



ORNL/GWPO-015

**OAK RIDGE  
NATIONAL  
LABORATORY**

**MARTIN MARIETTA**

**MURT User's Guide: A Hybrid  
Lagrangian-Eulerian Finite Element  
Model of Multiple-Pore-Region  
Solute Transport Through  
Subsurface Media**

J. P. Gwo  
P. M. Jardine  
G. T. Yeh  
G. V. Wilson

Environmental Sciences Division  
Publication No. 4375



MANAGED BY  
MARTIN MARIETTA ENERGY SYSTEMS, INC.  
FOR THE UNITED STATES  
DEPARTMENT OF ENERGY

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

5

This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from the Office of Scientific and Technical Information, P.O. Box 62, Oak Ridge, TN 37831; prices available from (615) 576-8401, FTS 626-8401.

Available to the public from the National Technical Information Service, U.S. Department of Commerce, 5285 Port Royal Rd., Springfield, VA 22161.

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

## **DISCLAIMER**

**Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.**

Environmental Sciences Division

**MURT USER'S GUIDE: A HYBRID LAGRANGIAN-EULERIAN FINITE ELEMENT  
MODEL OF MULTIPLE-PORE-REGION SOLUTE TRANSPORT THROUGH  
SUBSURFACE MEDIA**

J. P. Gwo, P. M. Jardine, G. T. Yeh<sup>1</sup>, and, G. V. Wilson<sup>2</sup>

---

<sup>1</sup>Department of Civil and Environmental Engineering, Pennsylvania State University

<sup>2</sup>Department of Plant and Soil Science, The University of Tennessee, Knoxville

Environmental Sciences Division

Publication No. 4375

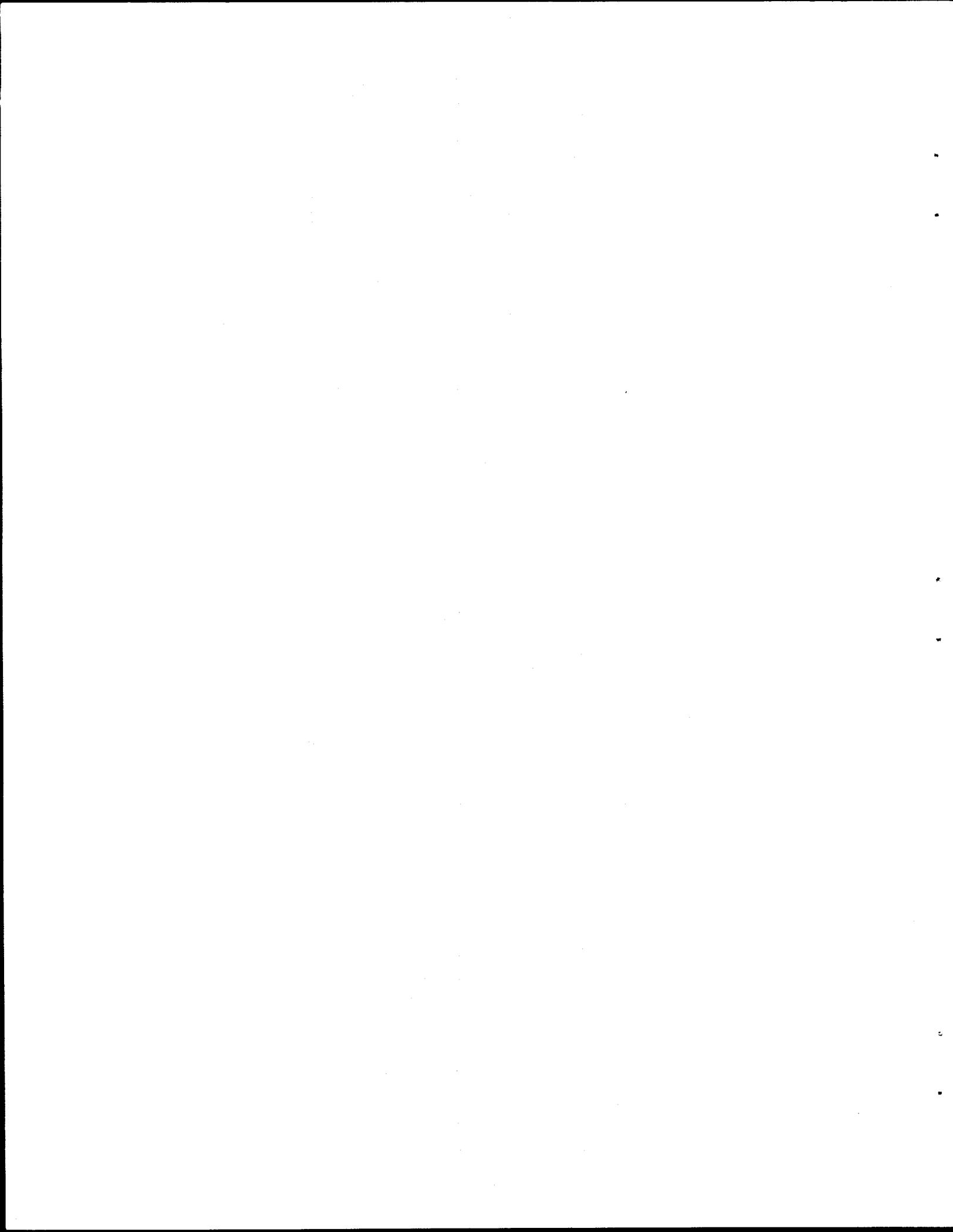
Date Published: April 1995

Prepared for the  
U.S. Department of Energy  
Office of Environmental Restoration and Waste Management  
EX 20 10 30 1

Prepared by  
OAK RIDGE NATIONAL LABORATORY  
Oak Ridge, Tennessee 37831-6285  
managed by  
LOCKHEED MARTIN ENERGY SYSTEMS, INC.  
for the  
U.S. DEPARTMENT OF ENERGY  
under contract DE-AC05-84OR21400

**MASTER**

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED  
59



# CONTENTS

	<u>Page</u>
FIGURES . . . . .	vii
TABLES . . . . .	ix
ACKNOWLEDGMENTS . . . . .	xi
EXECUTIVE SUMMARY . . . . .	xiii
1. INTRODUCTION . . . . .	1
1.1 REVIEW OF MULTIREGION SOLUTE TRANSPORT CONCEPTS . . . . .	1
1.2 MULTIREGION MASS TRANSFER MECHANISM . . . . .	2
2. MULTIREGION SOLUTE TRANSPORT AND INTERREGION MASS TRANSFER EQUATIONS . . . . .	5
2.1 MULTIREGION SOLUTE TRANSPORT EQUATIONS . . . . .	5
2.2 DIFFUSIVE MASS TRANSFER RELATIONS . . . . .	9
2.3 ADVECTIVE MASS TRANSFER RELATIONS . . . . .	11
2.4 WORKING GOVERNING EQUATIONS FOR MURT . . . . .	12
3. PROGRAM STRUCTURE OF MURT . . . . .	17
3.1 FUNCTIONS OF SUBROUTINES . . . . .	17
3.1.1 Subroutine <i>advbc</i> . . . . .	17
3.1.2 Subroutine <i>advtrn</i> . . . . .	21
3.1.3 Subroutine <i>afabta</i> . . . . .	21
3.1.4 Subroutine <i>asmlab</i> . . . . .	21
3.1.5 Subroutine <i>asmblc</i> . . . . .	21
3.1.6 Subroutine <i>bc</i> . . . . .	21
3.1.7 Subroutines <i>cbvfct, dbvfct, essfct, nbvfct, vbvfct, wssfct</i> . . . . .	22
3.1.8 Subroutine <i>datain</i> . . . . .	22
3.1.9 Subroutine <i>dgelg</i> . . . . .	22
3.1.10 Subroutine <i>dimes</i> . . . . .	23
3.1.11 Subroutine <i>dispc</i> . . . . .	23
3.1.12 Real Function <i>fcos</i> . . . . .	23
3.1.13 Subroutine <i>flux</i> . . . . .	23
3.1.14 Subroutine <i>gicrn</i> . . . . .	23
3.1.15 Subroutine <i>glre</i> . . . . .	23
3.1.16 Subroutine <i>glrn</i> . . . . .	24
3.1.17 Subroutine <i>gm2dxz</i> . . . . .	24
3.1.18 Subroutine <i>gmdata</i> . . . . .	25

## CONTENTS (Continued)

	<u>Page</u>
3.1.19 Integer Function <i>ialloc</i> . . . . .	25
3.1.20 Subroutine <i>iinit</i> . . . . .	25
3.1.21 Subroutine <i>intrbnd</i> . . . . .	25
3.1.22 Subroutine <i>ioctrl</i> . . . . .	25
3.1.23 Matrix Solver MA28 . . . . .	26
3.1.24 Program <i>main</i> . . . . .	26
3.1.25 Subroutine <i>mploc</i> . . . . .	26
3.1.26 Subroutine <i>ndtau</i> . . . . .	26
3.1.27 Subroutine <i>piss</i> . . . . .	27
3.1.28 Subroutine <i>ppcg</i> . . . . .	27
3.1.29 Subroutine <i>precond</i> . . . . .	27
3.1.30 Subroutine <i>printt</i> . . . . .	27
3.1.31 Subroutines <i>q2cb</i> and <i>q2vb</i> . . . . .	27
3.1.32 Subroutines <i>q3ab</i> and <i>q4ab</i> . . . . .	28
3.1.33 Subroutines <i>q3c</i> and <i>q4c</i> . . . . .	29
3.1.34 Subroutines <i>q3d</i> and <i>q4d</i> . . . . .	29
3.1.35 Subroutines <i>q3r</i> and <i>q4r</i> . . . . .	29
3.1.36 Subroutine <i>readn</i> . . . . .	30
3.1.37 Subroutine <i>readr</i> . . . . .	30
3.1.38 Subroutine <i>sflow</i> . . . . .	30
3.1.39 Subroutine <i>shape</i> . . . . .	32
3.1.40 Subroutine <i>store</i> . . . . .	32
3.1.41 Subroutine <i>surf</i> . . . . .	32
3.1.42 Subroutine <i>thnode</i> . . . . .	32
3.1.43 Subroutine <i>xsi2d</i> . . . . .	32
3.2 I/O UNITS . . . . .	33
3.2.1 Geometry in Unformatted I/O Unit 10 . . . . .	33
3.2.2 Boundary Arrays in Unformatted I/O Unit 11 . . . . .	34
3.2.3 Pointer Arrays in Unformatted I/O Unit 12 . . . . .	35
3.2.4 Restart Data in Unformatted I/O Unit 13 . . . . .	35
3.2.5 Simulation Results in Unformatted I/O Unit 14 . . . . .	35
3.2.6 Hydraulic Parameters Generated By Multiregion Flow Code MURF in Unformatted I/O Unit 16 . . . . .	36
4. SITE SPECIFIC APPLICATIONS OF MURT . . . . .	39
4.1 UNDISTURBED SOIL COLUMN TRACER INJECTIONS . . . . .	39
4.1.1 Problem Titles and Basic Control Parameters . . . . .	41
4.1.2 Material Properties . . . . .	42
4.1.3 Problem Domain, Initial Condition, and Boundary Conditions . . . . .	43
4.1.4 Hydrological Input Variables . . . . .	45

## CONTENTS (Continued)

	<u>Page</u>
4.1.5 Predicting BTC for Br Injection On Soil Column 2 . . . . .	45
4.2 SENSITIVITY ANALYSIS OF ADVECTIVE AND DIFFUSIVE MASS TRANSFER . . . . .	45
4.2.1 Numerical Model Configurations . . . . .	46
4.2.2 Effects of Advective Mass Transfer on Fluid Flow . . . . .	49
4.2.3 Effects of Advective-Diffusive Mass Transfer on Solute Transport . . . . .	51
4.3 HILLSLOPE PONDING INFILTRATION AND SECONDARY SOURCE . . . . .	55
4.3.1 Numerical Model Configurations . . . . .	55
4.3.2 Results and Discussion . . . . .	55
5. REFERENCES . . . . .	61
APPENDIX A. DERIVATION OF MULTIREGION SUBSURFACE SOLUTE TRANSPORT EQUATIONS . . . . .	65
A.1 THEORETICAL BACKGROUND . . . . .	65
A.2 DERIVING MULTIREGION SOLUTE TRANSPORT EQUATIONS BASED ON VOLUME-AVERAGE OVER HOMOGENIZED MEDIA . . . . .	65
A.3. REFERENCES . . . . .	71
APPENDIX B. NUMERICAL IMPLEMENTATION OF MULTIREGION SOLUTE TRANSPORT EQUATIONS . . . . .	73
B.1 OVERVIEW. . . . .	73
B.2 NUMERICAL APPROXIMATION OF MULTIREGION SOLUTE TRANSPORT EQUATIONS . . . . .	73
B.2.1 Finite Element Approximation of Solute Transport Equations . . . . .	73
B.2.2 Numerical Implementation of Mass Flux Equation . . . . .	81
B.2.3 Finite Difference Approximation of Time Derivative Term . . . . .	81
B.2.4 Numerical Implementation of Boundary Conditions . . . . .	82
B.2.5 Mass Lumping . . . . .	85
B.2.6 Matrix Solvers for Linearized System . . . . .	86
B.3 REFERENCES . . . . .	87
APPENDIX C. DATA INPUT GUIDE FOR MURT . . . . .	89

## CONTENTS (Continued)

	<u>Page</u>
C.1 DATA INPUT GUIDE FOR FILE 3: GEOMETRY OF PROBLEM DOMAIN AND MATERIAL TYPE CORRECTIONS . . . . .	89
C.2 DATA INPUT GUIDE FOR FILE 4: CONTROL AND OPTION PARAMETERS AND MATERIAL PROPERTIES . . . . .	92
C.3 DATA INPUT GUIDE FOR FILE 5: INITIAL CONDITION, EXTERNAL SOURCES/SINKS, AND BOUNDARY CONDITIONS . . . . .	95
C.4 DATA INPUT GUIDE FOR FILE 6: HYDRAULIC PARAMETERS	102
C.5 REFERENCES . . . . .	104

## FIGURES

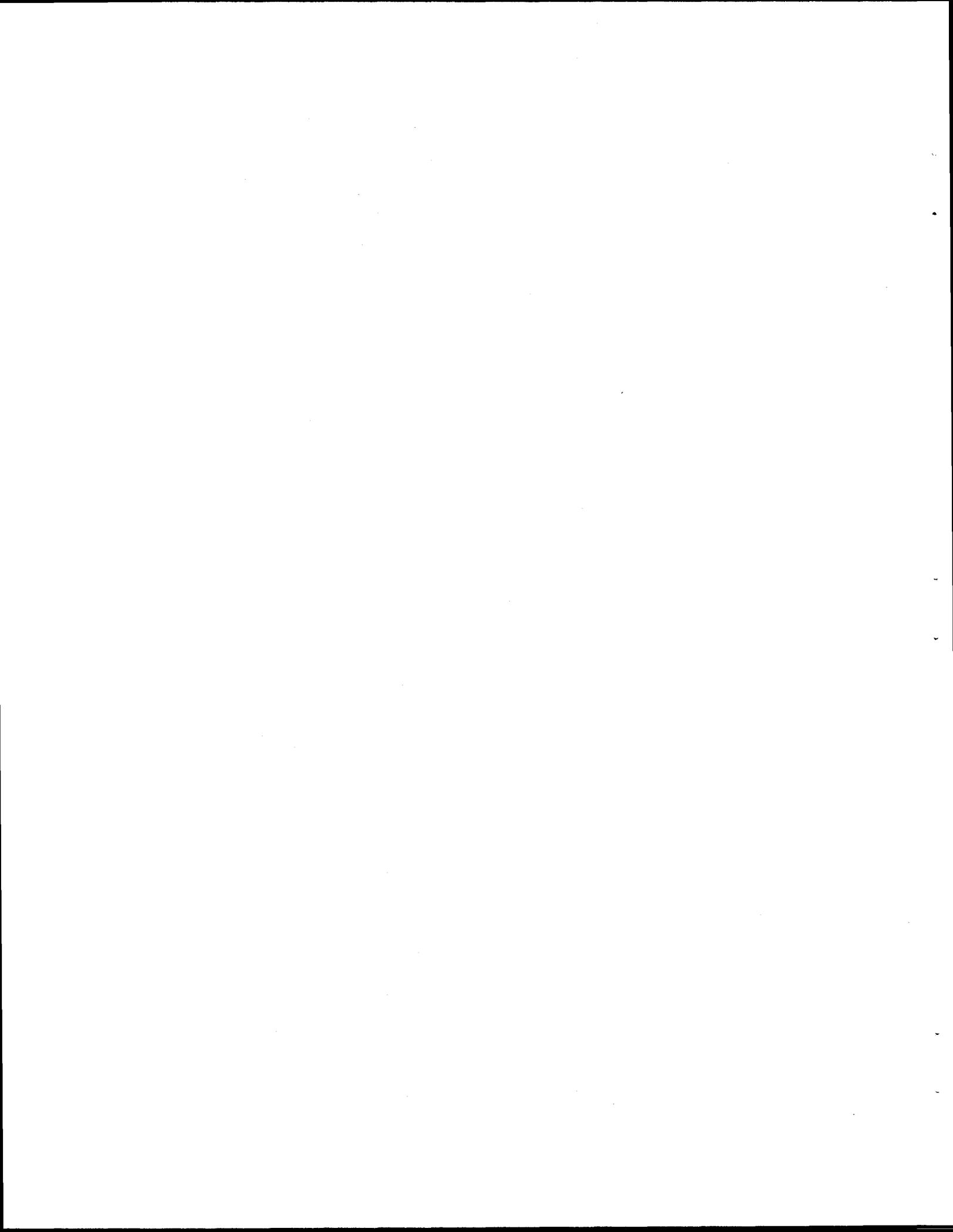
<u>Figure</u>	<u>Page</u>
1.1 A three-pore-region multiregion mass transfer mechanism . . . . .	3
2.1 A schematic representation of homogenization and interregion mass transfer . . . . .	7
3.1 Program structure of MURT . . . . .	18
4.1 Template specifying array dimensions for the undisturbed soil column application . . . . .	40
4.2 Template specifying I/O control parameters for the undisturbed soil column application . . . . .	41
4.3 I/O files containing material properties and modeling control and option parameters for the undisturbed soil column application . . . . .	42
4.4 Fitted BTC and experimental data for the Br <sup>-</sup> injection on soil column 1 . . . . .	43
4.5 I/O file defining the problem domain and element connectivity for the undisturbed soil column application . . . . .	44
4.6 I/O file defining initial and boundary conditions for the undisturbed soil column application . . . . .	44
4.7 I/O file specifying hydraulic parameter for the undisturbed soil column application . . . . .	45
4.8 Predicted and measured BTC of Br <sup>-</sup> injection on soil column 2 . . . . .	46
4.9 Velocity variations at $z = 95$ cm . . . . .	50
4.10 Effect of advective mass transfer on pressure head difference between pore regions at time = 5.23 h and $z = 95$ cm . . . . .	50
4.11 Times needed for pressure head difference between pore regions to become less than 1 cm, 0.5% of the length of the soil column, at $z = 95$ cm as function of advective mass transfer coefficients . . . . .	51
4.12 Solute concentration variations in the pore regions at $z = 95$ cm . . . . .	52
4.13 Effect of diffusive mass transfer on solute concentrations at $z = 95$ cm and time = 1.97 h . . . . .	52

## FIGURES (Continued)

<u>Figure</u>	<u>Page</u>
4.14 Times needed for pore-region concentration to become less than 1% of the source concentration at $z = 95$ cm as related to (a) diffusive mass transfer coefficients at $\epsilon_{vj}^f = 0.01 \text{ cm}^{-1} \text{ h}^{-1}$ and (b) advective mass transfer coefficients at $\epsilon_{vj}^t = 0.1 \text{ h}^{-1}$ . . . . .	53
4.15 Effect of advective mass transfer on pressure heads and solute concentrations at $z = 95$ cm and time = 1.97 h . . . . .	54
4.16 Problem domain of the hillslope ponding infiltration and secondary source problem . . . . .	56
4.17 Finite element discretization of the problem domain of the hillslope ponding infiltration and secondary source . . . . .	57
4.18 Pore region solute concentrations along the direction of flow . . . . .	58
4.19 Solute mass in the three pore regions at 61.6 h: (a) macropores, (b) mesopores, and (c) micropores . . . . .	59
4.20 Solute concentration in the three pore regions at 61.6 h: (a) macropores, (b) mesopores, and (c) micropores . . . . .	60
A. 1 A hypothetical three-region system with each region occupying a portion of the voids of the porous medium . . . . .	66
B.1. Local coordinate systems used for quadrilateral and triangular elements . . . . .	75
B.2 Weighting factor along a line element. . . . .	77
B.3 Upstream weighting factors along the sides of a quadrilateral element . . . . .	78

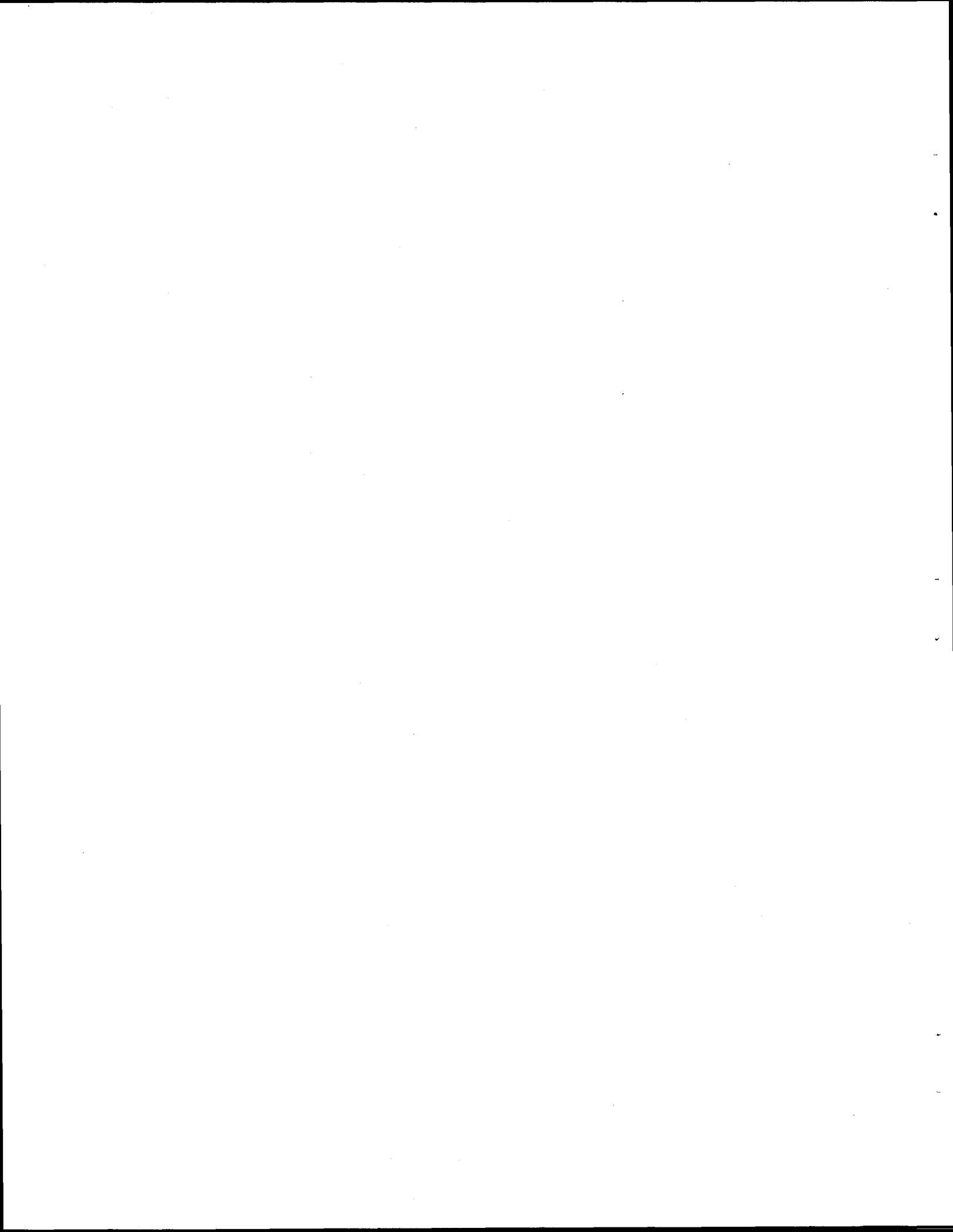
## TABLES

<u>Table</u>	<u>Page</u>
3.1 I/O units used by MURT . . . . .	33
4.1 Dimensions, material properties, and hydrological parameters of the soil columns . . . . .	40
4.2 Fitted model parameters for the soil column 1 . . . . .	43
4.3 Pore-region hydraulic and solute transport parameters for sensitivity analyses	47
4.4 Minimum and maximum mass transfer coefficients between pore regions . .	49
4.5 Advective and diffusive mass transfer coefficients for the hillslope ponding and secondary source problem . . . . .	58



## ACKNOWLEDGMENTS

The research was supported by the Subsurface Science Program, Office of Health and Environmental Research, U.S. Department of Energy. This work was also supported in part by an appointment to the Oak Ridge National Laboratory Postdoctoral Research Associates Program, which is administered jointly by the Oak Ridge National Laboratory and the Oak Ridge Institute for Science and Education.



