05
55
o.755
op.2

OR-NATURE: THE NUMERICAL ANALYSIS OF TRANSPORT AND WATER SOLUTES THROUGH SOIL AND PLANTS

VOUME III. ERROR MESSAGES AND NUMERICAL PROBLEMS

Oregon State University LIBRARY DEC 1985

Special Report 755
November 1985

Agricultual Experiment Station
Oregon State University, Corvallis

OR-NATURE: THE NUMERICAL ANALYSIS OF TRANSPORT OF WATER AND SOLUTES THROUGH SOIL AND PLANTS

VOLUME III. ERROR MESSAGES AND NUMERICAL PROBLEMS

M. J. UNGS, L. BOERSMA, S. AKRATANAKUL

This is one of five volumes about numerical analysis of water and solute transport through soil.

AUTHORS: M. J. Ungs, engineer, Tetra-Tech, Lafayette, California; L. Boersma, professor, Department of Soil Science, Oregon State University; and S. Akratanakul, professor, Department of Soil Science, Kasetsart University, Thailand.

FOREWORD

This report presents numerical solutions for the problems of transient, one-dimensional transfer of water and solutes in the layer of soil important for plant growth. It presents solutions to problems of infiltration of rainfall or irrigation water, evaporation, redistribution of water in the soil, and uptake of water and nutrients by plant roots. The numerical analysis presented in this report was prepared in response to the recognition that computer programs which deal with these problems are usually only applicable to very specific problems and are not easily generalized. This report was written on the premise that a manual should be available for the numerical analysis of these problems which can be used by persons who are not highly skilled in the use of computers. The program can be used by research workers who have a limited understanding of computer programming.

The report is presented in five volumes. The first volume gives the theoretical background of the program and should be of most interest to the research workers familiar with the mathematical analysis of the problem and computer programming. The second volume presents the manual for the use of the program. The user does not have to be familiar with, or understand, the content of the first volume in order to use the manual. A discussion of potential numerical problems and a listing of computer generated error messages is given in volume three. The fourth volume presents examples of the use of the program, and the fifth volume is a listing of the program.

ACKNOWLEDGMENTS

This publication reports work initiated under a Research Grant from the Pacific Northwest Forest and Range Experiment Station, Portland, Oregon, entitled "Physical and Chemical Factors Affecting the Transport and Distribution of Nutrient Ions in Soils Developed on Volcanic Materials" (FSPNW-GRANT Number 20). Dr. L. Boersma was project leader. Michael Ungs and Suntaree Akratanakul were graduate students at Oregon State University during the conduct of the studies reported here.

Completion of the project was supported by funds from the Oregon Agricultural Experiment Station in allotments from the Regional Research Fund established by the Hatch Act, as amended August 11, 1955 and administered by the Cooperative State Research Service, U.S. Department of Agriculture.

CONTENTS

Page

Foreword i
Acknowledgment i
List of Figures. iii
List of Tables iv

6. ERROR MESSAGES AND NUMERICAL PROBLEMS. 1
6.1 Error Messages Generated by Software. 2
6.1.1 Underflow. 2
6.1.2 End-of-File. 3
6.1.3 Negative Logarithm 3
6.2 Error Messages Generated by OR-NATURE 3
6.2.1 Inconsistent Data Parameters 4
6.2.2 Incorrect Data Arrays. 4
6.2.3 Incorrect Data Transformations 4
6.2.4 Extrapolation. 5
6.2.5 Description of Error Codes 5
6.2.6 Coded Names Used in Error Messages 26
6.2.7 Sample Output from an Error Message. 27
6.3 Failure to Converge 34
6.3.1 Definition 34
6.3.2 Reasons for Failure. 35
6.3.3 Corrections to be Made 36
6.3.4 Sample Output from a Failure to Converge 40
6.4 Stability 53
6.5 Numerical Accuracy of the Water Flow Equation 54
6.6 Numerical Accuracy of the Solute Displacement Equation. . . . 62

LIST OF FIGURES

		Page
6.1	Sample printout of error message	33
6.2	Sample printout of a failure to converge.	41
6.3	Computer output for Philip's solution at t = 2.777 hours.	56
6.4	Computer output for Philip's solution at t = 11.1Ø8 hours.	57
6.5	Computer output for Philip's solution at t = 27.77 hours.	58
6.6	Computer output for the analytical solution at t = 1 day.	65
6.7	Computer output for the analytical solution at t = 2 days.	66
6.8	Computer output for the analytical solution at t = 3 days.	67
6.9	Apparent error in the solute mass balance of program OR-NATURE in Example 7.5. Two methods of computing mass balance are shown.	71

LIST OF TABLES

		Page
6.1	Error messages.	6
6.2	A list of coded names used in error messages.	29
6.3	Number of interations allowed before failure to converge is declared.	35
6.4	Specified parameters for the computer program given by Vauclin et al. (1979) for Yolo light clay.	55
6.5	A comparison between Philip's and OR-NATURE'S numerical solutions at time = 2.777 hours.	59
6.6	A comparison between Philip's and OR-NATURE'S numerical solutions at time = 11.1Ø8 hours.	60
6.7	A comparison between Philip's and OR-NATURE'S numerical solutions at time = 27.77 hours.	60
6.8	A comparison between Philip's and OR-NATURE'S numerical solutions at time = 138.85 hours.	61
6.9	Comparing the peak concentrations computed by the analytical solution and OR-NATURE.	68
6.10	A comparison of the computational effort between the analytical solution and OR-NATURE at time = 3 days.	69

OR-NATURE: THE NUMERICAL ANALYSIS OF TRANSPORT OF WATER AND SOLUTES THROUGH SOIL AND PLANTS

VOLUME III. ERROR MESSAGES AND NUMERICAL PROBLEMS

6. ERROR MESSAGES AND NUMERICAL PROBLEMS

Various numerical difficulties may be encountered in solving the water flow equation and the solute displacement equation. Most of these numerical difficulties occur when using subroutines WATER and SOLUTE. This part of the report addresses some of these problems. Suggestions are given to help the user overcome some of the typical numerical problems that may arise in using the program. In addition, detailed descriptions of error messages printed out by the program are given.

The following error messages are discussed:

i) Error messages generated by the system software during execution of program OR-NATURE:
   - a) underflow;
   - b) end-of-file;
   - c) logarithm of a negative number.

ii) Error messages generated by program OR-NATURE:
   - a) specifying either non-existent or inconsistent data parameters;
   - b) incorrectly reading in tabulated data arrays;
   - c) specifying the wrong data transformation technique for a given data set;
   - d) extrapolation;
   - e) failure to converge.

Finally, the numerical accuracy of program OR-NATURE is discussed by comparing the finite-difference solutions against analytic solutions.

6.1 Error Messages Generated by Software

Each computer system has its own set of error message codes. If all data are correctly specified, the only execution error that might occur is that of underflow.

6.1.1 Underflow

The underflow condition occurs when the magnitude of a floating-point number is smaller than the permitted minimum. Underflow does not occur when equal numbers are subtracted.

Underflow has never occurred when running OR-NATURE on either a CDC 73 or Burroughs B69ØØ computer, but it has occurred when running the program on an IBM System/36Ø computer. The difference between these CDC, Burroughs, and IBM computers is in the size of their single-precision words. The CDC CYBER 17Ø, model 72Ø/73, computer has a floating-point minimum of about $1 \times 1Ø^{-293}$, the Burroughs B69ØØ has a floating-point minimum of $1 \times 1Ø^{-46}$ and the IBM System/36Ø has a floating-point minimum of $1 \times 1Ø^{-29}$. The CDC computer uses 6Ø bits per word with 14 significant digits, the B69ØØ uses 48 bits per word with 11 significant digits whereas, the IBM computer uses 36 bits per word with 8 significant digits.

The underflow "trap" can be turned off on some IBM FORTRAN compilers by calling special subroutine functions called either NOUNDF or ERRSET. Other computers may use different names. During execution of a program, subroutines NOUNDF or ERRSET will automatically set small "underflow" numbers to zero and will allow execution to continue. For example, if subroutine NOUNDF is available and underflow becomes a problem, insert the following statement, CALL NOUNDF, after the COMMON statements in program NATURE. This CALL statement should be located so that it becomes the first executable statement in program OR-NATURE. Users should consult with computer center personnel to determine which underflow statements are available for their particular computer system.

The underflow message is only a nuisance and does not indicate any program error. There is no easy way of eliminating the underflow error message except by directly suppressing it. Program OR-NATURE could be rewritten in double-precision, but this would substantially increase the required core size of the program.

6.1.2 End-Of-File

An end-of-file error message indicates that the program was reading input data when an end-of-file card occurred. This is caused by either inserting the wrong number of data cards or by incorrectly specifying the value of NUM(...) on one of the data set cards (Vol. II, Section 4.7). The end-of-file error is caused by incorrect data input.

6.1.3 Negative Logarithm

Negative logarithm errors occur when a semi-log or log-log data transform is specified for a data set in which a change in sign occurs. Program OR-NATURE is designed to handle log transforms of either positive or negative quantities, but not data that range from negative to positive values. The data transform to the base $\log_e$ (Naperian) is always done using absolute values.

Negative logarithm errors can also occur when a non-linear sorption model of the form $S = c^k$ is specified and the liquid phase solute concentration, c, goes negative. By rewriting the above model, $S = \exp\{k \cdot \ln(c)\}$ it becomes obvious that the solute concentration must have a positive value. The liquid phase solute concentration only goes negative when oscillations occur in the numerical solution of the solute displacement equation. To avoid negative logarithm errors, program OR-NATURE sets S equal to zero whenever c is less than or equal to zero.

6.2 Error Messages Generated by OR-NATURE

Program OR-NATURE contains one hundred diagnostic statements that check if there are any errors in the input data and in the operation of the program. The most exhaustive error analysis of the input data is obtained by setting program parameters XORDER = 1 and SCAN = 1 on program card #1. The error messages or error codes will indicate in which subroutine the error was diagnosed and which corrective action is necessary. A description of the error codes is given in Section 6.2.5.

6.2.1 Inconsistent Data Parameters

It is important to specify the program parameters correctly (Section 4.5.3 and Table 4.4 of Vol. II). The data parameters can take on only certain values, must be of a certain type (integer, floating-point), and must be consistent with each other. For example, the parameter GRAV can only take on the floating-point values of either Ø. or 1. In addition, when MODE = 1, then JSORB must be set to zero since sorption occurs only with the solute equation. The computer program will give an error message whenever a parameter is incorrectly specified.

6.2.2 Incorrect Data Arrays

Care must be taken in reading in data arrays. Particular attention is needed to make sure that the array specifying the independent variable, X(...), is in correct order. The values of X_1, X_2, ..., X_{NUM} must increase monotonically, i.e., $X_2 > X_1$, $X_3 > X_2$, etc. and $X_2 \neq X_1$, $X_3 \neq X_2$, etc. In most data sets, the independent and dependent array elements are read in as sequential pairs, such as X_1, Y_1, X_2, Y_2, X_3, Y_3, etc. (Fig. 4.6 and Table 4.5 of Vol. II). However, there are two exceptions, such as for data sets ZØ and ZF in which the array elements are fed in sequentially, e.g., Z_1, Z_2, Z_3, etc. For data written in either E or I format, the value must be right-justified when punched onto data cards.

6.2.3 Incorrect Data Transformations

Log transformations can be specified for either positive or negative values since the program uses the absolute value for transformations. However, a log transform cannot be specified for a data set containing a zero value or a data set in which the range of values changes sign, e.g., X(...) = -2, -1, +1, +1Ø. The type of interpolation and data transformation to be used on a particular data set are specified through the parameter TBL(...). The following values of TBL(...) indicate either a semi-log or log-log data transform, TBL(...) = 4, 5, 6, 7, 8, 9 (Vol. II, Section 4.8).

6.2.4 Extrapolation

The computer program does not allow extrapolation outside the range of any data set, only interpolation is allowed. Extrapolation can be completely avoided by extending the range of values of the data set. If extrapolation is attempted, the program will list the X(...) and Y(...) values of the data set, indicate the value of x at which extrapolation was tried, and then abort.

6.2.5 Description of Error Codes

When program OR-NATURE detects an error in the input data or if an error occurs during execution of the program, any one of one hundred error codes will be printed out. When the program prints out an error message, it lists the most recent values of the pertinent variables associated with the error. In addition, the program presents a brief diagnosis of the cause of the problem and suggests a possible correction. It must be realized that any given error can have many causes but that the program suggests only the most likely cause.

A detailed description of each error message is given in Table 6.1. If the error is concerned with a data set, a coded name of the data set is listed within the error message. Section 6.2.6 lists all coded names used. In Table 6.1, the symbols EXP, S.A., and U.A. are used. A brief description of each follows.

EXP: Explanation of why the error occurred.

S.A.: System Action--action taken by program OR-NATURE. A fatal error means that the program will abort after listing the error message. The name of the subroutine in which the error is detected is given.

U.A.: User Action--suggested action to be taken by the user to eliminate the error.

Table 6.1 Error Messages

Error Code	Description/Action
1	EXP - The volumetric water content specified for air-dry conditions is negative and/or the specified saturated condition is less than or equal to zero. S.A. - Fatal error occurred in program NATURE. U.A. - Respecify the value of either DRY and/or SAT on program card #3.
2	EXP - The range of data set PW is too limited. The value of DRY is either less than XPW(1) and/or SAT is greater than XPW(NUMPW). S.A. - Fatal error occurred in program NATURE. U.A. - Increase the range of matric potential as a function of water content in data set PW so that it encompasses the values of DRY and SAT.
3	EXP - The matric potential specified for air-dry conditions is greater than or equal to zero. S.A. - Fatal error occurred in program NATURE. U.A. - The matric potential at air-dry conditions must be a negative quantity. Respecify the value of DRY on program card #3.
4	EXP - The matric potential specified for saturated conditions is greater than zero. S.A. - Fatal error occurred in program NATURE. U.A. - The matric potential at saturation must be less than or equal to zero. Respecify the value of SAT on program card #3.
5	EXP - The range of data set WP is too limited. The value of DRY is less than XWP(1). S.A. - Fatal error occurred in program NATURE. U.A. - Increase the range of water content as a function of matric potential in data set WP so that it encompasses the value of DRY.
6	EXP - The range of data set WP is too limited. The value of SAT is greater than XWP(NUMWP). S.A. - Fatal error occurred in program NATURE. U.A. - Increase the range of water content as a function of matric potential in data set WP so that it encompasses the value of SAT.

Table 6.1, continued.

Error Code	Description/Action
7	EXP - Data set WP, water content as a function of matric potential, $\theta(\psi)$, must be specified when solving the water flow equation in terms of matric potential. S.A. - Fatal error occurred in subroutine ERROR. U.A. - Specify data set WP and specify TBLWP greater than or equal to zero.
8	EXP - Data set PW, matric potential as a function of water content, $\psi(\theta)$, must be specified when solving the water flow equation in terms of volumetric water content. S.A. - Fatal error occurred in subroutine ERROR. U.A. - Specify data set PW and specify TBLPW greater than or equal to zero.
9	EXP - When solving the water flow equation as a function of volumetric water content (JWORH = Ø) or when solving the solute displacement equation by itself (MODE = 2), you forgot to specify the initial water content, θ_0, of the soil. S.A. - Fatal error occurred in subroutine ERROR. U.A. - Specify data set WØ and specify TBLWØ greater than or equal to zero.
1Ø	EXP - When solving the water flow equation as a function of matric potential, you forgot to specify the initial potential, ψ_0, of the soil. S.A. - Fatal error occurred in subroutine ERROR. U.A. - Specify data set PØ and specify TBLPØ greater than or equal to zero.
11	EXP - When solving the solute displacement equation with a non-equilibrium model of sorption, you forgot to specify the initial concentration of the solid phase, S_0, in the soil. S.A. - Fatal error occurred in subroutine ERROR. U.A. - Specify data set SØ and specify TBLSØ greater than or equal to zero.

Table 6.1, continued.

Error Code	Description/Action
12	EXP - When solving the solute displacement equation, you forgot to specify the initial concentration of the liquid phase, c_0, in the soil. S.A. - Fatal error occurred in subroutine ERROR. U.A. - Specify data set CØ and specify TBLCØ greater than or equal to zero.
13	EXP - When solving the solute displacement equation, you forgot to specify the "type" of boundary condition, BCBTMC, that will be used on the lower boundary, $z = z_{max}$, of the soil column. S.A. - Fatal error occurred in subroutine ERROR. U.A. - Specify data set BCB and specify TBLBCB greater than or equal to zero.
14	EXP - When solving the solute displacement equation, you forgot to specify the "type" of boundary condition, BCTOPC, that will be used on the upper boundary, z = Ø, of the soil column. S.A. - Fatal error occurred in subroutine ERROR. U.A. - Specify data set BCT and specify TBLBCT greater than or equal to zero.
15	EXP - When solving the solute displacement equation, you forgot to specify the apparent dispersion coefficient, D_{sz}. S.A. - Fatal error occurred in subroutine ERROR. U.A. - Specify data set DSZ and specify TBLDSZ greater than or equal to zero.
16	EXP - When solving the solute displacement equation, you forgot to specify the value on the right-hand side of the lower boundary condition, FBTMC. S.A. - Fatal error occurred in subroutine ERROR. U.A. - Specify data set FCB and specify TBLFCB greater than or equal to zero.
17	EXP - When solving the solute displacement equation, you forgot to specify the value on the right-hand side of the upper boundary condition, FTOPC. S.A. - Fatal error occurred in subroutine ERROR. U.A. - Specify data set FCT and specify TBLFCT greater than or equal to zero.

Table 6.1, continued.

Error Code	Description/Action
18	EXP - When solving the water flow equation, you forgot to specify the "type" of boundary condition, BCBTMW, that will be used on the lower boundary, $z = z_{max}$. S.A. - Fatal error occurred in subroutine ERROR. U.A. - Specify data set BWB and specify TBLBWB greater than or equal to zero.
19	EXP - When solving the water flow equation, you forgot to specify the "type" of boundary condition, BCTOPW, that will be used on the upper boundary, $z = \emptyset$. S.A. - Fatal error occurred in subroutine ERROR. U.A. - Specify data set BWT and specify TBLBWT greater than or equal to zero.
2Ø	EXP - When solving the water flow equation as a function of volumetric water content (JWORH = Ø) or when solving the solute displacement equation by itself (MODE = 2), you forgot to specify the soil-water diffusivity coefficient, $D(\theta)$. S.A. - Fatal error occurred in subroutine ERROR. U.A. - Specify data set DWZ and specify TBLDWZ greater than or equal to zero.
21	EXP - When solving the water flow equation, you forgot to specify the value on the right-hand side of the lower boundary condition, FBTMW. S.A. - Fatal error occurred in subroutine ERROR. U.A. - Specify data set FWB and specify TBLFWB greater than or equal to zero.
22	EXP - When solving the water flow equation, you forgot to specify the value on the right-hand side of the upper boundary condition, FTOPW. S.A. - Fatal error occurred in subroutine ERROR. U.A. - Specify data set FWT and specify TBLFWT greater than or equal to zero.

Table 6.1, continued.

Error Code	Description/Action
23	EXP - You forgot to specify the soil water hydraulic conductivity, K(θ). This must be specified at all times, even if you are only solving the solute displacement equation. S.A. - Fatal error occurred in subroutine ERROR. U.A. - Specify data set KWZ and specify TBLKWZ greater than or equal to zero.
24	EXP - You forgot to specify the initial spatial locations of the nodes, z_i, i = 1,...,NUMZØ. S.A. - Fatal error occurred in subroutine ERROR. U.A. - Specify data set ZØ and specify TBLZØ greater than or equal to zero.
25	EXP - The value of TGRID was specified to be less than TIMEØ. However, if you wish to regrid or relocate the spatial locations of the nodes at time equal to TGRID, then TGRID must be larger than TIMEØ. S.A. - Fatal error occurred in subroutine ERROR. U.A. - Check to make sure that you want to regrid the nodes. If not, set TBLZF = -1, otherwise specify TGRID to be greater than or equal to TIMEØ on program card #5.
26	EXP - When solving only the water flow equation, MODE = 1, you cannot specify any sorption models. Only the solute displacement equation has a sorption term. S.A. - Fatal error occurred in subroutine ERROR. U.A. - Set JSORB = Ø on program card #1.
27	EXP - The parameter that converts the unit of matric potential to that of pressure head, CNVRSN, must have a value greater than zero. You specified a value less than or equal to zero. S.A. - Fatal error occurred in subroutine ERROR. U.A. - Check program card #3 for the value of CNVRSN. If you are not solving the water flow equation, you must still specify a value. If there is no conversion, set CNVRSN equal to one.

Table 6.1, continued.

Error Code	Description/Action
28	EXP - The START parameter must have an integer value greater than or equal to minus one. S.A. - Fatal error occurred in subroutine ERROR. U.A. - Check program card #2 and set START greater than or equal to minus one.
29	EXP - The START option, i.e., Kirchhoff transform, can only be used when solving the water flow equation. If MODE equals two, then START must equal zero. S.A. - Fatal error occurred in subroutine ERROR. U.A. - Check program card #2 and set START equal to zero.
3Ø	EXP - When solving the solute displacement equation all by itself, you must still specify the volumetric water content. If MODE equals two, then JWORH must equal zero. S.A. - Fatal error occurred in subroutine ERROR. U.A. - Check program card #1 and set JWORH equal to zero. Also make sure that data sets WØ, KWZ, and DWZ are also correctly specified.
31	EXP - The parameter JWORH can only be given the integer value of either zero or one. Any other value is not allowed. S.A. - Fatal error occurred in subroutine ERROR. U.A. - Check program card #1 and respecify the value of JWORH.
32	EXP - The plant potential at wilting is a negative, non-zero valued quantity. If transpiration is to be modeled, PWILT must be set to a value less than zero. S.A. - Fatal error occurred in subroutine ERROR. U.A. - Check program card #3. Set PWILT to a value less than zero. If transpiration is not being modeled, set TBLQPS equal to minus one.
33	EXP - The parameter BEST can only have the integer value of either zero or one. S.A. - Fatal error occurred in subroutine ERROR. U.A. - Check program card #2. BEST can only have the integer value of either zero or one.

