst
met unit 31 side 0 runtype tmrun dummy bce const bce scale bce 2
        nmetcyc 2 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. At the bottom of the file, the file is rewound and the file is read from the top again. The rewind cycle occurs twice (*nmetcyc* is 2). While reading, B_e is assumed constant for each period and the value in the file is multiplied by 2.

12/18

Table 4-1. The met command

Modifier Word	Argument(s)	Function
unit	*IOMET*	Unit for input
format	*ZMEFMT*	FORTRAN format for input
runtype	*ZMERUN*	Type of run
	richrun	Moisture only
	thrmrun	Energy only
	tmrun	Coupled energy/moisture
nmetcyc	NCYC	Number of rewinds for the input file
side	MMET	0/1 for side 0,1
[*ZNAME*]	—	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_t
windsp	—	Wind speed
const	*ZNAME*	Variable *ZNAME* is constant between entries
linear	*ZNAME*	Variable *ZNAME* is linear between entries
scale	*ZNAME*	Variable to scale
	scfac	Amount to scale
shift	*ZNAME*	Variable to shift
	shfac	Amount to shift
set	—	End of **met** input

*No more than one of *esiga* or *lwrad* should be provided.

5 MODIFICATIONS TO INPUT/OUTPUT CAPABILITIES

The data structure in BREATH Version 1.1 was not well suited to allowing the state of the simulation to be dumped at the end of the simulation. In addition, it was found that adding more variables was quite clumsy. Accordingly, the internal storage structure was streamlined with the express purpose of allowing all variables to be output in a format suitable to provide initial conditions on a restart basis.

The restart capability is straightforwardly implemented. A new variable, *iovar*, is used to hold the unit number of the file that has been opened to receive the dump of the variables. The segment of input-file code:

```
file 28 open vardump.out
iovar 28
```

is an example of how the restart capability is initiated. At the end of execution or when an interrupt signal is given, all variables are echoed into the file in a format that BREATH can read. However, the user must add all control commands into the file (i.e., anything other than the **set** commands) before the file can be used to run another BREATH simulation.

As a consequence of the new storage structure, the implementation of the **snap** and **trace** commands was simplified, allowing each nodal and elemental variable to be output with these commands. Thus, table A-8 in the BREATH Version 1.1 documentation is incomplete and should be replaced with table 5-1.

Table 5-1. The trace and snap commands

Modifier Word	Argument(s)	Function
nodeunit	*IOTRN*	Unit for nodal quantity output
elemunit	*IOTRE*	Unit for elemental quantity output
format	*ZFMT*	FORTRAN format for output
nperskip	*NSKIP*	Periods to skip between outputs; 0 for no skipping
allstep	—	Output each time step
all	—	Flag all variables
nodes	—	Flag all nodal variables
elems	—	Flag all elemental variables
tottim	—	Flag output of total time
*ZNAME**	—	Flag output of nodal or elemental variable *ZNAME*
on	—	Turn on output of flagged variables
off	—	Turn off output of flagged variables

**ZNAME* is a generic name that is replaced by the name of the variable of interest.

14/18

As another consequence of streamlining the data storage structure, internal vectors that could not previously be addressed using the input/output files can now be accessed. All of the variables that are available in BREATH Version 1.2 that did not appear in BREATH Version 1.1 are listed in table 5-2; variables listed in the tables in Appendix B of the BREATH Version 1.1 documentation are also available.

Table 5-2. New variables available in BREATH Version 1.2.