## EXECUTIVE SUMMARY

Matrix diffusion, a diffusive mass transfer process, in the structured soils and geological units at the Oak Ridge National Laboratory (ORNL) is believed to be a very important subsurface mass transfer mechanism (Solomon et al. 1992). It may affect the off-site movement of radioactive wastes and the remediation of waste disposal sites by locally exchanging wastes between soil/rock matrix and macropores/fractures. A second local-scale mass transfer mechanism, advective mass transfer, that also contributes to the movement of wastes is largely unattended by researchers. Similar to matrix diffusion, advective mass transfer can be theoretically derived and similarly characterized. This report presents the first documented two-dimensional multiregion solute transport code (MURT) that incorporates not only diffusive but also advective mass transfer and can be applied to heterogeneous porous media under transient flow conditions.

Advective mass transfer arises from the movement of fluids between pore domains and is active when subsurface flow systems are under transient conditions that generally prevail in the field. These conditions are of special concern at waste disposal sites like those at ORNL, where wastes are buried in the stormflow and vadose zones that often come in contact with transient subsurface stormflow. As fluids are moved between pore domains, solutes contained in the fluids are transferred in a process mechanistically similar to advection in terms of classical convection-dispersion concepts. This process is often neglected in available two-pore-domain or dual-porosity models, which mostly deal with steady-state flow conditions (Skopp et al. 1981; Parker and van Genuchten 1984).

Soil horizons in the stormflow and vadose zones are largely composed of heterogeneous porous media that are rich in microscopic features such as soil matrix and fractures. The movement of water and wastes through these media is difficult and expensive to characterize and the relevant mass transfer processes are poorly understood. Advective and diffusive mass transfer may be characterized separately in the laboratory; however, the complex effect resulting from both processes simultaneously is very difficult to quantify experimentally. Computer models therefore become an alternative tool to study the coupled effect of the mass transfer processes.

In this report, theoretical background is reviewed and the derivation of multiregion solute transport equations is presented. Similar to MURF (Gwo et al. 1994), a multiregion subsurface flow code, multiple pore domains as suggested by previous investigators (e.g., Wilson and Luxmoore 1988) can be implemented in MURT. The transient or steady-state flow fields of the pore domains can be either calculated by MURF or provided by modelers. We briefly discuss the mass transfer process through a three-pore-region multiregion solute transport mechanism. Mass transfer equations that describe mass flux across pore region interfaces are also presented, and the parameters that are needed to calculate mass transfer coefficients are detailed. Three applications of MURT -- a laboratory tracer injection problem, a sensitivity analysis on advective and diffusive mass transfer, and a hillslope ponding infiltration and secondary source problem -- were simulated and the results are discussed. We also discuss the program structure of MURT

and functions of MURT subroutines so that users can adapt the code to their specific applications. Finally, the guides for input data preparation are provided in the appendices for users' reference.

## 1. INTRODUCTION

Matrix diffusion, a microscopic nonequilibrium process that moves wastes among pore domains, is believed to be an important mechanism that controls movement of wastes in the subsurface media at ORNL (Solomon et al. 1992). Despite its importance, the effect of matrix diffusion on waste movement at the field-scale, which could be either positive or negative to waste site remediation, is poorly understood. Wastes could be retarded by slow diffusion into soil matrix during injection and could be withdrawn, again slowly, back into macropores and fractures during cleanup operations such as pump-and-treat. Slow removal of wastes resulting from slow matrix diffusion could induce high remediation costs. In addition to matrix diffusion, advective mass transfer driven by inter-pore-domain, or interregion, velocity fields may also contribute to movement of wastes into and out of soil matrix. This latter mass transfer mechanism is even less understood than matrix diffusion and was not incorporated into subsurface mass transport processes until recently (Gwo 1992; Gwo et al. 1994a; Gwo et al. 1995).

This report presents a multiple-pore-region, or multiregion, model that considers both advective and diffusive mass transfer processes, simultaneously and collectively, in subsurface solute transport systems. Both large-scale heterogeneities, which cause preferential flow, and local-scale heterogeneities, which result in advective mass transfer and matrix diffusion, can be implemented in the model. This report, similar to its companion report on a multiregion subsurface flow model that calculates pore-region velocity fields needed for multiregion solute transport modeling (Gwo et al. 1994b), represents one of our efforts in studying mass transport in highly fractured heterogeneous media. The conceptualization of multiregion mass transport processes owes tremendously to recent knowledge gained in the laboratory and field (Seyfried and Rao 1987; Jardine et al. 1988; Wilson et al. 1992; Wilson and Luxmoore 1988; Jardine et al. 1990; Vepraskas et al. 1991; Jardine et al. 1993). The mechanistical approach adopted in these reports is based largely on mechanisms revealed during experiments conducted on undisturbed soil columns and field sites with macroporous and fractured media (Jardine et al. 1993; Wilson et al. 1993).

### 1.1 REVIEW OF MULTIREGION SOLUTE TRANSPORT CONCEPTS

The effect of matrix diffusion on solute transport has been extensively studied since the 1960s (e.g., Maloszewski and Zuber 1990; Parker and van Genuchten 1984; Grisak and Pickens 1980; van Genuchten and Wierenga 1976; Coats and Smith 1964). Differential capacitance models (Coats and Smith 1964) and mobile-immobile models (van Genuchten and Wierenga 1976) were applied to displacement experiments in rock cores and soil columns (Coats and Smith 1964; Parker and van Genuchten 1984). Concepts employed by previous investigators (e.g., Grisak and Pickens 1980; Tang et al. 1981) to solve solute transport problems involving a single fracture closely resembled the dead-end pore and mobile-immobile models. Diffusion of solutes into rock matrix was either explicitly accounted for by considering fracture and rock matrix as two porous media

(Grisak and Pickens 1980) or by applying Fick's first law at the fracture-matrix interface (Tang et al. 1981). The former approach regarded the matrix domain as a pore region with a velocity field, while the latter treated the rock matrix as an immobile pore region with stagnant pore water. The latter approach was also parallel to the treatment of solute transport through soils with large cylindrical macropores by van Genuchten et al. (1984), which illustrated the resemblance between mass transport in fractured rocks and that in macroporous soils. The concept taken by the former approach also found its counterpart in the soil science literature (Skopp et al. 1981; Skopp and Gardner 1992) where micropores are treated as a mobile pore domain.

It was not until the 1980s that the significance of matrix diffusion in large-scale problems was emphasized by researchers (e.g., Solomon et al. 1992; Kimura and Munakata 1992; Jarvis et al. 1991a; Jarvis et al. 1991b; Jardine et al. 1990; and Bibby 1981). A dual-porosity mobile-immobile model was applied to the movement of saline water in a chalk aquifer in England (Bibby 1981). Jardine et al. (1990) clearly demonstrated the movement of solutes along pore domains and the effect of natural rainfall on the remobilization of solutes diffused into matrix pores. These observations in fact suggested not only that matrix diffusion was a process dynamically regulating storage and release of solutes but also that two flow paths in macroporous soils may be utilized by solute particles. Jarvis et al. (1991a) used a two-pore-region model to describe water and solute transport in a macroporous soil. Matrix diffusion was incorporated into a channel flow model by Kimura and Munakata (1992) to study tracer movement in a crystalline rock in Sweden. A study by Solomon et al. (1992) suggested that mass transfer between large and small pores may substantially reduce contaminant migration rates relative to water velocities in large pores.

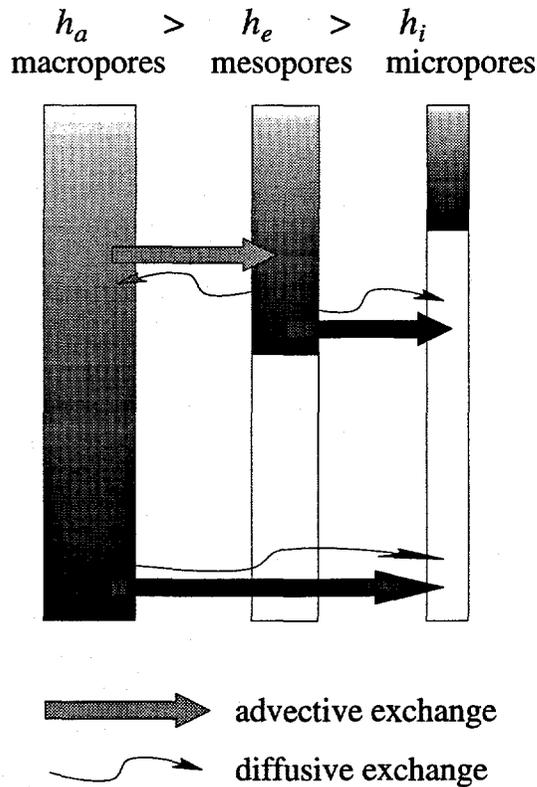
Because of the complex physicochemical processes in subsurface media, questions remain about whether it is sufficient to represent these processes conceptually by two-pore domains. Review of chemical transport through soils by Jury and Fluhler (1992) suggested that two flow paths, or pore domains, might not be sufficient to explain the movement of solutes in highly heterogeneous soils. Three-pore-domain solute transport models that accounted for multiple nonequilibrium processes were proposed by previous investigators (Brusseau et al. 1989). Field studies on a macroporous soil by Wilson and Luxmoore (1988) suggested that three porosities were required to describe the dynamics of water movement in the soil. Three-pore-region hydraulic conductivity and water retention curves were later obtained for the soil (Wilson et al. 1992). These soil properties were used by Gwo (1992) and Gwo et al. (1994a) in a multiregion model of solute transport on a forested watershed at ORNL.

## **1.2 MULTIREGION MASS TRANSFER MECHANISM**

Generally, interregion mass transfer is composed of two components, namely, advective and diffusive mass transfer. Advective mass transfer, similar to advection in porous medium or one-region models, results from interregion velocity fields induced by

interregion pressure gradient. In contrast, diffusive mass transfer is driven exclusively by interregion concentration gradients and is widely known as matrix diffusion. As will become obvious later, advective mass transfer may reduce the interregion concentration gradients and counteract diffusive mass transfer.

To illustrate the mechanism of multiregion mass transfer, let us look at an example that consists of three pore regions, namely, macropores, mesopores, and micropores (Fig. 1.1). The flow fields in the pore regions are represented by three interconnected flow paths with discernible fluid velocities. A pulse of solution is injected and distributions of solutes in the three flow fields result. Higher hydraulic conductivity results in faster pressure head build-up in larger pores, i.e.,  $h_a > h_e > h_i$ , where  $h_a$ ,  $h_e$ , and  $h_i$  are pressure heads in macropores, mesopores, and micropores, respectively. Advective mass transfer,



**Fig. 1.1. A three-pore-region multiregion mass transfer mechanism.** The interregion mass flux is composed of advective and diffusive mass transfer.

driven by interregion pressure gradient, then moves fluid and solutes together as solution parcels from large pores to small pores, without being dispersed along the interregion mass transfer paths. Diffusive mass transfer, however, moves only solutes between pore regions according to interregion concentration gradients. Therefore, the direction of solute movement has a lot to do with the point or region of interest. At upstream areas,

diffusive mass transfer moves solutes from small pores to large pores, while at downstream areas it moves solutes in the opposite directions. The directions of advective and diffusive solute movements may or may not be the same (Fig. 1.1).

Movement of solute by advective mass transfer could be either from high to low concentration areas or vice versa. When advective mass transfer moves solute from high to low concentrations areas, solute concentration in the target pore region increases and therefore reduces the interregion concentration gradient. Diffusive mass transfer now has a lower interregion concentration gradient and hence a slower mass transfer rate. When advective mass transfer moves solute from low to high concentration areas, the solution of the target pore region is diluted by the incoming low-concentration solution. The interregion concentration gradient is reduced and so is the interregion mass flux contributed by diffusive mass transfer. Therefore, advective mass transfer may counteract diffusive mass transfer. However, studies by Duguid and Lee (1977), Gwo (1992), and Bai et al. (1993) suggested that the period of transient interactive flow between pore domains are relatively small, and it has been shown that the dominant interregion mechanism is diffusive mass transfer (Gwo et al. 1994a).

In this report we presents a two-dimensional multiregion solute transport computer code, MURT (multiregion solute transport), which is applicable to variably saturated multiple-porosity-multiple-permeability media. Decay, degradation, and sorption of solutes are allowed to be independently associated with the pore regions. Models developed according to the multiregion concept are therefore multiple-domain and multiple-processes. Three site-specific examples are presented along with the description of the computer code and the user's input guide. Mass balance equations of a general multiregion solute transport model are presented in Sect. 2, and a detailed derivation of the equations can be found in Appendix A.

## 2. MULTIREGION SOLUTE TRANSPORT AND INTERREGION MASS TRANSFER EQUATIONS

The multiregion mass transfer mechanism depicted in Sect. 1.2 can be translated into mass balance equations either robustly by applying classic thermodynamic theories to macroporous media (Gwo 1992) or intuitively by starting from the one-region porous-medium solute transport or convection-dispersion equation. The latter approach is used in this section, and the former is presented in Appendix A. For both approaches, one needs to determine the mass transfer equations at pore-region interfaces, which are also discussed in this section.

### 2.1 MULTIREGION SOLUTE TRANSPORT EQUATIONS

The well-known classic convection-dispersion equation consists of two basic solute transport terms. The advection term accounts for movement of solute molecules in the velocity of fluid molecules as though they were moving in free-flowing surface water. The dispersion term accounts for solute movement along fluid pathways that diverge and converge when the fluid is moving along a pore space of various sizes. Molecular diffusion, velocity variation due to pore geometry, and meandering of flowline (tortuosity) are collectively accounted for by a dispersion coefficient. In addition to these mass transfer forces, physicochemical reactions are accounted for by chemical decay and degradation and by adsorption of solutes onto the solid phase. Including external sources/sinks to the system, one arrives at the following one-region porous-medium solute transport equations, in aqueous and solid phases, respectively:

$$\rho\theta \frac{\partial c}{\partial t} + \rho\theta v \cdot \nabla c = \nabla \cdot (\rho\theta D \cdot \nabla c) - \rho\theta(\lambda + K_w)c + \theta\rho qc_a^* - c \left( \frac{\partial \rho\theta}{\partial t} + \nabla \rho\theta v \right) \quad (2.1)$$

and

$$\rho_b \frac{\partial s}{\partial t} = -\rho_b s(\lambda + K_s), \quad (2.2)$$

where

- $\rho$  = aqueous phase density (ML<sup>-3</sup>),
- $\theta$  = water content of the porous medium,
- $c$  = concentration of the aqueous phase (ML<sup>-3</sup>),
- $t$  = time (T),
- $v$  = aqueous or pore water velocity (LT<sup>-1</sup>),
- $D$  = dispersion coefficient tensor (L<sup>2</sup>T<sup>-1</sup>),
- $\lambda$  = radioactive decay constant (T<sup>-1</sup>),
- $K_w$  = first-order degradation constant of the aqueous phase (T<sup>-1</sup>),
- $q$  = external source/sink rate (L<sup>3</sup>L<sup>-3</sup>T<sup>-1</sup>),

$c_q^*$  = source/sink concentration (ML<sup>-3</sup>),  
 $\rho_b$  = bulk density of the porous medium (ML<sup>-3</sup>),  
 $s$  = solid phase concentration of the solute (MM<sup>-1</sup>),  
 $K_s$  = degradation rate of the solid phase (T<sup>-1</sup>).

Depending on  $q$ ,  $c_q^*$  could be either the incoming solution concentration, if  $q$  is a source (>0), or the aqueous phase concentration at the location of interest, if  $q$  is a sink (<0). The initial and boundary conditions for the solute transport equations can be formulated as follows:

$$c(x, z, 0) = c_i(x, z, 0) \quad (2.3)$$

and

$$Ac_b + B\theta n \cdot \nu c_b - C\theta n \cdot D\nabla c_b = f_b(x, y, t) \quad (2.4)$$

where  $c_i(x, z, 0)$  is the initial concentration at  $(x, z)$ ;  $c_b$  is the concentration at the boundary;  $\mathbf{n}$  is the normal vector;  $A$ ,  $B$ , and  $C$ , are either 1 or 0 depending on the type of boundary given; and  $f_b(x, y, t)$  is specified mass flux or concentration at the boundary.

For macroporous and fractured subsurface media, a pore domain (e.g., macropores and fractures) can be treated as a porous medium. According to homogenization theory (Arbogast et al. 1989), Eqs. (2.1) through (2.4) can therefore be applied to the pore domain. Because pore domains are independently and individually homogenized, interaction between pore regions must be explicitly specified. This approach is schematically illustrated in Fig. 2.1, where rock matrix and fractures are homogenized into two types of media with interregion mass transfer.

Assuming that mass transfer occurs through the aqueous phase only, we arrive at the following multiregion solute transport equations for a system consisting of  $N$  pore regions:

$$\begin{aligned}
 \rho_\alpha \theta_\alpha \frac{\partial c_\alpha}{\partial t} + \rho_\alpha \theta_\alpha \nu_\alpha \cdot \nabla c_\alpha = \nabla \cdot (\rho_\alpha \theta_\alpha D_\alpha \cdot \nabla c_\alpha) - \rho_\alpha \theta_\alpha (\lambda_\alpha + K_{w\alpha}) c_\alpha \\
 + \theta_\alpha \rho_\alpha q_\alpha c_{q\alpha}^* - c_\alpha \left( \frac{\partial \rho_\alpha \theta_\alpha}{\partial t} + \nabla \cdot \rho_\alpha \theta_\alpha \nu_\alpha \right) + \rho_\alpha \theta_\alpha e_\alpha^t \quad \alpha = 1, 2, \dots, N
 \end{aligned} \quad (2.5)$$

and

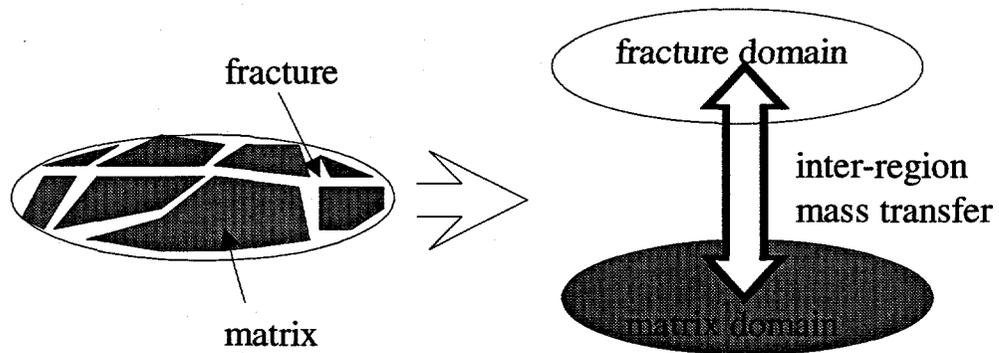
$$\rho_{b\alpha} \frac{\partial s_\alpha}{\partial t} = -\rho_{b\alpha} s_\alpha (\lambda_\alpha + K_{s\alpha}) \quad \alpha = 1, 2, \dots, N, \quad (2.6)$$

where the subscript  $\alpha$  stands for pore region  $\alpha$ , and  $e'_\alpha$  is the net mass transfer rate to pore region  $\alpha$  ( $\text{ML}^{-3}\text{T}^{-1}$ ). The dispersion coefficient tensor can be further represented by dispersivity, velocity, tortuosity, and molecular diffusion coefficient as follows (Nguyen et al. 1982):

$$D_\alpha = a_{T\alpha} |v_\alpha| \delta + (a_{L\alpha} - a_{T\alpha}) \frac{v_\alpha v_\alpha}{|v_\alpha|} + a_{m\alpha} \tau_\alpha \delta, \quad (2.7)$$

where

- $a_{T\alpha}$  = transverse dispersivity (L),
- $a_{L\alpha}$  = longitudinal dispersivity (L),
- $\delta$  = the Kronecker delta tensor,
- $|v_\alpha|$  = the magnitude of velocity  $v_\alpha$  ( $\text{LT}^{-1}$ ),
- $a_{m\alpha}$  = molecular diffusion coefficient ( $\text{L}^2\text{T}^{-1}$ ),
- $\tau$  = tortuosity.



**Fig. 2.1. A schematic representation of homogenization and interregion mass transfer.** Because the pore domains are homogenized independently, interregion interactions must be explicitly specified.