Table 6.1, continued.

Error Code	Description/Action
34	EXP - When solving the solute displacement equation by itself, i.e., MODE equals two, you must still define the soil water diffusivity function, D(θ). S.A. - Fatal error occurred in subroutine ERROR. U.A. - Specify data set DWZ and specify TBLDWZ greater than or equal to zero.
35	EXP - When solving the solute displacement equation by itself, i.e., MODE equals two, you must still define the soil water hydraulic conductivity function, K(θ). S.A. - Fatal error occurred in subroutine ERROR. U.A. - Specify data set KWZ and specify TBLKWZ greater than or equal to zero.
36	EXP - The arrays associated with data sets BCB, BCT, BWB, and BWT are dimensioned to a value of MAXSET. If larger arrays are desired, redimension all of the COMMON statements to accommodate the largest of these arrays. S.A. - Fatal error occurred in subroutine ERROR. U.A. - Either eliminate some of the data or redimension the size of the COMMON statements for data sets BCB, BCT, BWB, and BWT. Also reset the value of MAXSET in program NATURE if the COMMON statements are redimensioned.
37	EXP - The arrays associated with data sets CST, DSZ, and DWZ are dimensioned to the value of MAXSET. If larger arrays are desired, redimension all of the COMMON statements to accommodate the largest of these arrays. S.A. - Fatal error occurred in subroutine ERROR. U.A. - Either eliminate some of the data or redimension the size of the COMMON statements for data sets CST, DSZ, and DWZ. Also reset the value of MAXSET in program NATURE if the COMMON statements are redimensioned.
38	EXP - The arrays associated with data sets FCB, FWB, KWZ, and PW are dimensioned to a value of MAXSET. If larger arrays are desired, redimension all the COMMON statements to accommodate the largest of these arrays. S.A. - Fatal error occurred in subroutine ERROR. U.A. - Either eliminate some of the data or redimension the size of the COMMON statements for data sets FCB, FWB, KWZ, and PW. Also reset the value of MAXSET in program NATURE if the COMMON statements are redimensioned.

Table 6.1, continued.

Error Code	Description/Action
39	EXP - The arrays associated with data sets RAC, RAW, RD, and WP are dimensioned to a value of MAXSET. If larger arrays are desired, redimension all of the COMMON statements to accommodate the largest of these arrays. S.A. - Fatal error occurred in subroutine ERROR. U.A. - Either eliminate some of the data or redimension the size of the COMMON statements for data sets RAC, RAW, RD, and WP. Also reset the value of MAXSET in program NATURE if the COMMON statements are redimensioned.
4Ø	EXP - The arrays associated with data sets CG, CTX, and RPF are dimensioned to a value of MAXSET. If larger arrays are desired, redimension all of the COMMON statements to accommodate the largest of these arrays. S.A. - Fatal error occurred in subroutine ERROR. U.A. - Either eliminate some of the data or redimension the size of the COMMON statements for data sets CG, CTX, and RPF. Also reset the value of MAXSET in program NATURE if the common statements are redimensioned.
41	EXP - The arrays associated with data sets CØ, PØ, SØ, WØ, ZØ, and ZF are dimensioned to a value of NUMMAX. If larger arrays are desired, redimension all of the COMMON statements to accommodate the largest of these arrays. S.A. - Fatal error occurred in subroutine ERROR. U.A. - Either eliminate some of the data or redimension the size of the COMMON statements for data sets CØ, PØ, SØ, WØ, ZØ, and ZF. Also reset the value of NUMMAX in program NATURE if the COMMON statements are redimensioned.
42	EXP - The arrays associated with data sets FCT, FWT, and QPS are dimensioned to a value of NUMMAX. If larger arrays are desired, redimension all of the COMMON statements to accommodate the largest of these arrays. S.A. - Fatal error occurred in subroutine ERROR. U.A. - Either eliminate some of the data or redimension the size of the COMMON statements for data sets FCT, FWT, and QPS. Also reset the value of NUMMAX in program NATURE if the COMMON statements are redimensioned.

Table 6.1, continued.

Error Code	Description/Action
43	EXP - The GRAV parameter can only have either the value of zero or the value of one. No other value is allowed. S.A. - Fatal error occurred in subroutine ERROR. U.A. - Check program card #3. Set GRAV equal to zero if only horizontal flow is considered or if gravity is considered to have negligible effect on Darcian flux. Otherwise set GRAV equal to one.
44	EXP - The parameters TIMEF, DTMIN, and DTMAX must always be greater than zero. The parameter TIMEØ must always be greater than or equal to zero. S.A. - Fatal error occurred in subroutine ERROR. U.A. - Correctly specify the parameters TIMEØ, TIMEF, DTMIN, and DTMAX on program card #5.
45	EXP - The final time, TIMEF, must be greater than or equal to the initial time, TIMEØ. S.A. - Fatal error occurred in subroutine ERROR. U.A. - Make the value of TIMEF greater than or equal to the value of TIMEØ on program card #5.
46	EXP - The maximum duration allowed for the time increment, DTMAX, must be greater than or equal to the minimum duration allowed for the time increment, DTMIN. S.A. - Fatal error occurred in subroutine ERROR. U.A. - Make the value of DTMAX greater than or equal to the value of DTMIN on program card #5.
47	EXP - The acceleration parameter that is used to make initial extrapolation guesses, ACLRAT, must be greater than zero. Normally, one sets ACLRAT equal to one. S.A. - Fatal error occurred in subroutine ERROR. U.A. - Make the value of ACLRAT greater than zero on parameter card #3.
48	EXP - The maximum number of iterations allowed, ITRMAX, in any given iterator scheme must be greater than zero. S.A. - Fatal error occurred in subroutine ERROR. U.A. - Make the value of ITRMAX larger than zero on program card #3.

Table 6.1, continued.

Error Code	Description/Action
49	EXP - All of the convergence ratios, ARATIO, QRATIO, SRATIO, and WRATIO, must have values greater than or equal to zero, even if a particular ratio is not used. S.A. - Fatal error occurred in subroutine ERROR. U.A. - Make the values of ARATIO, QRATIO, SRATIO, and WRATIO greater than or equal to zero on program card #4.
5Ø	EXP - The parameter K1DELT can only have the integer value of either zero, one, or two. S.A. - Fatal error occurred in subroutine ERROR. U.A. - Set K1DELT equal to either zero, one, or two on program card #1.
51	EXP - The time increment coefficients ATCOEF, BTCOEF, and CTCOEF must have values greater than or equal to zero when K1DELT equals one. For the (k+1) time step, the general formula is DTK1 = ATCOEF + BTCOEF * DTK + CTCOEF * TIME (6.1) S.A. - Fatal error occurred in subroutine ERROR. U.A. - Make each of the coefficients ATCOEF, BTCOEF, and CTCOEF greater than or equal to zero on program card #7. If the user wants the time-step size to decrease with time, use the K1DELT=2 option.
52	EXP - The time increment coefficients ATCOEF, BTCOEF, and CTCOEF cannot all be simultaneously be zero when K1DELT equals one. S.A. - Fatal error occurred in subroutine ERROR. U.A. - Make at least one of the coefficients ATCOEF, BTCOEF, and CTCOEF greater than zero on program card #7 when K1DELT equals one.
53	EXP - When specifying the final node spacings, ZF, there must be at least five nodes in the soil column. S.A. - Fatal error occurred in subroutine ERROR. U.A. - Check data set ZF and make sure that there are at least five nodes specified over the soil column.

Table 6.1, continued.

Error Code	Description/Action
54	EXP - When the top two nodes of the soil column become air-dry, it is impossible for the program to satisfy any evaporative surface flux condition. S.A. - Fatal error occurred in subroutine WATER. U.A. - Either make the upper surface flux of water greater than or equal to zero or increase the initial water content of the soil column.
55	EXP - The algorithm to compute the water content failed to converge after ITRMAX iterations. S.A. - Warning occurred in subroutine WATER. For the next 25 iterations, the program will print out the node at which the worst guess of water content occurred. It is possible for the program to recover. U.A. - A careful examination of what the program prints out will often pinpoint the cause of this failure to converge, see Section 6.3.
56	EXP - The algorithm to compute the water content failed to converge after ITRMAX plus 25 iterations. S.A. - Fatal error occurred in subroutine WATER. The final values of 28 variables are printed out. U.A. A careful examination of the printout may indicate the cause of the failure to converge (Section 6.3).
57	EXP - The algorithm to compute the plant surface potential, ψ_{ps}, failed to converge after ITRMAX iterations. S.A. - Fatal error occurred in subroutine WATER. The program will print out the last computed value of plant surface potential and the last estimated value of plant surface potential. U.A. - Normally more iterations, ITRMAX, need to be specified, or the convergence criterion ARATIO needs to be increased.
58	EXP - The rate of plant transpiration, TR, must be less than or equal to zero. S.A. - Fatal error occurred in subroutine WATER. U.A. - The last computed values of plant surface potential and transpiration are printed out. Make sure that the atmospheric demand term, QPS, is correctly specified and is negative. Also check to make sure that the stomatal efficiency term, CST, is greater than or equal to zero, where TR = QPS*CST.

Table 6.1, continued.

Error Code	Description/Action
59	EXP - There are only three "types" of boundary conditions allowed at the upper boundary of the water flow equation. The parameter BCTOPW can only have the values of one, two, and four. S.A. - Fatal error occurred in subroutine WATER. U.A. - Check data set BWT to make sure that only a first, second, or fourth type boundary condition is specified.
6∅	EXP - There are only three "types" of boundary conditions allowed at the lower boundary of the water flow equation. The parameter BCBTMW can only have the values of one, two, and four. S.A. - Fatal error occurred in subroutine WATER. U.A. - Check data set BWB to make sure that only a first, second, or fourth type boundary condition is specified.
61	EXP - The algorithm to compute the upper surface flux of water is failing to converge. S.A. - Warning occurred in subroutine WATER. For the next 25 iterations, the program will print out the values of 14 different variables that are used in the iterator routine. It is possible for the program to recover. U.A. - A careful examination of what the program prints out may pinpoint the cause of this failure to converge (Section 6.3).
62	EXP - The algorithm to compute the upper surface flux of water failed to converge after an additional 25 iterations. S.A. - Fatal error occurred in subroutine WATER. The final values of 41 variables are printed out. U.A. - A careful examination of the printout may pinpoint the cause of this failure to converge (Section 6.3).
63	EXP - The algorithm to compute the sorbed or solid phase solute concentration is failing to converge after ITRMAX iterations. S.A. - Warning occurred in subroutine SOLUTE. For the next 25 iterations, the program will print out the values of eight different variables that are used in the iterator routine. It is possible for the program to recover. U.A. - A careful examination of what the program prints out may pinpoint the cause of this failure to converge.

Table 6.1, continued.

Error Code	Description/Action
64	EXP - The algorithm to compute the sorbed or solid phase solute concentration failed to converge after an additional 25 iterations. S.A. - Fatal error occurred in subroutine SOLUTE. The final values of 17 variables are printed out. U.A. - A careful examination of the printout may pinpoint the cause of this failure to converge.
65	EXP - There are only three "types" of boundary conditions allowed at the upper boundary of the solute displacement equation. The parameter BCTOPC can only have the value of either one, two, or three. S.A. - Fatal error occurred in subroutine SOLUTE. U.A. - Check data set BCT to make sure that only a first, second, or third type boundary condition is specified.
66	EXP - There are only three "types" of boundary conditions allowed at the lower boundary of the solute displacement equation. The parameter BCBTMC can only have the values of one, two, and three. S.A. - Fatal error occurred in subroutine SOLUTE. U.A. - Check data set BCB to make sure that only a first, second, or third type boundary condition is specified.
67	EXP - The algorithm to compute the sorbed solute concentration from a general non-equilibrium, non-linear, isotherm, JSORB=9, failed to converge after ITRMAX iterations. S.A. - Fatal error occurred in subroutine SOLUTE. U.A. - The sorption model was probably not programmed correctly between statements 9Ø4 to 9Ø5 in subroutine SOLUTE. If all else fails, linearize the sorption model or make it less complicated.
68	EXP - If X and Y are tabulated in one-dimensional arrays, and if the values of X are the independent variables, Y(X), then the values of array X must be in ascending order. S.A. - Fatal error occurred in subroutine ORDER but the name of the data set that caused the error is also printed out. U.A. - Make sure that X(2) is greater than X(1), that X(3) is greater than X(2), etc. Rearrange the data until X(i) increases monotonically, i = 1,...,NUM. Note that X(i) cannot equal X(i-1), i = 2,...,NUM.

Table 6.1, continued.

Error Code	Description/Action
69	EXP - All data set arrays must have at least two values in them. S.A. - Fatal error occurred in subroutine ORDER but the name of the data array that caused the error is also printed out. U.A. - Make sure this array was specified correctly and, if not, add more values to the X and Y arrays.
7Ø	EXP - When specifying the final node spacings, ZF, there must be at least five nodes in the soil column. S.A. - Fatal error occurred in subroutine REGRID. U.A. - Check data set ZF and make sure that there are at least five nodes specified over the soil column.
71	EXP - When changing the nodal spacing from ZØ to ZF, the last node must be the same distance as before, i.e., ZØ(NUMZØ) = ZF(NUMZF). (6.2) S.A. - Fatal error occurred in subroutine REGRID. U.A. - Make the spatial location of the last node of data set ZF equal to the spatial location of the last node of data set ZØ. Note that the number of nodes does not have to be the same, only the length of the soil column.
72	EXP - When integrating function Y as a function of the independent variable X, i.e., Y(X), the tabulated values of X must increase monotonically. S.A. - Fatal error occurred in subroutine INTGRT. A listing of the improperly specified data set is given along with the name of the data set. U.A. - Make sure the independent variable is correctly specified and in proper order, i.e., X(2) is greater than X(1), X(3) is greater than X(2), etc.

Table 6.1, continued.

Error Code	Description/Action
73	EXP - Only two methods of integration are allowed in subroutine INTGRT, namely Type one and Type two. Type one integration is a piecewise-constant integration and Type two integration is based on the integration of a function that has been expanded by a second-order Taylor series. Either method will integrate over unequally or equally spaced values. S.A. - Fatal error occurred in subroutine INTGRT. A listing of the data set being integrated is given along with the name of the data set. U.A. - The parameter TYPE can only have an integer value of either one or two. If the data is piecewise-constant, set TYPE equal to one, otherwise set TYPE equal to two.
74	EXP - In order to integrate a data set, there must be at least two data points specified. S.A. - Fatal error occurred in subroutine INTGRT. The name of the data set that failed is given. U.A. - Specify at least two data points for this data set for it to be integrated.
75	EXP - Extrapolation outside the range of a data set is not allowed. The range of the independent variable X is defined in the data set as X(1) and X(NUM). To evaluate the dependent variable Y at point x_i, $Y(x_i)$, without extrapolation, then x_i must lie between or at X(1) and X(NUM). S.A. - Fatal error occurred in subroutine INTRP. The name of the data set that caused the error is printed out. U.A. - Extend the range of values for the independent variable, X, of this data set so that it encompasses the value of x_i.
76	EXP - When the parameter TBL equals zero, an empirical formula is to be used to evaluate a given function. There is no reason for subroutine INTRP to be called. S.A. - Fatal error occurred in subroutine INTRP. The name of the data set in which the problem occurred is printed out. U.A. - If an empirical formula is already written, then don't call subroutine INTRP. If there is a tabulated array, then specify the parameter TBL with an integer value between one and 10, inclusively.

Table 6.1, continued.

Error Code	Description/Action
77	EXP - When the parameter TBL is set equal to minus one, then the corresponding data set is not used by the program. Thus, there is no need to call subroutine INTRP. S.A. - Fatal error in subroutine INTRP. The name of the defective data set is printed out. U.A. - Make sure that parameter TBL was correctly specified. If TBL is supposed to equal minus one, then don't call subroutine INTRP.
78	EXP - Parameter TBL must have an integer value between minus one and 10, inclusive. If any other interpolation routine is desired, the user must modify subroutine INTRP. S.A. - Fatal error occurred in subroutine INTRP. U.A. - Make sure parameter TBL was correctly specified. If it was, then modify subroutine INTRP to handle this new interpolation procedure.
79	EXP - For linear interpolation, at least two data points are needed in the data set. S.A. - Fatal error occurred in subroutine INTRP. U.A. - Specify more data points for this data set.
8Ø	EXP - For cubic spline interpolation, at least three data points are needed in the data set. S.A. - Fatal error occurred in subroutine INTRP. U.A. - Specify more data points for this data set.
81	EXP - For quadratic interpolation, at least three data points are needed in the data set. S.A. - Fatal error occurred in subroutine INTRP. U.A. - Specify more data points for this data set.
82	EXP - When specifying a logarithmic transform of the independent variable X, it is not possible to interpolate at the point x_i equal to zero. S.A. - Fatal error occurred in subroutine LINEAR. U.A. - If you want the point x_i equal to zero to be used by the interpolation routine, then specify a non-logarithmic transform for the independent variable X of this data set.

Table 6.1, continued.

Error Code	Description/Action
83	EXP - When specifying a logarithmic transform of the dependent variable Y, it is not possible for one of the values of Y to equal zero. S.A. - Fatal error occurred in subroutine LINEAR. U.A. - If the value of zero must be included in the range of variable Y, then specify a non-logarithmic transform for the dependent variable Y of this data set.
84	EXP - When specifiying a logarithmic transform of the independent variable X, it is not possible for one of the values of X to equal zero. S.A. - Fatal error occurred in subroutine LINEAR. U.A. - If the value of zero must be included in the range of variable X, then specify a non-logarithmic transform for the independent variable X of this data set.
85	EXP - The values of the independent variable X must be unique and arranged in ascending order. S.A. - Fatal error occurred in subroutine QUAD. U.A. - Make sure that X(2) is greater than X(1), that X(3) is greater than X(2), etc. Rearrange data set values until X(i) increases monotonically, i = 1,...,NUM. Note that X(i) cannot equal X(i-1), i = 2,...,NUM.
86	EXP - For quadratic interpolation, at least three data points are needed in the data set. S.A. - Fatal error occurred in subroutine QUAD. U.A. - Specify more data points for this data set.
87	EXP - The values of the independent variable X must be unique and arranged in ascending order. S.A. - Fatal error occurred in subroutine CUBIC. U.A. - Make sure that X(2) is greater than X(1), that X(3) is greater than X(2), etc. Rearrange data set values until X(i) increases monotonically, i = 1,...,NUM. Note that X(i) cannot equal X(i-1), i = 2,...,NUM.

Table 6.1, continued.

Error Code	Description/Action
88	EXP - When specifying a logarithmic transform of the dependent variable Y, it is not possible for one of the values of Y to equal zero. S.A. - Fatal error occurred in subroutine CUBIC. U.A. - If the value of zero must be included in the range of variable Y, then specify a non-logarithmic transform for the dependent variable Y of this data set.
89	EXP - When specifying a logarithmic transform of the independent variable X, it is not possible for one of the values of X to equal zero. S.A. - Fatal error occurred in subroutine CUBIC. U.A. - If the value of zero must be included in the range of variable X, then specify a non-logarithmic transform for the independent variable X of this data set.
9Ø	EXP - The values of the independent variable X must be unique and arranged in ascending order. S.A. - Fatal error occurred in subroutine LINEAR. U.A. - Make sure that X(2) is greater than X(1), that X(3) is greater than X(2), etc. Rearrange data set values until X(i) increases monotonically, i = 1,...,NUM. Note that X(i) cannot equal X(i-1), i = 2,...,NUM.
91	EXP - When specifying the initial node spacings, ZØ, there must be at least five nodes in the soil column. S.A. - Fatal error occurred in subroutine ERROR. U.A. - Check data set ZØ and make sure that there are at least five nodes specified over the soil column.
92	EXP - Parameter SAVE can only have the integer values of zero and one. S.A. - Fatal error occurred in subroutine ERROR. U.A. - Correctly specify parameter SAVE with a value of either zero or one on program card #2.

Table 6.1, continued.

Error Code	Description/Action
93	EXP - Parameter OUTPUT can only have the integer value of either zero, one, or two. S.A. - Fatal error occurred in subroutine ERROR. U.A. - Correctly specify parameter OUTPUT with a value of either zero, one, or two on program card #1.
94	EXP - Parameter SCAN can only have the integer values of zero and one. S.A. - Fatal error occurred in subroutine ERROR. U.A. - Correctly specify parameter SCAN with a value of either zero or one on program card #2.
95	EXP - Parameter XORDER can only have the integer values of zero and one. S.A. - Fatal error occurred in subroutine ERROR. U.A. - Correctly specify parameter XORDER with a value of either zero or one on program card #2.
96	EXP - Parameter MODE can only have the integer values of zero, one, and two. S.A. - Fatal error occurred in subroutine ERROR. U.A. - Correctly specify parameter MODE with a value of either zero, one, or two on program card #1.
97	EXP - The internally dimensioned size of arrays Z, RHO, and TAU is too small. Increase their dimensioned size in subroutine CUBIC. S.A. - Fatal error occurred in subroutine CUBIC. The program prints out the dimension size that it needs, NUM. U.A. - Increase the dimensioned size of arrays Z, RHO, and TAU to at least that of Z(NUM), RHO(NUM), and TAU(NUM) in subroutine CUBIC. Also set parameter NUMSET equal to the newly dimensioned size used. Note that array Z in subroutine CUBIC is equal to the slope of the cubic function and is unrelated to array Z, i.e., the space coordinates, that is used in the rest of the program.