Name	Size	Units	Default	Function
amobw	nnd	LTM^{-1}	0	Current value of mobility
dvdp	nnd	$L^{-2}T^{-2}$	0	Derivative of ρ_w with respect to $\boldsymbol{P}_w$
dvdt	nnd	$ML^{-3}\ K^{-1}$	0	Derivative of ρ_w with respect to $\boldsymbol{T}$
dmwdp	nnd	$L^2T^3M^{-2}$	0	Derivative of λ with respect to $\boldsymbol{P}_w$
dmcdpw	nnd	$L\ T^2M^{-1}$	0	Derivative of θ_w with respect to $\boldsymbol{P}_w$
amptp	nnd	K^{-1}	0	$\theta_w' \partial T' / \partial x$
aperd	nnd	$(M\ L^{-3})^2$	0	Time-averaged $\left(\rho_v'\right)^2$
aperp	nnd	$(M\ L^{-1}\ T^{-2})^2$	0	Time-averaged $\left(\boldsymbol{P}_w'\right)^2$
aprssw	nnd	$M\ L^{-1}T^{-2}$	0	Time averaged $\boldsymbol{P}_w$
aved	nnd	$M\ L^{-3}$	0	$\hat{\rho}_v$ to calculate perturbations about
avep	nnd	$M\ L^{-1}T^{-2}$	0	$\hat{\boldsymbol{P}}_w$ to calculate perturbations about
aves	nnd	[–]	0	$\hat{\theta}_w$ to calculate perturbations about
avet	nnd	K	0	$\hat{\boldsymbol{T}}$ to calculate perturbations about
couw	nnd	[–]	0	Grid Courant number
pecw	nnd	[–]	0	Grid Peclet number
avmobw	nnd	LTM^{-1}	0	Time-averaged *amobw*
avdmdp	nnd	$L\ T^2M^{-1}$	0	Time-averaged *dmcdpw*
avdmwdp	nnd	$L^2T^3M^{-2}$	0	Time-averaged *dmwdp*
avmptp	nnd	K^{-1}	0	Time-averaged *amptp*
avcouw	nnd	[–]	0	Time-averaged *couw*
avpecw	nnd	[–]	0	Time-averaged *pecw*
iovar	1	[–]	6	Unit number for variable-dump (restart) output

15/18

khdres	1	[-]	0	1/0 to output header in element snap file
khdret	1	[-]	0	1/0 to output header in element trace file
khdrmb	1	[-]	0	1/0 to output header in mass balance file
khdrns	2	[-]	0	1/0 to output header in node snap file
khdrnt	1	[-]	0	1/0 to output header in node trace file
khdrvd	1	[-]	0	1/0 to output header in variable dump file
ztime	1	[-]		Character string with clock time at start of simulation
zversion	1	[-]		Character string with BREATH version

Technical Operating Procedure (TOP)-018 governs the development of computer software at CNWRA. One of the requirements of TOP-018 is that individual executions of a code are uniquely identified. Individual BREATH runs may be identified with a header in each output file, consisting of 3 lines of output: (i) BREATH version, (ii) time of the start of execution, and (iii) purpose of the output file. The default execution mode is to not provide these headers, but they can be turned on by setting variables *khdres*, *khdret*, *khdrmb*, *khrns*, *khdrnt*, and *khdrvd* to 1 (see Table 3) for descriptions). The version number is provided as a character string call *zversion* and the time at the start of execution is provided as a character string called *ztime*; these can be output using **echo** like any other variable.

16/18

6 ERRATA IN BREATH VERSION 1.1 DOCUMENTATION

Several corrections in the BREATH Version 1.1 documentation are identified:

p. x Units for both k_v and k_h should be $[LT^{-1}]$

p. 4-6 The correct expression for the *densva* boundary condition in Table 4-2 is $q_{vs} = k_v(B_v - \rho_v)/\rho_v$

p. 4-9 Units for atmospheric density ρ should be $[ML^{-3}]$

17/18

7 SUMMARY

The current document is an addendum to the BREATH Version 1.1 user's manual (Stothoff, 1995) so only modifications unique to the 1D finite-element code BREATH are reported here. Modifications to BREATH Version 1.1 are relatively minor, primarily involved with enhancing the data flow and making the internal variables more accessible to the user.

In section 2 various options for calculating second-derivative terms were presented. BREATH Version 1.1 provided only one option; BREATH Version 1.2 provides seven options for the flow equation and four options for the energy equation.

Additional simulation statistics are discussed in section 3. These statistics are primarily concerned with time-averaged behavior of state and derived variables, and perturbations of state variables. Diagnostic variables (i.e., grid Courant and Peclet numbers) are also presented.

Enhanced input capabilities for meteorological parameters are presented in section 4. The new capabilities enable the user to repeat the meteorological input deck and to modify the values systematically upon input.

The revised data storage structure and consequences of the revision are examined in section 5. Consequences include the ability to output the state of all variables at the end of a simulation, improved capabilities for the **trace** and **snap** commands, and better access to internal variables. Unique headers are available for each output file, with software version and time at start of execution.In addition, all of the variables added to BREATH Version 1.2 are listed.

18/18

8 REFERENCES

Stothoff, S.A. 1995. *BREATH Version 1.1—Coupled Flow and Energy Transport in Porous Media: Simulator Description and User Guide*. NUREG/CR-6333. Washington, DC: Nuclear Regulatory Commission.