Note that we need to relate Eqs. (2.5) and (2.6) to form a closed mathematical system. In this study, we adopt equilibrium adsorption/desorption relations, including linear, Freundlich, and Langmuir isotherms, such that the sorbed phase concentration can be expressed as analytical functions of aqueous phase concentration as follows:

$$s_\alpha = K_d c_\alpha \quad \text{for linear isotherms,} \quad (2.8a)$$

$$s_\alpha = K c_\alpha^n \quad \text{for Freundlich isotherms,} \quad (2.8b)$$

$$s_{\alpha} = \frac{s_{\alpha,m} K c_{\alpha}}{1 + K c_{\alpha}} \quad \text{for Langmuir isotherms,} \quad (2.8c)$$

where  $K_d$  is the distribution coefficient ( $L^3/M$ ),  $s_{\alpha,m}$  is the maximum concentration allowed on the solid surface ( $M/L^3$ ),  $n$  is the order of Freundlich isotherms, and  $K$  is a functional coefficient for Freundlich and Langmuir isotherms. The initial and boundary conditions of a multiregion system can be formulated similarly as follows:

$$c_{\alpha}(x, z, 0) = c_{i\alpha}(x, z, 0) \quad \alpha = 1, 2, \dots, N \quad (2.9)$$

and

$$A c_{b\alpha} + B \theta_{\alpha} \mathbf{n} \cdot \mathbf{v}_{\alpha} c_{b\alpha} - C \theta_{\alpha} \mathbf{n} \cdot \mathbf{D}_{\alpha} \nabla c_{b\alpha} = f_{b\alpha}(x, y, t) \quad \alpha = 1, 2, \dots, N, \quad (2.10)$$

Four types of boundary conditions are available in the computer code MURT, namely, Dirichlet, variable, Cauchy, and Neumann. The Dirichlet boundary is a boundary with prescribed solute concentrations. Using the notation of Eq. (2.10), we have

$$A = 1 \quad (2.11a)$$

$$B = 0 \quad (2.11b)$$

$$C = 0 \quad (2.11c)$$

$$f_{b\alpha}(x, z, t) = c_{D\alpha}(x, z, t) \quad \alpha = 1, 2, \dots, N, \quad (2.11d)$$

where  $c_{D\alpha}(x, z, t)$  is the prescribed Dirichlet boundary concentration for region  $\alpha$ . Both external and internal Dirichlet concentrations can be specified, for example, at the interface of aquifers and streams for the former and wells and waste trenches for the latter.

At boundaries where the direction of fluid flux varies, either into or out of the area of interest, the boundary condition varies. If the fluid flux is going out, no input to the area occurs and therefore dispersive flux is set to zero. On the other hand, if fluid flux is coming in, the boundary condition needs to be adjusted to accommodate incoming solute flux. For pore region  $\alpha$ , these boundary conditions can be implemented as follows (Yeh et al. 1994):

(a) For the run-in boundary, when  $\mathbf{n} \cdot \mathbf{v}_{\alpha} \leq 0$ , using the notation of Eq. (2.10),

$$A = 0 \quad (2.12a)$$

$$B = 1 \quad (2.12b)$$

$$C = 1 \quad (2.12c)$$

$$f_{b\alpha}(x, z, t) = \mathbf{n} \cdot \mathbf{v}_{\alpha} c_{\alpha}(x, z, t) \quad (2.12d)$$

(b) For the seep-out boundary, when  $\mathbf{n} \cdot \mathbf{v}_{\alpha} > 0$ , using the notation of Eq. (2.10),

$$A = 0 \quad (2.13a)$$

$$B = 0 \quad (2.13b)$$

$$C = 1 \quad (2.13c)$$

$$f_{b\alpha}(x, z, t) = 0 \quad (2.13d)$$

where  $c_v(x, z, t)$  is the solute concentration in the incoming water at variable boundaries. At boundaries where advection also contributes to solute flux and fluid is generally flowing into the area of interest, for example, artificial recharge results from surface ponding water, a Cauchy-type boundary condition can be used. Using the notation of Eq. (2.10), the boundary condition can be described as

$$A = 0 \quad (2.14a)$$

$$B = 1 \quad (2.14b)$$

$$C = 1 \quad (2.14c)$$

$$f_{b\alpha}(x, z, t) = q_{c\alpha}(x, z, t), \quad (2.14d)$$

where  $q_{c\alpha}(x, z, t)$  is the Cauchy boundary flux for pore region  $\alpha$ . Similarly, at boundaries where advection does not contribute to boundary flux and the solute flux is outward in general, a Neumann-type boundary can be prescribed. Using the notation of Eq. (2.10), the boundary condition can be described as

$$A = 0 \quad (2.15a)$$

$$B = 0 \quad (2.15b)$$

$$C = 1 \quad (2.15c)$$

$$f_{b\alpha}(x, z, t) = q_{n\alpha}(x, z, t), \quad (2.15d)$$

where  $q_{n\alpha}(x, z, t)$  is the Neumann boundary flux for pore region  $\alpha$ .

Before numerically implementing Eqs. (2.5) through (2.10), one needs to determine the appropriate mass transfer relation for the problem. A steady-state, a transient, and a quasi-transient diffusive mass transfer relations are reviewed in the following section.

## 2.2 DIFFUSIVE MASS TRANSFER RELATIONS

Diffusion of solute into and out of spherical aggregates (e.g., Rao et al. 1980) and non-spherical aggregates (e.g., Rao et al. 1982; van Genuchten and Dalton 1986) has been studied to account for interregion mass transfer. These studies suggested that diffusive mass transfer coefficients can be expressed explicitly by pore-region geometry and fluid properties (van Genuchten 1985). By comparing a spherical aggregate model with a mobile-immobile model, one can obtain (Rao et al. 1980)

$$q_{fa} = \eta(c_a - c_f), \quad (2.16a)$$

$$\eta = \frac{D_e \theta_a \phi \sigma_1}{a^2 \sigma_2}, \quad (2.16b)$$

where  $q_{fa}$  is the solute flux across the aggregate surface;  $\eta$  is the diffusive mass transfer coefficient;  $c_a$  and  $c_f$  are the concentrations in the aggregate and macropores, respectively;  $D_e$  is the effective molecular diffusion coefficient in the porous sphere;  $\theta_a$  is the water content of the sphere;  $a$  is the sphere radius,  $\phi = \theta_f / (\theta_f + \theta_a)$ , where  $\theta_f$  is the water content of the macropores, and  $\sigma_1$  and  $\sigma_2$  are functions of time, the sphere radius, the ratio of water contents in the macropores to that in the aggregates, and the effective molecular diffusion coefficient in the aggregates (Rao et al. 1980). For nonspherical aggregates, interregion mass transfer can be obtained accordingly. Similar approaches using shape factors were devised (e.g., van Genuchten 1985) to account for the different aggregate shapes and sizes in soils.

In a study of fresh water exchanging saline water within dispersed clay minerals, Gvirtzman et al. (1988) suggested that the initial mass transfer coefficient  $\eta$  in Eq. (2.16a) may decay exponentially after a characteristic time was reached:

$$\eta(x, t) = \eta_0 e^{-\beta(t-x/V_m)} \quad \text{if } t - x/V_m \geq 0, \quad (2.17a)$$

$$\eta(x, t) = \eta_0 \quad \text{if } t - x/V_m < 0, \quad (2.17b)$$

where  $\eta_0$  is the initial mass transfer rate,  $t$  is time,  $x$  is depth,  $\beta$  is a decay constant, and  $V_m$  is the pore water velocity in the mobile region. The time-dependency of diffusive mass transfer has been accounted for by the concentration difference term in Eq. (2.16a). Equation (2.17), in fact, suggests that Eq. (2.16a), a first-order equation, may not be sufficient to describe diffusive mass transfer processes in the field. Second order equations have also been obtained for mobile-immobile models (Dykhuizen 1990):

$$q_{fm} = \frac{\theta_m D_m}{l^2} \left( \frac{\pi}{2} - 1 \right) \frac{c_f^2}{c_m} \quad \text{if } c_m < c_f \left( 1 - \frac{\pi}{2} \right), \quad (2.18a)$$

$$q_{fm} = \frac{\theta_m D_m \pi^2}{2l^2} (c_f - c_m) \quad \text{if } c_m > c_f \left( 1 - \frac{\pi}{2} \right), \quad (2.18b)$$

where  $\theta_m$  is the water content of the immobile region,  $l$  is the half-dimension of the immobile region,  $D_m$  is the molecular diffusion coefficient in the immobile region, and  $c_f$  and  $c_m$  are the solute concentrations in the mobile and immobile regions, respectively.

A generic first-order finite-volume diffusive mass transfer relation is numerically implemented in MURT (see Appendix B) as follows:

$$q_{\alpha}^{et} = - \sum_{\substack{j=1 \\ j \neq \alpha}}^N \varepsilon_{\alpha j}^t (c_{\alpha} - c_j) \quad (2.19)$$

where  $q_{\alpha}^{et}$  is the net diffusive flux to pore region  $\alpha$  ( $ML^{-3}T^{-1}$ ) and  $\varepsilon_{\alpha j}^t$  is the diffusive mass transfer coefficient between pore regions  $\alpha$  and  $j$  ( $T^{-1}$ ). Equation (2.19), in general, considers diffusive mass transfer only, where interregion mass transfer is driven by molecular diffusion. Interregion velocity fields (Gwo 1992; Gwo et al. 1994a) also affect interregion mass flux. This mass transfer process is mechanistically similar to advection in one-region solute transport models. In the next section we discuss advective mass transfer relations derived or proposed by previous investigators.

### 2.3 ADVECTIVE MASS TRANSFER RELATIONS

Previous investigations of advective mass transfer include microscopic and macroscopic approaches. Both microscopic and macroscopic approaches employ mass conservation principles across pore-region interfaces. The former approach incorporates microscopic geometry of pore regions in the solution procedure and the latter lumps the microscopic geometry into a mass transfer coefficient (e.g., Duguid and Lee 1977). By assuming that interregion flux along a characteristic length is perpendicular to the fracture wall, Duguid and Lee (1977) obtained the following interregion flow equation:

$$q_{\alpha j}^{ef} = -\varepsilon_{\alpha j}^f \left[ (h_{\alpha} - h_j) + 2 \sum_{n=1}^{\infty} (-1)^n h_{\alpha} e^{-D} - 2 \sum_{n=1}^{\infty} h_j e^{-D} \right], \quad (2.20)$$

where

$q_{\alpha j}^{ef}$  = the fluid flux between pore regions  $\alpha$  and  $j$  ( $T^{-1}$ ),

$\varepsilon_{\alpha j}^f = 4K_{js}n_{\alpha} / \pi cl$  is the steady-state advective mass transfer coefficient between pore regions  $\alpha$  and  $j$  ( $L^{-1}T^{-1}$ ),

$h_{\alpha}$  = pressure head in pore region  $\alpha$  (L),

$h_j$  = pressure head in pore region  $j$  (L),

$K_{js}$  = saturated hydraulic conductivity of pore region  $j$  (L/T),

$n_{\alpha}$  = effective porosity of pore region  $\alpha$ ,

$c$  = half-spacing of parallel plates (L),

$l$  = half-dimension of a primary block (L),

$D = K_{js}n^2\pi^2t / 2l^2\rho_{\alpha}gn_j\beta$ , where  $n_j$  is the effective porosity of pore region  $j$  and  $\beta$  is the compressibility of water.

A variation of Eq. (2.19) was also proposed by Duguid and Abel (1974):

$$q_{\alpha j}^{ef} = -\varepsilon_{\alpha j}^f \left( 1 + \sum_{n=1}^{\infty} e^{-D} \right) (h_{\alpha} - h_j) . \quad (2.21)$$

At large times, the summation terms of Eqs. (2.20) and (2.21) become insignificant and the equations can be reduced to

$$q_{\alpha j}^{ef} = -\varepsilon_{\alpha j}^f (h_{\alpha} - h_j) . \quad (2.22)$$

Results from microscopic approaches have also been compared to those using macroscopic mass transfer equations (Zimmerman et al. 1993). Using an integral method to derive interregion mass flux between fractures and spherical matrix blocks, Zimmerman et al. (1993) arrived at:

$$q_{\alpha j}^{ef} = -\frac{\pi^2 K_{sj}}{2a_j^2} \frac{(h_{\alpha} - h_i)^2 - (h_j - h_i)^2}{h_j - h_i} , \quad (2.23)$$

where  $a_j$  is the radius of a spherical matrix block and  $h_i$  is the initial pressure head in the fracture.

Determining the appropriate advective mass transfer scheme depends on the microscopic geometry of pore regions and the time scales of the application. Studies by Duguid and Lee (1977) on a fractured aquifer, in fact, suggested that the difference in results using transient and steady mass transfer coefficients was negligible. A similar conclusion was reached by Bai et al. (1993), who suggested that, in a dual-porosity or fractured porous medium, the period of transient interregion flow can be relatively short. Therefore, only first-order finite-volume advective mass transfer schemes are implemented in MURT. Given fracture spacing and pore region porosities, one can estimate the advective mass transfer coefficients  $\varepsilon_{\alpha j}^f$  using Eq. (2.22) or (2.23). In the next section, we conclude the discussion of the mathematical derivation of the multiregion solute transport equations by explicitly referencing to first-order finite-volume advective and diffusive mass transfer schemes. We then obtain the working equation for MURT by writing the multiregion equations in Lagrangian-Eulerian form.

## 2.4 WORKING GOVERNING EQUATIONS FOR MURT

To conclude the derivation of the multiregion solute transport equations, one notices that the terms in the last parentheses on the right-hand side of Eq. (2.6) are essentially the multiregion flow continuity equations without the source and mass transfer terms. By similarly applying the homogenization theory to subsurface fluid flow, Gwo et al. (1994b) arrived at the following multiregion flow equations:

$$F_\alpha \frac{\partial h_\alpha}{\partial t} = \nabla \cdot \mathbf{K}_\alpha \nabla (h_\alpha + z) + \theta_\alpha (q_\alpha + e_\alpha^f) \quad \alpha = 1, 2, \dots, N \quad (2.24a)$$

or

$$\frac{\partial \rho_\alpha \theta_\alpha}{\partial t} + \nabla \cdot \rho_\alpha \theta_\alpha \mathbf{v}_\alpha = \rho_\alpha \theta_\alpha (q_\alpha + e_\alpha^f) \quad \alpha = 1, 2, \dots, N, \quad (2.24b)$$

where, for pore region  $\alpha$ ,

$F_\alpha$  = the storage coefficient ( $L^{-1}$ ),

$z$  = the elevation head ( $L$ ),

$\mathbf{K}_\alpha$  = is hydraulic conductivity tensor ( $LT^{-1}$ ),

$e_\alpha^f$  = net fluid transfer rate to pore region  $\alpha$  ( $T^{-1}$ ).

Expanding the first term on the right hand side of Eq. (2.5), neglecting the second-order term  $\nabla \rho_\alpha \theta_\alpha \cdot \mathbf{D}_\alpha \cdot \nabla c_\alpha$ , and substituting Eq. (2.24b) into Eq. (2.5), we have

$$\begin{aligned} \rho_\alpha \theta_\alpha \frac{\partial c_\alpha}{\partial t} + \rho_\alpha \theta_\alpha \mathbf{v}_\alpha \cdot \nabla c_\alpha &= \rho_\alpha \theta_\alpha \nabla \cdot (\mathbf{D}_\alpha \cdot \nabla c_\alpha) - \rho_\alpha \theta_\alpha (\lambda_\alpha + K_{w\alpha}) c_\alpha \\ + \theta_\alpha \rho_\alpha q_\alpha c_{q\alpha}^* - c_\alpha \rho_\alpha \theta_\alpha (q_\alpha + e_\alpha^f) + \rho_\alpha \theta_\alpha e_\alpha^f &\quad \alpha = 1, 2, \dots, N. \end{aligned} \quad (2.25)$$

Combining Eqs. (2.6) and (2.25) and rearranging the resultant equation, we arrive at

$$\begin{aligned} \theta_\alpha \frac{\partial c_\alpha}{\partial t} + \rho_{b\alpha} \frac{\partial s_\alpha}{\partial t} + \theta_\alpha \mathbf{v}_\alpha \cdot \nabla c_\alpha &= \theta_\alpha \nabla \cdot (\mathbf{D}_\alpha \cdot \nabla c_\alpha) - \theta_\alpha (\lambda_\alpha + K_{w\alpha}) c_\alpha \\ - \rho_{b\alpha} s_\alpha (\lambda_\alpha + K_{s\alpha}) + \theta_\alpha q_\alpha c_{q\alpha}^* - c_\alpha \theta_\alpha (q_\alpha + e_\alpha^f) + \theta_\alpha e_\alpha^f &\quad \alpha = 1, 2, \dots, N. \end{aligned} \quad (2.26)$$

The mass transfer term  $\theta_\alpha e_\alpha^f$  is therefore represented by a combination of first-order finite-volume advective and diffusive mass transfer components as follows:

$$\theta_\alpha e_\alpha^f = q_\alpha^{ef} c_\alpha^* + q_\alpha^{et} = - \sum_{\substack{j=1 \\ j \neq \alpha}}^N \epsilon_{\alpha j}^f (h_\alpha - h_j) c_\alpha^* - \sum_{\substack{j=1 \\ j \neq \alpha}}^N \epsilon_{\alpha j}^t (c_\alpha - c_j), \quad (2.27)$$

where  $c_\alpha^*$  is the solute concentration of the upstream pore region, that is,  $c_\alpha^* = c_\alpha$  if  $h_\alpha > h_j$  and  $c_\alpha^* = c_j$  if  $h_\alpha < h_j$ . The advective mass transfer term is repeated here for the reader's reference:

$$\theta_{\alpha} e_{\alpha}^f = - \sum_{\substack{j=1 \\ j \neq \alpha}}^N \varepsilon_{\alpha j}^f (h_{\alpha} - h_j) . \quad (2.28)$$

Substituting Eqs. (2.27) and (2.28) into Eq.(2.26), we have

$$\begin{aligned} \theta_{\alpha} \frac{\partial c_{\alpha}}{\partial t} + \rho_{b\alpha} \frac{\partial s_{\alpha}}{\partial t} + \theta_{\alpha} \mathbf{v}_{\alpha} \cdot \nabla c_{\alpha} &= \theta_{\alpha} \nabla \cdot (\mathbf{D}_{\alpha} \cdot \nabla c_{\alpha}) - \theta_{\alpha} (\lambda_{\alpha} + K_{w\alpha}) c_{\alpha} \\ - \rho_{b\alpha} s_{\alpha} (\lambda_{\alpha} + K_{s\alpha}) + \theta_{\alpha} q_{\alpha} c_{q\alpha}^* - c_{\alpha} \theta_{\alpha} q_{\alpha} - \sum_{\substack{j=1 \\ j \neq \alpha}}^N \varepsilon_{\alpha j}^f (h_{\alpha} - h_j) c_{\alpha}^* \\ - \sum_{\substack{j=1 \\ j \neq \alpha}}^N \varepsilon_{\alpha j}^f (c_{\alpha} - c_j) - \sum_{\substack{j=1 \\ j \neq \alpha}}^N \varepsilon_{\alpha j}^f (h_{\alpha} - h_j) &\quad \alpha = 1, 2, \dots, N . \end{aligned} \quad (2.29)$$

Before the equations can be numerically implemented, we need to linearize the terms with solid phase concentration,  $s_{\alpha}$ , such that

$$s_{\alpha} = s_{\alpha,0} + (c_{\alpha} - c_{\alpha,0}) \frac{ds_{\alpha}}{dc_{\alpha}} , \quad (2.30)$$

where  $c_{\alpha,0}$  and  $s_{\alpha,0}$  are the aqueous and solid phase concentrations at the previous time step or iterate. Substituting Eq. (2.30) into Eq. (2.29), we can rearrange the resultant to the flow working equation, which is in Lagrangian-Eulerian form,

$$\begin{aligned} \left( \theta_{\alpha} + \rho_{b\alpha} \frac{ds_{\alpha}}{dc_{\alpha}} \right) \frac{Dc_{\alpha}}{Dt} &= \theta_{\alpha} \nabla \cdot (\mathbf{D}_{\alpha} \cdot \nabla c_{\alpha}) - \lambda_{\alpha} \left( \theta_{\alpha} + \rho_{b\alpha} \frac{ds_{\alpha}}{dc_{\alpha}} \right) c_{\alpha} \\ - \left( K_{w\alpha} + K_{s\alpha} \rho_{b\alpha} \frac{ds_{\alpha}}{dc_{\alpha}} \right) c_{\alpha} &+ \theta_{\alpha} q_{\alpha} c_{q\alpha}^* - c_{\alpha} \theta_{\alpha} q_{\alpha} - \sum_{\substack{j=1 \\ j \neq \alpha}}^N \varepsilon_{\alpha j}^f (h_{\alpha} - h_j) c_{\alpha}^* \\ - \sum_{\substack{j=1 \\ j \neq \alpha}}^N \varepsilon_{\alpha j}^f (c_{\alpha} - c_j) + \sum_{\substack{j=1 \\ j \neq \alpha}}^N \varepsilon_{\alpha j}^f (h_{\alpha} - h_j) c_{\alpha} &- \rho_{b\alpha} (\lambda_{\alpha} + K_{s\alpha}) \left( s_{\alpha,0} - \frac{ds_{\alpha}}{dc_{\alpha}} c_{\alpha,0} \right) , \\ \alpha = 1, 2, \dots, N , & \end{aligned} \quad (2.31)$$

with the pore water velocity calculated by

$$v_{\alpha}^* = \frac{v_{\alpha}^*}{1 + \frac{\rho_{b\alpha}}{\theta_{\alpha}} \frac{ds_{\alpha}}{dc_{\alpha}}} \quad (2.32)$$

Equations (2.31) and (2.32) are implemented in the numerical code MURT using a hybrid Lagrangian-Eulerian finite element method. A backward particle tracking technique (Baptista 1987) is used to determine the Lagrangian concentration. The program structure and subroutines of MURT are described in the next section. Three applications of MURT are described in Sect. 4. Detailed instructions for input data preparation can be found in Appendix C. With the first application, a step by step guide for preparing the input data sets is also provided.



### 3. PROGRAM STRUCTURE OF MURT

The computer code MURT is a vertical two-dimensional multiregion subsurface solute transport model that accounts for large-scale intraregion advection-dispersion and local-scale interregion advective and diffusive mass transfer. The primary structure of the model, including the program *main* and the driver routine *gm2dxz*, is shown in Fig. 3.1. The computer code consists of one main program, 55 subroutines, 5 user functions, and a sparse matrix solver, MA28 (Duff 1990). Both steady-state and transient-state solute transport simulations can be executed by the code. A hybrid Lagrangian-Eulerian finite element method is used in conjunction with backward particle tracking to reduce numerical dispersion and to increase computational efficiency by allowing large time-stepping. An upstream weighting method is also implemented as a modeling option in MURT. If transient source/sink functions and/or transient boundary conditions are to be represented analytically, users need to supply analytical functions in subroutines *dbvfct*, *vbvfct*, *cbvfct*, *nbvfct*, and *wssfct*. The functions of the main program and the subroutines are described in the following sections.

#### 3.1 FUNCTIONS OF SUBROUTINES

FORTRAN programs, subroutines, and functions related to memory allocation and solute transport simulations are discussed in detail in this section. These code units can be categorized into memory allocation routines (E. F. D'Azevedo, personal communication, 1993) and the core routines of MURT. Memory allocation routines are discussed only when relevant, but all of the core routines of MURT are discussed in detail. Description of the functions of the code units are presented in the following sections in alphabetical order by the name.