Table 6.1, continued.

Error Code	Description/Action
98	EXP - Either the program is attempting to extrapolate outside the range of data set X or the values of array X, i.e., the independent variable, are not properly arranged in monotonically increasing value. S.A. - Fatal error occurred in subroutine LINEAR. U.A. - Rearrange the values of data set X in monotonically increasing value.
99	EXP - Either the program is attempting to extrapolate outside the range of the data set X or the values of array X, i.e., the independent variable, are not properly arranged in monotonically increasing value. S.A. - Fatal error occurred in subroutine QUAD. U.A. - Rearrange data set X in monotonically increasing value.
1ØØ	EXP - Either the program is attempting to extrapolate outside the range of the data set X or the values of array X, i.e., the independent variable, are not properly arranged in monotonically increasing value. S.A. - Fatal error occurred in subroutine CUBIC. U.A. - Rearrange data set X in monotonically increasing value.

6.2.6 Coded Names Used In Error Messages

In about 25 of the error messages given in Section 6.2.5, a coded name is printed out along with other information pertinent to the analysis of the error message. A coded name is defined as a string of three characters. Any group of these alpha-numeric characters can be used, for example CZ1, WP, WZ3, etc. Blanks are considered as characters. In an error message, the coded name is usually embedded within other words, such as NUM(...), TBL(...), X(...), Y(...), where the ellipses (...) are replaced by the coded name. Thus the coded name CZ1 may be printed out as NUMCZ1, TBLCZ1, XCZ1, or YCZ1.

A coded name is used to identify which set of data was being analyzed when an error was detected. Subroutines INTRP, INTGRT, ORDER, and TABLED may utilize any one or all 60 sets of data used by program OR-NATURE. The "NAME" of the data set is passed, in addition to the values of the X and Y arrays when these subroutines are called.

A list of 128 coded names is given in Table 6.2. For each coded name, a brief comment or description is given of the data set. Only four types of comments are given: 1) interpolate array Y versus array X, where X is the independent variable and Y is the dependent variable; 2) integrate YdX; 3) check the order of array X so that all values of the array are arranged in increasing value; and 4) tabulate arrays X and Y with three additional values between each array entry by interpolation. In addition to the comments, Table 6.2 gives the name of the CALL statement that passed the defective data, the name of the subroutine in which the CALL statement originated and the identification number or card number on which the CALL statement occurred. The identification number refers to Versions 1.1 and 1.2 of OR-NATURE.

The coded name precisely defines the data set that contains the defective data. Upon examination of the accompanying error message that is printed out at the time of the error, the user can determine in which subroutine the error was detected. Upon examination of Table 6.2, the user can then determine the exact card from which the error originated in the program.

6.2.7 Sample Output From An Error Message

A typical example of the printed output from an error message is given in Figure 6.1. Ten points of discussion are marked in Figure 6.1. An analysis of these 10 points will fully reveal the purpose of this error message and action to be taken by the user to correct the error. A detailed discussion of the 10 points follows.

Point 1) The number of the error code is number 75. Error code 75 is described in Section 6.2.5 as an attempt to extrapolate outside the range of the specified data set. Additional discussion on extrapolation is given in Section 6.2.4.

Point 2) The error was detected and the error message was printed from subroutine INTRP.

Point 3) Since parameter TBL equals one, the program is supposed to interpolate arrays Y and X linearly (Vol. II, Section 4.8).

Point 4) The coded name of the data set involved in this error message is "DSZ". It is found upon examination of Section 6.2.6 and Table 6.2 that subroutine INTRP is called from subroutine DSZ. In subroutine DSZ, the apparent solute dispersion coefficient, D_{sz}, is interpolated as a function of pore-water velocity, v. Additional details of this data set are given in Vol. II, Section 4.7.18.

Point 5) The error message states that the program was trying to extrapolate, using data set DSZ.

Point 6) The program was looking for the value of the solute dispersion coefficient, given that the pore-water velocity equaled 94.37.

Point 7) However, the pore-water velocity, array X, is only defined from zero to 10. Thus the value of 94.37 lies outside the range of array X and extrapolation is needed to obtain the corresponding value of the solute dispersion coefficient, array Y.

Point 8) Extend the range of data set DSZ so that array X goes from zero to at least 100.

Point 9) Arrays X and Y contained three values (NUM = 3) and a linear interpolation scheme was specified (TBL = 1).

Point 1Ø) The X and Y arrays of data set DSZ are printed.

In this example, the originally specified range of the pore-water velocity in data set DSZ is too limited. When the program was first run, the range from 0 to 10 was assumed to have been sufficient. However, upon running the program, it became obvious that the pore-water velocity actually ranged from 0 to 100. The relationship between pore-water velocity and solute dispersion is the same as that given in the example of Vol. II, Section 4.7.18, $D_{sz} = .216 * v + .192$. Data set DSZ was repunched such that:

XDSZ(1) = Ø	YDSZ(1) = .192
XDSZ(2) = 1.	YDSZ(2) = .4Ø8
XDSZ(3) = 1ØØ.	YDSZ(3) = 21.792

Upon resubmitting the program with the above data, no additional error messages were encountered.

Table 6.2 A list of coded names used in error messages

Coded name	Comments	CALL statement	Subroutine called from	Card Id number
BCB	check order of data set BCB	ORDER	ERROR	1349
BCB	interpolate BCB vs t	INTRP	BCCBTM	5Ø32
BCT	check order of data set BCT	ORDER	ERROR	135Ø
BCT	interpolate BCT vs t	INTRP	BCCTOP	5Ø97
BWB	check order of data set BWB	ORDER	ERROR	1351
BWB	interpolate BWB vs t	INTRP	BCWBTM	5163
BWT	check order of data set BWT	ORDER	ERROR	1352
BWT	interpolate BWT vs t	INTRP	BCWTOP	522Ø
CØ	check order of data set CØ	ORDER	ERROR	1353
CØ	tabulate data set CØ	TABLED	ERROR	1436
CØ	interpolate CØ vs ZØ	INTRP	INTLC	4912
CG	check order of data set CG	ORDER	ERROR	1354
CG	tabulate data set CG	TABLED	ERROR	1448
CG	interpolate CG vs C	INTRP	SINKC	5289
CST	check order of data set CST	ORDER	ERROR	1355
CST	tabulate data set CST	TABLED	ERROR	145Ø
CST	interpolate CST vs ψ_{ps}	INTRP	STMATA	5318
CTX	check order of data set CTX	ORDER	ERROR	1356
CTX	tabulate data set CTX	TABLED	ERROR	1452
CTX	interpolate CTX vs θ	INTRP	KWCRTX	5351
CZ1	integrate $c\theta dz$	INTGRT	NATURE	588
CZ2	integrate $c\theta dz$	INTGRT	NATURE	897
CZ3	integrate $c\theta dz$	INTGRT	NATURE	898
DF	interpolate D vs F	INTRP	NATURE	872
DF1	interpolate D vs F	INTRP	NATURE	883
DF2	interpolate D vs F	INTRP	WATER	2247
DF3	interpolate D vs F	INTRP	WATER	2393
DF4	interpolate D vs F	INTRP	WATER	2427
DF5	interpolate D vs F	INTRP	WATER	25ØØ
DF6	interpolate D vs F	INTRP	WATER	2536
DF7	interpolate D vs F	INTRP	WATER	257Ø
DSZ	check order of data set DSZ	ORDER	ERROR	1357
DSZ	tabulate data set DSZ	TABLED	ERROR	1454
DSZ	interpolate D_{sz} vs v	INTRP	DSZ	539Ø
DW	interpolate $\partial\theta/\partial F$ vs F	INTRP	NATURE	876
DW1	interpolate $\partial\theta/\partial F$ vs F	INTRP	NATURE	887
DW2	interpolate $\partial\theta/\partial F$ vs F	INTRP	WATER	2431
DW3	interpolate $\partial\theta/\partial F$ vs F	INTRP	WATER	251Ø
DW4	interpolate $\partial\theta/\partial F$ vs F	INTRP	WATER	2574
DW5	interpolate $\partial\theta/\partial\psi$ vs ψ	INTRP	NATURE	847
DW6	interpolate $\partial\theta/\partial\psi$ vs ψ	INTRP	NATURE	855
DW7	interpolate $\partial\theta/\partial\psi$ vs ψ	INTRP	WATER	2263
DW8	interpolate $\partial\theta/\partial\psi$ vs ψ	INTRP	WATER	2419
DW9	interpolate $\partial\theta/\partial\psi$ vs ψ	INTRP	WATER	2486

Table 6.2, continued.

Coded name	Comments	CALL statement	Subroutine called from	Card Id number
DWP	interpolate $\partial\theta/\partial\psi$ vs ψ	INTRP	WATER	2562
DWZ	check order of data set DWZ	ORDER	ERROR	1358
DWZ	tabulate data set DWZ	TABLED	ERROR	1456
DWZ	interpolate D vs θ	INTRP	DWZ	5419
FCB	check order of data set FCB	ORDER	ERROR	1359
FCB	tabulate data set FCB	TABLED	ERROR	1441
FCB	interpolate FCB vs t	INTRP	BCCBTM	5Ø37
FCT	check order of data set FCT	ORDER	ERROR	136Ø
FCT	tabulate data set FCT	TABLED	ERROR	1443
FCT	interpolate FCT vs t	INTRP	BCCTOP	51Ø4
FC1	interpolate FCT vs t	INTRP	BCCTOP	51Ø8
FP	interpolate F vs ψ	INTRP	WATER	2Ø23
FP1	interpolate F vs ψ	INTRP	WATER	2Ø29
FP2	integrate K$d\psi$	INTGRT	TRNSFM	4617
FP3	interpolate F vs ψ	INTRP	TRNSFM	4678
FP4	interpolate F vs ψ	INTRP	TRNSFM	4679
FP5	interpolate F vs ψ	INTRP	TRNSFM	468Ø
FP6	interpolate F vs ψ	INTRP	TRNSFM	4681
FW	interpolate F vs θ	INTRP	WATER	2Ø21
FW1	interpolate F vs θ	INTRP	WATER	2Ø27
FW2	integrate D$d\theta$	INTGRT	TRNSFM	4536
FW3	interpolate F vs θ	INTRP	TRNSFM	4576
FW4	interpolate F vs θ	INTRP	TRNSFM	4577
FW5	interpolate F vs θ	INTRP	TRNSFM	4578
FW6	interpolate F vs θ	INTRP	TRNSFM	4579
FWB	check order of data set FWB	ORDER	ERROR	1361
FWB	tabulate data set FWB	TABLED	ERROR	1445
FWB	interpolate FWB vs t	INTRP	BCWBTM	5167
FWT	check order of data set FWT	ORDER	ERROR	1362
FWT	tabulate data set FWT	TABLED	ERROR	1447
FWT	interpolate FWT vs t	INTRP	BCWTOP	5225
KF	interpolate K vs F	INTRP	NATURE	862
KF1	interpolate K vs F	INTRP	NATURE	87Ø
KF2	interpolate K vs F	INTRP	NATURE	881
KF3	interpolate K vs F	INTRP	WATER	2395
KF4	interpolate K vs F	INTRP	WATER	2425
KF5	interpolate K vs F	INTRP	WATER	2496
KF6	interpolate K vs F	INTRP	WATER	2497
KF7	interpolate K vs F	INTRP	WATER	2538
KF8	interpolate K vs F	INTRP	WATER	2568
KP	interpolate K vs ψ	INTRP	NATURE	612
KPZ	interpolate K vs ψ	INTRP	NATURE	836
KP1	interpolate K vs ψ	INTRP	NATURE	843
KP2	interpolate K vs ψ	INTRP	NATURE	851
KP3	interpolate K vs ψ	INTRP	WATER	226Ø
KP4	interpolate K vs ψ	INTRP	WATER	2416

Table 6.2, continued.

Coded name	Comments	CALL statement	Subroutine called from	Card Id number
KP5	interpolate K vs ψ	INTRP	WATER	2487
KP6	interpolate K vs ψ	INTRP	WATER	2488
KP7	interpolate K vs ψ	INTRP	WATER	2559
KP8	interpolate K vs ψ	INTRP	TRNSFM	4613
KP9	interpolate K vs ψ	INTRP	TRNSFM	467Ø
KWZ	check order of data set KWZ	ORDER	ERROR	1363
KWZ	tabulate data set KWZ	TABLED	ERROR	1458
KWZ	interpolate K vs θ	INTRP	KWZ	5451
PØ	check order of data set PØ	ORDER	ERROR	1364
PØ	tabulate data set PØ	TABLED	ERROR	1437
PØ	interpolate PØ vs ZØ	INTRP	INTLP	494Ø
PF	interpolate ψ vs F	INTRP	NATURE	78Ø
PF1	interpolate ψ vs F	INTRP	WATER	2Ø44
PF2	interpolate ψ vs F	INTRP	WATER	2259
PF3	interpolate ψ vs F	INTRP	WATER	23Ø6
PW	check order of data set PW	ORDER	ERROR	1365
PW	tabulate data set PW	TABLED	ERROR	1459
PW	interpolate ψ vs θ	INTRP	WATER	2915
PW1	interpolate ψ vs θ	INTRP	PSIW	5483
QPS	check order of data set QPS	ORDER	ERROR	1366
QPS	tabulate data set QPS	TABLED	ERROR	1461
QPS	interpolate q_{ps} vs t	INTRP	PTRNS	551Ø
R1	interpolate ${}^{*}c^{k+1}$ vs ZF	INTRP	REGRID	4385
R2	interpolate ${}^{*}\psi^{k+1}$ vs ZF	INTRP	REGRID	4389
R3	interpolate ${}^{*}S^{k+1}$ vs ZF	INTRP	REGRID	4393
R4	interpolate ${}^{*}\theta^{k+1}$ vs ZF	INTRP	REGRID	4397
R5	interpolate c^{k+1} vs ZF	INTRP	REGRID	44Ø1
R6	interpolate c^{k-1} vs ZF	INTRP	REGRID	44Ø7
R7	interpolate c^{k} vs ZF	INTRP	REGRID	4411
R8	interpolate ψ^{k-1} vs ZF	INTRP	REGRID	4415
R8B	interpolate ψ^{k} vs ZF	INTRP	REGRID	4419
R9	interpolate S^{k-1} vs ZF	INTRP	REGRID	4423
R1Ø	interpolate S^{k} vs ZF	INTRP	REGRID	4427
R11	interpolate θ^{k+1} vs ZF	INTRP	REGRID	4429
R12	interpolate θ^{k-1} vs ZF	INTRP	REGRID	4435
R13	interpolate θ^{k} vs ZF	INTRP	REGRID	4439
R14	interpolate ${}^{*}F^{k+1}$ vs ZF	INTRP	REGRID	4441
R15	interpolate F^{k-1} vs ZF	INTRP	REGRID	4447

Table 6.2, continued.

Coded name	Comments	CALL statement	Subroutine called from	Card Id number
R16	interpolate F^k vs ZF	INTRP	REGRID	4451
R17	interpolate q_z^{k+1} vs ZF	INTRP	REGRID	4453
RAC	check order of data set RAC	ORDER	ERROR	1367
RAC	tabulate data set RAC	TABLED	ERROR	1463
RAC	interpolate $R_a(c)$ vs c	INTRP	RACTYC	554Ø
RAW	check order of data set RAW	ORDER	ERROR	1368
RAW	tabulate data set RAW	TABLED	ERROR	1465
RAW	interpolate $R_a(\theta)$ vs θ	INTRP	RACTYW	5566
RD	check order of data set RD	ORDER	ERROR	1369
RD	tabulate data set RD	TABLED	ERROR	1466
RD	interpolate R_d vs z	INTRP	RDNSTY	5596
RPF	check order of data set RPF	ORDER	ERROR	137Ø
RPF	tabulate data set RPF	TABLED	ERROR	1468
RPF	interpolate RPF vs z	INTRP	RPF	5625
SØ	check order of data set SØ	ORDER	ERROR	1371
SØ	tabulate data set SØ	TABLED	ERROR	1438
SØ	interpolate SØ vs ZØ	INTRP	INTLS	4967
SZ1	integrate Sdz	INTGRT	NATURE	59Ø
SZ2	integrate Sdz	INTGRT	NATURE	9Ø2
SZ3	integrate Sdz	INTGRT	NATURE	9Ø3
T1	interpolate using TBL = 1	INTRP	TABLED	4182
T2	interpolate using TBL = 2	INTRP	TABLED	4183
T3	interpolate using TBL = 3	INTRP	TABLED	4184
T4	interpolate using TBL = 4	INTRP	TABLED	4185
T5	interpolate using TBL = 5	INTRP	TABLED	4186
T6	interpolate using TBL = 6	INTRP	TABLED	42Ø7
T7	interpolate using TBL = 7	INTRP	TABLED	42Ø8
T8	interpolate using TBL = 8	INTRP	TABLED	42Ø9
T9	interpolate using TBL = 9	INTRP	TABLED	421Ø
T1Ø	interpolate using TBL = 1Ø	INTRP	TABLED	4211
WØ	check order of data set WØ	ORDER	ERROR	1372
WØ	tabulate data set WØ	TABLED	ERROR	1439
WØ	interpolate WØ vs ZØ	INTRP	INTLW	4994
WF	interpolate θ vs F	INTRP	NATURE	778
WF1	interpolate θ vs F	INTRP	WATER	23Ø7
WF2	interpolate θ vs F	INTRP	WATER	2913
WF3	interpolate θ vs F	INTRP	TRNSFM	465Ø
WP	check order of data set WP	ORDER	ERROR	1373
WP	tabulate data set WP	TABLED	ERROR	1469
WP	interpolate θ vs ψ	INTRP	WATER	2127
WP1	interpolate θ vs ψ	INTRP	WPSI	566Ø
WZ1	integrate θdz	INTGRT	NATURE	582
WZ2	integrate θdz	INTGRT	NATURE	941
WZ3	integrate θdz	INTGRT	NATURE	942
ZØ	check order of data set ZØ	ORDER	NATURE	368
ZF	check order of data set ZF	ORDER	NATURE	375

```
                              ERROR CODE  75 (1)

                    ERROR FLAG FROM SUBROUTINE  INTRP (2)

                                                     (3)                  (4)
              USING THE INTERPOLATION SCHEME OF  TBL =  1 WITH DATA SET DSZ
(5)
   YOU TRIED TO EXTRAPOLATE OUTSIDE OF THE GIVEN RANGE OF FUNCTION  Y VERSUS VARIABLE X
   ONLY INTERPOLATION IS PERMITTED
                                        (6)
YOU WANTED TO EVALUATE AT VARIABLE  XI =  .94375E+02 BUT TABULAR DATA OF VARIABLE X IS LIMITED TO VALUES

              BETWEEN XDSZ(  1) =  0.  (7)   AND  XDSZ(  3) =   .1000E+02

         (8) RESUBMIT THE PROGRAM AFTER EXTENDING THE RANGE OF DATA SET DSZ



                    DATA VARIABLES AT TIME OF ERROR MESSAGE

                              NUMDSZ =   3 (9)
                              TBLDSZ =   1

             THE FOLLOWING IS A LIST OF THE X AND Y VALUES FROM DATA SET DSZ

                                        (10)
       XDSZ(  1) =  0.                       YDSZ(  1) =   .1920E+00
       XDSZ(  2) =   .1000E+01               YDSZ(  2) =   .4080E+00
       XDSZ(  3) =   .1000E+02               YDSZ(  3) =   .2352E+01
```

Figure 6.1 Sample printout of error message.

6.3 Failure to Converge

"Failure to converge" is the most difficult error message to respond to. It only occurs during the computation of an iteration algorithm (Vol. I, Section 3) and indicates that the iteration algorithm is failing to converge.

6.3.1 Definition

The water flow equation is non-linear because the conductivity and diffusivity coefficients are functions of the variable to be calculated. So, to solve the equation for a new (k+1) time level, the value of the water content is guessed for each space node, the non-linear cofficients are evaluated using the guessed values of water content, the centered Crank-Nicolson formulation is set up, and by means of the Thomas algorithm, a computed value of water content is obtained for each space node at time level (k+1). The most recently computed value is compared with the guessed value and the largest absolute difference in percent between these two values is identified. Convergence is assumed if the largest difference is less than the specified criterion, WRATIO. If this criterion is not met, a new guess is made and the iteration process is repeated. If more than ITRMAX iterations are necessary, the program declares that the routine is failing to converge. Twenty-five additional iterations are tried, during which the program prints out the values of various variables pertinent to the iteration scheme. If the program does not converge within these last 25 iterations, a failure to converge error message is printed and the program aborts. A similar procedure is used in the surface flux iterator scheme using QRATIO, in the plant surface potential iterator scheme using ARATIO, and in the solid phase solute iterator scheme using SRATIO.

In Table 6.3, a summary is given of all variables with which a failure to converge error message can originate. In the column marked "failure status", the word FAILING indicates that additional iterations will be allowed but the program will print out the values of pertinent iteration variables. The word FAILED means that the iterator scheme will be terminated and the program aborted. Error codes are explained in Section 6.2.5.

Table 6.3 Number of iterations allowed before failure to converge is declared

Iteration variable	Failure status	Error code	Maximum number of iterations allowed	Convergence criterion
θ	FAILING	55	M = ITRMAX	WRATIO
θ	FAILED	56	M = ITRMAX + 25	WRATIO
ψ_{ps}	FAILED	57	MR = ITRMAX	ARATIO
water flux	FAILING	61	ITR = ITRMAX * 1Ø - 25	QRATIO
water flux	FAILED	62	ITR = ITRMAX * 1Ø	QRATIO
S	FAILING	63	MS = ITRMAX	SRATIO
S	FAILED	64	MS = ITRMAX + 25	SRATIO
Non-linear S when JSORB = 9	FAILED	67	M9 = ITRMAX	SRATIO

6.3.2 Reasons for Failure

There are many possible causes for failure to converge in subroutines WATER and SOLUTE. Four general types of failure modes are identified.

i) Excessive numerical oscillations or ripples are produced in the solution;

ii) The guess-compute procedure becomes locked into a cyclic pattern. The program uses the same guess value every other iteration, hence the iterator becomes trapped in an "endless loop" (Vol. I, Section 3.1.1);

iii) The guess-compute process becomes very evasive as the guess tends to remain "out of step" with the ever wandering computed values;

iv) The guess-compute process is converging but it is doing so at a very slow rate.

The above failure modes are not mutually exclusive and the cause is not always identifiable. Some of the circumstances which can aggravate or cause the above failures are:

i) a set of boundary conditions, which are not physically possible. This occurs particularly with "step" inputs;

ii) using poorly defined or discontinuous functions to represent the soil water characteristic curve, the hydraulic conductivity function, or the diffusivity function. Problems usually occur in either nearly dry or nearly saturated sandy soils;

iii) using empirical formulas to represent functions, i.e., when TBL(...) = Ø, which are not continuous at "junction points." This occurs for example when different formulas are used over different ranges of a function;

iv) specifying the minimum time step-size DTMIN too large. The program computes the time step-size, DTK1, in subroutine STEP but it is overridden if the computed value is smaller than DTMIN or greater than DTMAX;

v) specifying the maximum number of iterations, ITRMAX, too small;

vi) setting the first few space nodes ZØ either too close together or too far apart, when a variable space step-size is used;

vii) specifying the convergence criterions ARATIO, QRATIO, SRATIO, and WRATIO either too small or too large. The criterions are expressed as a ratio, thus .Ø1 means a 1 percent convergence criterion;

viii) specifying the wrong type of interpolation to be used with the data sets or using too few data points to define a function, particularly when using either the quadratic or cubic spline interpolation.

6.3.3 Corrections to be Made

Determining the cause of failure and recommending a solution to a failure to converge problem is at best an art, but a number of helpful steps are suggested below.

Step 1) Check to make sure that all specified data and empirical functions are correctly specified. It is pointless to proceed until this is done. The program has a data error analysis routine built into it, which is activated by setting XORDER = 1 and SCAN = 1, but even this may fail to catch all input data errors.

Step 2) Try using the BEST option by setting BEST = 1 in program card #2. The purpose of this option is to give the user a feel for the data sets and the effect of using different types of interpolation schemes in the input data tables. In many instances the program can be sensitive to the number and spacing of the entries in a tabulated data set (X_i, Y_i) and the type of interpolation routine used. For each tabulated data point, X_i, three additional intermediate table values of X are generated and the corresponding value of Y and its derivative dY/dX are computed by all 10 types of interpolation schemes. One can determine immediately if either a cubic spline or quadratic interpolation scheme is desirable by looking for oscillations in the interpolated values of Y and dY/dX. It can also be determined if log-log or semi-log data transforms of Y vs X are of any value. After using the BEST option, the spacing or number of data points in a data set may be changed or the most suitable interpolation technique for the data may be determined. In addition, one can determine the most suitable data transform by plotting the data on log-log and linear graph paper. The best data transform is that one which maps the data onto a straight line or most closely to a straight line.

Step 3) Try varying the time step-size, particularly the minimum and maximum specified values. Also vary the space step-size, particularly near the soil surface, z = Ø.

Step 4) When subroutines WATER and SOLUTE fail to converge, the last 25 iterations, before failure, are printed. The program will only print the worst guess-compute node during each of these iterations. One can use this information by plotting the guess versus computed value on graph paper. This graph may suggest the mode of failure.

Step 5) Every NSKIP time step, the program prints the accumulative mass balance of water expressed as WBLNCE and of solute expressed as CBLNCE. A great deal of insight can be gained by plotting the mass balance as a function of time. Any sudden deviation in the mass balance indicates approximately when the program started to fail.

Step 6) When a flux type boundary condition (4th) is specified on the upper surface for water flow, the program prints out the following information at the end of each time step:

TIME	ITERATIONS		SURFACE WATER FLUX	
	MAXI	MINI	POTENTIAL	ACTUAL
.	.	.	.	.
.	.	.	.	.
.	.	.	.	.
2.Ø	1	5	1.Ø	1.Ø
3.Ø	1	4	1.Ø	1.Ø
4.Ø	1	5	1.Ø	1.Ø
5.Ø	1	12	2.Ø	2.Ø
6.Ø	1	15	3.Ø	2.998
7.Ø	2	23	4.Ø	3.8Ø
8.Ø	3	28	5.Ø	3.42

It is clear that few iterations are needed by the flux iterator when the potential flux is small. MAXI and MINI indicate - the total number of iterations needed to compute the actual surface flux. The number of flux iterations increases as the potential flux increases. The flux iterator is not very sensitive to the magnitude of the flux, rather it is more sensitive to the rate at which it changes with respect to time. When the actual flux becomes less than the potential flux, a dramatic increase in the number of iterations required for convergence occurs.

When the water flow equation fails to converge with a flux type upper boundary condition (a 4th type) a complete listing of all pertinent variables used in the Golden Section Search interator is given (Vol. I, Section 3.4). An extensive listing is also given when the solute equation fails to converge with a 3rd type upper boundary condition.

Step 7) Try using the START option (Kirchhoff transform, Vol. I, Section 1.2.3). This may be used during the first "k" time steps by setting START = k, where k is equal to the number of time steps desired or it may be used to run the entire program in terms of the transform by setting START = -1. The pressure based, as well as the water based, flow equations can be solved by means of the START option. Except for the START parameter of program card #2, no additional information is needed to use this transform technique.

Step 8) If the program fails after it has finished using the START option and has switched back to the non-transformed equation, try using the START option again but allow more time steps under START, e.g., double the value of START. One may also try running all time steps with the Kirchhoff transform by setting START = -1. An example of continuously using the Kirchhoff transform is given in Vol. IV, Section 7.4.

Step 9) If a solution to the water flow equation has been attempted in terms of matric potential (JWORH = 1), try solving the problem in terms of volumetric water content (JWORH = Ø). The water flow equation based on water content almost always behaves better. Do not forget to respecify DRY, SAT, WØ, PW, etc. again after switching to water content.

Step 10) Bypass the guess-correct iteration scheme by setting ARATIO, QRATIO, SRATIO, or WRATIO to a large positive value. Both Hanks and Bowers (1962) and Rubin and Steinhardt (1963) used a single guess for each new time step, with the guess based on a linear extrapolation of the previous time step. This guess is not corrected. To use such a scheme, set ACLRAT = 1. and WRATIO = 1ØØØ. or any other large, positive, value.

Step 11) Reduce the magnitude of the boundary condition for the first few time steps, then set it back to the desired level when the water flow or solute equation has stabilized. This is particularly useful when working with initially very dry, sandy, soils. This approach produces inaccurate results for the first few time steps but the error disappears rapidly in most cases. The purpose of this scheme is to insure convergence by building the boundary condition up from an "easy to converge" value to the desired "difficult" value. This scheme can be used regardless of whether the water flow equation is being solved in terms of water content, matric potential, or diffusivity potential and regardless of the type of boundary condition.

6.3.4 Sample Output From A Failure To Converge

A typical example of the output generated by program OR-NATURE for a failure to converge is given in Figure 6.2. Notice in Figure 6.2 the circled numbers or "points", of which there are 26. Each of these points will be discussed in detail below. Each point will describe the variables being printed and their significance to the analysis of the failure to converge.

A complete description of the data used in this example is given in Vol. IV, Section 7.1. The problem involves cyclic infiltration, redistribution, and evaporation of water into a 158-cm soil column of clay loam over a period of 144 hours. The soil has an initial uniform water content of .15 cm^3/cm^3, saturation is specified at .38 cm^3/cm^3 and air-dry at .Ø7 cm^3/cm^3. The water flow equation is solved as a function of volumetric water content. A constant time increment of .2 hours is used. The program failed at time equal to 1Ø6 hours. From hour 1Ø2 to hour 1Ø6, the program was in an infiltration phase. Infiltration was increasing from Ø to 2.1 cm/hr.

TIME (HR)	ITERATIONS MAXI	ITERATIONS MINI	SURFACE MOISTURE FLUX (CM/HR) AT (1/2,K+1) POTENTIAL FLUX	ACTUAL FLUX
.104E+03	1	18	.3000E+00	.2973E+00
.104E+03	1	8	.5000E+00	.4963E+00
.105E+03	1	6	.7000E+00	.6954E+00
.105E+03	1	6	.9000E+00	.8945E+00
.105E+03	1	8	.1100E+01	.1094E+01
.105E+03	1	8	.1300E+01	.1293E+01
.105E+03	1	25	.1500E+01	.1492E+01
.106E+03	2	14	.1700E+01	.1691E+01
.106E+03	4	35	.1900E+01	.1886E+01

(1)

(2)

ERROR CODE 61

(3)

FLUX ITERATOR IN SUBROUTINE WATER IS FAILING TO CONVERGE

PROGRAM WILL CONTINUE FOR 25 ADDITIONAL ITERATIONS (4)

AFTER EACH FLUX ITERATION THE PROGRAM WILL LIST THE FOLLOWING DATA

(5) ITR	QGAK	(6) QGBK	QG	(7) QC	(8) FTOP	(9) RATIOG	(10) RATIOQ	(11) RWORST	(12) IWORST	(13) FLOW(IWORST)	(14) M	(15) MQ	(16) MQFAIL	(17) NITR
475	.179E+01	.181E+01	.180E+01	.177E+01	.180E+01	.63E-01	.16E+00	.30E-03	3	.3780E+00	0	-13	0	12
476	.177E+01	.181E+01	.177E+01	.177E+01	.177E+01	.51E-11	.16E+00	.60E-03	2	.3796E+00	0	-1	0	10
477	.177E+01	.181E+01	.179E+01	.178E+01	.179E+01	.19E-01	.15E+00	.23E-03	1	.3800E+00	0	-2	0	10
478	.177E+01	.181E+01	.178E+01	.178E+01	.178E+01	.61E-11	.15E+00	.92E-04	9	.3674E+00	0	-3	0	10
479	.177E+01	.181E+01	.178E+01	.178E+01	.178E+01	.48E-12	.15E+00	.12E-03	2	.3797E+00	0	-4	0	10
480	.178E+01	.181E+01	.180E+01	.168E+01	.180E+01	.32E+00	.20E+00	.47E-03	2	.3799E+00	0	-5	0	10
481	.178E+01	.180E+01	.178E+01	.178E+01	.178E+01	.58E-11	.15E+00	.31E-03	2	.3797E+00	0	-6	0	10
482	.178E+01	.180E+01	.179E+01	.175E+01	.179E+01	.12E+00	.17E+00	.18E-03	2	.3798E+00	0	-7	0	10
483	.178E+01	.179E+01	.178E+01	.178E+01	.178E+01	.28E-12	.15E+00	.21E-03	2	.3797E+00	0	-8	0	10
484	.178E+01	.179E+01	.178E+01	.159E+01	.178E+01	.53E+00	.24E+00	.41E-02	2	.3813E+00	1	1	2	100
485	.142E+01	.210E+01	.178E+01	.178E+01	.178E+01	.48E-11	.15E+00	.37E-02	3	.3773E+00	0	-1	0	12
486	.178E+01	.210E+01	.198E+01	.157E+01	.198E+01	.10E+01	.25E+00	.56E-02	2	.3816E+00	0	-2	2	12
487	.178E+01	.198E+01	.178E+01	.178E+01	.178E+01	.27E-11	.15E+00	.34E-02	3	.3775E+00	0	-3	0	12
488	.178E+01	.198E+01	.190E+01	.157E+01	.190E+01	.87E+00	.25E+00	.35E-02	2	.3809E+00	0	-4	2	12
489	.178E+01	.190E+01	.178E+01	.178E+01	.178E+01	.96E-11	.15E+00	.23E-02	3	.3776E+00	0	-5	0	12
490	.178E+01	.190E+01	.186E+01	.158E+01	.186E+01	.75E+00	.25E+00	.21E-02	2	.3804E+00	0	-6	2	12
491	.178E+01	.186E+01	.178E+01	.178E+01	.178E+01	.34E-11	.15E+00	.14E-02	3	.3777E+00	0	-7	0	12

Figure 6.2 Sample printout of a failure to converge. This printout is continued on pages 42, 43, 44, and 45.

492	.178E+01	.186E+01	.183E+01	.158E+01	.183E+01	.67E+00	.25E+00	.16E-02	1	.3805E+00	0 -8	1	12
493	.178E+01	.183E+01	.178E+01	.178E+01	.178E+01	.62E-12	.15E+00	.87E-03	3	.3778E+00	0 -9	0	12
494	.178E+01	.183E+01	.181E+01	.160E+01	.181E+01	.59E+00	.24E+00	.67E-03	2	.3800E+00	0 -10	0	12
495	.178E+01	.181E+01	.178E+01	.178E+01	.178E+01	.48E-11	.15E+00	.53E-03	3	.3779E+00	0 -11	0	12
496	.178E+01	.181E+01	.180E+01	.168E+01	.180E+01	.34E+00	.20E+00	.33E-03	2	.3799E+00	0 -12	0	12
497	.178E+01	.180E+01	.179E+01	.179E+01	.179E+01	.81E-11	.15E+00	.34E-03	2	.3797E+00	0 -13	0	12
498	.179E+01	.180E+01	.179E+01	.159E+01	.179E+01	.54E+00	.24E+00	.41E-02	2	.3814E+00	1 1	2	100
499	.143E+01	.210E+01	.179E+01	.179E+01	.179E+01	.30E-11	.15E+00	.38E-02	3	.3774E+00	0 -1	0	12
500	.179E+01	.210E+01	.198E+01	.157E+01	.198E+01	.10E+01	.25E+00	.55E-02	2	.3816E+00	0 -2	2	12

(18)

ERROR CODE 62

(19)

FLUX ITERATOR FAILED TO CONVERGE IN SUBROUTINE WATER

AT THE TIME OF FAILURE, THE FOLLOWING VARIABLES HAD THE VALUE OF

(20)

```
     M    =   0
     MQ   =   -2
     MR   =   0
    ITR   = 500
   MAXI   =  28
   MINI   = 472
  IQPOS   =  20
 MQFAIL   =   2
  MQMAX   = 500
  NMINI   =  12
   NITR   =  12
  SHORT   =   0
 GOLDEN   =  22
   TIME   =   .10600E+03
  DTMIN   =   .20000E+00
  DTMAX   =   .20000E+00
    DTK   =   .20000E+00
   DTK1   =   .20000E+00
     K1   =   530
 ARATIO   =   .10000E-01
 QRATIO   =   .10000E-01
 WRATIO   =   .10000E-01
     QP   =   .21000E+01
     QG   =   .19799E+01
     QC   =   .15737E+01
 BCTOPW   =   4
   FTOP   =   .19799E+01
   QGAK   =   .17855E+01
   QGTK   =   .17953E+01
 QGTKP    =   .19799E+01
```

Figure 6.2 (continued)

```
       QGBK  =    .21000E+01
  QCOLD(1)  =    .16907E+01
  QCOLD(2)  =    .18861E+01
  QPOLD(1)  =    .17000E+01
  QPOLD(2)  =    .19000E+01
    RATIOG  =    .10257E+01
    RATIOQ  =    .25061E+00
    RWORST  =    .54981E-02
    IWORST  =    2
       MIN  =    .70000E-01
       MAX  =    .38000E+00
```

GIVEN BELOW IS A LIST OF 7 ARRAYS FROM SUBROUTINE WATER CONTAINING THE VALUES OF MOISTURE FROM THE LAST TWO INTEGRATION STEPS, THE GUESSED AND COMPUTED VALUES OF THE LAST TWO ITERATION STEPS AND THE MOST CURRENT ITERATICN GUESS VALUES
EVALUATED ON ITERATION NUMBER 500 OF SUBROUTINE WATER AT TIME = .10600E+03 (HR)

COMPUTED AND GUESSED VALUES OF VOLUMETRIC MOISTURE CONTENT (CM**3/CM**3) (21)

TIME ITERATION STATUS (22)	K-1 ACTUAL	K ACTUAL	K+1 498 GUESS	K+1 498 COMPUTED	K+1 499 GUESS	K+1 499 COMPUTED	K+1 500 GUESS
NODE I	OLD1(I) (23)	OLD2(I) (24)	FGGOLD(I) (25)	FCCOLD(I)	FGOLD(I)	FCOLD(I)	GUESS(I)
1	.3709E+00	.3744E+00	.3800E+00	.3794E+00	.3797E+00	.3800E+00	.3798E+00
2	.3699E+00	.3734E+00	.3799E+00	.3792E+00	.3795E+00	.3800E+00	.3798E+00
3	.3673E+00	.3739E+00	.3788E+00	.3774E+00	.3781E+00	.3796E+00	.3788E+00
4	.3645E+00	.3744E+00	.3769E+00	.3756E+00	.3762E+00	.3775E+00	.3769E+00
5	.3638E+00	.3726E+00	.3772E+00	.3759E+00	.3766E+00	.3776E+00	.3771E+00
6	.3655E+00	.3682E+00	.3795E+00	.3787E+00	.3791E+00	.3800E+00	.3795E+00
7	.3629E+00	.3678E+00	.3786E+00	.3773E+00	.3780E+00	.3785E+00	.3782E+00
8	.3544E+00	.3729E+00	.3716E+00	.3704E+00	.3710E+00	.3713E+00	.3712E+00
9	.3497E+00	.3740E+00	.3684E+00	.3673E+00	.3679E+00	.3679E+00	.3679E+00
10	.3544E+00	.3656E+00	.3745E+00	.3733E+00	.3739E+00	.3738E+00	.3738E+00
11	.3606E+00	.3552E+00	.3800E+00	.3800E+00	.3800E+00	.3800E+00	.3800E+00
12	.3576E+00	.3534E+00	.3800E+00	.3800E+00	.3800E+00	.3800E+00	.3800E+00
13	.3449E+00	.3605E+00	.3693E+00	.3729E+00	.3693E+00	.3729E+00	.3711E+00
14	.3314E+00	.3677E+00	.3494E+00	.3622E+00	.3494E+00	.3623E+00	.3558E+00
15	.3263E+00	.3661E+00	.3455E+00	.3592E+00	.3455E+00	.3592E+00	.3523E+00
16	.3300E+00	.3552E+00	.3521E+00	.3646E+00	.3521E+00	.3646E+00	.3584E+00
17	.3356E+00	.3416E+00	.3607E+00	.3722E+00	.3607E+00	.3722E+00	.3665E+00
18	.3350E+00	.3326E+00	.3644E+00	.3748E+00	.3644E+00	.3748E+00	.3696E+00
19	.3233E+00	.3326E+00	.3589E+00	.3683E+00	.3589E+00	.3684E+00	.3636E+00
20	.3017E+00	.3393E+00	.3463E+00	.3548E+00	.3463E+00	.3549E+00	.3506E+00
21	.2757E+00	.3468E+00	.3320E+00	.3398E+00	.3320E+00	.3398E+00	.3359E+00
22	.2421E+00	.3310E+00	.3287E+00	.3352E+00	.3287E+00	.3352E+00	.3320E+00
23	.2314E+00	.2871E+00	.3421E+00	.3470E+00	.3421E+00	.3471E+00	.3446E+00
24	.2301E+00	.2514E+00	.3331E+00	.3372E+00	.3331E+00	.3372E+00	.3351E+00
25	.2311E+00	.2372E+00	.2968E+00	.2989E+00	.2968E+00	.2989E+00	.2979E+00
26	.2326E+00	.2341E+00	.2610E+00	.2615E+00	.2610E+00	.2615E+00	.2613E+00

Figure 6.2 (continued)