##### 3.1.1 Subroutine *advbc*

Subroutine *advbc* is called by *gm2dxz* to implement the Lagrangian boundary conditions. For Dirichlet boundaries, the Lagrangian concentration is already specified. For variable boundaries, if the flow is directed out of the region of interest, the fictitious particle associated with the boundary node must come from the interior nodes. Hence the Lagrangian concentration for the boundary node has already been computed by subroutine *advtrn* and the implementation for this node is bypassed. If the flow is directed into the region of interest, the Lagrangian concentration is calculated according to

$$c_i^* = \frac{\int_{B_v} N_i^e V_v c_{iv} dB}{\int_{B_v} N_i^e V_v dB}, \quad (3.1)$$

where  $c_i^*$  is the Lagrangian concentration at the boundary node  $i$ ,  $V_v$  is the normal velocity to the variable boundary side, and  $c_{iv}$  is the concentration of the incoming fluid through

(a) Program *main*

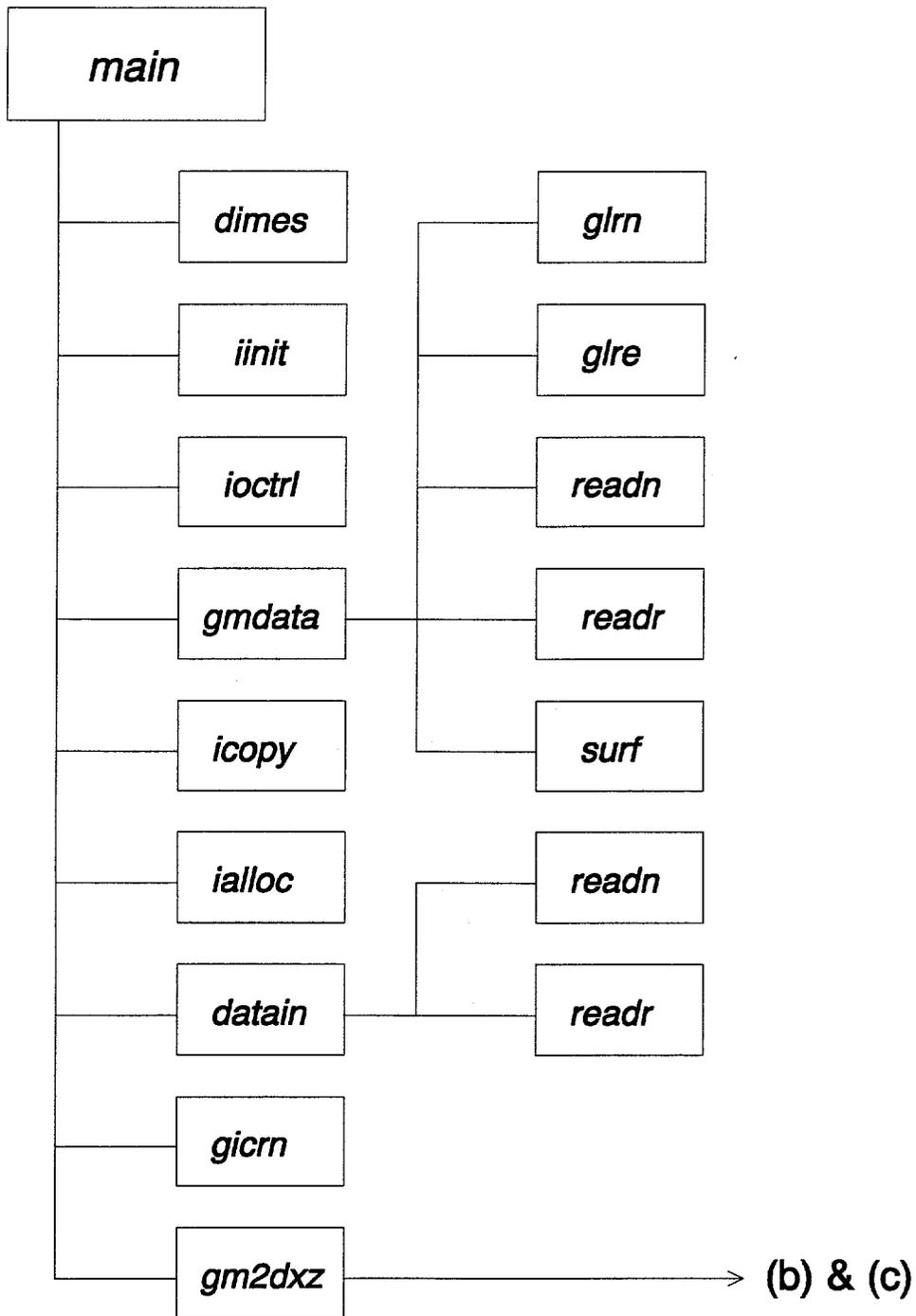


Fig. 3.1. Program structure of MURT (1 of 3).

(b) Subroutine *gm2dxz*, steady state

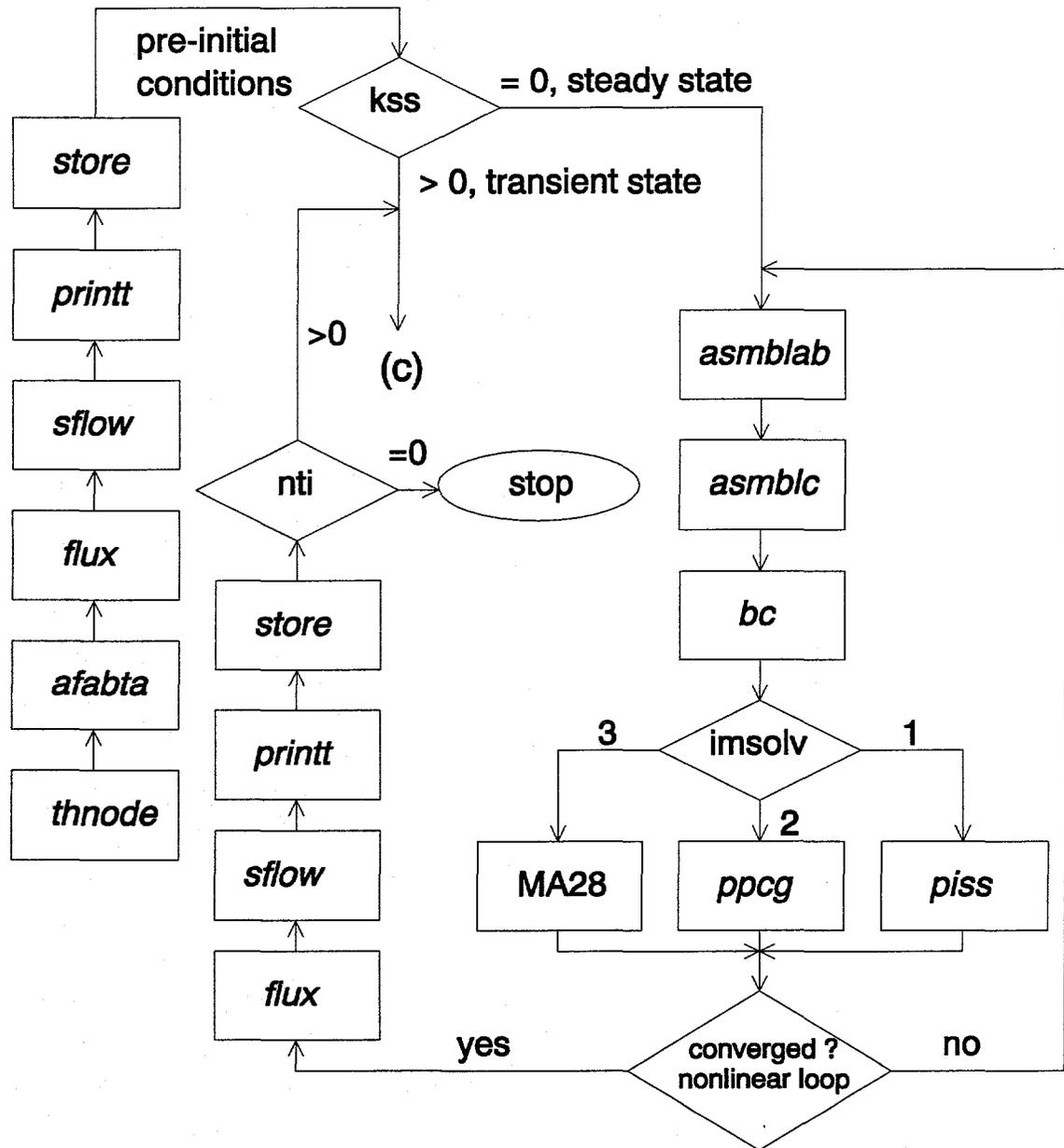


Fig. 3.1 (continued, 2 of 3)

(c) Subroutine *gm2dxz*, transient state

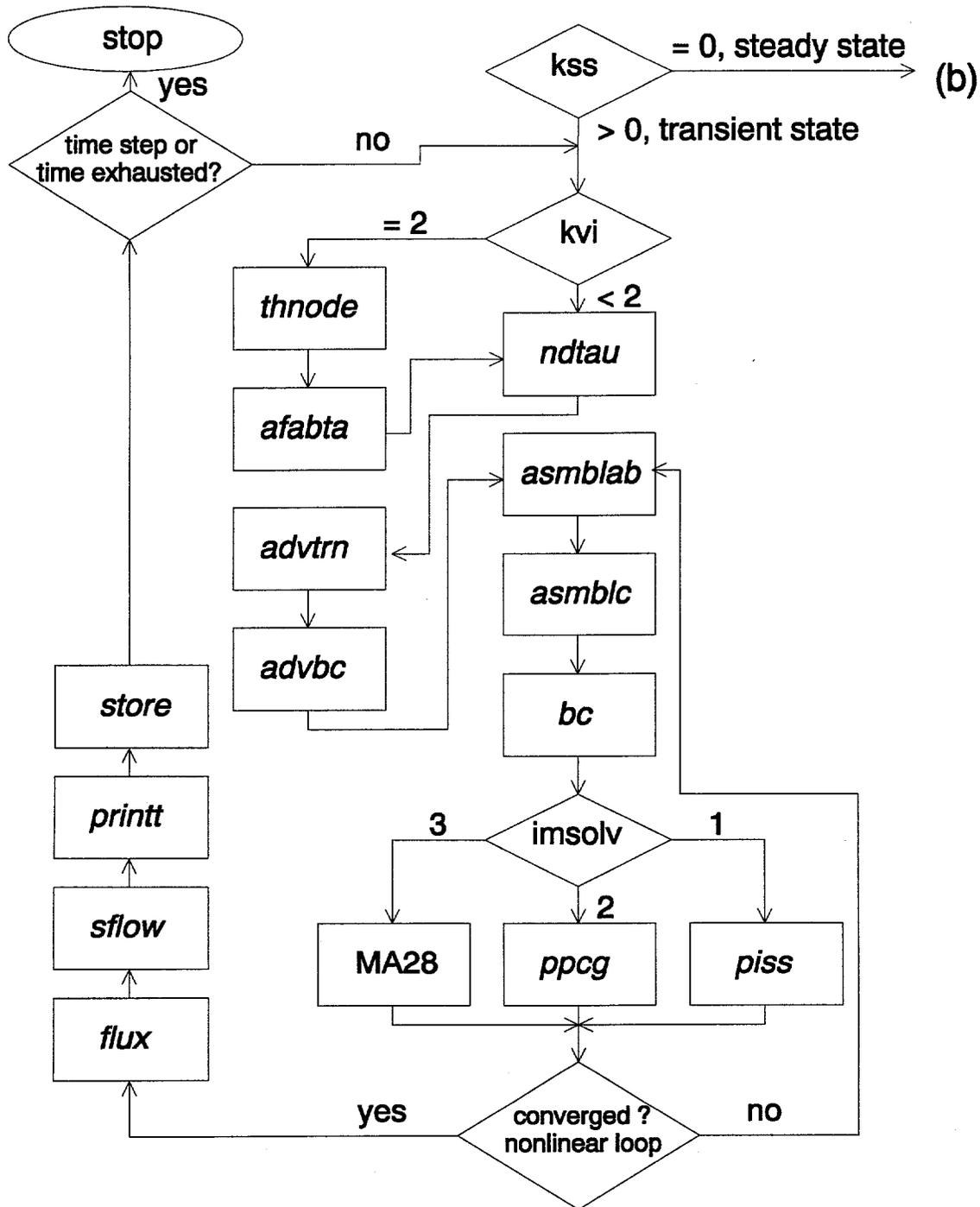


Fig. 3.1 (continued, 3 of 3)

the variable boundary node. Cauchy boundary conditions are normally applied to the boundaries where flow is directed into the region. The Lagrangian concentration can thus be computed similarly by

$$c_i^* = \frac{\int_{B_c} N_i^e q_{ic} dB}{\int_{B_c} N_i^e V_c dB}, \quad (3.2)$$

where  $q_{ic}$  is the Cauchy flux of the incoming fluid at Cauchy node  $i$  and  $V_c$  is the normal velocity to the Cauchy boundary side.

### 3.1.2 Subroutine *advtrn*

Subroutine *advtrn* computes the Lagrangian concentration at all nodes. *Advtrn* calls *mploc* to find the element where a particle is located. *Xsi2d* is then called to compute the local coordinates given the global coordinates of the fictitious particle. The concentration at the location of the fictitious particle is then interpolated using the concentrations of the nodal points associated with the element.

### 3.1.3 Subroutine *afabta*

Subroutine *afabta* calculates the values of upstream weighting factors along four sides of all elements. The weighting factors thus obtained are used for steady-state simulations or when upstream methods are required by modelers.

### 3.1.4 Subroutine *asmlab*

Subroutine *asmlab* calls subroutines *q4ab* and *q3ab* to compute the element matrices derived from the storage, advection, and dispersion terms and element load vectors from the (non-point) source/sink terms in the governing equations, for quadrilateral and triangular elements, respectively. The element matrices and vectors are assembled into a global matrix equation, which is the linearized version of the governing equations.

### 3.1.5 Subroutine *asmblc*

Subroutine *asmblc* calls subroutines *q4c* and *q3c* to compute the element matrices derived from the mass transfer terms in the governing equations, for quadrilateral and triangular elements, respectively. The element matrices are assembled into the global matrix equation obtained in subroutine *asmlab*.

### 3.1.6 Subroutine *bc*

Subroutine *bc* implements Dirichlet, Cauchy, Neumann, and variable boundary conditions. The Dirichlet boundary conditions are implemented by normalizing the

associated diagonal elements of the global matrix. The associated off-diagonal elements are then set to zero and the associated elements of the load vector are set to the assigned concentrations. For Cauchy boundaries, the integration of the Cauchy flux is added to the global load vector and the integration of the normal velocity is added to the matrix. For Neumann boundaries, the integration of the gradient-driven flux is added to the global load vector. For variable boundaries, the integration of the normal velocity times the incoming concentration is added to the load vector and the integration of the normal velocity is added to the matrix.

### 3.1.7 Subroutines *cbvfct*, *dbvfct*, *essfct*, *nbvfct*, *vbvfct*, *wssfct*

Temporal variations of boundary conditions and source strength are described by transient profiles in MURT. In each of these six subroutines, transient profiles can be either analytical or tabular. If an analytical profile is to be used, the user needs to supply the analytical function and modify the subroutine.

Subroutine	Function
<i>cbvfct</i>	Computes the normal flux for Cauchy boundaries and returns the flux as a one-dimensional array to subroutine <i>gm2dxz</i> .
<i>dbvfct</i>	Computes Dirichlet concentrations for Dirichlet boundaries and returns them as a one-dimensional array to subroutine <i>gm2dxz</i> .
<i>essfct</i>	Computes the distributed source/sink strength, including flow rates and concentrations, for distributed source/sink elements and returns the results as a one-dimensional array to subroutine <i>gm2dxz</i> .
<i>nbvfct</i>	Computes the normal flux for Neumann boundaries and returns the flux as a one-dimensional array to subroutine <i>gm2dxz</i> .
<i>vbvfct</i>	Computes the run-in-seep-out mass fluxes at variable boundaries and returns the fluxes as a one-dimensional array to subroutine <i>gm2dxz</i> .
<i>wssfct</i>	Computes the point (well) source/sink strength, including flow rates and concentrations, for point source/sink nodes and returns the results as a one-dimensional array to subroutine <i>gm2dxz</i> .

### 3.1.8 Subroutine *datain*

Subroutine *datain* reads data contained in two input data files, namely, the control and option parameters, material property, and material type change file and the initial and boundary condition file (see Appendix C). Subroutines *readn* and *readr* are called to read input data of types integer and real, respectively, with patterns as defined in Appendix C.

### 3.1.9 Subroutine *dgelg*

Subroutine *dgelg* is called by the pointwise iterative solver *piss* to solve the diagonal blocks of the global matrix equation.

### 3.1.10 Subroutine *dimes*

Subroutine *dimes* reads user-specified dimensions of arrays that are required for a simulation. These dimensions are necessary for MURT to determine the size of the problem and allocate computer memory for the arrays. Array dimensions input through this subroutine include numbers of nodes and elements, numbers of parameters that define material properties, number of material type changes, and numbers that define boundary conditions. The input data file corresponding to this subroutine is the array dimension file (see Fig. 4.1 in the next section for an example).

### 3.1.11 Subroutine *dispc*

Subroutine *dispc* calculates the harmonic means of dispersion coefficients at all nodes based on the areas and dispersion coefficients of the elements connecting to a node.

### 3.1.12 Real Function *fcos*

FORTRAN user function *fcos* computes the size and direction of the normal vector of a surface formed by the vector from an arbitrary point to one of the end points of the boundary side and the vector formed by the two end points of the side. The sign of the normal vector is used by subroutine *mploc* to determine if a fictitious particle is inside an element.

### 3.1.13 Subroutine *flux*

Subroutine *flux* calls *q4d* or *q3d* to compute element matrices and the derivatives of the concentrations. The element matrices are then assembled over the problem domain to form a matrix equation governing the material flux components at all nodal points. The coefficient matrix is lumped to save compute time and the matrix equation is solved point by point.

### 3.1.14 Subroutine *gicrn*

Subroutine *gicrn* generates indices arrays ICN(i) and IRN(i) for the coefficient matrix C(i) when sparse matrix solver MA28 is to be used. Under this circumstance, C(i) is a one-dimensional array and therefore needs row index IRN(i) and column index ICN(i) to represent a two-dimensional array.

### 3.1.15 Subroutine *glre*

Subroutine *glre* generates the array LRE(i,j) that contains the global numbers of elements connected to a nodal point. This array is used later to generate LRN(i,j) that contains the global numbers of nodes connected to a nodal point. LRN(i,j), in turn, is used to generate the index arrays ICN(i) and IRN(i) for sparse matrix solver MA28 and, more importantly, is needed to assemble local element matrices into global matrices.

### 3.1.16 Subroutine *glrn*

Subroutine *glrn* generates the array  $LRN(i,j)$  that contains the global numbers of nodes connected to a nodal point using the information contained in  $LRE(i,j)$ .  $LRN(i,j)$ , in turn, is used to generate the index arrays  $ICN(i)$  and  $IRN(i)$  for sparse matrix solver MA28 and, more importantly, is needed to assemble local element matrices into global matrices. Two by-products, the internal boundary marker arrays  $NTBND(i,j)$  and  $NEBT(i,j)$ , are used by subroutine *surf* to identify boundary sides and to calculate length and directional cosines at these sides.

### 3.1.17 Subroutine *gm2dxz*

Subroutine *gm2dxz* controls the flow of execution for both steady-state and transient simulations. For parameter  $NTI > 0$ , the simulations are transient-state without regard to the values of  $KSS$ . However, if  $KSS = 0$  and  $NTI > 0$ , the user input initial condition will be ignored. The result of a steady-state simulation will be used as the initial condition for the subsequent transient-state simulation.

Subroutine *gm2dxz* consists of two main iteration loops to compute steady-state and transient solute concentrations, respectively. Within the steady-state iteration loop, the nonlinear solute transport equation is linearized when Freundlich and/or Langmuir isotherms are used, and the linearized equation is passed to the sparse matrix solver MA28, a preconditioned conjugate gradient matrix solver, or a pointwise iteration matrix solver for solutions. The structure of the transient-state loop is similar to the steady-state loop except that outside the nonlinear loop there is a time-marching loop to advance the computations along the time axis.

Include files *common.h*, *sparse.h*, and *para.h* are included in subroutine *gm2dxz*. Subroutine *gm2dxz* calls *dbvfct*, *vbfct*, *cbvfct*, and *nbvfct* to calculate boundary concentration or flux values and *essfct* and *wssfct* to calculate external source/sink strength. Subroutines *flux* and *sflow* are called to calculate mass fluxes and surface fluxes, respectively. Subroutines *printt* and *store* print and store computational results to disk files, respectively. Subroutine *thnode* is called to calculate nodal water contents given the elemental water contents calculated by MURF (Gwo et al. 1994b) or input by users. Subroutine *afabta* is called to calculate upstream weighting functions when steady-state simulations are intended or the upstream weighting method is chosen. *Gm2dxz* also calls subroutine *ndtau* to calculate the maximum sub-time-step sizes allowable for backward particle tracking (see Sect. 3.1.26), subroutine *advtrn* to obtain Lagrangian concentration (see Sect. 3.1.2), and subroutine *advbc* to implement Lagrangian boundary conditions (see Sect. 3.1.1). Subroutines *asmlab* and *asmlbc* are called to assemble element matrices and vectors related to (1) advection, dispersion, mass storage and adsorption/desorption, and sources/sinks and (2) element mass transfer matrices, respectively, into the global coefficient matrix and load vector. Subroutine *bc* is then called to apply boundary conditions. There are three matrix solvers: (1) *piss* invokes a pointwise iteration matrix solver; (2) *ppcg* invokes a polynomial preconditioned

conjugate gradient matrix solver; and (3) *ma28a* decomposes the global coefficient matrix and *ma28c* solves the decomposed matrix equation by forward and backward substitutions.

### **3.1.18 Subroutine *gmdata***

Subroutine *gmdata* is the driver routine for reading geometry data from the input data file: geometry of problem domain and material type corrections (see Appendix C). The routine calls subroutine *readr* to read *x* and *z* coordinates, subroutine *glre* and *glrn* to generate arrays that contain the numbers of elements and nodes connecting to a node, respectively, and subroutine *surf* to generate boundary side information.

### **3.1.19 Integer Function *ialloc***

Integer function *ialloc* allocates space in the array MEM(mxmem) to a dimensional array by determining the location of the dimensional array in MEM(mxmem). An integer is returned indicating the beginning position of the array in MEM(mxmem). Subroutine *assert* is called to print out error messages if errors occur during an allocation. Integrity of the space allocation is checked by calling subroutine *icheck*.

### **3.1.20 Subroutine *iinit***

Subroutine *iinit* initializes the integer dimension array MEM(mxmem) that is used for memory allocation, where mxmem is the maximum length of the array. All of the global dimension arrays used in MURT, both integer and real, are allocated using the memory space claimed by MEM(mxmem). A function, *ialloc* (see Sect. 3.1.19), is called to allocate space for and to determine the locations of dimensional arrays within MEM(mxmem).

### **3.1.21 Subroutine *intrbnd***

Subroutine *intrbnd* is called by subroutine *glrn* to mark internal boundary sides between elements. These internal boundary sides are needed to identify elements that do not have common sides with the global boundary and are used in subroutine *surf* to identify external boundary sides.

### **3.1.22 Subroutine *ioctrl***

Subroutine *ioctrl* reads the three parameters IGEOM, IMOD, and KSTR that determine how geometry data will be input, how much of these data will be printed to disk files, and whether the job is a restart job, respectively. The input data file corresponding to this subroutine is the I/O control parameters file (see Fig. 4.2 in the next section for an example), which is provided through an input template.

### 3.1.23 Matrix Solver MA28

MA28 is a sparse matrix solver for large linear symmetric or asymmetric systems (Duff 1990). The subroutine *ma28a* is called by *gm2dxz* to decompose the coefficient matrix of the target linear system; *ma28c* is then called to do forward and backward substitutions and to obtain the solution of the linear system. Users are referred to Duff (1990) for a detailed description of the package.