27	.2341E+00	.2344E+00	.2431E+00	.2433E+00	.2431E+00	.2433E+00	.2432E+00
28	.2356E+00	.2356E+00	.2380E+00	.2381E+00	.2380E+00	.2381E+00	.2380E+00
29	.2370E+00	.2369E+00	.2375E+00	.2375E+00	.2375E+00	.2375E+00	.2375E+00
30	.2383E+00	.2382E+00	.2383E+00	.2383E+00	.2383E+00	.2383E+00	.2383E+00
31	.2395E+00	.2394E+00	.2394E+00	.2394E+00	.2394E+00	.2394E+00	.2394E+00
32	.2417E+00	.2417E+00	.2416E+00	.2416E+00	.2416E+00	.2416E+00	.2416E+00
33	.2438E+00	.2437E+00	.2436E+00	.2436E+00	.2436E+00	.2436E+00	.2436E+00
34	.2456E+00	.2455E+00	.2454E+00	.2454E+00	.2454E+00	.2454E+00	.2454E+00
35	.2472E+00	.2471E+00	.2470E+00	.2470E+00	.2470E+00	.2470E+00	.2470E+00
36	.2486E+00	.2486E+00	.2485E+00	.2485E+00	.2485E+00	.2485E+00	.2485E+00
37	.2499E+00	.2499E+00	.2498E+00	.2498E+00	.2498E+00	.2498E+00	.2498E+00
38	.2511E+00	.2510E+00	.2510E+00	.2510E+00	.2510E+00	.2510E+00	.2510E+00
39	.2521E+00	.2521E+00	.2520E+00	.2520E+00	.2520E+00	.2520E+00	.2520E+00
40	.2530E+00	.2530E+00	.2529E+00	.2529E+00	.2529E+00	.2529E+00	.2529E+00
41	.2538E+00	.2537E+00	.2536E+00	.2536E+00	.2536E+00	.2536E+00	.2536E+00
42	.2544E+00	.2544E+00	.2543E+00	.2543E+00	.2543E+00	.2543E+00	.2543E+00
43	.2549E+00	.2549E+00	.2548E+00	.2548E+00	.2548E+00	.2548E+00	.2548E+00
44	.2553E+00	.2553E+00	.2552E+00	.2552E+00	.2552E+00	.2552E+00	.2552E+00
45	.2556E+00	.2555E+00	.2555E+00	.2555E+00	.2555E+00	.2555E+00	.2555E+00
46	.2557E+00	.2557E+00	.2556E+00	.2556E+00	.2556E+00	.2556E+00	.2556E+00
47	.2557E+00	.2556E+00	.2556E+00	.2556E+00	.2556E+00	.2556E+00	.2556E+00
48	.2555E+00	.2555E+00	.2554E+00	.2554E+00	.2554E+00	.2554E+00	.2554E+00
49	.2552E+00	.2551E+00	.2551E+00	.2551E+00	.2551E+00	.2551E+00	.2551E+00
50	.2547E+00	.2546E+00	.2546E+00	.2546E+00	.2546E+00	.2546E+00	.2546E+00
51	.2539E+00	.2539E+00	.2538E+00	.2538E+00	.2538E+00	.2538E+00	.2538E+00
52	.2530E+00	.2530E+00	.2529E+00	.2529E+00	.2529E+00	.2529E+00	.2529E+00
53	.2518E+00	.2518E+00	.2517E+00	.2517E+00	.2517E+00	.2517E+00	.2517E+00
54	.2504E+00	.2503E+00	.2503E+00	.2503E+00	.2503E+00	.2503E+00	.2503E+00
55	.2486E+00	.2485E+00	.2485E+00	.2485E+00	.2485E+00	.2485E+00	.2485E+00
56	.2464E+00	.2464E+00	.2463E+00	.2463E+00	.2463E+00	.2463E+00	.2463E+00
57	.2438E+00	.2438E+00	.2438E+00	.2438E+00	.2438E+00	.2438E+00	.2438E+00
58	.2406E+00	.2406E+00	.2406E+00	.2406E+00	.2406E+00	.2406E+00	.2406E+00
59	.2369E+00	.2369E+00	.2369E+00	.2369E+00	.2369E+00	.2369E+00	.2369E+00
60	.2323E+00	.2323E+00	.2323E+00	.2323E+00	.2323E+00	.2323E+00	.2323E+00
61	.2267E+00	.2267E+00	.2268E+00	.2268E+00	.2268E+00	.2268E+00	.2268E+00
62	.2198E+00	.2199E+00	.2200E+00	.2200E+00	.2200E+00	.2200E+00	.2200E+00
63	.2113E+00	.2114E+00	.2116E+00	.2115E+00	.2116E+00	.2115E+00	.2115E+00
64	.2006E+00	.2008E+00	.2010E+00	.2010E+00	.2010E+00	.2010E+00	.2010E+00
65	.1875E+00	.1878E+00	.1880E+00	.1880E+00	.1880E+00	.1880E+00	.1880E+00
66	.1728E+00	.1731E+00	.1734E+00	.1734E+00	.1734E+00	.1734E+00	.1734E+00
67	.1604E+00	.1606E+00	.1608E+00	.1608E+00	.1608E+00	.1608E+00	.1608E+00
68	.1536E+00	.1537E+00	.1538E+00	.1538E+00	.1538E+00	.1538E+00	.1538E+00
69	.1510E+00	.1511E+00	.1511E+00	.1511E+00	.1511E+00	.1511E+00	.1511E+00
70	.1503E+00	.1503E+00	.1503E+00	.1503E+00	.1503E+00	.1503E+00	.1503E+00
71	.1501E+00	.1501E+00	.1501E+00	.1501E+00	.1501E+00	.1501E+00	.1501E+00
72	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00
73	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00
74	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00
75	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00
76	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00
77	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00
78	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00
79	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00
80	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00
81	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00
82	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00
83	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00
84	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00
85	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00
86	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00
87	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00
88	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00
89	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00

Figure 6.2 (continued)

90	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00
91	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00
92	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00
93	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00
94	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00
95	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00
96	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00
97	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00
98	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00
99	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00
100	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00	.1500E+00


```
CHECK IF THE SPACE AND TIME STEP SIZES VIOLATE THE CRITERIA THAT

DIFFUSIVITY *DEL(T)/DEL(Z)**2  SHOULD BE LESS THAN  .5

ON THIS PARTICULAR TIME STEP AND AT THE SOIL SURFACE

    DIFFUSIVITY =  .3662E+03 (CM**2/HR )
         DEL(Z) =  .5000E+00 (CM)                (26)
         DEL(T) =  .2000E+00 (HR )

HENCE, DIFFUSIVITY*DEL(T)/DEL(Z)**2 =   .2929E+03
```

Figure 6.2 (continued)

Point 1) Program OR-NATURE prints out additional information about the flux iterator whenever a flux type (4th) boundary condition is specified for the water flow equation at the upper soil surface, z = Ø. It is apparent in this example that as the specified, i.e., potential, value of the surface flux increased from .3 to 1.9 cm/hr, the program required more and more iterations to find the true or actual surface flux. Under most conditions only one or two maxi iterations are needed and less than 20 mini iterations are needed for each time step. Details of the mini-iterator are given in Vol. I, Section 3.4.2. Near the time of failure, there is a dramatic increase in the number of iterations. The failure in this example occurred when the potential flux became very large, that is, larger than 1.9 cm/hr.

Point 2) The error code is number 61. Error code 61 is described in Section 6.2.5 as a failure of the water flux iterator to converge.

Point 3) The error message originated in the flux iterator scheme of subroutine WATER. Additional details of the iterator scheme are given in Vol. I, Section 3.4.

Point 4) During this phase of the error message, the program will allow 25 additional iterations to occur before stopping. It is possible for the program to recover during these last few iterations if convergence is obtained.

Point 5) The variable ITR is defined as the total number of iterations executed by the program since the start of the current time step. It includes both mini and maxi iterations. Thus for this example, the Golden Section Search scheme has gone through a total of 475 iterations since the beginning of the current time step.

Point 6) As the program searches for the actual water flux, the Golden Section Search method "brackets" the numerical value of fluxes over which it will search. The upper limit of this flux bracket, based on the absolute value, is called QGBK and the lower limit of this flux bracket is called QGAK. For example, on iteration number 477, the iterator searched over the range of fluxes between 1.77 and 1.81 cm/hr. This bracket will normally shrink with each additional iteration. On iteration number 484, the bracket was reduced to the range of 1.78 to 1.79 cm/hr. However, the program computed an unfeasible value of water content at node number two during this iteration. Since the range of the flux bracket was less than one percent, i.e.,

$$|(QGBK-QGAK)/QG| \leq QRATIO, \qquad (6.3)$$

the program assumed the actual value of flux was outside the bracketed range. On iteration number 485, the bracket was enlarged 20 percent above and below the previous "bracket ends" to give a new range of 1.42 to 2.1 cm/hr. Once again the Golden Section Search routine reduced the range of the flux bracket. Additional details are given in Vol. I, Section 3.4.

Point 7) During each flux iteration, the program must guess the actual value of the flux boundary condition to solve the water flow equation. This guessed value of flux is called QG. After the water flow equation has been solved, the program numerically recomputes the boundary flux, given the computed value of water content. This computed flux is called QC. For example, on iteration number 475, the program guessed the flux to be 1.8 cm/hr but after solving the water flow equation for this iteration, it recomputed the flux to be 1.77 cm/hr.

Point 8) The numerical value of the boundary condition used during a given iteration is called FTOP. For fourth type boundary conditions, FTOP equals QG.

Point 9) The variable RATIOG is defined in Vol. II, Section 5.2 as the percent difference between the guessed and computed values of water flux during any given iteration, where

$$RATIOG = |(QC-QG)/QG|*5. \qquad (6.4)$$

The computation of RATIOG is multiplied by five to "exaggerate the difference between the guessed and computed flux values. This "exaggeration" forces the iterator to give greater accuracy between QG and QC. During the last 25 iterations of this example, f this example, RATIOG ranged from a high of 1. to a low of 2.8×10^{-13}. The condition that RATIOG must be less than QRATIO is a necessary but not sufficient condition for convergence.

Point 1∅) The variable RATIOQ is defined in Vol. II, Section 5.2, as the percent difference between the potential and computed values of water flux during any given iteration, where

$$RATIOQ = |(QP-QC)/QP|. \qquad (6.5)$$

Vol. I, Section 3.4 states that the iteration scheme attempts to make RATIOQ as small as possible without making any of the computed water contents become unfeasible. From time equal to 1∅4 hours until the current time step, RATIOQ was less than QRATIO, i.e., .∅1. However, during the current time step, RATIOQ is greater than .15. It is obvious that the specified or potential infiltration rate of 2.1 cm/hr exceeds the rate of soil infiltrability. The soil has gone from a "flux-controlled" condition to a "profile-controlled" condition.

Point 11) The variable RWORST is defined in Vol. II, Section 5.2, as the largest percent difference between the computed and guessed values of water content of any node during a given iteration. Thus,

$$RWORST = |\{\theta_i(\text{compute})-\theta_i(\text{guess})\}/\theta_i(\text{guess})|_{max} \qquad (6.6)$$

$$i = 1,\ldots,NITR.$$

The node i at which the maximum occurs is called IWORST. A necessary but not sufficient condition for convergence is that RWORST must be less than WRATIO. During the last 25 iterations, RWORST was always less than WRATIO, i.e., .∅1. Thus, failure to converge is not related to RWORST.

Point 12) The variable IWORST is defined above in point 11).

Point 13) The computed value of water content at node IWORST is defined as

$$FLOW(IWORST) = \theta_i(\text{compute}),\ i = IWORST \qquad (6.7)$$

Note that the IWORST nodes are approaching saturation, where $\theta_{sat} = .38$. Additional discussion of the significance of

FLOW(IWORST) during iterations 484, 488, 490, 492, 498, and 500 is given below in point 16) with the discussion of MQFAIL.

Point 14) The variable M is defined in Vol. II, Section 5.2, as the number of iterations required to compute the water content for any node at a given step and boundary condition. Since RWORST was always smaller than WRATIO during the last 25 iterations, there never was any need to iterate on water content. Thus, M is less than or equal to one.

Point 15) The variable MQ is defined in Vol. II, Section 5.2, as the current number of iterations used by the flux iterator. A negative value of MQ indicates that the "mini-iterator" scheme is being used to solve for the water content, see Vol. I, Section 3.4.2. A positive value of MQ indicates the program is in the "maxi" mode of iteration for water content. The value of MQ is reinitiated to zero when the program switches from the mini to the maxi mode and vice versa. Such reversals occurred at iteration numbers 484 and 498. The user can tell whether the program is in a mini or maxi mode by looking at the value of NITR (point 17 below). At iteration number 475, MQ equaled -13, i.e., the thirteenth mini iteration. However, the mini-iterator scheme was reinitiated because of two special conditions: the percent difference between QGAK and QGBK was less than QRATIO (see point 6) and the computed flux QC was less than the lower limit QGAK. Thus, the Golden Section Search routine had suddenly jumped outside of what it had thought was the "optimal" flux bracket. On iteration number 476, the lower limit of the flux bracket, QGAK, was lowered from 1.79 to 1.77 cm/hr, MQ was set equal to -1, and the guessed flux QG was set equal to QGAK. On iteration 483, the program was in the mini mode when the flux iterator converged "locally". The variable RATIOG is less than QRATIO and the flux bracket QGBK-QGAK is less than QRATIO. Hence, the actual surface flux equals 1.78 cm/hr. The program, however, did not stop iterating because "global" convergence can only occur when in the maxi mode. During the mini mode, only the first 10 nodes are being solved for, whereas, during the maxi mode all 100 nodes are solved for. The program switched to the maxi mode on iteration number 484 but failed to converge. The water content of the second node became unfeasible, that is, greater than saturation, when $\theta_2 = .3813$. The iterator returned to the mini mode on the next iteration and enlarged the lower and upper flux limits by 20 percent.

Point 16) The variable MQFAIL is defined in Vol. II, Section 5.2. It indicates for which of the top three nodes, if any, a water content greater than saturation or less than air-dry conditions was computed. On iteration number 484, the second node had a computed volumetric water content of FLOW(IWORST) = .3813. Since saturation is equal to .38, an unfeasible solution was obtained during the maxi mode of iteration. An unfeasible solution during the maxi mode is not tolerated and the program switched to the mini mode for the next iteration. The program assumes that the true surface flux is not yet known and that it is a waste of time to iterate for the true flux while using all of the nodes. Additional unfeasible solutions occurred during iterations 486, 488, 490, 492, 498, and 500.

Point 17) The variable NITR is defined in Vol. II, Section 5.2, as the number of nodes being solved by the water flow routine during any given iteration. During the maxi mode, such as during iterations number 484 and 498, the program solves the water flow equation using all specified spatial nodes, i.e., 100. During the mini mode, only the first 10 or 12 spatial nodes are used in the water flow routine. Additional details are given in Vol. I, Section 3.4.2.

Point 18) The error code is number 62. Error code 62 is defined in Section 6.2.5 as a fatal error. The flux iterator failed to converge and the program aborted.

Point 19) The error message originated in the flux iterator scheme of subroutine WATER.

Point 20) The most recent values of 41 variables are printed out at the end of iteration number 5ØØ. Some of the variables merely "echo" what the user specified and others give details of the iteration scheme. The user should carefully check the "echo" variables to make sure that their values are correct. All variables are defined in Vol. II, Section 5.2. Additional discussion of some of these variables is given above in points 5) to 17).

Point 21) The units in which the water flow equation is being solved are listed. The water flow equation can be solved in terms of volumetric water content, matric potential, and diffusivity potential.

Point 22) Four variables are listed across the page, at the top of each of the seven listed arrays. The first variable indicates the TIME step relative to the current one. The current time step is K+1 and its actual numerical value is defined in the program as K1, i.e., K1 = 53Ø. The previous time step is called K, etc. Each of the seven arrays was generated during a particular ITERATION or time step. For example, the guessed and computed values of water listed in columns three and four were generated during iteration number 498, the guessed and computed values listed in columns five and six came from iteration number 499, and the guessed values of the seventh column came from the last iteration before failure, iteration number 5ØØ. The STATUS label indicates which variable is being printed. In the first column, the final computed or actual values of water are given for time step (k-1). In the second column, the actual water values of the previous time step are given. In column three, the guessed values of water for iteration number 498 are given for the current time step. In column four, the computed values of water for iteration number 498 are given for the current time, etc. On the line labeled NODE I, the actual names of the variables used in subroutine WATER are listed. Thus, the water content of time step (k-1) are stored in array OLD1(i), i = 1,...,N, the water contents of the previous time step are stored in array OLD2(i), etc.

Point 23) In the first column the volumetric water contents during time step (k-1) are listed. Saturation is defined at $\theta_{sat} = .38$. Note that there is a monotonically decreasing water content from the first node to that of the fifth node. The sixth node has a higher water content than those of nodes above and below it. Since there had been a continuous infiltration into the soil surface for the last several time steps, the higher value at node six is probably due to numerical error. Nevertheless, this apparent error is well within the one percent error criterion specified by parameter WRATIO. Other apparent anomalies appear at nodes 16, 25, and 48 but these are probably from the cyclic nature of the specified surface flux. Earlier in the computation of this example, cyclic periods of evaporation and infiltration were specified.

Point 24) On the next to last time step, (k), additional oscillations appear at nodes 4, 8, 13, and 21. In addition, there is a dramatic increase in water content at node 21, when compared to the computed value at time step (k-1). It is obvious that the computational scheme is beginning to break down.

Point 25) It is obvious during iteration 498 of the current time step (k+1) that the program will fail. Both the computed and guessed water contents of nodes 11 and 12 have become saturated. It is clear that numerical oscillations have become a dominant driving force in the program and convergence will become impossible.

Point 26) After printing the water content values of the seven arrays, the program checks if a certain stability criterion for space and time discretization was violated. The criterion

$$\Delta t * D / \Delta z^2 \leq .5 \qquad (6.8)$$

is discussed further in Section 6.4. Strictly speaking, the Crank-Nicolson method is unconditionally stable for all ratios of space and time increments. However, the relative effect of the space and time increments can be better assessed with this criterion. For the problem given in Fig. 6.2, the combination of space and time step-sizes gives a value of 293. This is several hundred times larger than the suggested criterion of .5. Either the time increment should be reduced or the space increment at the upper surface should be increased.

This example of failure to converge is very typical in that the solution to correct the problem is not readily apparent. A great deal of information can be obtained from the data listed by the program at the time of failure but the user is still faced with the problem of deciphering it. The key to the solution of this problem can be found in examining points 1), 6), 13), 15), 24), 25), and 26). The failure to converge occurred when the magnitude of the infiltration became very large. Many possible explanations can be discarded by the fact that the program had been running a long time before failing. In addition, the Golden Section Search iterator was converging to a flux value of about 1.78 cm/hr but it could never really zero in on an "optimal" value. Some of the nodes were becoming saturated near the surface. Oscillations were beginning to develop in the numerical solution and were becoming larger and larger with each additional time step. Upon comparing the stability criterion, it was also obvious that the relative magnitude between the space and time increments was too extreme.

The solution to this failure to converge example was simple. The time increment was reduced from Ø.2 to Ø.1 days. Upon resubmitting the program, the example ran without difficulty. The larger time increment was causing numerical oscillations to build up in the solution. This in turn made the Golden Section Search iterator unstable.

6.4 Stability

A finite-difference method is considered to be "stable" if the amplification of the numerical errors is in some sense restricted as the computation moves forward (Remson et al., 1971). Errors are introduced by the truncation of the Taylor series which are used to represent the derivative in the process of replacing the differential equation by finite-difference equations. Both the Crank-Nicolson method and the backward-difference method used on the boundary conditions are unconditionally stable for all ratios of Δz to Δt (von Rosenberg, 1969). However, no guarantee is made about either the occurrence of numerical oscillations or the convergence of the guess-compute iteration routine.

To reduce numerical oscillations, the following criterion for Δz and Δt should be satisfied:

$$\Delta t * D_{max} / \Delta z^2 \leq 0.5 \qquad (6.9)$$

where D_{max} is the maximum (i.e., saturated) value of soil water diffusivity. The above criterion is obtained from the "Fourier series method" analysis for explicit finite-difference solutions. The Crank-Nicolson method is a central-difference approximation and the backward-difference method is an implicit scheme. Hence, the above criterion is not required for convergence and stability but it does give an approximate or "desirable" criterion for specifying Δz and Δt.

6.5 Numerical Accuracy of the Water Flow Equation

To check the numerical accuracy of the finite-difference solution of the water flow equation, a comparison was made against the quasi-analytical solution of Philip (1957). However, the exact accuracy of OR-NATURE cannot be computed because Philip's solution is also numerical. Philip's solution serves only as a relative benchmark from which comparisons can be made.

When a constant, saturated boundary condition is specified, Equation 1.3a, Vol. I, can be solved and its solution expressed in the form of a power series:

$$z(\theta,t) = \sum_{n=1}^{N} f_n(\theta)\, t^{n/2} \qquad (6.10)$$

The term $z(\theta,t)$ is the calculated depth for a given value of volumetric water content, θ; time, t; and the coefficients $f_n(\theta)$ are solutions to a series of simultaneous, ordinary differential equations. Details of the numerical scheme and a FORTRAN computer program to solve Philip's coefficients are given in Vauclin et al. (1979).

The sample problem chosen to compare with Philip's solution is one of flow irrigation into Yolo Light clay. Flow irrigation corresponds to a constant, saturated boundary condition at the upper soil surface (θ_s = .495, ψ_s = ∅). The lower boundary is assumed to be semi-infinite ($\partial\psi/\partial z$ = ∅). The soil column is assumed to be 96 cm long and is initially assumed to have a matric potential of -629.55 cm H_2O (θ_0 = .239).

The soil properties and the numerical solution of OR-NATURE are given in Vol. IV, Section 7.4, for three simulation times, t = 2.777, 11.1∅8, and 27.77 hours. A list of all the specified parameters for the computer program given by Vauclin et al. (1979) for Philip's quasi-analytical solution is given in Table 6.4. Computer output listings of Philip's solution for simulation times 2.777, 11.1∅8, and 27.77 hours are given in Figures 6.3, 6.4, and 6.5. OR-NATURE's solution for these same simulation times are given in Figures 7.54, 7.55, and 7.56 of Vol. IV.