### 3.1.24 Program *main*

The main program performs the following tasks:

- opens I/O files for input data and computation output;
- determines the array dimensions and allocates memory space for arrays according to user specified problem sizes;
- reads problem geometry data, including nodal coordinates and connectivity, from input files;
- reads computation control parameters, soil properties, material types of elements, and initial and boundary conditions;
- generates array indices for sparse matrix solver MA28 if necessary; and
- passes control to subroutine *gm2dxz* for job execution.

The main program includes three include files, namely, *common.h*, *sparse.h*, and *para.h*, and calls the subroutines *dimes*, *iinit*, *ioctrl*, *gmdata*, *icopy*, *datain*, *gicrn*, *gm2dxz*, and a function *ialloc*. The include files *common.h* and *sparse.h* contain FORTRAN COMMON statements needed for MURT and sparse matrix solver MA28, respectively. The include file *para.h* specifies parameters used by MURT. Memory of dimensional arrays is allocated by the code automatically according to the dimensions given in the array dimension file (see Appendix C). Therefore, virtually no change of code is necessary unless users are to use analytical transient boundary profiles of their own.

### 3.1.25 Subroutine *mploc*

Subroutine *mploc* is called by *ndtau* and *advtrn* to identify the element in which a fictitious particle is located. A real function *fcos* is called to compute the cross product of the a vector formed by the two end points of a boundary side and the vector from a node on the surface to the fictitious particle. The value of the product is then used to determine if the fictitious particle is inside the element.

### 3.1.26 Subroutine *ndtau*

Subroutine *ndtau* is called by *gm2dxz* to compute the sub-time-step size and the number of this time-step size so that no fictitious particle would travel over an element within the sub-time-step size. This information will then be used in *advtrn* to do the backward particle tracking.

### 3.1.27 Subroutine *piss*

Subroutine *piss* implements a pointwise iterative solver for the linearized equation. The diagonal blocks of the matrix equation are solved by calling subroutine *dgelg*, which is a full pivoting matrix solver. A sweep therefore consists of moving the upper triangle blocks to the right-hand side, using updated values for the lower triangle blocks, and solving the diagonal blocks.

### 3.1.28 Subroutine *ppcg*

Subroutine *ppcg* implements a polynomial preconditioned conjugate gradient matrix solver by calling the preconditioner subroutine *precond*.

### 3.1.29 Subroutine *precond*

Given a coefficient matrix, subroutine *precond* calculates a polynomial preconditioner for subroutine *ppcg*. Optimization coefficients for the polynomial are estimated from the coefficient matrix and the highest order of the polynomial is set to 2.

### 3.1.30 Subroutine *printt*

Subroutine *printt* prints the modeling results, including surface flux rates at all types of boundaries, solute concentrations, and material flux components in  $x$  and  $y$  directions. The amount of printout is controlled by  $KPR0$  and  $KPR(itm)$ , where  $itm$  is the time step number (see Appendix C.2).

### 3.1.31 Subroutines *q2cb* and *q2vb*

Subroutine *q2cb* is called by subroutine *bc* to compute the normal flux along a Cauchy boundary side according to the following relation:

$$BQ(i, j) = \int_{B_c} N_i^e \mathbf{n} \cdot \mathbf{V} N_j^e dB . \quad (3.3)$$

Subroutine *q2vb* is called by subroutine *bc* to compute the normal flux along a variable boundary side as follows:

$$BQ(i, j) = \int_{B_v} N_i^e \mathbf{n} \cdot \mathbf{V} N_j^e dB, \quad \mathbf{n} \cdot \mathbf{V} < 0 , \quad (3.4a)$$

$$BQ(i, j) = 0, \quad \mathbf{n} \cdot \mathbf{V} \geq 0 . \quad (3.4b)$$

### 3.1.32 Subroutines *q3ab* and *q4ab*

Subroutines *q3ab* and *q4ab* calculate the element matrices derived from the storage, advection, and dispersion terms and element load vectors derived from the (non-point) source/sink terms in the governing equations, for quadrilateral and triangular elements, respectively. The integration in subroutine *q4ab* is computed using four-point Gaussian quadrature, while the integration in subroutine *q3ab* is obtained analytically. The storage terms are computed according to the following equations:

$$QA(i, j) = \int_{R_e} N_i^e \theta N_j^e dR, \quad (3.5)$$

$$QAA(i, j) = \int_{R_e} N_i^e \rho_b N_j^e dR. \quad (3.6)$$

The dispersion term is calculated according to the following relation:

$$QB(i, j) = \int_{R_e} \nabla N_i^e \cdot \theta D \nabla N_j^e dR. \quad (3.7)$$

The advection term is computed using the following relation:

$$QV(i, j) = \int_{R_e} N_i^e V \cdot \nabla N_j^e dR. \quad (3.8)$$

In Eqs. (3.5) through (3.8),  $\theta$  is water content,  $\rho_b$  is bulk density,  $N_i^e$  is the base function associated with node  $i$  of element  $e$ ,  $D$  is the dispersion coefficient tensor, and  $V$  is the darcy velocity or specific discharge vector. The first-order decay terms are calculated as follows:

$$QC(i, j) = \int_{R_e} N_i^e [\lambda \theta - K_w] N_j^e dR + \int_{R_e} N_i^e [\lambda + K_s] \rho_b \frac{ds}{dc} N_j^e dR, \quad (3.9)$$

where  $\lambda$  is the first-order radioactive decay constant,  $K_w$  is the first-order biodegradation constant in the fluid phase,  $K_s$  is the first-order biodegradation constant in the solid phase, and  $s$  is the concentration in the solid phase. The load vectors of the element matrix equations are calculated according to the following relation:

$$RQ(i) = \int_{R_e} N_i^e \left[ \theta q c_{in} - \rho_b (\lambda + K_s) (s_0 - \frac{ds}{dc} c_0) \right] N_j^e dR, \quad (3.10)$$

where  $q$  is the external non-point source/sink flux rate,  $c_{in}$  is the solute concentration in the incoming fluid,  $s_0$  and  $c_0$  are the solid and fluid phase concentrations at previous iterate or time step.

### 3.1.33 Subroutines *q3c* and *q4c*

Subroutines *q4c* and *q3c* calculate the element matrices derived from the mass transfer terms in the governing equations for quadrilateral and triangular elements, respectively. The integration in subroutine *q4c* is computed using four-point Gaussian quadrature, while the integration in subroutine *q3c* is obtained analytically. The advective mass transfer terms between regions *ir* and *jr* are computed according to

$$QCF(i, j, ir, jr) = \int_{R_e} N_i^e \epsilon_{ir,jr}^f N_j^e dR, \quad (3.11)$$

and the diffusive mass transfer terms between regions *ir* and *jr* are calculated according to

$$QCC(i, j, ir, jr) = \int_{R_e} N_i^e \epsilon_{ir,jr}^t N_j^e dR, \quad (3.12)$$

where  $\epsilon_{ir,jr}^f$  and  $\epsilon_{ir,jr}^t$  are the advective and diffusive mass transfer coefficients between pore regions *ir* and *jr*, respectively.

### 3.1.34 Subroutines *q3d* and *q4d*

Subroutines *q3d* and *q4d* are called by subroutine *flux* to compute the element matrix given by the following relation:

$$QB(i, j) = \int_{R_e} N_i^e N_j^e dR, \quad (3.13)$$

The load vectors of the material flux components are calculated according to

$$QRX(i) = - \int_{R_e} N_i^e \mathbf{i} \cdot \theta \mathbf{D} \cdot (\nabla N_j^e) c_j dR, \quad (3.14)$$

$$QRZ(i) = - \int_{R_e} N_i^e \mathbf{k} \cdot \theta \mathbf{D} \cdot (\nabla N_j^e) c_j dR, \quad (3.15)$$

where  $c_j$  is the concentration at local node *j* and  $\mathbf{i}$  and  $\mathbf{k}$  are the unit vectors along *x* and *z* coordinate axes, respectively.

### 3.1.35 Subroutines *q3r* and *q4r*

Subroutines *q3r* and *q4r* are used to compute the contribution of material accumulation rates of quadrilateral and triangular elements, respectively, as follows:

$$QRM = \int_R \theta c dR , \quad (3.16)$$

$$QDM = \int_R s dR , \quad (3.17)$$

$$SOSM = \int_R q c_{in} dR . \quad (3.18)$$

### 3.1.36 Subroutine *readn*

Given appropriate parameters, *readn* automatically generates sequences of integer numbers and returns the numbers as a one-dimensional array to the calling routines. A flag that signals the calling routines as to whether or not the number of records read and generated is in agreement with that specified is also returned.

### 3.1.37 Subroutine *readr*

Given appropriate parameters, *readr* automatically generates sequences of real numbers and returns the numbers as a one-dimensional array to the calling routines. A flag that signals the calling routines as to whether or not the number of records read and generated is in agreement with that specified is also returned.

### 3.1.38 Subroutine *sflow*

Subroutine *sflow* computes the fluxes through various types of boundaries and the increasing rates of mass in the region of interest. *FRATE(7)* stores the flux through the entire boundary enclosing the problem domain. It is calculated by

$$FRATE(7) = \int_B (F_x n_x + F_z n_z) dB , \quad (3.19)$$

where  $B$  is the global boundary of the problem domain,  $F_x$  and  $F_z$  are the material flux components in  $x$  and  $z$  directions, and  $n_x$  and  $n_z$  are the directional cosines of the outward unit vector normal to the boundary  $B$ . *FRATE(1)* to *FRATE(5)* are the fluxes through Dirichlet boundaries  $B_d$ , Cauchy boundaries  $B_c$ , Neumann boundaries  $B_n$ , and the incoming and outgoing flux through the variable boundaries  $B_v$ , respectively. These fluxes are calculated according to the following relations:

$$FRATE(1) = \int_{B_d} (F_x n_x + F_z n_z) dB , \quad (3.20)$$

$$FRATE(2) = \int_{B_c} (F_x n_x + F_z n_z) dB , \quad (3.21)$$

$$FRATE(3) = \int_{B_n} (F_x n_x + F_z n_z) dB , \quad (3.22)$$

$$FRATE(4) = \int_{B_v} (F_x n_x + F_z n_z) dB \quad n \cdot V_v < 0, \quad (3.23)$$

$$FRATE(5) = \int_{B_v} (F_x n_x + F_z n_z) dB \quad n \cdot V_v \geq 0, \quad (3.24)$$

where  $V_v$  is velocity vector at a variable boundary side.  $FRATE(6)$ , which is related to numerical loss, is given by

$$FRATE(6) = FRATE(7) - \sum_{i=1}^5 FRATE(i). \quad (3.25)$$

$FRATE(8)$  and  $FRATE(9)$  are the accumulating rates in dissolved and adsorbed phases within the problem domain, respectively, and are calculated according to

$$FRATE(8) = \int_R \theta \frac{\partial c}{\partial t} dR \quad (3.26)$$

and

$$FRATE(9) = \int_R \rho_b \frac{\partial s}{\partial t} dR. \quad (3.27)$$

$FRATE(10)$  stores the rate of loss due to decay according to the following relations:

$$FRATE(10) = \int_R \lambda(\theta c + \rho_b s) dR, \quad (3.28)$$

and

$$FRATE(11) = 0. \quad (3.29)$$

$FRATE(12)$  and  $FRATE(13)$  are the loss due to biodegradation through dissolved and adsorbed phases, respectively, according to the following relations:

$$FRATE(12) = \int_R K_w c dR, \quad (3.30)$$

and

$$FRATE(13) = \int_R K_s \rho_b s dR. \quad (3.31)$$

$FRATE(14)$  is for the source/sink rates:

$$FRATE(14) = \int_R q c_{in} dR. \quad (3.32)$$

If there is no numerical error, the following relation should be satisfied:

$$\sum_{i=7}^{14} FRATE(i) = 0 , \quad (3.33)$$

and  $FRATE(6)$  should be zero. Equation (3.33) states that the net rate of material going out of and coming into the region through the boundaries and the sources/sinks should be equal to the rate of material accumulation in the region. In other words, mass must be conserved.

### 3.1.39 Subroutine *shape*

Subroutine *shape* is called by subroutines *q4ab* and *q4c* to compute the base functions at the Gaussian points. For MURT, a bilinear base function is used for all of the quadrilateral elements.

### 3.1.40 Subroutine *store*

Subroutine *store* stores the computation results in a designated logical unit. The information stored includes domain geometry, concentrations, and material flux components.

### 3.1.41 Subroutine *surf*

Subroutine *surf* identifies the boundary sides, sequences the boundary nodes, and computes length and the directional cosines of the surface sides. Boundary side information is necessary for calculating boundary fluxes in routine *bc* and for computing mass balance error in routine *sflow*.

### 3.1.42 Subroutine *thnode*

Subroutine *thnode* computes the harmonic means of water contents at nodal points by using the water contents at the Gaussian points given by users or calculated by MURF (Gwo et al. 1994b). The result is returned to *gm2dxz* for the computation of particle or fluid velocities during backward particle tracking.

### 3.1.43 Subroutine *xsi2d*

Given the global coordinates of a particle, *xsi2d* computes the local coordinates associated with the element where the particle is located. These coordinates are needed when nodal concentrations are used to interpolate the concentration at the location of the particle.

### 3.2 I/O UNITS

Fourteen I/O units are used by MURT and are related as shown in Table 3.1. I/O units 7, 8, 15, 21, 23, and 24 are the five standard input data files in ASCII format (FORTRAN formatted files). I/O unit 16 is reserved for the hydraulic parameter file generated by MURF (Gwo et al. 1994b). I/O units 9 and 22 contain the echo output of geometry and other input data as described in Appendix C. Five FORTRAN unformatted files, I/O units 10 to 14, are generated each time a simulation is executed. These I/O units store geometry data, boundary arrays, pointer arrays, restart information, and simulation results. Except for the unit containing simulation results, these I/O units will be used if geometry is to be input through an unformatted file or if restart jobs are requested.

**Table 3.1. I/O units used by MURT**

Unit	File name	Data contained and format
7	<sup>a</sup>	Control and option parameters, material properties and types, ASCII (FORTRAN formatted)
8	<sup>a</sup>	Problem domain geometry and element connectivity, ASCII (FORTRAN formatted)
9	<sup>a</sup>	simulation results, ASCII (FORTRAN formatted)
10	<i>murt.geo</i>	Problem domain geometry and element connectivity, FORTRAN unformatted
11	<i>murt.bnd</i>	Boundary arrays, FORTRAN unformatted
12	<i>murt.pnt</i>	Pointer arrays, FORTRAN unformatted
13	<i>murt.rst</i>	Restart information, FORTRAN unformatted
14	<i>murt.sto</i>	Simulation results, FORTRAN unformatted
15	<sup>a</sup>	Hydraulic parameters, ASCII (FORTRAN formatted)
16	<sup>a</sup>	Hydraulic parameters generated by MURF, FORTRAN unformatted
21	<sup>a</sup>	Array dimensions, ASCII (FORTRAN formatted)
22	<sup>a</sup>	Echo output of input data, ASCII (FORTRAN formatted)
23	<sup>a</sup>	I/O control parameters, ASCII (FORTRAN formatted)
24	<sup>-a</sup>	Initial and boundary conditions, ASCII (FORTRAN formatted)

<sup>a</sup>File name specified by users.

#### 3.2.1 Geometry in Unformatted I/O Unit 10

The file should always be named *murt.geo*, and the data are written to a FORTRAN unformatted file according to the following FORTRAN statements:

```
write (ioge) nnp, (x(n), z(n), n=1, nnp)
write (ioge) nel, ((ie(m, iq), m=1, nel), iq=1, 5) ,
```

where

*iogeo* = I/O unit number of the unformatted file, FORTRAN integer number, 4 bytes;  
*nnp* = total number of finite element nodes, FORTRAN integer number, 4 bytes;  
*x*(*n*) = the *x* coordinate at node *n*, FORTRAN real number, 8 bytes;  
*z*(*n*) = the *z* coordinate at node *n*, FORTRAN real number, 8 bytes;  
*nel* = total number of elements, FORTRAN integer number, 4 bytes;  
*ie*(*m*, *iq*=1, . . . , 4) = the vertex node numbers associated with element *m*; if the element is a triangle, *ie*(*m*, 4) should be set to integer 0, FORTRAN integer number, 4 bytes;  
*ie*(*m*, 5) = the material type of element *m*, FORTRAN integer number, 4 bytes.

### 3.2.2 Boundary Arrays in Unformatted I/O Unit 11

Data written to this I/O unit are generated in subroutine *surf*, which is called by subroutine *gmdata*. The file should always be named *murt.bnd* and the data written to a FORTRAN unformatted file according to the following FORTRAN statement:

```

write (iobnd) nbn,nbel, (dlb(m),m=1,nbel),
  (dcosxb(m),m=1,nbel), (dcoszb(m),m=1,nbel),
  (nbe(m),m=1,nbel), ((isb(iq,m),iq=1,4),m=1,nbel),
  (npb(n),n=1,nbn) ,
  
```

where

*iobnd* = I/O unit number of the boundary array file, FORTRAN integer number, 4 bytes;  
*nbn* = the total number of boundary nodes, FORTRAN integer number, 4 bytes;  
*nbel* = the total number of boundary sides, FORTRAN integer number, 4 bytes;  
*dlb*(*m*) = the length of boundary side *m*, FORTRAN real number, 8 bytes;  
*dcosxb*(*m*) = directional cosine along *x* axis for boundary side *m*, FORTRAN real number, 8 bytes;  
*dcoszb*(*m*) = directional cosine along *z* axis for boundary side *m*, FORTRAN real number, 8 bytes;  
*nbe*(*m*) = global element number associated with boundary side *m*, FORTRAN integer number, 4 bytes;  
*npb*(*n*) = global node number of boundary node *n*, FORTRAN integer number, 4 bytes;  
*isb*(1, *m*) = the global node number of the first node associated with boundary side *m*, FORTRAN integer number, 4 bytes;  
*isb*(2, *m*) = the global node number of the second node associated with boundary side *m*, FORTRAN integer number, 4 bytes;  
*isb*(3, *m*) = the local node number of the first node associated with boundary side *m*, FORTRAN integer number, 4 bytes;  
*isb*(4, *m*) = the local node number of the second node associated with boundary side *m*, FORTRAN integer number, 4 bytes.

### 3.2.3 Pointer Arrays in Unformatted I/O Unit 12

Data written to this I/O unit are generated in subroutines *glre* and *glrn*, which are called by subroutine *gmdata*. The file should always be named *murt.pnt* and the data are written to a FORTRAN unformatted file according to the following FORTRAN statements:

```
write (iopnt) nlre, ((lre(nj,ne),ne=1,nnp),nj=0,nlre)
write (iopnt) nlrn, ((lrn(nj,np),np=1,nnp),nj=0,nlrn) ,
```

where

*iopnt* = I/O unit number of the unformatted file, FORTRAN integer number, 4 bytes,  
*nlre* = the number of elements connected to a node, FORTRAN integer number, 4 bytes,  
*nlrn* = the number of nodes connected to a node, FORTRAN integer number, 4 bytes,  
*lre(nj,ne)* = the *nj*th element connected to node *ne*, FORTRAN integer number, 4 bytes,  
*lrn(nj,np)* = the *nj*th node connected to node *np*, FORTRAN integer number, 4 bytes.

### 3.2.4 Restart Data in Unformatted I/O Unit 13

The file should always be named *murt.rst*, and the data contained in the file are written to a FORTRAN unformatted file according to the following FORTRAN statements:

```
write (iostr) time, ((rp(n,idf),n=1,nnp), (fx(n,idf),
n=1,nnp), (fz(n,idf),n=1,nnp), idf=1,ndf) ,
```

where

*iostr* = I/O unit number of the unformatted file, FORTRAN integer number, 4 bytes;  
*time* = time of the current time step, FORTRAN real number, 8 bytes;  
*rp(n,idf)* = concentration at global node *n* and pore region *idf*, FORTRAN real number, 8 bytes;  
*fx(n,idf)* = *x* component of the material flux at global node *n* and pore region *idf*, FORTRAN integer number, 4 bytes;  
*fz(n,idf)* = *z* component of the material flux at global node *n* and pore region *idf*, FORTRAN integer number, 4 bytes;  
*ndf* = number of pore regions.

### 3.2.5 Simulation Results in Unformatted I/O Unit 14

Data written to this I/O unit are generated in subroutine *gm2dxz* and are passed to subroutine *store* to write to the I/O unit. The file should always be named *murt.sto* and

the data written to a FORTRAN unformatted file according to the following FORTRAN statements:

- (1) The following data are written only once:

```
write(iosto) title,nprob,nnp,nel,ndf,nti
write(iosto) (x(n),n=1,nnp), (z(n),n=1,nnp),
  ((ie(m,i),m=1,nel),i=1,5) ,
```

- (2) The following data are written for the initial time step at time = 0.0 and the time steps with  $k_{dsk}(itm)$  greater than zero, where  $k_{dsk}(itm)$  is the storage control variable at time step  $itm$ :

```
write(iosto) itm,time,((r(n,idf),n=1,nnp), (fx(n,idf),
  n=1,nnp), (fz(n,idf),n=1,nnp), idf=1,ndf) ,
```

where

$iosto$  = I/O unit number of the unformatted file, FORTRAN integer number, 4 bytes;

$title$  = the title of the problem set, FORTRAN character string, 72 bytes;

$nprob$  = problem number of the simulation, FORTRAN integer number, 4 bytes;

$nti$  = number of time steps, FORTRAN integer number, 4 bytes;

$r(n, idf)$  = concentration at global node  $n$  and pore region  $idf$ , FORTRAN real number, 8 bytes.

### 3.2.6 Hydraulic Parameters Generated by Multiregion Flow Code MURF in Unformatted I/O Unit 16

Data written to this I/O unit are generated by the subsurface multiregion flow code MURF. Users must provide the file and path names of this I/O unit in the file specifying the I/O file names (see Appendix C). The data are written to a FORTRAN unformatted file according to the following FORTRAN statements:

- (1) The following data are written only once:

```
write(iosto) title,nprob,nnp,nel,ndf,nbn,nbel,nti
write(iosto) (x(n),n=1,nnp), (z(n),n=1,nnp),
  ((ie(m,i),m=1,nel),i=1,5), (dlb(m),m=1,nbel),
  (dcosxb(m),m=1,nbel), (dcoszb(m),m=1,nbel),
  (nbe(m),m=1,nbel), ((isb(i,m),i=1,4),m=1,nbel),
  (npb(n),n=1,nbn) ,
```

- (2) the following data are written for the initial time step at time = 0.0 and the time steps with  $k_{dsk}(itm)$  greater than zero, where  $k_{dsk}(itm)$  is the storage control variable at time step  $itm$ :

```
write(iosto) time, ((h(n,idf),n=1,nnp),  
  (ht(n,idf),n=1,nnp), ((th(i,m,idf),i=1,4),m=1,nel),  
  (vx(n,idf),n=1,nnp), (vz(n,idf),n=1,nnp) ,
```

where

ht(n, idf) = hydraulic head at global node n and pore region idf, FORTRAN  
real number, 8 bytes;

th(i, m, idf) = water content at local node i, element m, and pore region idf,  
FORTRAN real number, 8 bytes;

vx(n, idf) = x direction darcy velocity at global node n, pore region idf,  
FORTRAN real number, 8 bytes;

vz(n, idf) = z direction darcy velocity at global node n, pore region idf,  
FORTRAN real number, 8 bytes.



#### 4. SITE SPECIFIC APPLICATIONS OF MURT

Three applications of MURT are presented in this section. The first application is a soil column tracer injection problem, where a nonreactive tracer was injected into two undisturbed soil columns. The second example is a sensitivity analysis on advective and diffusive mass transfer. A transient infiltration-evaporation problem was simulated using a homogeneous soil column consisting of three pore regions. Fracture spacing data at ORNL were used to calculate the range of advective and diffusive mass transfer coefficients, and the models MURF (Gwo et al. 1994b) and MURT were used to study the movement of water and a nonreactive solute. The effects of advective and diffusive mass transfer on local-scale nonequilibrium were compared and discussed. The third example is a hillslope ponding infiltration problem. Transient flow and solute transport are simulated to study the leaching of contaminated water into underlying groundwater aquifers. Preparation of input data is detailed for the first problem only. Input data for the example problems described here are available upon request.

##### 4.1 UNDISTURBED SOIL COLUMN TRACER INJECTIONS

Two undisturbed soil columns were obtained from the Melton Branch watershed at ORNL, on which tracer injections under partially saturated conditions were conducted (Jardine et al. 1993). Bromide (Br<sup>-</sup>) solution was injected when the pressure head in the soil column was maintained at -10 cm and the flow condition had reached steady-state. The soils were composed of fractured saprolite and weathered, interbedded shale and limestone. Previous field studies at the watershed (Wilson and Luxmoore 1988) suggested that three types of porosities might be responsible for the flow dynamics observed at the site. A macropore-mesopore-micropore conceptual pore-domain model was proposed by these authors.

Two Br<sup>-</sup> tracer injections were conducted in the laboratory, one on each of the soil columns. A breakthrough curve was fitted to the effluent concentrations of one soil column (soil column 1) and model parameters were obtained. These model parameters were then used to predict the effluent concentrations of the other soil column (soil column 2). The three-pore-region conceptual model suggested by Wilson and Luxmoore (1988) was used for both sets of computations. Soil column dimensions, material properties, and hydrological data are listed in Table 4.1.

A one-region model (Parker and van Genuchten 1984) was fitted to the breakthrough concentration (BTC) of soil column 1 and a one-region dispersion coefficient was obtained (Table 4.1). The one-region dispersion coefficient was used as an initial guess for the three-region curve-fitting simulations. The simulations were stopped when the agreement between laboratory and simulation results was visually satisfactory. No optimization of the model parameters was attempted. Six input data files (see Appendix C) were needed to represent a computer model using MURT. For this application, array

dimensions and I/O control parameters were provided through templates, as shown in Figs. 4.1 and 4.2, respectively. A detailed guide to preparing the other four input data files follows.

**Table 4.1 Dimensions, material properties, and hydrological parameters of the soil columns**

Parameters	Soil column 1	Soil column 2
Length, cm	40	40
Water content	0.419	0.415
Bulk density, g/cm <sup>3</sup>	1.51	1.56
Pore water velocity, cm/h	0.174	0.466
Specific discharge, cm/h	0.0729	0.193
Pulse duration, pore volume	0.118	0.219
One-region dispersion coefficient, cm <sup>2</sup> /h	1.084	3.517

* These data specify array dimensions. Please follow the template.	
number of nodal points . . . . .	82
number of elements . . . . .	40
number of degrees of freedom . . . . .	3
number of time increments . . . . .	40
no. of times to reset time step size . . . . .	2
number of different materials . . . . .	1
number of material properties . . . . .	12
number of correction materials . . . . .	0
no. of source element . . . . .	0
no. of source profiles . . . . .	0
no. of data points for each source prof . . . . .	0
no. of well source/sink nodes . . . . .	0
no. of well source/sink profiles . . . . .	0
no. of data points in well profile . . . . .	0
no. of dirichlet nodal points . . . . .	2
no. of dirichlet head profiles . . . . .	2
no. of data points for dirichlet prof . . . . .	4
no. of variable element sides . . . . .	1
no. of variable boundary nodes . . . . .	2
no. of variable boundary profiles . . . . .	1
no. of data points for each VB profile . . . . .	2
no. of cauchy element sides . . . . .	0
no. of cauchy nodal points . . . . .	0
no. of cauchy flux profiles . . . . .	0
no. of data point for cauchy profiles . . . . .	0
no. of neumann element sides . . . . .	0
no. of neumann nodal points . . . . .	0
no. of neumann flux profiles . . . . .	0
no. of data points for neumann profile . . . . .	0

**Fig. 4.1. Template specifying array dimensions for the undisturbed soil column application.**

\* This is a template for io control variables IGEOM, KSTR, and IMOD. Please  
\* follow the line and change the appropriate data only as necessary.

1. Geometry data input control:

- 0 = geometry data read through the formatted file given by the user;
- 1 = geometry data already in an unformatted file, but boundary geometry and pointer arrays will be computed by the code;
- 2 = geometry, boundary, pointer arrays are in an unformatted file.

IGEOM = 0

2. Geometry data print out control:

- negative integers = NO
- positive integers = YES

IMOD = -1

3. Job restart control:

- negative integers = NO
- positive integers = YES

KSTR = -1

**Fig. 4.2. Template specifying I/O control parameters for the undisturbed soil column application.**

#### 4.1.1 Problem Titles and Basic Control Parameters

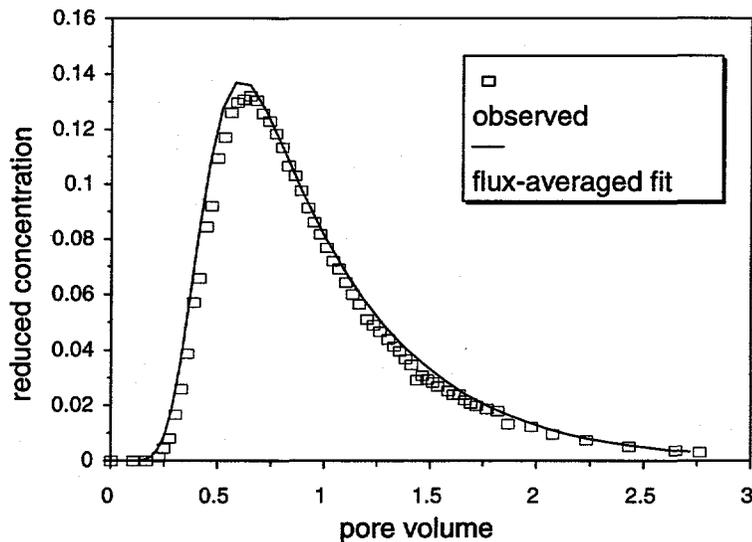
Problem titles and basic control parameters are entered through the file that contains material properties and modeling control and option parameters. The first data set in this file is a problem identification card that consists of a problem number (must be greater than 0) and a single line that describe the application (Fig. 4.3).

The second and third data sets define universal control parameters and control parameters for simulations and output to disk files, respectively. For this application, debugging information was printed (IBUG = 1), mass matrix was lumped (ILUMP = 1), Lagrangian-Eulerian method was used, and upstream weighting parameters were ignored. Forty time steps (NTI = 40) were needed to simulate the tracer injection. Because the tracer was nonreactive, the problem was linear and only one iteration of the nonlinear loop was required. The tracer was injected as a single pulse, and a time-step size reset was specified at 27.126, at which injection of Br<sup>-</sup> solution was terminated and ionized water was switched in. The initial time-step size was 5 h and was increased by 10% for each subsequent time step. Maximum time step size allowed was 22 h. Detailed printouts of concentration and mass flux were required every 10 time steps (Fig. 4.3).



**Table 4.2 Fitted model parameters for soil column 1**

Parameters	Pore region 1	Pore region 2	Pore region 3
Dispersivity, cm	4.068	4.227	6.192
Diffusive mass transfer coefficient decay rate, h <sup>-1</sup>	0.475	0.475	0.475
Diffusive mass transfer coefficient, from region 1, h <sup>-1</sup>	0.0	0.5	1.5
Diffusive mass transfer coefficient, from region 2, h <sup>-1</sup>	0.5	0.0	1.5
Diffusive mass transfer coefficient, from region 3, h <sup>-1</sup>	1.5	1.5	0.0
Specific discharge, cm/h	0.0034	0.0438	0.02571
water content	0.025	0.175	0.219



**Fig. 4.4. Fitted BTC and experimental data for the Br<sup>-</sup> injection on soil column 1.**

#### 4.1.3 Problem Domain, Initial Condition, and Boundary Conditions

The soil column was treated as a vertical one-dimensional flow field with 40 elements and 82 nodes, with each element equally sized. The automatic generation mechanism provided by MURT was used to generate nodal coordinates and element connectivity (Fig. 4.5). The soil column was initially free of Br<sup>-</sup> and water and tracer were injected at the bottom of the column at time 0 (Fig. 4.6). The pulse of injection lasted for 27.13 h and was simulated by a Dirichlet boundary with a fixed reduced concentration 1.0 at the bottom of the column (Fig. 4.6). The upper boundary was set up as a variable boundary



#### 4.1.4 Hydrological Input Variables

The flow condition of the application was assumed steady-state and fluid velocity was provided in an ASCII data file with KVI set to -1 (Fig. 4.7). To maintain mass balance, the total water content and specific discharge of the pore regions must equal to the observed water content and specific discharge of the entire soil column. Because the injection was conducted at -10 cm pressure head, most of the water content and specific discharge were attributed to pore regions 2 and 3. These two parameters for the fitted BTC are listed in Table 4.2.

```
-1
C ***** DATA SET 3: VELOCITY
1 81 1 3.4D-3 0.0D0 0.0D0 0.0D0
0 0 0 0.0 0.0 0.0 0.0      END OF VELOCITY, region 1
1 81 1 0.043800 0.0D0 0.0D0 0.0D0
0 0 0 0.0 0.0 0.0 0.0      END OF VELOCITY, region 2
1 81 1 0.025706 0.0D0 0.0D0 0.0D0
0 0 0 0.0 0.0 0.0 0.0      END OF VELOCITY, region 3
C ***** DATA SET 4: WATER CONTENT
1 39 1 0.025D0 0.0
0 0 0 0.0 0.0      END OF TH, region 1
1 39 1 0.175D0 0.0
0 0 0 0.0 0.0      END OF TH, region 2
1 39 1 0.219D0 0.0
0 0 0 0.0 0.0      END OF TH, region 3
```

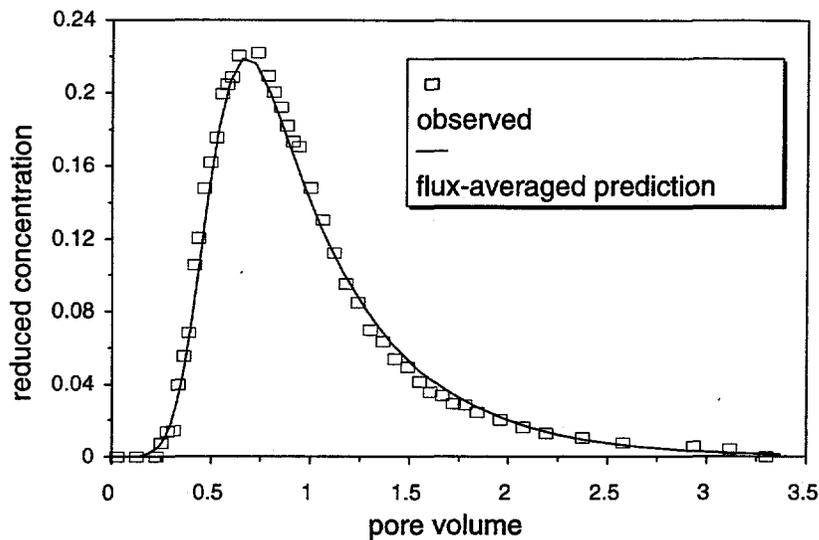
**Fig. 4.7. I/O file specifying hydraulic parameter for the undisturbed soil column application.**

#### 4.1.5 Predicting BTC for Br- Injection On Soil Column 2

Because of similarity in soil structures and hydrological conditions during Br- injections on soil columns 1 and 2, Br- BTC from soil column 2 can be predicted, theoretically, using the parameters obtained from soil column 1. The water content and specific discharge of soil column 2 were slightly different from those of soil column 1. They were distributed to the three pore regions according to the proportion obtained for soil column 1. Other model parameters were maintained the same as those used for soil column 1. The prediction showed a very good agreement with the experimental data (Fig. 4.8).

## 4.2 SENSITIVITY ANALYSIS OF ADVECTIVE AND DIFFUSIVE MASS TRANSFER

Field studies have suggested that diffusive mass transfer, or matrix diffusion, in fractured porous media is a very important solute transport process (Solomon et al. 1992). This



**Fig. 4.8. Predicted and measured BTC of Br<sup>-</sup> injection on soil column 2.**

local-scale nonequilibrium process may cause the retention of wastes in high porosity rocks, which may later become the secondary sources of groundwater contamination. To understand the effects of advective and diffusive mass transfer, we conducted a sensitivity analysis on the two mass transfer processes. A multiregion subsurface fluid flow model, MURF (Gwo et al. 1994b), was used to compute the pore-region flow fields, which were then used as input to the multiregion solute transport model MURT. The following section describes the model configuration for this study, including water retention relations, relative hydraulic conductivities, and advective and diffusive mass transfer coefficients. Input data files for this application are available upon request.

#### 4.2.1 Numerical Model Configurations

The conceptual model for this study was a one-dimensional vertical soil column that consisted of three pore regions, namely, macropores, mesopores, and micropores. The soil was homogeneous, and the flow was in the vertical, or  $z$ , direction only. The height of the soil column was 200 cm, with 40 elements and 82 nodes. The bottom of the soil column was maintained at 0 cm pressure head to establish variably saturated transient flow fields in the pore regions, and water with relative solute concentration of 1.0 was injected at the top at a rate of  $9 \text{ cm}^3 \text{ h}^{-1} \text{ m}^{-2}$  for 2 h. Solute-free infiltration water at the same rate was then switched in and maintained for another 3 h before the infiltration was terminated. No water was injected after time = 5 h, and all of the simulations were stopped at time = 10 h. The initial pressure head in the soil column was -97 cm. The solute was nonreactive, and the initial solute concentration in the fluid phase was zero.

Solute transport parameters for the study were determined from miscible displacement experiments under partially saturated conditions conducted on undisturbed soil columns (Jardine et al. 1993). In the experiments, negative pressure heads, or tensions, were maintained at the ends of the soil column to establish steady-state fluid flow. Tracers were then injected at the top of the soil column. Breakthrough data from these experiments were used to calibrate a multiregion solute transport model and to obtain model parameters (Gwo et al. 1995). These parameters were used in this study and are listed in Table 4.3.

**Table 4.3 Pore-region hydraulic and solute transport parameters for sensitivity analyses**

Properties	Macropores	Mesopores	Micropores
Saturated water content, $\theta_{sa}$ , (cm <sup>3</sup> /cm <sup>3</sup> )	0.39	-	-
Residual water content, $\theta_{sr}$ , (cm <sup>3</sup> /cm <sup>3</sup> )	0.35	-	-
$\gamma_a$ , cm <sup>-1</sup>	0.47	-	-
$h_{\theta a}$ , cm	-5	-	-
Saturated water content, $\theta_{s\alpha}$	-	0.35	0.31
Residual water content, $\theta_{r\alpha}$	-	0.31	0.0
$\beta_\alpha$ , cm <sup>-1</sup>	-	0.2212	0.0028
$n_\alpha$	-	1.1456	1.379
Saturated hydraulic conductivity, $K_{s\alpha}$ , cm h <sup>-1</sup>	8.28	0.72	0.000234
$\xi_\alpha$ , m <sup>-1</sup>	8.5	7.32	1000
$\kappa_\alpha$ , cm <sup>-1</sup>	0.65	0.03	0.00075
$h_{\kappa\alpha}$ , cm	-15	-250	-15000
Longitudinal dispersivity, $a_{L\alpha}$ , cm	8.0046	12.007	26.682

Because a pore region consists of an associated flow field, conductivity and water content relations  $K_\alpha(h_\alpha)$  and  $\theta_\alpha(h_\alpha)$ , respectively, are needed for each pore region to calculate the velocity fields. For this application, we used the following Fermi function (Wilson et al. 1992) to represent the  $\theta_\alpha(h_\alpha)$  relation of the macropore region:

$$\theta_a = \frac{\theta_{sa} - \theta_{sr}}{1 + \exp[-\gamma_a(h_a - h_{\theta a})]}, \quad (4.2)$$

where  $\theta_a$  is the water content in macropores,  $\theta_{sa}$  and  $\theta_{sr}$  are the saturated and residual water contents of the macropores,  $\gamma_a$  and  $h_{\theta a}$  are the function parameters, and  $h_a$  is the pressure head in the macropores. For mesopores and micropores, the following equation was used for the  $\theta_\alpha(h_\alpha)$  relation (van Genuchten 1980):

$$\theta_{\alpha} = \frac{\theta_{s\alpha} - \theta_{r\alpha}}{\left[1 + (-\beta_{\alpha} h_{\alpha})^{n_{\alpha}}\right]^{m_{\alpha}}}, \quad (4.3)$$

where  $\theta_{s\alpha}$  and  $\theta_{r\alpha}$  are the saturated and residual water contents in region  $\alpha$ , respectively, and  $\beta_{\alpha}$ ,  $n_{\alpha}$ , and  $m_{\alpha} = 1 - 1/n_{\alpha}$  are equation parameters. To account for the dramatic decrease in hydraulic conductivities in the pore regions, especially the macropores, near the air entry point, we used the following Fermi function (Ward et al. 1987) to represent the  $K_{\alpha}(h_{\alpha})$  relation for the pore regions:

$$\log_{10}(K_{a\alpha} / K_{s\alpha}) = \frac{\xi_{\alpha}}{1 + \exp[-\kappa_{\alpha}(h_{\alpha} - h_{\kappa\alpha})]} - \xi_{\alpha}, \quad (4.4)$$

where  $K_{a\alpha}$  and  $K_{s\alpha}$  are the absolute and saturated hydraulic conductivities of region  $\alpha$  and  $\xi_{\alpha}$ ,  $\kappa_{\alpha}$ , and  $h_{\kappa\alpha}$  are equation parameters. Water retention and hydraulic conductivity data obtained from field experiments at the Melton Branch watershed and laboratory experiments using soils collected on the watershed were fitted by the Fermi and van Genuchten equations. The equation parameters thus obtained are listed in Table 4.3. Model parameters not listed in Table 4.3 were all set to zero.

Fracture spacing on the Oak Ridge Reservation ranges between 0.51 and 158.75 cm (Hatcher et al. 1992). Using Eqs. (2.22) and (2.18b) and the fracture spacing as the dimensions of the matrix block, one can calculate the ranges of advective and diffusive mass transfer coefficients. For the advective mass transfer between macropores and mesopores,  $K_{js} = 0.72 \text{ cm h}^{-1}$ ,  $n_{\alpha} = 0.04$ , and  $c = l = 0.26$  and 79.38 cm for minimum and maximum fracture spacing, respectively, we have the maximum advective mass transfer coefficient  $\epsilon_{\alpha j}^f = 0.54 \text{ cm}^{-1} \text{ h}^{-1}$ , and the minimum advective mass transfer coefficient  $\epsilon_{\alpha j}^f = 5.82 \times 10^{-6} \text{ cm}^{-1} \text{ h}^{-1}$ . For the advective mass transfer between macropores and micropores and that between mesopores and micropores,  $K_{js} = 2.34 \times 10^{-4} \text{ cm h}^{-1}$ ,  $n_{\alpha} = 0.04$ , and  $c = l = 0.26$  and 79.38 cm for minimum and maximum fracture spacing, respectively, we have the maximum advective mass transfer coefficient  $\epsilon_{\alpha j}^f = 1.76 \times 10^{-4} \text{ cm}^{-1} \text{ h}^{-1}$ , and the minimum advective mass transfer coefficient  $\epsilon_{\alpha j}^f = 2.0 \times 10^{-9} \text{ cm}^{-1} \text{ h}^{-1}$ . Therefore, the advective mass transfer coefficients between macropores and mesopores were varied by 5 orders of magnitude in the simulations, from  $1.0 \times 10^{-4}$  to  $1.0 \text{ cm}^{-1} \text{ hr}^{-1}$ . The advective mass transfer coefficients between macropores and micropores and those between mesopores and micropores were maintained 4 orders of magnitude lower than those between macropores and mesopores. The diffusive mass transfer coefficients can be similarly obtained using Eq. (2.18b) and a molecular diffusion coefficient of  $3.6 \times 10^{-2} \text{ cm}^2 \text{ h}^{-1}$ . The maximum and minimum diffusive mass transfer coefficients between macropores and mesopores were 0.11 and  $1.0 \times 10^{-6} \text{ h}^{-1}$ , respectively. The maximum and minimum diffusive mass transfer coefficients between macropores and micropores and

those between mesopores and micropores were  $0.85$  and  $1.0 \times 10^{-5} \text{ h}^{-1}$ , respectively. For the sensitivity analysis, all of the diffusive mass transfer coefficients are systematically varied between  $1.0$  and  $1.0 \times 10^{-6} \text{ h}^{-1}$ . The maximum and minimum mass transfer coefficients between pore regions are listed in Table 4.4.

**Table 4.4 Minimum and maximum mass transfer coefficients between pore regions.**

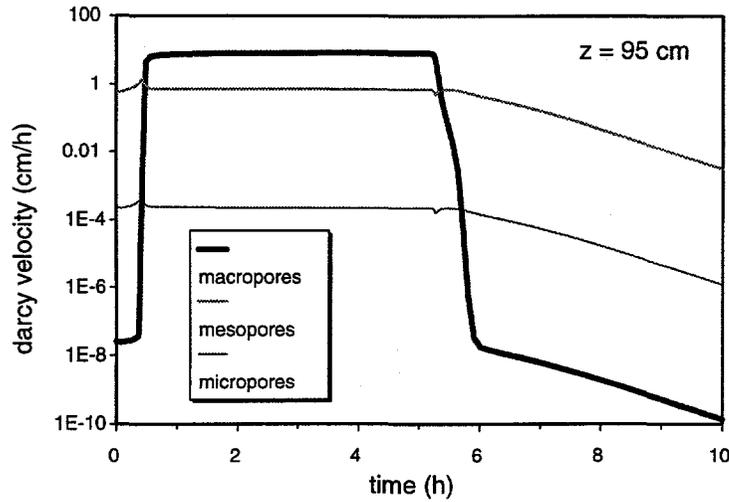
	advective mass transfer		diffusive mass transfer
	macropores- mesopores ( $\text{cm}^{-1} \text{ h}^{-1}$ )	macropores-micropores, mesopores-micropores ( $\text{cm}^{-1} \text{ h}^{-1}$ )	macropores-mesopores mesopores-micropores macropores-micropores ( $\text{h}^{-1}$ )
minimum	$1.0 \times 10^{-4}$	$1.0 \times 10^{-8}$	$1.0 \times 10^{-6}$
maximum	$1.0$	$1.0 \times 10^{-4}$	$1.0$

#### 4.2.2 Effects of Advective Mass Transfer on Fluid Flow

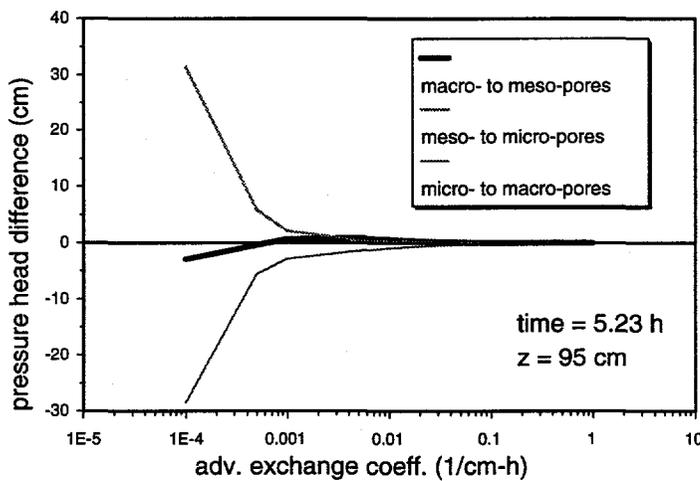
Subsurface flow systems usually go through perturbation and equilibration cycles in response to change of boundary conditions. In multiregion media, the change is perceived not only as perturbation to the system as a whole but also as physical nonequilibrium among pore regions. In search of another equilibrium state, fluid is moved by interregion pressure gradients from one region to another until pore-region pressure heads are balanced.