Table 6.4 Specified parameters for the computer program given by Vauclin et al. (1979) for Yolo Light clay.

Computer Program Parameters
A = 124.6
AKS = .∅4428
ALFA = 738.8
B = 1.77
BETA = 3.98
DT = 2.777
KEY = 2
N = 1∅∅
TETA1 = .495
TETAI = .2376
TETAR = .125
TETAS = .495

In Figures 6.3 to 6.5, there are five columns of numbers on each side of the page labeled Z1, Z2, Z3, Z, and TETA. Z1 corresponds to the first term of the series in Equation 6.1∅, Z2 corresponds to the sum of the first and second terms of Equation 6.1∅, and Z3 is the sum of the first, second, and third terms. Z is the computed depth of soil corresponding to the volumetric water content, TETA (cm^3/cm^3). Z is equal

TIME = 2.777 (HOURS) CUMI = 1.20222(CM) IRATE = 0.22437(CM/H)

Z1	Z2	Z3	Z (CM)	TETA	Z1	Z2	Z3	Z (CM)	TETA
0.0000	0.0000	0.0000	0.0000	0.4950	0.4484	0.5053	0.5108	0.5112	0.4924
0.7264	0.8149	0.8229	0.8234	0.4899	0.9528	1.0644	1.0738	1.0744	0.4873
1.1506	1.2805	1.2909	1.2914	0.4847	1.3292	1.4738	1.4847	1.4853	0.4821
1.4932	1.6499	1.6612	1.6617	0.4796	1.6456	1.8123	1.8236	1.8241	0.4770
1.7881	1.9631	1.9744	1.9748	0.4744	1.9223	2.1041	2.1153	2.1157	0.4718
2.0490	2.2365	2.2475	2.2478	0.4693	2.1692	2.3613	2.3721	2.3724	0.4667
2.2836	2.4794	2.4899	2.4902	0.4641	2.3925	2.5913	2.6016	2.6018	0.4615
2.4966	2.6978	2.7078	2.7079	0.4590	2.5962	2.7992	2.8089	2.8091	0.4564
2.6917	2.8960	2.9054	2.9056	0.4538	2.7834	2.9886	2.9978	2.9979	0.4512
2.8715	3.0774	3.0863	3.0864	0.4487	2.9564	3.1625	3.1712	3.1713	0.4461
3.0383	3.2444	3.2528	3.2529	0.4435	3.1173	3.3232	3.3314	3.3315	0.4409
3.1937	3.3992	3.4072	3.4073	0.4384	3.2676	3.4726	3.4803	3.4804	0.4358
3.3391	3.5434	3.5510	3.5511	0.4332	3.4086	3.6120	3.6194	3.6195	0.4307
3.4759	3.6785	3.6857	3.6858	0.4281	3.5414	3.7430	3.7500	3.7501	0.4255
3.6051	3.8055	3.8125	3.8125	0.4229	3.6671	3.8664	3.8732	3.8732	0.4204
3.7275	3.9256	3.9322	3.9323	0.4178	3.7864	3.9832	3.9898	3.9898	0.4152
3.8439	4.0394	4.0459	4.0459	0.4126	3.9001	4.0943	4.1006	4.1007	0.4101
3.9550	4.1479	4.1541	4.1542	0.4075	4.0088	4.2003	4.2064	4.2065	0.4049
4.0614	4.2516	4.2576	4.2577	0.4023	4.1130	4.3018	4.3077	4.3078	0.3998
4.1636	4.3510	4.3568	4.3569	0.3972	4.2133	4.3993	4.4050	4.4051	0.3946
4.2621	4.4467	4.4524	4.4525	0.3920	4.3101	4.4933	4.4989	4.4990	0.3895
4.3574	4.5391	4.5446	4.5447	0.3869	4.4039	4.5842	4.5896	4.5898	0.3843
4.4497	4.6286	4.6340	4.6341	0.3817	4.4949	4.6724	4.6777	4.6778	0.3792
4.5395	4.7155	4.7209	4.7210	0.3766	4.5835	4.7582	4.7634	4.7636	0.3740
4.6271	4.8003	4.8055	4.8056	0.3714	4.6701	4.8420	4.8471	4.8472	0.3689
4.7127	4.8832	4.8883	4.8884	0.3663	4.7549	4.9240	4.9291	4.9292	0.3637
4.7967	4.9645	4.9695	4.9696	0.3612	4.8382	5.0046	5.0096	5.0097	0.3586
4.8794	5.0444	5.0493	5.0495	0.3560	4.9203	5.0839	5.0888	5.0890	0.3534
4.9610	5.1233	5.1281	5.1282	0.3509	5.0014	5.1624	5.1672	5.1673	0.3483
5.0417	5.2013	5.2061	5.2062	0.3457	5.0818	5.2401	5.2449	5.2450	0.3431
5.1219	5.2788	5.2836	5.2837	0.3406	5.1618	5.3175	5.3222	5.3223	0.3380
5.2018	5.3561	5.3607	5.3608	0.3354	5.2417	5.3947	5.3993	5.3994	0.3328
5.2816	5.4333	5.4379	5.4380	0.3303	5.3216	5.4720	5.4765	5.4767	0.3277
5.3617	5.5108	5.5153	5.5154	0.3251	5.4020	5.5497	5.5542	5.5544	0.3225
5.4424	5.5889	5.5933	5.5935	0.3200	5.4831	5.6282	5.6327	5.6328	0.3174
5.5241	5.6679	5.6723	5.6724	0.3148	5.5654	5.7079	5.7122	5.7124	0.3122
5.6070	5.7482	5.7526	5.7527	0.3097	5.6492	5.7890	5.7933	5.7934	0.3071
5.6918	5.8303	5.8346	5.8347	0.3045	5.7350	5.8722	5.8764	5.8765	0.3020
5.7789	5.9147	5.9189	5.9190	0.2994	5.8235	5.9579	5.9621	5.9622	0.2968
5.8689	6.0020	6.0061	6.0063	0.2942	5.9153	6.0470	6.0511	6.0512	0.2917
5.9628	6.0930	6.0971	6.0972	0.2891	6.0114	6.1402	6.1443	6.1444	0.2865
6.0614	6.1888	6.1928	6.1929	0.2839	6.1130	6.2389	6.2429	6.2430	0.2814
6.1663	6.2907	6.2946	6.2947	0.2788	6.2217	6.3445	6.3484	6.3485	0.2762
6.2793	6.4005	6.4044	6.4045	0.2736	6.3396	6.4592	6.4630	6.4631	0.2711
6.4031	6.5210	6.5247	6.5249	0.2685	6.4703	6.5864	6.5901	6.5902	0.2659
6.5420	6.6562	6.6598	6.6600	0.2633	6.6190	6.7313	6.7349	6.7350	0.2608
6.7027	6.8129	6.8164	6.8166	0.2582	6.7949	6.9028	6.9063	6.9065	0.2556
6.8982	7.0037	7.0071	7.0072	0.2530	7.0165	7.1193	7.1227	7.1228	0.2505
7.1565	7.2563	7.2595	7.2596	0.2479	7.3305	7.4266	7.4297	7.4299	0.2453
7.5654	7.6568	7.6598	7.6599	0.2427	7.9418	8.0263	8.0290	8.0291	0.2402
9.0477	9.1142	9.1163	9.1164	0.2376	0.0000	0.0000	0.0000	0.0000	0.0000

Figure 6.3 Computer output for Philip's solution at t = 2.777 hours.

TIME = 11.108 (HOURS) CUMI = 2.49569(CM) IRATE = 0.12086(CM/H)

Z1	Z2	Z3	Z (CM)	TETA	Z1	Z2	Z3	Z (CM)	TETA
0.0000	0.0000	0.0000	0.0000	0.4950	0.8969	1.1244	1.1680	1.1745	0.4924
1.4529	1.8067	1.8704	1.8789	0.4899	1.9055	2.3520	2.4277	2.4367	0.4873
2.3013	2.8206	2.9037	2.9127	0.4847	2.6584	3.2368	3.3243	3.3328	0.4821
2.9864	3.6133	3.7032	3.7109	0.4796	3.2911	3.9580	4.0487	4.0558	0.4770
3.5762	4.2762	4.3668	4.3731	0.4744	3.8445	4.5719	4.6615	4.6671	0.4718
4.0981	4.8480	4.9361	4.9411	0.4693	4.3385	5.1069	5.1932	5.1976	0.4667
4.5671	5.3504	5.4347	5.4386	0.4641	4.7850	5.5803	5.6624	5.6659	0.4615
4.9932	5.7979	5.8778	5.8809	0.4590	5.1924	6.0044	6.0820	6.0847	0.4564
5.3834	6.2007	6.2761	6.2786	0.4538	5.5667	6.3878	6.4610	6.4633	0.4512
5.7431	6.5664	6.6376	6.6397	0.4487	5.9128	6.7373	6.8065	6.8084	0.4461
6.0765	6.9011	6.9684	6.9702	0.4435	6.2346	7.0584	7.1238	7.1255	0.4409
6.3873	7.2095	7.2733	7.2749	0.4384	6.5351	7.3551	7.4172	7.4188	0.4358
6.6783	7.4955	7.5561	7.5576	0.4332	6.8171	7.6310	7.6902	7.6917	0.4307
6.9519	7.7621	7.8199	7.8214	0.4281	7.0828	7.8890	7.9455	7.9470	0.4255
7.2102	8.0120	8.0673	8.0688	0.4229	7.3341	8.1314	8.1856	8.1871	0.4204
7.4550	8.2473	8.3005	8.3020	0.4178	7.5728	8.3602	8.4123	8.4138	0.4152
7.6878	8.4700	8.5213	8.5228	0.4126	7.8001	8.5771	8.6275	8.6290	0.4101
7.9100	8.6816	8.7311	8.7327	0.4075	8.0175	8.7836	8.8324	8.8340	0.4049
8.1228	8.8834	8.9314	8.9330	0.4023	8.2260	8.9810	9.0284	9.0300	0.3998
8.3273	9.0767	9.1233	9.1250	0.3972	8.4266	9.1704	9.2165	9.2181	0.3946
8.5243	9.2624	9.3079	9.3096	0.3920	8.6203	9.3527	9.3977	9.3994	0.3895
8.7148	9.4415	9.4859	9.4876	0.3869	8.8078	9.5289	9.5728	9.5745	0.3843
8.8994	9.6149	9.6583	9.6600	0.3817	8.9898	9.6996	9.7426	9.7444	0.3792
9.0790	9.7832	9.8257	9.8275	0.3766	9.1671	9.8657	9.9078	9.9096	0.3740
9.2541	9.9471	9.9888	9.9907	0.3714	9.3402	10.0277	10.0690	10.0708	0.3689
9.4254	10.1073	10.1483	10.1501	0.3663	9.5098	10.1862	10.2268	10.2287	0.3637
9.5935	10.2643	10.3046	10.3065	0.3612	9.6764	10.3419	10.3818	10.3837	0.3586
9.7588	10.4188	10.4584	10.4603	0.3560	9.8406	10.4952	10.5344	10.5364	0.3534
9.9219	10.5711	10.6101	10.6120	0.3509	10.0028	10.6466	10.6853	10.6873	0.3483
10.0834	10.7219	10.7603	10.7622	0.3457	10.1637	10.7968	10.8349	10.8369	0.3431
10.2438	10.8716	10.9094	10.9114	0.3406	10.3237	10.9462	10.9838	10.9857	0.3380
10.4035	11.0207	11.0580	11.0600	0.3354	10.4833	11.0953	11.1323	11.1343	0.3328
10.5032	11.1699	11.2067	11.2087	0.3303	10.6432	11.2447	11.2812	11.2832	0.3277
10.7235	11.3197	11.3559	11.3579	0.3251	10.8040	11.3949	11.4309	11.4330	0.3225
10.8849	11.4706	11.5063	11.5084	0.3200	10.9662	11.5467	11.5822	11.5842	0.3174
11.0482	11.6234	11.6586	11.6606	0.3148	11.1307	11.7007	11.7357	11.7377	0.3122
11.2141	11.7787	11.8135	11.8155	0.3097	11.2983	11.8577	11.8922	11.8942	0.3071
11.3836	11.9376	11.9718	11.9739	0.3045	11.4700	12.0186	12.0526	12.0547	0.3020
11.5578	12.1010	12.1347	12.1367	0.2994	11.6470	12.1847	12.2182	12.2202	0.2968
11.7379	12.2701	12.3033	12.3053	0.2942	11.8306	12.3573	12.3902	12.3923	0.2917
11.9256	12.4465	12.4792	12.4812	0.2891	12.0229	12.5381	12.5705	12.5725	0.2865
12.1229	12.6323	12.6644	12.6664	0.2839	12.2260	12.7295	12.7613	12.7633	0.2814
12.3326	12.8301	12.8616	12.8636	0.2788	12.4433	12.9345	12.9657	12.9677	0.2762
12.5586	13.0434	13.0743	13.0763	0.2736	12.6793	13.1575	13.1880	13.1900	0.2711
12.8062	13.2776	13.3078	13.3098	0.2685	12.9406	13.4049	13.4347	13.4367	0.2659
13.0839	13.5408	13.5701	13.5721	0.2633	13.2380	13.6870	13.7159	13.7179	0.2608
13.4054	13.8461	13.8746	13.8765	0.2582	13.5898	14.0216	14.0495	14.0514	0.2556
13.7963	14.2184	14.2457	14.2476	0.2530	14.0330	14.4443	14.4710	14.4728	0.2505
14.3131	14.7120	14.7380	14.7398	0.2479	14.6611	15.0454	15.0704	15.0722	0.2453
15.1307	15.4964	15.5202	15.5219	0.2427	15.8836	16.2216	16.2436	16.2452	0.2402
18.0954	18.3613	18.3781	18.3793	0.2376	0.0000	0.0000	0.0000	0.0000	0.0000

Figure 6.4 Computer output for Philip's solution at t = 11.1Ø8 hours.

TIME = 27.770 (HOURS) CUMI = 4.12775(CM) IRATE = 0.08361(CM/H)

Z1	Z2	Z3	Z (CM)	TETA	Z1	Z2	Z3	Z (CM)	TETA
0.0000	0.0000	0.0000	0.0000	0.4950	1.4181	1.9869	2.1594	2.2001	0.4924
2.2972	3.1818	3.4334	3.4866	0.4899	3.0129	4.1291	4.4282	4.4849	0.4873
3.0386	4.9370	5.2655	5.3214	0.4847	4.2033	5.6493	5.9952	6.0481	0.4821
4.7220	6.2891	6.6443	6.6930	0.4796	5.2037	6.8709	7.2295	7.2736	0.4770
5.6545	7.4045	7.7624	7.8019	0.4744	6.0787	7.8972	8.2514	8.2864	0.4718
6.4796	8.3545	8.7028	8.7339	0.4693	6.8598	8.7807	9.1219	9.1493	0.4667
7.2212	9.1796	9.5127	9.5370	0.4641	7.5658	9.5541	9.8786	9.9001	0.4615
7.8949	9.9068	10.2225	10.2416	0.4590	8.2099	10.2398	10.5467	10.5638	0.4564
8.5118	10.5551	10.8533	10.8687	0.4538	8.8018	10.8544	11.1440	11.1580	0.4512
9.0806	11.1390	11.4204	11.4333	0.4487	9.3490	11.4103	11.6837	11.6957	0.4461
9.6079	11.6693	11.9352	11.9465	0.4435	9.8577	11.9172	12.1759	12.1866	0.4409
10.0992	12.1548	12.4067	12.4169	0.4384	10.3329	12.3829	12.6284	12.6383	0.4358
10.5593	12.6023	12.8418	12.8514	0.4332	10.7788	12.8136	13.0474	13.0568	0.4307
10.9919	13.0174	13.2459	13.2552	0.4281	11.1989	13.2143	13.4378	13.4470	0.4255
11.4003	13.4048	13.6235	13.6328	0.4229	11.5963	13.5893	13.8036	13.8129	0.4204
11.7873	13.7683	13.9784	13.9877	0.4178	11.9736	13.9421	14.1483	14.1577	0.4152
12.1554	14.1110	14.3136	14.3231	0.4126	12.3331	14.2755	14.4746	14.4842	0.4101
12.5068	14.4358	14.6316	14.6413	0.4075	12.6768	14.5921	14.7848	14.7947	0.4049
12.8433	14.7448	14.9346	14.9446	0.4023	13.0065	14.8940	15.0811	15.0912	0.3998
13.1665	15.0401	15.2246	15.2348	0.3972	13.3237	15.1831	15.3652	15.3755	0.3946
13.4781	15.3234	15.5031	15.5136	0.3920	13.6299	15.4610	15.6386	15.6492	0.3895
13.7792	15.5962	15.7717	15.7824	0.3869	13.9263	15.7291	15.9026	15.9134	0.3843
14.0712	15.8599	16.0315	16.0424	0.3817	14.2141	15.9887	16.1585	16.1695	0.3792
14.3552	16.1156	16.2837	16.2949	0.3766	14.4944	16.2409	16.4073	16.4186	0.3740
14.6321	16.3646	16.5294	16.5408	0.3714	14.7682	16.4868	16.6501	16.6616	0.3689
14.9029	16.6076	16.7695	16.7811	0.3663	15.0363	16.7273	16.8878	16.8994	0.3637
15.1686	16.8458	17.0049	17.0167	0.3612	15.2998	16.9633	17.1211	17.1329	0.3586
15.4300	17.0799	17.2365	17.2484	0.3560	15.5593	17.1958	17.3510	17.3630	0.3534
15.6879	17.3109	17.4650	17.4770	0.3509	15.8159	17.4254	17.5783	17.5904	0.3483
15.9433	17.5394	17.6912	17.7033	0.3457	16.0702	17.6530	17.8037	17.8159	0.3431
16.1968	17.7663	17.9159	17.9281	0.3406	16.3232	17.8795	18.0279	18.0402	0.3380
16.4494	17.9925	18.1399	18.1523	0.3354	16.5756	18.1055	18.2519	18.2643	0.3328
16.7019	18.2187	18.3640	18.3765	0.3303	16.8284	18.3321	18.4764	18.4889	0.3277
16.9553	18.4458	18.5891	18.6016	0.3251	17.0826	18.5600	18.7023	18.7149	0.3225
17.2105	18.6748	18.8161	18.8287	0.3200	17.3391	18.7903	18.9306	18.9432	0.3174
17.4687	18.9067	19.0460	19.0586	0.3148	17.5992	19.0241	19.1624	19.1751	0.3122
17.7310	19.1426	19.2800	19.2927	0.3097	17.8642	19.2626	19.3989	19.4116	0.3071
17.9991	19.3840	19.5194	19.5321	0.3045	18.1357	19.5072	19.6416	19.6543	0.3020
18.2744	19.6324	19.7657	19.7784	0.2994	18.4155	19.7598	19.8921	19.9048	0.2968
18.5592	19.8898	20.0210	20.0337	0.2942	18.7059	20.0225	20.1527	20.1654	0.2917
18.8560	20.1584	20.2875	20.3002	0.2891	19.0098	20.2980	20.4259	20.4386	0.2865
19.1680	20.4416	20.5684	20.5810	0.2839	19.3310	20.5897	20.7154	20.7280	0.2814
19.4996	20.7432	20.8676	20.8802	0.2788	19.6746	20.9026	21.0258	21.0384	0.2762
19.8569	21.0689	21.1908	21.2034	0.2736	20.0477	21.2433	21.3638	21.3763	0.2711
20.2484	21.4270	21.5461	21.5585	0.2685	20.4609	21.6217	21.7393	21.7517	0.2659
20.6875	21.8297	21.9457	21.9580	0.2633	20.9311	22.0537	22.1680	22.1802	0.2608
21.1958	22.2977	22.4101	22.4222	0.2582	21.4874	22.5668	22.6773	22.6892	0.2556
21.8139	22.8691	22.9772	22.9891	0.2530	22.1881	23.2162	23.3218	23.3335	0.2505
22.6310	23.6283	23.7309	23.7423	0.2479	23.1812	24.1419	24.2408	24.2520	0.2453
23.9238	24.8379	24.9320	24.9428	0.2427	25.1141	25.9593	26.0459	26.0560	0.2402
28.6114	29.2761	29.3423	29.3502	0.2376	0.0000	0.0000	0.0000	0.0000	0.0000

Figure 6.5 Computer output for Philip's solution at t = 27.77 hours.

to the sum of the 1st, 2nd, 3rd and 4th series terms of Equation 6.1Ø. For example, at time equal to 2.777 hours, the volumetric water content is equal to .4924 (cm^3/cm^3) at a depth of .5112 cm. At the top of Figures 6.3 to 6.5, the term CUMI equals the accumulated mass (cm) of water that has entered the soil column since time zero and IRATE equals the rate of water infiltration (cm/hr) into the upper soil surface.

A comparison between Philip's and OR-NATURE's numerical computations of CUMI and IRATE is given in Tables 6.5 to 6.8 for various simulation times, space discretizations, and time intervals. The variable CUMI is computed by program OR-NATURE by subtracting the value of WCMPTE at any given simulation time from that of WCMPTE at time zero ($WCMPTE_0$ = 22.9292 cm). The variable IRATE is assumed to equal the Darcian flux of variable QZ(1) in program OR-NATURE. QZ(1) is located at node (1+1/2) and is evaluated at time step (k+1/2). In contrast, IRATE is equivalently defined in Philip's program at node (1) and at time step (k+1).