Balanced pressure fields, however, do not imply balanced velocity fields. Under equilibrium state, intraregion pressure gradients are nearly the same in all of the pore regions, but pore-region velocities may vary proportionately according to their respective hydraulic conductivities. For example, at  $z = 95 \text{ cm}$ , during the periods 2 to 5 h and 6 to 10 h, the pressure fields were balanced (not shown), but the velocity fields were not (Fig. 4.9). Interference of velocity fields was manifested through the peaks and valleys in the velocity curves of mesopores and micropores. Increase of fluid velocity and pressure in the macropores resulted in recharge to the other two pore regions and consequently the rises of their local pressure gradients and velocities. On the other hand, at the moment the velocity and pressure in the macropore region decreased dramatically, local pressure gradients were reduced in the other two pore regions and two small valleys were created on their velocity curves. Recharge from mesopores and micropores resulted in two deflection points on the macropore velocity curve.

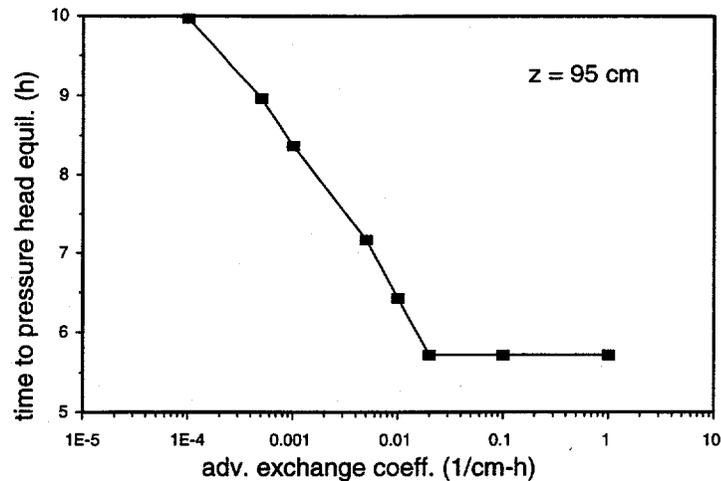
Higher advective mass transfer coefficients imply less time to equilibrium. For instance, at time = 5.23 h and  $z = 95 \text{ cm}$ , the pressure head difference among pore regions is smaller at higher advective mass transfer coefficients (Fig. 4.10). The times needed for pressure fields to reach equilibrium decrease as mass transfer coefficients increase (Fig. 4.11).



**Fig. 4.9. Velocity variations at  $z = 95$  cm.** The interference of velocity fields was manifested through the deflection of the macropore velocity curve and the peaks and valleys in the velocity curves of micropores and mesopores. The advective mass transfer coefficient was  $0.1 \text{ cm}^{-1} \text{ h}^{-1}$  between macropores and mesopores, and  $1 \times 10^{-5} \text{ cm}^{-1} \text{ h}^{-1}$  between micropores and the other two pore regions.



**Fig. 4.10. Effect of advective mass transfer on pressure head difference between pore regions at time = 5.23 h and  $z = 95$  cm.** Higher mass transfer coefficients resulted in less difference in pressure heads.

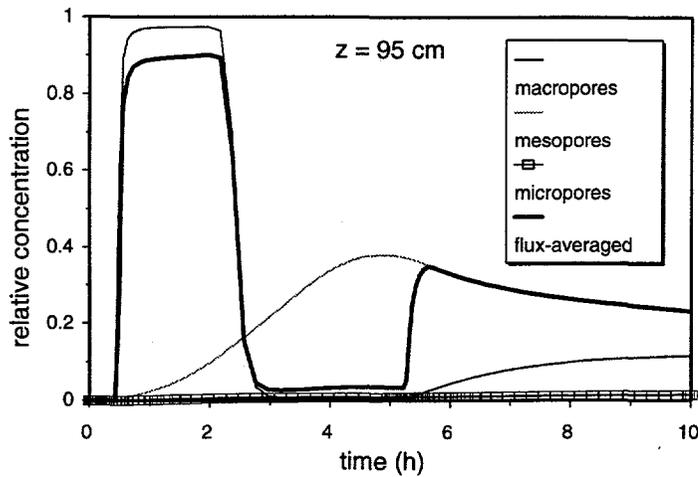


**Fig. 4.11.** Times needed for pressure head difference between pore regions to become less than 1 cm, 0.5% of the length of the soil column, at  $z = 95$  cm as function of advective mass transfer coefficients.

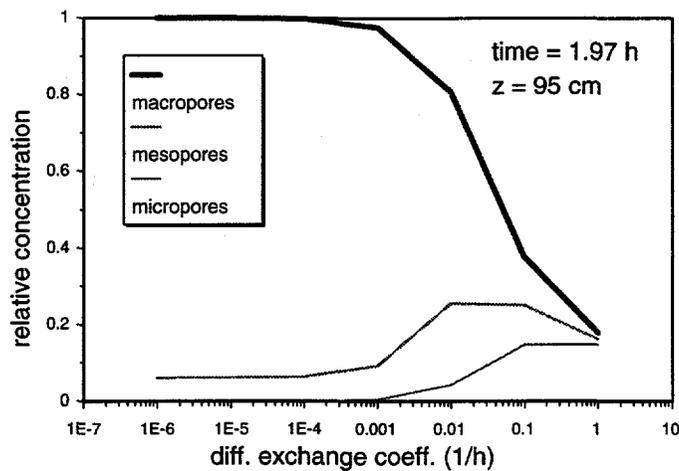
#### 4.2.3 Effects of Advective-Diffusive Mass Transfer on Solute Transport

A unique feature of multiple-pore-region media is secondary solute sources. While solute transport may be initiated for all pore regions at the same moment near an external source, the fronts of concentration fields may move at different rates. Consequently, local nonequilibrium results and interregion mass transfer may continually move solutes between pore regions until the concentration fields are balanced. After external sources are terminated, solutes may be moved from mesopores to macropores in upstream areas, while in downstream areas they may be moved in the opposite direction. Solutes in mesopores, therefore, become a secondary source to macropores, which may explain the decrease and then increase of effluent solute concentrations after the termination of external solute sources (Fig. 4.12). The first breakthrough of solute is mostly contributed by macropores; the second one is mostly by mesopores.

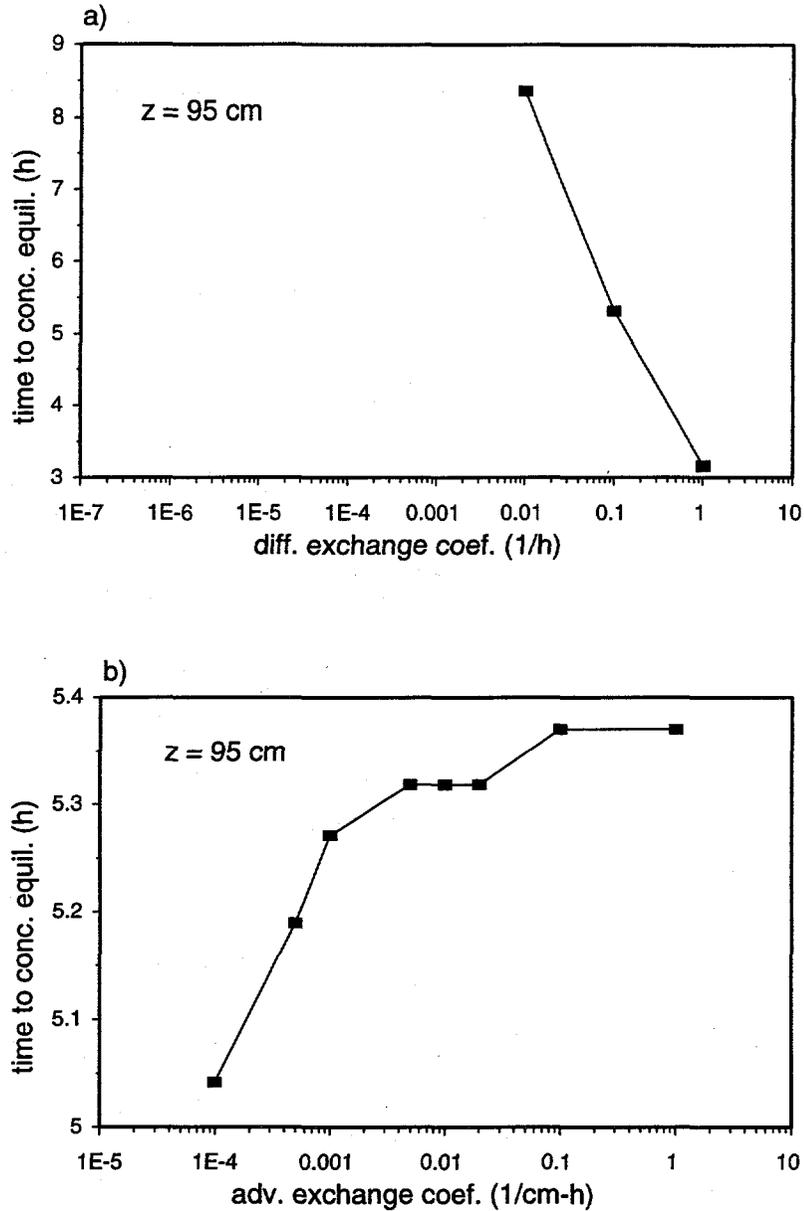
Similar to the effect of advective mass transfer on pressure head distributions, high diffusive mass transfer accelerates the equilibration of pore-region solute concentrations. For instance, at  $z = 95$  cm and time = 1.97 h (Fig. 4.13), the solute concentrations in the pore regions are not balanced until the diffusive mass transfer coefficient reaches  $1.0 \text{ h}^{-1}$ . The times needed for concentration differences to become less than 1% of the source concentration have a relation with diffusive mass transfer coefficients similar to that shown in Fig. 4.11; however, no equilibrium of pore-region concentrations have been observed within 10 h for diffusive mass transfer coefficients less than  $0.01 \text{ h}^{-1}$  (Fig. 4.14a). Times required to reduce the concentration difference between macropores and mesopores to 1% of the source concentration are positively related to advective mass transfer coefficients (Fig. 4.14b); that is, more time is needed to equilibrate pore-region



**Fig. 4.12.** Solute concentration variations in the pore regions at  $z = 95$  cm. The diffusive mass transfer coefficient is  $1 \times 10^{-3} \text{ h}^{-1}$ ; the advective mass transfer coefficient is  $0.01 \text{ cm}^{-1} \text{ h}^{-1}$  between macropores and mesopores,  $1 \times 10^{-6} \text{ cm}^{-1} \text{ h}^{-1}$  between micropores and the other two pore regions.

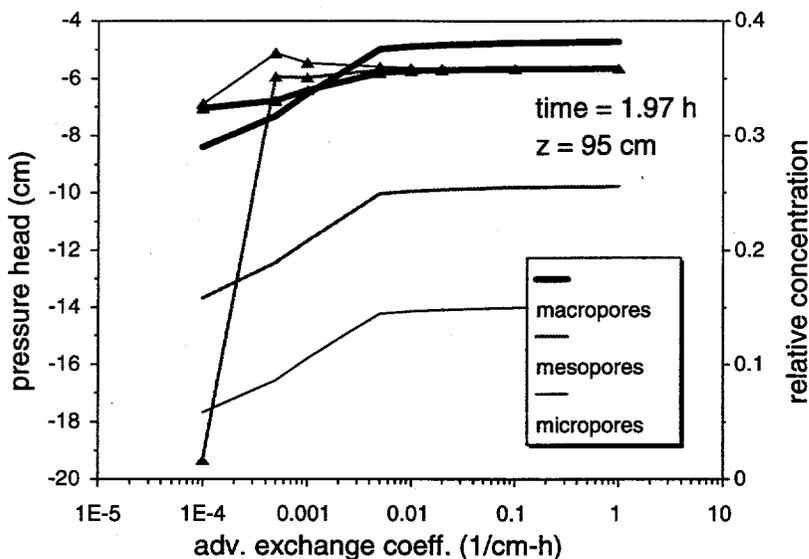


**Fig. 4.13.** Effect of diffusive mass transfer on solute concentrations at  $z = 95$  cm and time = 1.97 h. Pore-region concentrations are not balanced until the diffusive mass transfer coefficient reaches  $1.0 \text{ h}^{-1}$ .



**Fig. 4.14. Times needed for pore-region concentration to become less than 1% of the source concentration at  $z = 95$  cm as related to (a) diffusive mass transfer coefficients at  $\epsilon_{\text{oj}}^f = 0.01 \text{ cm}^{-1} \text{ h}^{-1}$  and (b) advective mass transfer coefficients at  $\epsilon_{\text{oj}}^f = 0.1 \text{ h}^{-1}$ . No solute concentration equilibrium has been observed within 10 h for diffusive mass transfer coefficients less than  $0.01 \text{ h}^{-1}$ .**

concentration fields with larger advective mass transfer coefficients. It appears that advective mass transfer may dilute the concentration in the down-pressure-gradient pore region, which happens to be the up-concentration-gradient pore region. This effect is better illustrated through the comparison of pore-region pressure heads and concentrations by fixing the diffusive mass transfer coefficient at  $0.1 \text{ h}^{-1}$  and varying advective mass transfer from  $1 \times 10^{-4}$  to  $1.0 \text{ cm}^{-1} \text{ h}^{-1}$  (Fig. 4.15). At advective mass transfer coefficient  $0.0005 \text{ cm}^{-1} \text{ h}^{-1}$ , the concentration gradient moves solute from macropores to mesopores, while pressure gradient moves low-concentration water in mesopores into high-concentration macropores (Fig. 4.15). The latter process dilutes the water in macropores because it increases the water content and decreases the solute concentration in the macropore region. And, for cases where high-concentration water is moved into low-concentration pore region by advective mass transfer, the interregion concentration gradient is also reduced. In general, therefore, advective mass transfer tends to smooth the interregion concentration gradient, which happens to be the driving force of diffusive mass transfer.



**Fig. 4.15.** Effect of advective mass transfer on pressure heads and solute concentrations at  $z = 95 \text{ cm}$  and  $\text{time} = 1.97 \text{ h}$ . Lines with markers are pore-region pressure heads, and those without are pore-region concentration. At advective mass transfer coefficient  $0.0005 \text{ cm}^{-1} \text{ h}^{-1}$ , water is moving from small pores to large pores while solute is being diffused from large pores to small pores. The movements of solute result from advective and diffusive mass transfer are in opposite directions.

### 4.3 HILLSLOPE PONDING INFILTRATION AND SECONDARY SOURCE

ORNL is located in a high precipitation area with structured soils and geological formations. Waste trenches are usually located in the stormflow and vadose zones that experience seasonal dry and wet cycles. During storms, a waste trench can be quickly saturated with rainwater, and leaching and transport of waste to local groundwater and surface water results. The saturated waste trenches result in effects such as bathtubting (Solomon and Yeh, 1987), where rainwater ponds in the trenches and becomes a continuous source of groundwater contamination. In this study, three pore regions were considered, in contrast to the one-region study of bathtubting trenches by Solomon and Yeh (1987).

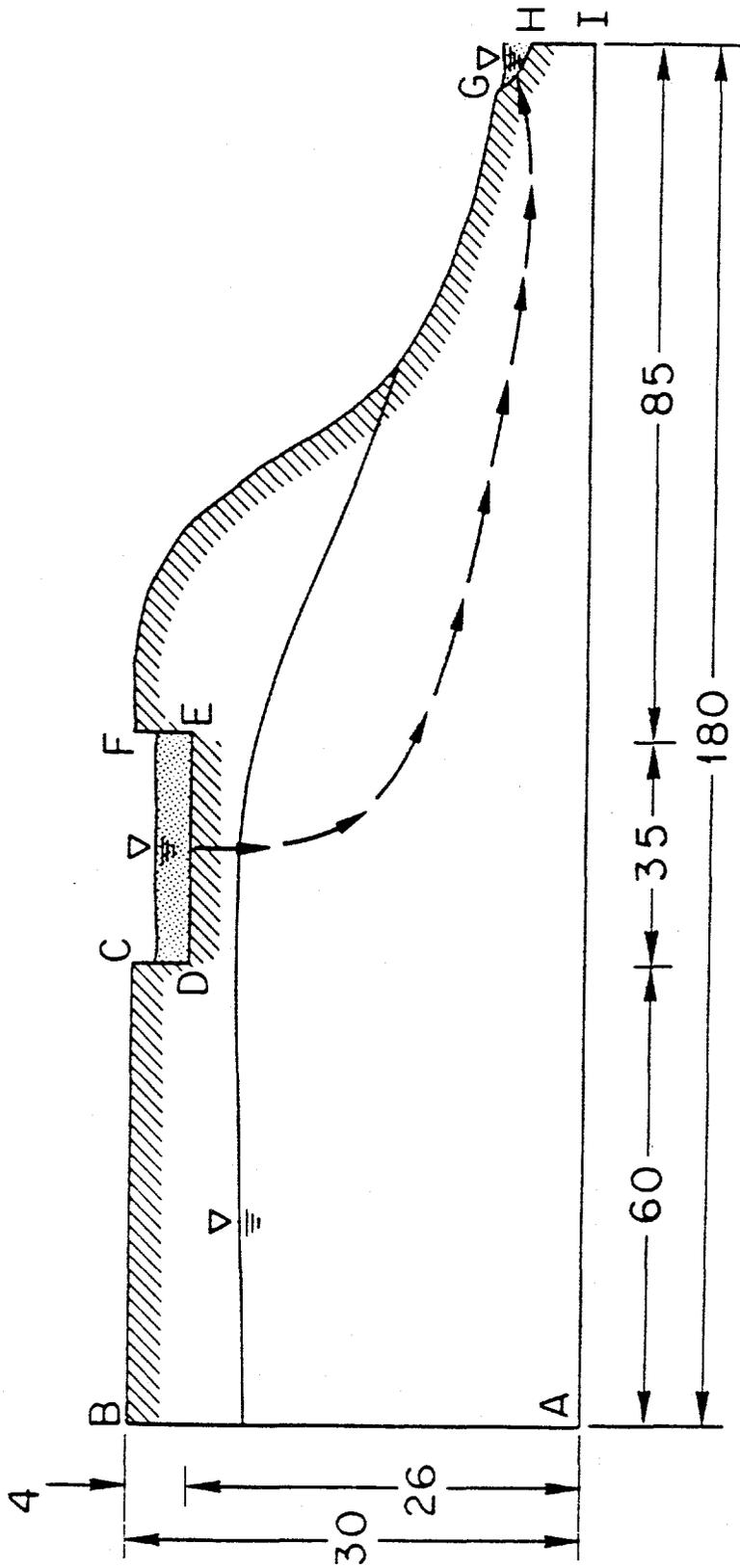
#### 4.3.1 Numerical Model Configurations

The problem domain was 180 m wide and 30 m high, with a seepage pond (representing a bathtubting waste trench) located on the top of the hill (Fig. 4.16). The cross section was discretized into 528 quadrilateral elements that consisted of 595 nodes (Fig. 4.17). A three-pore-region conceptual model and parameters described earlier were used again here (Wilson and Luxmoore 1988; Wilson et al. 1992). Each of the elements and nodes in the problem domain, therefore, consisted of three degrees of freedom. For the details of preparing input data files for MURF to obtain pore-region transient flow fields, readers are referred to Gwo et al (1994b). The boundary and initial conditions and the soil property functions, including water retention curves and relative hydraulic conductivity relations, are also detailed in Gwo et al. (1994b) and are not repeated here.

To simulate a transient solute transport problem, we assumed that the solute was nonreactive and the soil in the hillslope was characteristically similar to those used in Sect. 4.1. The fitted parameters in Sect. 4.1, except for the diffusive mass transfer coefficients, were used for the current application (Table 4.2). Advective and diffusive mass transfer coefficients for the current application are listed in Table 4.5, which was determined using values reported in Table 4.4. The darcy velocities calculated by MURF, which was written to FORTRAN unformatted files, are read directly by MURF. There is no solute in the soil initially. At the bottom of the pond, a tracer of unit relative concentration is applied for 16 h and is removed thereafter.

#### 4.3.2 Results and Discussion

Fast breakthrough of nonreactive solutes is believed to be one of the characteristics of macroporous media. However, once the primary source is removed, the secondary source may remain in place for a much longer period of time (Wickliff et al. 1991). The slow movement of the secondary source can be explained by the current application, which uses the geometrical mean of diffusive and advective mass transfer coefficients at ORNL (Gwo et al. 1994b). Figure 4.18 shows the pore-region concentrations along a cross section parallel to the direction of flow at 61.6 h, which is 45.6 hours after the primary



DIMENSIONS IN METERS

Fig. 4.16. Problem domain of the hillslope ponding infiltration and secondary source problem.