Table 6.5 A comparison between Philip's and OR-NATURE's numerical solutions at time = 2.777 hours

	Δz (cm)	Δt (hr)	CUMI (cm)	IRATE (cm/hr)
Philips	*	*	1.2Ø22	.2236
OR-NATURE	.2	.Ø6943	1.23Ø3	.2374
	.5	.Ø6943	1.2928	.2396
	.5	.1388	1.2983	.2369
	.5†	.2777	1.3Ø39	.2384
	.5	.5554	1.3299	.2642

* $\Delta\theta = (\theta_s - \theta_n)/100$ = .ØØ257 where θ_n = .2376 and θ_s = .495.

† example problem given in Section 7.4, Vol. IV.

Table 6.6 A comparison between Philip's and OR-NATURE's numerical solutions at time = 11.1Ø8 hours

	Δz	Δt	CUMI	IRATE
	cm	--hr--	--cm--	cm/hr
Philip	*	*	2.4957	.12Ø9
OR-NATURE	.2	.Ø6943	2.7189	.1275
	.5	.Ø6943	2.6686	.1282
	.5	.1388	2.6719	.128Ø
	.5†	.2777	2.6748	.1279
	.5	.5554	2.69Ø1	.1276

* $\Delta\theta = (\theta_s - \theta_n)/100$ = .ØØ257 where θ_n = .2376 and θ_s = .495.

† example problem given in Section 7.4, Vol. IV.

Table 6.7 A comparison between Philip's and OR-NATURE's numerical solutions at time = 27.77 hours

	Δz	Δt	CUMI	IRATE	COMPUTATIONAL EFFORT**
	cm	--hr--	--hr--	cm/hr	
Philip	*	*	4.1277	.Ø836	.ØØ29
OR-NATURE	.2	.Ø6943	4.4Ø27	.Ø8824	.ØØ6Ø7
	.5	.Ø6943	4.4ØØ8	.Ø885	.ØØ642
	.5	.1388	4.4Ø32	.Ø885	.ØØ831
	.5†	.2777	4.4Ø5Ø	.Ø884	.Ø1297
	.5	.5554	4.416Ø	.Ø884	.Ø1729

† example problem given in Section 7.4, Vol. IV.

* $\Delta\theta = (\theta_s - \theta_n)/100$ = .ØØ257 where θ_n = .2376 and θ_s = .495

**computational effort = (CPU time)/(# of space nodes • # of time steps) where CPU is the central processing units or total execution time of the program in seconds.

Table 6.8 A comparison between Philip's and OR-NATURE's numerical solutions at time = 138.85 hours

	Δz	Δt	CUMI	IRATE
	cm	--hr--	--cm--	cm/hr
Philip	*	*	1Ø.8555	.Ø514
OR-NATURE	.5†	.2777	11.519	.0536

† example problem given in Section 7.4, Vol. IV.

* $\Delta\theta = (\theta - \theta_n)/1ØØ = .ØØ257$ where $\theta_n = .2376$ and $\theta_s = .495$

In Philip's solution, the initial range of water content is subdivided into one hundred increments. In program OR-NATURE, four different constant time increments and two different variable space increments were tried. In Tables 6.5 to 6.8, a space increment of $\Delta z = .2$ indicates the following spatial discretization in program OR-NATURE:

Δz =	.2 cm	$Ø \le z \le 5$
	.5 cm	$5 \le z \le 2Ø$
	1. cm	$20 \le z \le 4Ø$
	2. cm	$4Ø \le z \le 96$

A space increment of $\Delta z = .5$ in Tables 6.5 to 6.8 indicates a spatial discretization in program OR-NATURE of:

Δz =	.5 cm	$Ø \le z < 5$
	1. cm	$5 \le z < 4Ø$
	2. cm	$4Ø \le z \le 96$

Upon examining Tables 6.5 to 6.8, it is apparent that both Philip and OR-NATURE give similar results. In fact, plotting Philip's volumetric water content from Figures 6.3 to 6.5 onto OR-NATURE's Figure 7.57

of Vol. IV gives almost identical profiles. However, when comparing the variable CUMI of Tables 6.5 to 6.8, Philip's solution has 2 to 7% less accumulated mass of water in the soil column. The infiltration rates, IRATE, also differ about 6% from that of OR-NATURE. The differences between Philip and OR-NATURE remain more or less the same over a simulation period of 138.85 hours.

A comparison of the computational effort between Philip's solution and OR-NATURE is given in Table 6.7 for the simulation time of 27.77 hours. The computational effort is defined as the total time to execute the program divided by the number of space nodes and then dividing this by the number of time increments. Philip's solution is approximately 2 to 6 times more efficient than program OR-NATURE. In addition, using small space and time increments is more efficient than using large time and space increments in program OR-NATURE because fewer iterations are needed for convergence.

As mentioned previously, the solutions of Philip and OR-NATURE both involve numerical integrations. Hence, they are both subject to various types of numerical errors. It is not possible to define the absolute accuracy of either program since no "exact" solution is known. However, for all practical purposes, both Philip and OR-NATURE give identical results. Philip's solution is obviously more computationally efficient than program OR-NATURE because Philip's solution is valid for only a special set of boundary conditions. Program OR-NATURE is a general purpose routine that can solve a large number of practical problems.

6.6 Numerical Accuracy of the Solute Displacement Equation

To check the numerical accuracy of the finite-difference solution of the solute displacement equation, a comparison was made against the analytical solution given by van Genuchten and Cleary (1979) and van Genuchten and Alves (1982). Unlike Philip's (1957) quasi-analytic solution in Section 6.5 for the water flow equation, van Genuchten gives a solution that can be used to compute the absolute accuracy of OR-NATURE.

Van Genuchten solves analytically the solute displacement equation, Eq. 1.34, Vol. I, for the special case of a semi-infinite soil column where the water content, pore-water velocity and dispersion coefficients are held constant. In addition, he allows for first-order decay and sorption where sorption is modeled as a linear equilibrium process. The upper soil surface is specified with a flux type boundary condition, whose value varies in a piecewise-constant manner. A pulse of solute with a concentration of $c_f=1$ g/cm^3 is applied to the upper soil surface for a period of $t_o=1.25$ days. After t_o, solute-free water is applied until the beginning of the third day, when the experiment is stopped. Complete details of the example are given in Vol. IV, Section 7.5. The final form of the solute displacement equation is given in Eq. 7.27, Vol. IV. Van Genuchten solves this equation by an integral transform technique. The final solution is given below in Eqs. 6.11 and 6.12, where c(z,t) is the solute concentration in the liquid phase. The terms exp(.) and erfc(c) represent the exponential function and the complementary error function, respectively.

$$c(z,t) = \begin{cases} c_f B(z,t) & 0 \le t \le t_o \quad [\mathrm{g/cm^3}] \\ c_f B(z,t) - c_f B(z,t-t_o) & t > t_o \quad [\mathrm{g/cm^3}] \end{cases} \qquad (6.11)$$

where

$$B(z,t) = \frac{v}{(v+u)} \exp \frac{(v-u)z}{2D_{sz}} \operatorname{erfc} \frac{Rz-ut}{2(D_{sz}Rt)^{1/2}} + \frac{v}{(v-u)} \exp \frac{(v+u)z}{2D_{sz}} \operatorname{erfc} \frac{Rz+ut}{2(D_{sz}Rt)^{1/2}} + \frac{v^2}{2D_{sz}k_1\theta} \exp \left(\frac{vz}{D_{sz}} - \frac{tk_1\theta}{R} \right) \operatorname{erfc} \frac{Rz+vt}{2(D_{sz}Rt)^{1/2}} \qquad (6.12)$$

The magnitude of the pore-water velocity is given by $v = |q_z/\theta|$, the retardation factor R is $R = (1+k_2/\theta)$, and where $u = v(1+4D_{sz}k_1/v^2)^{1/2}$. The remaining parameters are defined in Vol. IV, Table 7.18.

The solute concentration in the solid or sorbed phase is given by Eq. 7.26, Vol. IV as

$$S(z,t) = .28c(z,t) \qquad [g/cm^3] \qquad (6.13)$$

and the rate of solute decay is given by Eq. 7.12, Vol. IV as

$$G(z,t) = -.08c(z,t) \qquad [g/cm^3/day] \qquad (6.14)$$

The analytical solutions given by Eqs. 6.11 and 6.12 were programmed and the final results of the analytical solution for simulation times 1, 2, and 3 days are listed in Figures 6.6, 6.7, and 6.8. The numerical solution of OR-NATURE is given in Vol. IV, Section 7.5, for the same simulation times. OR-NATURE's solution for these simulation times are given in Figures 7.61, 7.62, and 7.63 of Vol. IV.

In Figures 6.6 to 6.8, there are six columns of numbers, showing the depth and concentration of each node. At the bottom of these figures, there are five variables, listed from (1) to (5). Line (1) corresponds to CCMPTE, which is the integral of θc versus soil depth, using a piecewise-constant integration scheme (Eq. 7.14, Vol. IV). Line (2) is the integral of θc versus soil depth using a second-order Taylor series expansion routine (Vol. I, Section 3.7.2). Line (3) corresponds to SCMPTE, which is the integral of S versus soil depth, using a piecewise-constant integration scheme (Eq. 7.3Ø, Vol. IV). Line (4) is the integral of S versus soil depth, using a second-order Taylor series expansion routine (Vol. I, Section 3.7.2). Line (5) is the integral of the solute flux that is entering the soil column versus time, (Eq. 7.33, Vol. IV).

Plotting the analytical solutions from Figures 6.6 to 6.8 onto OR-NATURE's solution in Vol. IV, Figure 7.64, gives essentially identical concentration profiles. The maximum or peak concentrations given by the analytical solution and OR-NATURE are compared in Table 6.9 at three different simulation times and at four different time increments. The peak concentrations and their spatial locations are identical when OR-NATURE uses a time step of .Ø1 days or smaller. When OR-NATURE uses a time increment of .Ø5 days or larger, numerical oscillations occur and the solution becomes unreliable.

ANALYTIC SOLUTION TO THE UNSTEADY-STATE,ONE-DIMENSIONAL CONVECTIVE-DISPERSIVE MASS TRANSPORT EQUATION
WITH LINEAR EQUILIBRIUM ADSORPTION AND DECAY

THE PARAMETERS ARE

D(DISPERSION)= 30.00(CM**2/DAY) K(DECAY)=.200(PER DAY) K(SORPTION)=.200(CM**3/G SOIL) Q(DARCIAN FLUX)= 16.00(CM/DAY)
RHO(BULK DENSITY)= 1.400(G SOIL/CM**3) TO(PULSE DURATION)= 1.250(DAYS) THETA(WATER CONTENT)=.4000(CM**3/CM**3)

AT TIME = 1.00 (DAYS)

Z= 0.0(CM)	C= .99627E+00	(G/CM**3)	Z= 0.5(CM)	C= .99379E+00	(G/CM**3)	Z= 1.0(CM)	C= .99131E+00	(G/CM**3)
Z= 1.5(CM)	C= .98883E+00	(G/CM**3)	Z= 2.0(CM)	C= .98635E+00	(G/CM**3)	Z= 2.5(CM)	C= .98387E+00	(G/CM**3)
Z= 3.0(CM)	C= .98138E+00	(G/CM**3)	Z= 3.5(CM)	C= .97888E+00	(G/CM**3)	Z= 4.0(CM)	C= .97637E+00	(G/CM**3)
Z= 4.5(CM)	C= .97383E+00	(G/CM**3)	Z= 5.0(CM)	C= .97126E+00	(G/CM**3)	Z= 6.0(CM)	C= .96597E+00	(G/CM**3)
Z= 7.0(CM)	C= .96038E+00	(G/CM**3)	Z= 8.0(CM)	C= .95427E+00	(G/CM**3)	Z= 9.0(CM)	C= .94739E+00	(G/CM**3)
Z= 10.0(CM)	C= .93936E+00	(G/CM**3)	Z= 11.0(CM)	C= .92970E+00	(G/CM**3)	Z= 12.0(CM)	C= .91783E+00	(G/CM**3)
Z= 13.0(CM)	C= .90304E+00	(G/CM**3)	Z= 14.0(CM)	C= .88459E+00	(G/CM**3)	Z= 15.0(CM)	C= .86172E+00	(G/CM**3)
Z= 16.0(CM)	C= .83372E+00	(G/CM**3)	Z= 17.0(CM)	C= .80002E+00	(G/CM**3)	Z= 18.0(CM)	C= .76032E+00	(G/CM**3)
Z= 19.0(CM)	C= .71462E+00	(G/CM**3)	Z= 20.0(CM)	C= .66330E+00	(G/CM**3)	Z= 22.0(CM)	C= .54734E+00	(G/CM**3)
Z= 24.0(CM)	C= .42285E+00	(G/CM**3)	Z= 26.0(CM)	C= .30315E+00	(G/CM**3)	Z= 28.0(CM)	C= .20022E+00	(G/CM**3)
Z= 30.0(CM)	C= .12111E+00	(G/CM**3)	Z= 32.0(CM)	C= .66779E-01	(G/CM**3)	Z= 34.0(CM)	C= .33441E-01	(G/CM**3)
Z= 36.0(CM)	C= .15165E-01	(G/CM**3)	Z= 38.0(CM)	C= .62137E-02	(G/CM**3)	Z= 40.0(CM)	C= .22960E-02	(G/CM**3)
Z= 42.0(CM)	C= .76400E-03	(G/CM**3)	Z= 44.0(CM)	C= .22866E-03	(G/CM**3)	Z= 46.0(CM)	C= .61496E-04	(G/CM**3)
Z= 48.0(CM)	C= .14850E-04	(G/CM**3)	Z= 50.0(CM)	C= .32175E-05	(G/CM**3)	Z= 52.0(CM)	C= .62517E-06	(G/CM**3)
Z= 54.0(CM)	C= .10889E-06	(G/CM**3)	Z= 56.0(CM)	C= .16993E-07	(G/CM**3)	Z= 58.0(CM)	C= .23754E-08	(G/CM**3)
Z= 60.0(CM)	C= .29733E-09	(G/CM**3)	Z= 62.0(CM)	C= .33318E-10	(G/CM**3)	Z= 64.0(CM)	C= .33416E-11	(G/CM**3)
Z= 66.0(CM)	C= .29991E-12	(G/CM**3)	Z= 68.0(CM)	C= .24083E-13	(G/CM**3)	Z= 70.0(CM)	C= .17300E-14	(G/CM**3)
Z= 72.0(CM)	C= .11116E-15	(G/CM**3)	Z= 74.0(CM)	C= .63875E-17	(G/CM**3)	Z= 76.0(CM)	C= .32824E-18	(G/CM**3)
Z= 78.0(CM)	C= .15082E-19	(G/CM**3)	Z= 80.0(CM)	C= .61959E-21	(G/CM**3)	Z= 82.0(CM)	C= .22756E-22	(G/CM**3)
Z= 84.0(CM)	C= .74714E-24	(G/CM**3)	Z= 86.0(CM)	C= .21927E-25	(G/CM**3)	Z= 88.0(CM)	C= .57521E-27	(G/CM**3)
Z= 90.0(CM)	C= .13487E-28	(G/CM**3)	Z= 92.0(CM)	C= .28261E-30	(G/CM**3)	Z= 94.0(CM)	C= .52925E-32	(G/CM**3)
Z= 96.0(CM)	C= .88573E-34	(G/CM**3)	Z= 98.0(CM)	C= .13246E-35	(G/CM**3)	Z=100.0(CM)	C= .17702E-37	(G/CM**3)
Z=102.0(CM)	C= .21138E-39	(G/CM**3)	Z=104.0(CM)	C= .22554E-41	(G/CM**3)	Z=106.0(CM)	C= .21502E-43	(G/CM**3)
Z=108.0(CM)	C= .18315E-45	(G/CM**3)	Z=110.0(CM)	C=0.	(G/CM**3)	Z=112.0(CM)	C=0.	(G/CM**3)
Z=114.0(CM)	C=0.	(G/CM**3)	Z=116.0(CM)	C=0.	(G/CM**3)			

TOTAL LIQUID PHASE SOLUTE MASS IN SOIL COLUMN	INTEGRAL(THETA*C*DZ) =	9.21420	(G/CM**2)	(PIECE)	(1)
TOTAL LIQUID PHASE SOLUTE MASS IN SOIL COLUMN	INTEGRAL(THETA*C*DZ) =	8.87934	(G/CM**2)	(TAYLOR)	(2)
TOTAL SOLID PHASE SOLUTE MASS IN SOIL COLUMN	INTEGRAL(S*DZ) =	6.44994	(G/CM**2)	(PIECE)	(3)
TOTAL SOLID PHASE SOLUTE MASS IN SOIL COLUMN	INTEGRAL(S*DZ) =	6.21554	(G/CM**2)	(TAYLOR)	(4)
TOTAL SOLUTE MASS THAT HAS ENTERED THE SOIL COLUMN	INTEGRAL(FLUX*DT) =	16.00000	(G/CM**2)		(5)

Figure 6.6 Computer output for the analytical solution at t = 1 day.

```
ANALYTIC SOLUTION TO THE UNSTEADY-STATE,ONE-DIMENSIONAL CONVECTIVE-DISPERSIVE MASS TRANSPORT EQUATION
                         WITH LINEAR EQUILIBRIUM ADSORPTION AND DECAY

                                         THE PARAMETERS ARE

D(DISPERSION)= 30.00(CM**2/DAY)   K(DECAY)=.200(PER DAY)   K(SORPTION)=.200(CM**3/G SOIL)   Q(DARCIAN FLUX)= 16.00(CM/DAY)
RHO(BULK DENSITY)= 1.400(G SOIL/CM**3)   TO(PULSE DURATION)= 1.250(DAYS)   THETA(WATER CONTENT)=.4000(CM**3/CM**3)


                                         AT TIME =  2.00 (DAYS)

Z=   0.0(CM)  C= .67594E-04 (G/CM**3)   Z=   0.5(CM)  C= .12460E-03 (G/CM**3)   Z=   1.0(CM)  C= .21226E-03 (G/CM**3)
Z=   1.5(CM)  C= .34328E-03 (G/CM**3)   Z=   2.0(CM)  C= .53449E-03 (G/CM**3)   Z=   2.5(CM)  C= .80784E-03 (G/CM**3)
Z=   3.0(CM)  C= .11915E-02 (G/CM**3)   Z=   3.5(CM)  C= .17209E-02 (G/CM**3)   Z=   4.0(CM)  C= .24404E-02 (G/CM**3)
Z=   4.5(CM)  C= .34039E-02 (G/CM**3)   Z=   5.0(CM)  C= .46764E-02 (G/CM**3)   Z=   6.0(CM)  C= .84704E-02 (G/CM**3)
Z=   7.0(CM)  C= .14596E-01 (G/CM**3)   Z=   8.0(CM)  C= .24032E-01 (G/CM**3)   Z=   9.0(CM)  C= .37930E-01 (G/CM**3)
Z=  10.0(CM)  C= .57527E-01 (G/CM**3)   Z=  11.0(CM)  C= .84012E-01 (G/CM**3)   Z=  12.0(CM)  C= .11835E+00 (G/CM**3)
Z=  13.0(CM)  C= .16106E+00 (G/CM**3)   Z=  14.0(CM)  C= .21210E+00 (G/CM**3)   Z=  15.0(CM)  C= .27067E+00 (G/CM**3)
Z=  16.0(CM)  C= .33525E+00 (G/CM**3)   Z=  17.0(CM)  C= .40365E+00 (G/CM**3)   Z=  18.0(CM)  C= .47328E+00 (G/CM**3)
Z=  19.0(CM)  C= .54135E+00 (G/CM**3)   Z=  20.0(CM)  C= .60525E+00 (G/CM**3)   Z=  22.0(CM)  C= .71243E+00 (G/CM**3)
Z=  24.0(CM)  C= .78538E+00 (G/CM**3)   Z=  26.0(CM)  C= .82514E+00 (G/CM**3)   Z=  28.0(CM)  C= .83908E+00 (G/CM**3)
Z=  30.0(CM)  C= .83549E+00 (G/CM**3)   Z=  32.0(CM)  C= .82008E+00 (G/CM**3)   Z=  34.0(CM)  C= .79532E+00 (G/CM**3)
Z=  36.0(CM)  C= .76141E+00 (G/CM**3)   Z=  38.0(CM)  C= .71762E+00 (G/CM**3)   Z=  40.0(CM)  C= .66340E+00 (G/CM**3)
Z=  42.0(CM)  C= .59918E+00 (G/CM**3)   Z=  44.0(CM)  C= .52665E+00 (G/CM**3)   Z=  46.0(CM)  C= .44882E+00 (G/CM**3)
Z=  48.0(CM)  C= .36958E+00 (G/CM**3)   Z=  50.0(CM)  C= .29319E+00 (G/CM**3)   Z=  52.0(CM)  C= .22347E+00 (G/CM**3)
Z=  54.0(CM)  C= .16328E+00 (G/CM**3)   Z=  56.0(CM)  C= .11414E+00 (G/CM**3)   Z=  58.0(CM)  C= .76207E-01 (G/CM**3)
Z=  60.0(CM)  C= .48531E-01 (G/CM**3)   Z=  62.0(CM)  C= .29443E-01 (G/CM**3)   Z=  64.0(CM)  C= .16999E-01 (G/CM**3)
Z=  66.0(CM)  C= .93327E-02 (G/CM**3)   Z=  68.0(CM)  C= .48684E-02 (G/CM**3)   Z=  70.0(CM)  C= .24116E-02 (G/CM**3)
Z=  72.0(CM)  C= .11338E-02 (G/CM**3)   Z=  74.0(CM)  C= .50564E-03 (G/CM**3)   Z=  76.0(CM)  C= .21384E-03 (G/CM**3)
Z=  78.0(CM)  C= .85730E-04 (G/CM**3)   Z=  80.0(CM)  C= .32570E-04 (G/CM**3)   Z=  82.0(CM)  C= .11723E-04 (G/CM**3)
Z=  84.0(CM)  C= .39969E-05 (G/CM**3)   Z=  86.0(CM)  C= .12905E-05 (G/CM**3)   Z=  88.0(CM)  C= .39450E-06 (G/CM**3)
Z=  90.0(CM)  C= .11417E-06 (G/CM**3)   Z=  92.0(CM)  C= .31276E-07 (G/CM**3)   Z=  94.0(CM)  C= .81088E-08 (G/CM**3)
Z=  96.0(CM)  C= .19895E-08 (G/CM**3)   Z=  98.0(CM)  C= .46190E-09 (G/CM**3)   Z= 100.0(CM)  C= .10146E-09 (G/CM**3)
Z= 102.0(CM)  C= .21085E-10 (G/CM**3)   Z= 104.0(CM)  C= .41452E-11 (G/CM**3)   Z= 106.0(CM)  C= .77086E-12 (G/CM**3)
Z= 108.0(CM)  C= .13560E-12 (G/CM**3)   Z= 110.0(CM)  C= .22559E-13 (G/CM**3)   Z= 112.0(CM)  C= .35497E-14 (G/CM**3)
Z= 114.0(CM)  C= .52823E-15 (G/CM**3)   Z= 116.0(CM)  C= .74337E-16 (G/CM**3)

     TOTAL LIQUID PHASE SOLUTE MASS IN SOIL COLUMN   INTEGRAL(THETA*C*DZ) = 10.13190  (G/CM**2)  (PIECE ) (1)
     TOTAL LIQUID PHASE SOLUTE MASS IN SOIL COLUMN   INTEGRAL(THETA*C*DZ) = 10.01635  (G/CM**2)  (TAYLOR) (2)

      TOTAL SOLID PHASE SOLUTE MASS IN SOIL COLUMN   INTEGRAL(   S*DZ   ) =  7.09233  (G/CM**2)  (PIECE ) (3)
      TOTAL SOLID PHASE SOLUTE MASS IN SOIL COLUMN   INTEGRAL(   S*DZ   ) =  7.01145  (G/CM**2)  (TAYLOR) (4)

 TOTAL SOLUTE MASS THAT HAS ENTERED THE SOIL COLUMN  INTEGRAL( FLUX*DT  ) = 20.00000  (G/CM**2)           (5)
```

Figure 6.7 Computer output for the analytical solution at t = 2 days.

ANALYTIC SOLUTION TO THE UNSTEADY-STATE,ONE-DIMENSIONAL CONVECTIVE-DISPERSIVE MASS TRANSPORT EQUATION
WITH LINEAR EQUILIBRIUM ADSORPTION AND DECAY

THE PARAMETERS ARE

D(DISPERSION)= 50.00(CM**2/DAY) K(DECAY)=.200(PER DAY) K(SORPTION)=.200(CM**3/G SOIL) Q(DARCIAN FLUX)= 16.00(CM/DAY)
RHO(BULK DENSITY)= 1.400(G SOIL/CM**3) TO(PULSE DURATION)= 1.250(DAYS) THETA(WATER CONTENT)=.4000(CM**3/CM**3)

AT TIME = 3.00 (DAYS)

Z= 0.0(CM)	C= .80563E-08 (G/CM**3)	Z= 0.5(CM)	C= .14923E-07 (G/CM**3)	Z= 1.0(CM)	C= .25742E-07 (G/CM**3)
Z= 1.5(CM)	C= .42399E-07 (G/CM**3)	Z= 2.0(CM)	C= .67585E-07 (G/CM**3)	Z= 2.5(CM)	C= .10508E-06 (G/CM**3)
Z= 3.0(CM)	C= .16017E-06 (G/CM**3)	Z= 3.5(CM)	C= .24018E-06 (G/CM**3)	Z= 4.0(CM)	C= .35517E-06 (G/CM**3)
Z= 4.5(CM)	C= .51885E-06 (G/CM**3)	Z= 5.0(CM)	C= .74984E-06 (G/CM**3)	Z= 6.0(CM)	C= .15223E-05 (G/CM**3)
Z= 7.0(CM)	C= .29902E-05 (G/CM**3)	Z= 8.0(CM)	C= .57059E-05 (G/CM**3)	Z= 9.0(CM)	C= .10608E-04 (G/CM**3)
Z= 10.0(CM)	C= .19253E-04 (G/CM**3)	Z= 11.0(CM)	C= .34166E-04 (G/CM**3)	Z= 12.0(CM)	C= .59357E-04 (G/CM**3)
Z= 13.0(CM)	C= .10105E-03 (G/CM**3)	Z= 14.0(CM)	C= .16869E-03 (G/CM**3)	Z= 15.0(CM)	C= .27634E-03 (G/CM**3)
Z= 16.0(CM)	C= .44443E-03 (G/CM**3)	Z= 17.0(CM)	C= .70208E-03 (G/CM**3)	Z= 18.0(CM)	C= .10898E-02 (G/CM**3)
Z= 19.0(CM)	C= .16628E-02 (G/CM**3)	Z= 20.0(CM)	C= .24946E-02 (G/CM**3)	Z= 22.0(CM)	C= .53430E-02 (G/CM**3)
Z= 24.0(CM)	C= .10729E-01 (G/CM**3)	Z= 26.0(CM)	C= .20236E-01 (G/CM**3)	Z= 28.0(CM)	C= .35911E-01 (G/CM**3)
Z= 30.0(CM)	C= .60059E-01 (G/CM**3)	Z= 32.0(CM)	C= .94837E-01 (G/CM**3)	Z= 34.0(CM)	C= .14166E+00 (G/CM**3)
Z= 36.0(CM)	C= .20061E+00 (G/CM**3)	Z= 38.0(CM)	C= .26997E+00 (G/CM**3)	Z= 40.0(CM)	C= .34618E+00 (G/CM**3)
Z= 42.0(CM)	C= .42429E+00 (G/CM**3)	Z= 44.0(CM)	C= .49874E+00 (G/CM**3)	Z= 46.0(CM)	C= .56436E+00 (G/CM**3)
Z= 48.0(CM)	C= .61720E+00 (G/CM**3)	Z= 50.0(CM)	C= .65497E+00 (G/CM**3)	Z= 52.0(CM)	C= .67704E+00 (G/CM**3)
Z= 54.0(CM)	C= .68407E+00 (G/CM**3)	Z= 56.0(CM)	C= .67744E+00 (G/CM**3)	Z= 58.0(CM)	C= .65880E+00 (G/CM**3)
Z= 60.0(CM)	C= .62976E+00 (G/CM**3)	Z= 62.0(CM)	C= .59178E+00 (G/CM**3)	Z= 64.0(CM)	C= .54634E+00 (G/CM**3)
Z= 66.0(CM)	C= .49496E+00 (G/CM**3)	Z= 68.0(CM)	C= .43939E+00 (G/CM**3)	Z= 70.0(CM)	C= .38162E+00 (G/CM**3)
Z= 72.0(CM)	C= .32374E+00 (G/CM**3)	Z= 74.0(CM)	C= .26786E+00 (G/CM**3)	Z= 76.0(CM)	C= .21585E+00 (G/CM**3)
Z= 78.0(CM)	C= .16918E+00 (G/CM**3)	Z= 80.0(CM)	C= .12884E+00 (G/CM**3)	Z= 82.0(CM)	C= .95233E-01 (G/CM**3)
Z= 84.0(CM)	C= .68265E-01 (G/CM**3)	Z= 86.0(CM)	C= .47419E-01 (G/CM**3)	Z= 88.0(CM)	C= .31897E-01 (G/CM**3)
Z= 90.0(CM)	C= .20765E-01 (G/CM**3)	Z= 92.0(CM)	C= .13077E-01 (G/CM**3)	Z= 94.0(CM)	C= .79623E-02 (G/CM**3)
Z= 96.0(CM)	C= .46858E-02 (G/CM**3)	Z= 98.0(CM)	C= .26643E-02 (G/CM**3)	Z=100.0(CM)	C= .14632E-02 (G/CM**3)
Z=102.0(CM)	C= .77596E-03 (G/CM**3)	Z=104.0(CM)	C= .39726E-03 (G/CM**3)	Z=106.0(CM)	C= .19630E-03 (G/CM**3)
Z=108.0(CM)	C= .93605E-04 (G/CM**3)	Z=110.0(CM)	C= .43066E-04 (G/CM**3)	Z=112.0(CM)	C= .19115E-04 (G/CM**3)
Z=114.0(CM)	C= .81833E-05 (G/CM**3)	Z=116.0(CM)	C= .33789E-05 (G/CM**3)		

TOTAL LIQUID PHASE SOLUTE MASS IN SOIL COLUMN INTEGRAL(THETA*C*DZ) = 8.90522 (G/CM**2) (PIECE) (1)
TOTAL LIQUID PHASE SOLUTE MASS IN SOIL COLUMN INTEGRAL(THETA*C*DZ) = 8.90483 (G/CM**2) (TAYLOR) (2)

TOTAL SOLID PHASE SOLUTE MASS IN SOIL COLUMN INTEGRAL(S*DZ) = 6.23365 (G/CM**2) (PIECE) (3)
TOTAL SOLID PHASE SOLUTE MASS IN SOIL COLUMN INTEGRAL(S*DZ) = 6.23338 (G/CM**2) (TAYLOR) (4)

TOTAL SOLUTE MASS THAT HAS ENTERED THE SOIL COLUMN INTEGRAL(FLUX*DT) = 20.00000 (G/CM**2) (5)

Figure 6.8 Computer output for the analytical solution at t = 3 days.

Table 6.9. Comparing the peak concentrations computed by the analytical solution and OR-NATURE

	Peak concentration (g/cm^3)		
	t = 1 day	t = 2 days	t = 3 days
Analytic	.9963	.8391	.6841
OR-NATURE†			
Δt = .01 days‡	.9975	.8389	.6837
Δt = .02 days	.9975	.8357	.6808
Δt = .05 days	.9973	.8268§	.6678
Δt = .10 days	1.024	.8248§	.6679
Location of peak (cm)	z = 0	z = 28.	z = 54.

† Space increment Δz = .5cm for $0 \leq z \leq 5$, z=1.cm for $5 < z \leq 20$, and Δz = 2.cm for $20 < z \leq 116$cm.

‡ Example problem given in Section 7.5, Vol. IV.

§ Peak concentration occurred at z = 30cm.

A comparison of the computational effort between the analytical solution and OR-NATURE is given in Table 6.10 for the simulation time of three days. The computational effort is defined as the total time to execute the program divided by the number of space nodes and then dividing this by the number of time increments. The analytical solution is approximately 10 times less efficient than program OR-NATURE. This is because of the costly execution of the complementary error functions in Eq. 6.12. In addition, using large time steps is less efficient than using small time steps in program OR-NATURE. This is caused in part by the proportion of time that the program spends printing out information. In constructing Table 6.10, all the OR-NATURE programs were programmed to print out the same number of pages. For a time increment of .01 days, a printout occurred every twentieth time step, but for a time increment of .05 days, a printout occurred every fourth time step.

Table 6.1Ø A comparison of the computational effort between the analytical solution and program OR-NATURE at time = 3 days

	Number of time steps	CPU seconds	Computational effort†
Analytic	3	8.1	.Ø3649
OR-NATURE			
Δt = .Ø1 days‡	3ØØ	81.6	.ØØ3676
Δt = .Ø2 days	15Ø	48.5	.ØØ4369
Δt = .Ø5 days	6Ø	26.2	.ØØ59Ø1
Δt = .1Ø days	3Ø	18.8	.ØØ8468

† Computational effort = (CPU time)/(no. of space nodes•no. of time steps) where CPU is the Central Processing Units or the total execution time of the program in seconds in a Burroughs B69ØØ computer. There were a total of 74 space nodes.

‡ Taken from Example 7.5 of Vol. IV, Section 7.5.

The apparent error in the mass balance for program OR-NATURE is shown in Figure 6.9. The apparent error is defined as the computed sum of solute mass in the soil column (Eq. 6.16) minus the theoretical sum of solute mass in the soil column (Eq. 6.17). This is then all divided by the theoretical sum of solute mass and then multiplied by 1ØØ to convert it to a percent basis.

$$\text{Apparent Error} = (\text{computed} - \text{theoretical}) \cdot 1\text{ØØ}/\text{theoretical}. \quad [\%] \qquad (6.15)$$

A positive error (+) indicates a numerical excess of mass and a negative error (-) indicates a numerical loss of mass.

The computed sum of solute mass is mathematically defined as

$$\text{Computed Sum } (g/cm^2) = \text{CCMPTE} + \text{SCMPTE} + |\text{CTRSUM}|, \qquad (6.16)$$

where variable CCMPTE equals the total solute mass in the liquid phase at time (k+1), SCMPTE is the total solute mass in the solid phase at time (k+1), and CTRSUM equals the liquid phase solute mass that has been removed by first-order decay at time (k+1/2). The total solute mass that should theoretically be in the system is computed by the formula

$$\text{Theoretical Sum } (g/cm^2) = \begin{cases} f(t)\cdot t & t \leq 1.25 \text{ days} \\ 2\emptyset & t > 1.25 \text{ days} \end{cases} \qquad (6.17)$$

where t is time and f(t) is the solute flux boundary condition, f(t) = 16 g/cm^2/day.

The apparent error of program OR-NATURE is plotted in Figure 6.9 using two different methods to calculate the "Computed Sum" of solute mass in Eq. 6.16. The upper curve, labeled "Piecewise-constant Integration" in Figure 6.9, starts with a +2.25% error and then drops to a low of +.ØØ2% on the third day. The lower curve, labeled "Taylor Series Integration" in Figure 6.9, starts with a -2.56% error, changes sign after 1.25 days, goes to a +.Ø57%, and then drops to a low of +.ØØØ15 on the third day. The piecewise-constant integration curve represents the method currently utilized by program OR-NATURE. Whenever a flux type boundary condition is specified at the upper soil surface, the program computes all mass balances of Eq. 6.16 with the piecewise-constant integration routine (Vol. I, Section 3.7.1). The piecewise-constant integration routines are called on lines 897 and 9Ø2 of OR-NATURE. These are given as

```
IF(BCTOPC.EQ.3) CALL INTGRT(1,N,Z,DUMMYB,CCMPTE,3HCZ2)             897
                                                                 (6.18)
IF(JSORB.GE.1.AND.BCTOPC.EQ.3) CALL INTGRT(1,N,Z,S,SCMPTE,3HSZ2) 9Ø2
```

To make a comparison, program OR-NATURE was modified so that the second-order accurate Taylor series integration scheme (Vol. I, Section 3.7.2) was used to integrate all of the mass balances of Eq. 6.16. The modification is very simple: change the one to a two on lines 897 and 9Ø2.

```
IF(BCTOPC.EQ.3) CALL INTGRT(2,N,Z,DUMMYB,CCMPTE,3HCZ2)             897
                                                                 (6.19)
IF(JSORB.GE.1.AND.BCTOPC.EQ.3) CALL INTGRT(2,N,Z,S,SCMPTE,3HSZ2) 9Ø2
```

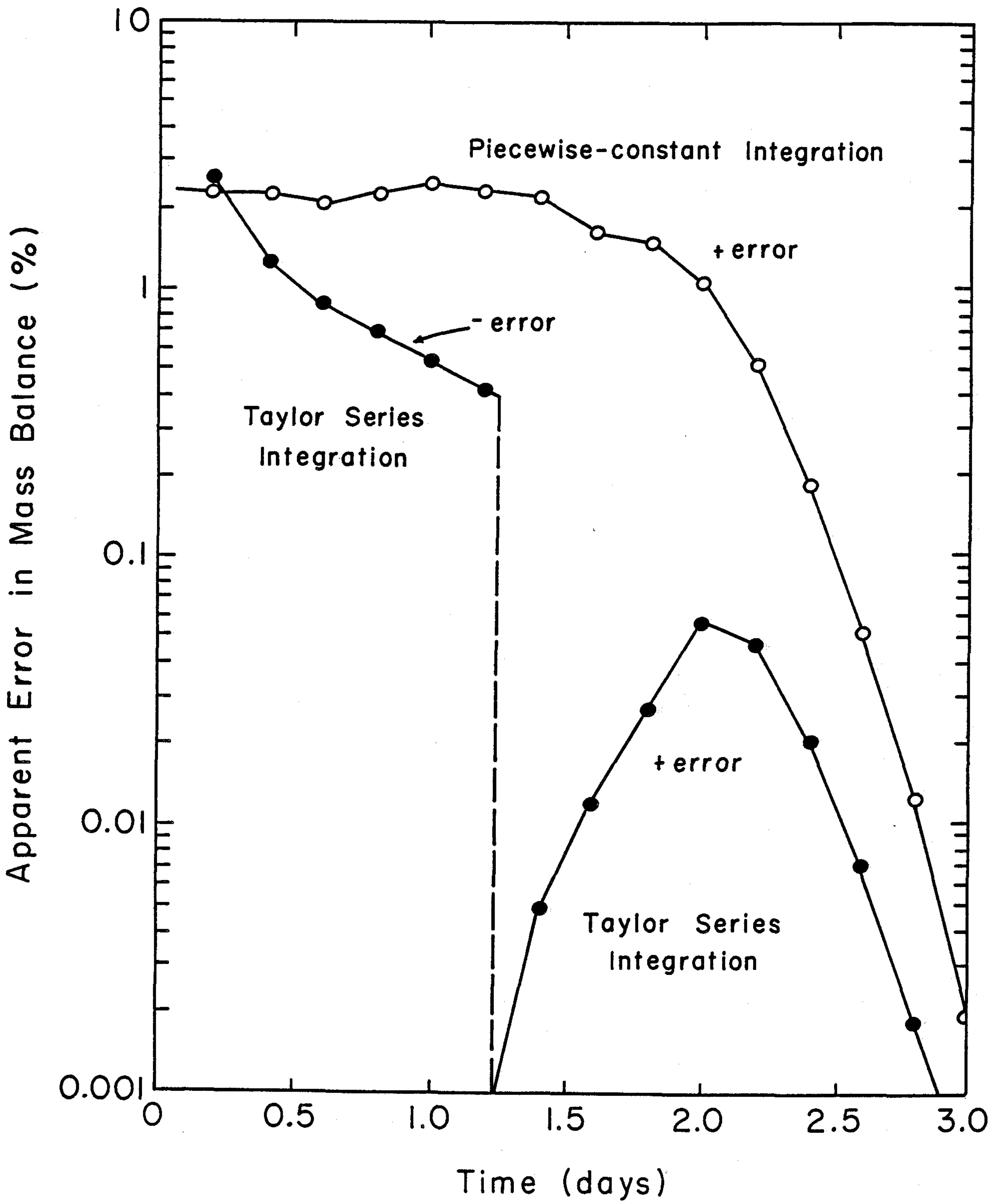

Figure 6.9 Apparent error in the solute mass balance of program OR-NATURE in Example 7.5. Two methods of computing the mass balance are shown.

It should be noted that the apparent error of Figure 6.9 depends on several factors. The mass balance terms CCMPTE and SCMPTE are sensitive to the method of integrating the solute concentrations c and S versus soil depth. The mass balance term CTRSUM is evaluated at time level (k+1/2), whereas, CCMPTE and SCMPTE are evaluated at time level (k+1). These factors are independent of the finite-difference solution technique used by program OR-NATURE to compute the concentrations c and S. Finally, the apparent error depends on the time and space increments used in the finite-difference solution of the mass transport equation for variables c and S.

Neither the piecewise-constant nor the Taylor series mass balance integration routines are perfect. Unfortunately, there is no way to specify an integration routine that will be optimal for all applications. For example, in Example 7.1 of Vol. IV, Section 7.1, the initial solute concentration is a "step input". Only the piecewise-constant integration routine will correctly compute the mass balance of the solute concentration profiles. Most "high-order" integration routines will "smear" a steep front or other discontinuities. However, in no case do the piecewise-constant or the Taylor series integration routines ever affect the accuracy of the finite-difference solution of the solute concentrations or water contents. Only the mass balance terms, CCMPTE, SCMPTE, and WCMPTE, are affected.

If the user wishes to modify or to use a different mass balance integration routine, the following program lines should be checked in program OR-NATURE.

For the liquid phase solute mass balance variable SCMPTE, check program lines 59Ø, 9Ø2, and 9Ø3.

For the water content mass balance variable WCMPTE, check program lines 582, 941, and 942.

In addition, modify line 3498 in subroutine INTGRT if an additional mass balance integration routine is added to subroutine INTGRT, i.e., TYPE=3.

In conclusion, the solute routine of program OR-NATURE was compared against an analytical solution for a problem with dispersion, convection, first-order decay, and a linear equilibrium sorption term. The concentration profiles of program OR-NATURE were found to be indistinguishable from those of the analytical solution when plotted, and the maximum or peak concentrations were identical to at least two place accuracy. A solute mass balance of program OR-NATURE was found to have an apparent excess of 2.3% mass, but this exponentially went to zero with time. The apparent excess mass is probably caused by the method of computing the mass balance than the finite-difference routine used to compute the solute concentrations.