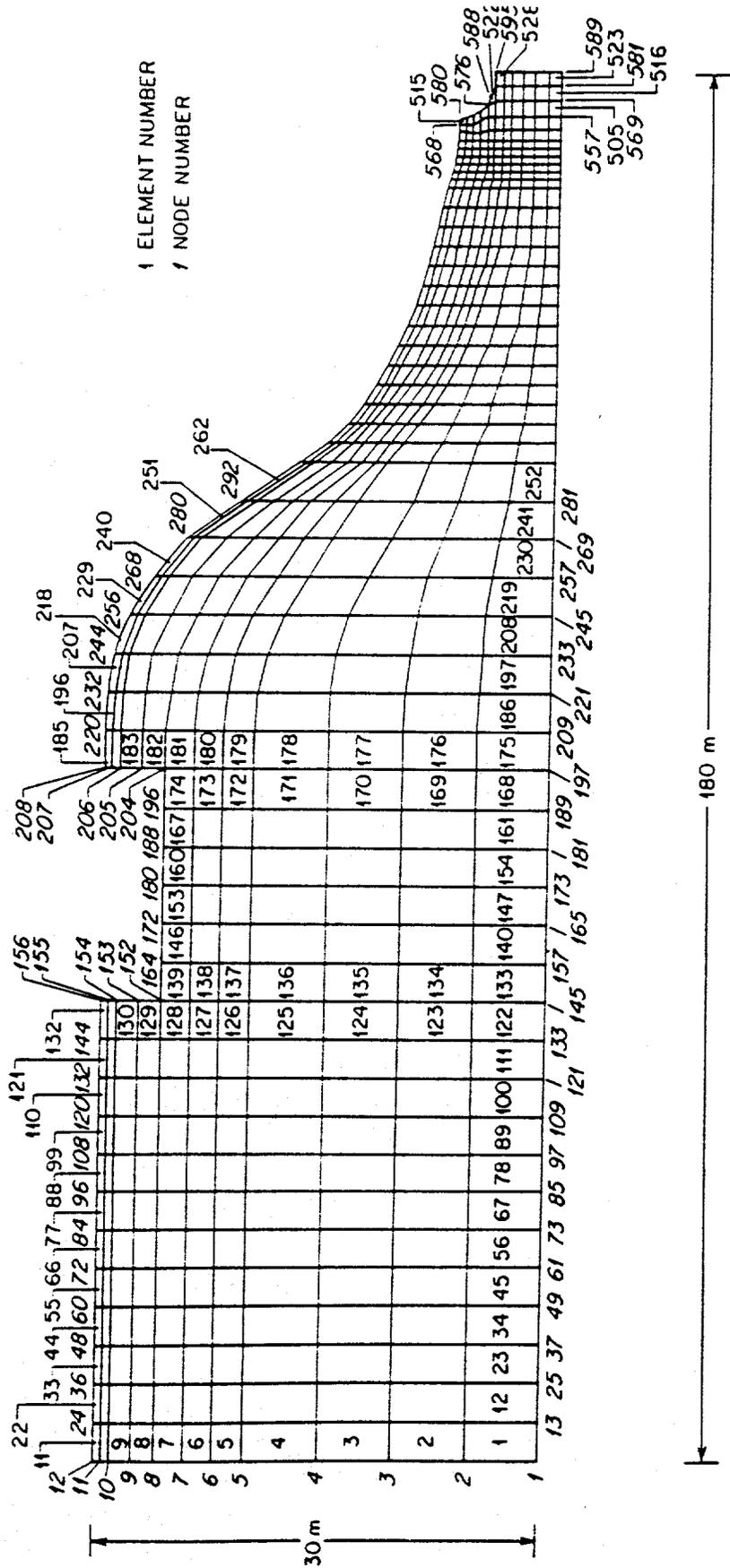
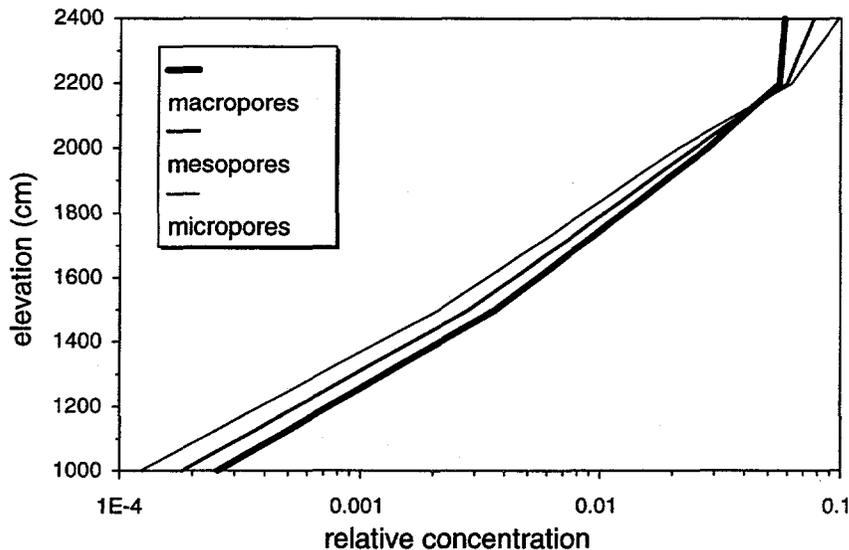


Fig. 4.17. Finite element discretization of the problem domain of the hillslope ponding infiltration and secondary source.

**Table 4.5 Advective and diffusive mass transfer coefficients for the hillslope ponding and secondary source problem**

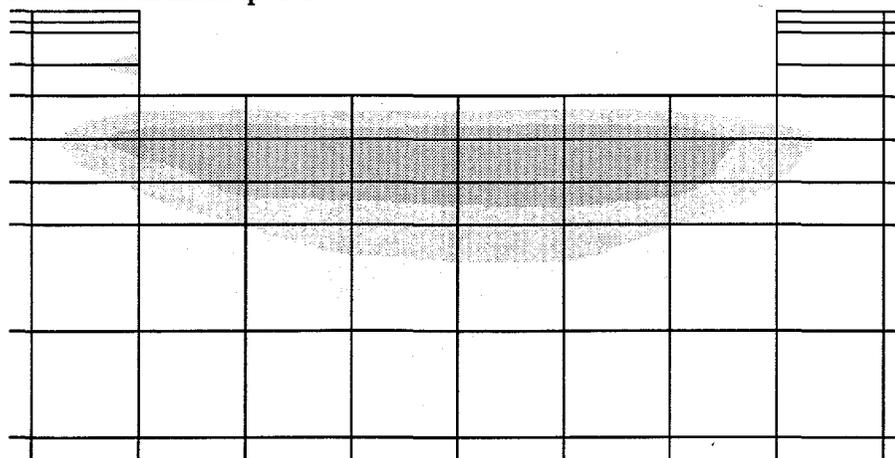
Mass transfer coefficients	Macropores	Mesopores	Micropores
$\epsilon_{1j}^f$ , advective, macropores, $\text{cm}^{-1} \text{h}^{-1}$	-	$1.0 \times 10^{-2}$	$1.0 \times 10^{-5}$
$\epsilon_{2j}^f$ , advective, mesopores, $\text{cm}^{-1} \text{h}^{-1}$	$1.0 \times 10^{-2}$	-	$1.0 \times 10^{-5}$
$\epsilon_{3j}^f$ , advective, micropores, $\text{cm}^{-1} \text{h}^{-1}$	$1.0 \times 10^{-5}$	$1.0 \times 10^{-5}$	-
Diffusive, all pore-regions, $\text{h}^{-1}$	$1.0 \times 10^{-2}$	$1.0 \times 10^{-2}$	$1.0 \times 10^{-2}$

source was removed. In upstream areas (elevation  $\sim 2400$  cm), concentration in micropores is higher than those in macropores and mesopores. By definition, solutes in micropores have become the secondary source at this location. Most of the solutes are stored in the large pore volume in the micropores (Fig. 4.19) and behave as a source to the other two pore regions. However, in downstream areas (elevation  $< 2000$  cm), solute concentrations in the two large pore regions are higher than that in micropores (Fig. 4.18). This concentration difference results in the movement of solutes into the large-storage micropore region in this neighborhood. The spread of solutes in the large pore regions is only slightly higher than that in the micropores (Fig. 4.20). In essence, the spreading of solutes through the fast-flowing pore regions, especially the macropores, is restrained by mass transfer into the micropores.

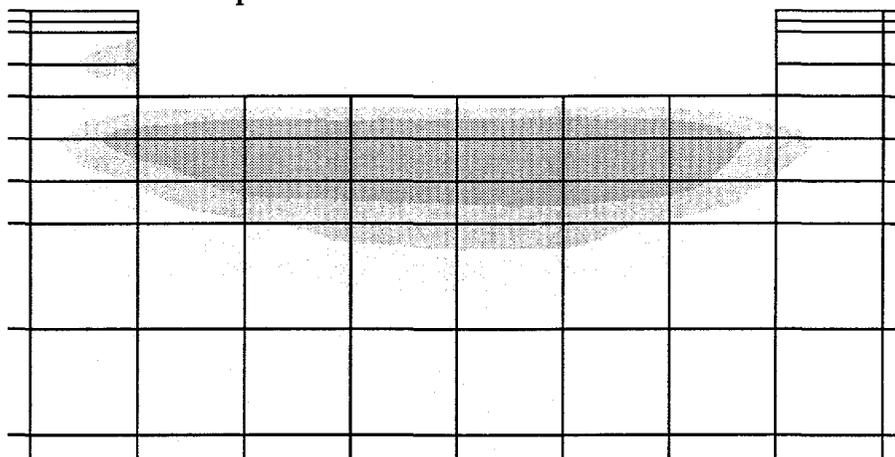


**Fig. 4.18. Pore region solute concentrations along the direction of flow. In upstream areas, solutes are moving from micropores to the other two pore regions, while they are moving in the opposite direction in downstream areas.**

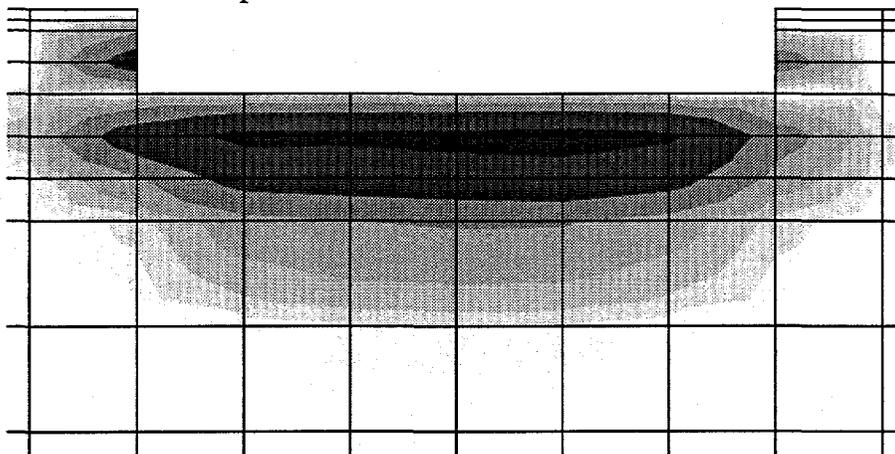
(a) Solute mass in macropores



(b) Solute mass in mesopores

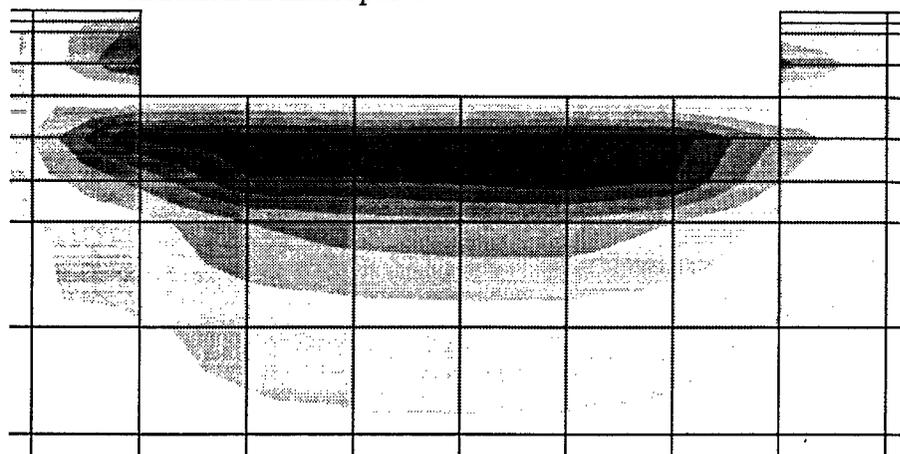


(c) Solute mass in micropores

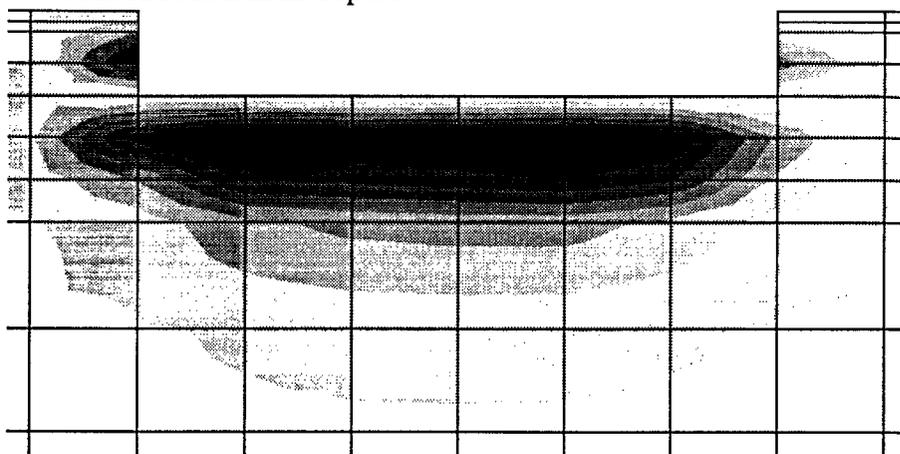


**Fig. 4.19. Solute mass in the three pore regions at 61.6 h: (a) macropores, (b) mesopores, and (c) micropores.**

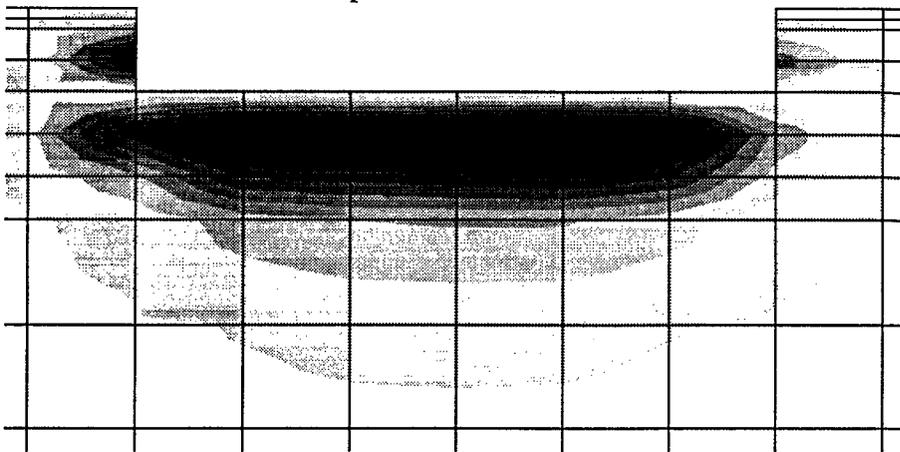
(a) Solute concentration in macropores



(b) Solute concentration in mesopores



(c) Solute concentration in micropores



**Fig. 4.20. Solute concentration in the three pore regions at 61.6 h: (a) macropores, (b) mesopores, and (c) micropores.**

## 5. REFERENCES

- Arbogast, T., J. Douglas, Jr., and U. Hornung. 1989. Derivation of the double porosity model of single phase flow via homogenization theory. *SIAM J. Math. Anal.* 21(4):823-836.
- Bai, M., D. Elsworth, and J.-C. Roegiers. 1993. Multiporosity/multipermeability approach to the simulation of naturally fractured reservoirs. *Water Resour. Res.* 29(6):1621-1633.
- Baptista, A. M. 1987. Solution of advection-dominated transport by Eulerian-Lagrangian methods using the backwards methods of characteristics. Ph.D. dissertation. M.I.T., Cambridge.
- Bibby, R. 1981. Mass transport of solutes in dual-porosity media. *Water Resour. Res.* 17(4):1075-1081.
- Brusseu, M. L., R. E. Jessup, and P. S. C. Rao. 1989. Modeling the transport of solutes influenced by multiprocess nonequilibrium. *Water Resour. Res.* 25(9):1971-1988.
- Coats, K. H. and B. D. Smith. 1964. Dead-end pore volume and dispersion in porous media. *Soc. Petr. Eng. J.* 4:73-84.
- Duff, I. S. 1990. MA28 - A set of FORTRAN subroutines for sparse unsymmetric linear equations. Computer Science and System Division, AERE, Harwell, Oxfordshire, England.
- Duguid, J. O. and J. F. Abel. 1974. Finite element Galerkin method for flow in fractured porous media. pp. 599-615. In J. T. Oden, O. C. Zienkiewicz, R. H. Gallagher, and C. Tayler (eds.), *Finite Element Methods in Flow Problems*. University of Alabama Press, Huntsville.
- Duguid, J. O. and P. C. Y. Lee. 1977. Flow in fractured porous media. *Water Resour. Res.* 13(3):558-566.
- Dykhuizen, R. C. 1990. A new coupling term for dual-porosity models. *Water Resour. Res.* 26(2):351-356.
- Grisak, G. E. and J. F. Pickens. 1980. Solute transport through fractured media. 1. the effect of matrix diffusion. *Water Resour. Res.* 16(4):719-730.
- Gvirtzman, H., N. Paldor, M. Magaritz, and Y. Bachmat. 1988. Mass exchange between mobile freshwater and immobile saline water in the unsaturated zone. *Water Resour. Res.* 27(10):1638-1644.

Gwo, J. P. 1992. Multi-region flow and transport modeling in subsurface media. Ph.D. dissertation. Pennsylvania State University, University Park.

Gwo, J. P., P. M. Jardine, G. V. Wilson, and G. T. Yeh. 1994a. Modeling small-scale physical non-equilibrium and large-scale preferential fluid and solute transport in a structured soil. pp. 465-472. In A. Peters, G. Wittum, B. Herrling, U. Meissner, C. A. Brebbia, W. G. Gray, and G. F. Pinder (eds.), *Computational Methods in Water Resources X*, Vol. 1. Kluwer Academic, Dordrecht.

Gwo, J. P., P. M. Jardine, G. T. Yeh, and G. V. Wilson. 1994b. MURF user's guide: a finite element model of multiple-pore-region flow through variably saturated subsurface media. ORNL/GWPO-011. Oak Ridge National Laboratory, Tenn.

Gwo, J. P., P. M. Jardine, G. V. Wilson, and G. T. Yeh. 1995. A multiple-pore-region concept to modeling mass transfer in subsurface media. *J. Hydrol.* 164:217-237.

Hatcher, Jr., R. D., P. J. Lemiszki, R. B. Dreier, R. H. Ketelle, R. R. Lee, D. A. Leitzke, W. M. McMaster, J. L. Forman, and S. Y. Lee. 1992. Status report on the geology of the Oak Ridge reservation. ORNL/TM-12074. Oak Ridge National Laboratory, Tenn.

Jardine, P. M., G. V. Wilson, and R. J. Luxmoore. 1988. Modeling the transport of inorganic ions through undisturbed soil columns from two contrasting watersheds. *Soil Sci. Soc. Am. J.* 52:1252-1259.

Jardine, P. M., G. V. Wilson, and R. J. Luxmoore. 1990. Unsaturated solute transport through a forested soil during rain storm events. *Geoderma* 46:103-118.

Jardine, P. M., G. K. Jacobs, and G. V. Wilson. 1993. Unsaturated transport processes in undisturbed heterogeneous porous media: I. Inorganic contaminants. *Soil Sci. Soc. Am. J.* 57:945-953.

Jarvis, N. J., P.-E. Jansson, P. E. Dik, and I. Messing. 1991a. Modeling water and solute transport in macroporous soil. I. Model description and sensitivity analysis. *J. Soil Sci.* 42:59-70.

Jarvis, N. J., L. Bergstrom, and P. E. Dik. 1991b. Modelling water and solute transport in macroporous soil. II. Chloride breakthrough under non-steady flow. *J. Soil Sci.* 42:71-81.

Jury, W. A. and H. Fluhler. 1992. Transport of chemicals through soil: mechanisms, models, and field applications. pp. 141-201. In D. L. Sparks (ed.), *Advances in Agronomy*. Harcourt Brace Jovanovich, New York.

Kimura, H. and M. Munakata. 1992. Validation studies of tracer tests in a fracture zone at the Finnsjon research area. *Adv. in Water Resour.* 15:63-74.

- Maloszewski, P. and A. Zuber. 1990. Mathematical modeling of tracer behavior in short-term experiments in fissured rocks. *Water Resour. Res.* 26(7):1517-1528.
- Nguyen, V. V., W. G. Gray, G. F. Pinder, J. F. Botha, and D. A. Crerar. 1982. A theoretical investigation on the transport of chemicals in reactive porous media. *Water Resour. Res.* 18(4):1149-1156.
- Parker, J. C. and M. Th. van Genuchten. 1984. Determining transport parameters from laboratory and field tracer experiments. Virginia Agricultural Experiment Station Bulletin 84-3. Virginia Polytechnic Institute and State University, Blacksburg.
- Rao, P. S. C., R. E. Jessup, D. E. Rolston, J. M. Davidson, and D. P. Kilcrease. 1980. Experimental and mathematical description of nonadsorbed solute transfer by diffusion in spherical aggregates. *Soil Sci. Soc. Am. J.* 44:684-688.
- Rao, P. S. C., R. E. Jessup, and T. M. Addiscott. 1982. Experimental and theoretical aspects of solute diffusion in spherical and nonspherical aggregates. *Soil Sci.* 133(6):342-349.
- Seyfried, M. S. and P. S. C. Rao. 1987. Solute transport in undisturbed columns of an aggregated tropical soil: Preferential flow effects. *Soil Sci. Soc. Am. J.* 51:1434-1444.
- Skopp, J., W. R. Gardner, and E. J. Tyler. 1981. Solute movement in structured soils: Two-region model with small interaction. *Soil Sci. Soc. Am. J.* 45:837-842.
- Skopp, J. and W. R. Gardner. 1992. Miscible displacement: An interacting flow region model. *Soil Sci. Soc. Am. J.* 56:1680-1686.
- Solomon, D. K., G. K. Moore, L. E. Toran, R. B. Dreier, and W. M. McMaster. 1992. Status report: A hydrologic framework for the Oak Ridge reservation. ORNL/TM-12026. Oak Ridge National Laboratory, Tenn.
- Solomon, D. K. and G. T. Yeh. 1987. Application of 3DFEMWATER to the study of trench "bathtubbing". ORNL/RAP/LTR-87/89. Oak Ridge National Laboratory, Tenn.
- Tang, D. H., E. O. Frind, and E. A. Sudicky. 1981. Contaminant transport in fractured porous media: analytical solution for a single fracture. *Water Resour. Res.* 17(3):555-564.
- van Genuchten, M. Th. and P. J. Wierenga. 1976. Mass transfer studies in sorbing porous media. I. Analytical solutions. *Soil Sci. Soc. Am. J.* 40:473-480.
- van Genuchten, M. Th. 1980. A closed-form equation for predicting the hydraulic conductivity of unsaturated soils. *Soil Sci. Soc. Am. J.* 44:892-898.

van Genuchten, M. Th., D. H. Tang, and R. Guennelon. 1984. Some exact solutions for solute transport through soils containing large cylindrical macropores. *Water Resour. Res.* 20(3):335-346.

van Genuchten, M. Th. 1985. A general approach for modeling solute transport in structured soils. pp. 513-526. In *Proceedings Hydrogeology of Rocks Low With Hydraulic Conductivity*. *Memoirs of the IAH*, 17(1).

van Genuchten, M. Th. and F. N. Dalton. 1986. Models for simulating salt movement in aggregated field soils. *Geoderma* 38:165-183.

Vepraskas, M. J., A. G. Jongmans, M. T. Hoover, and J. Bouma. 1991. Hydraulic conductivity of saprolite as determined by channels and porous groundmass. *Soil Sci. Soc. Am. J.* 55:932-938.

Ward, R. C., R. J. Luxmoore, and G. T. Yeh. 1987. HYSPEC-Simulating hydraulic flow in spatially correlated soils. ORNL-5879. Oak Ridge National Laboratory, Tenn.

Wickliff, D. S., D. K. Solomon, and N. D. Farrow. 1991. Preliminary investigation of processes that affect source term identification. ORNL/ER-59. Oak Ridge National Laboratory, Tenn.

Wilson, G. V. and R. J. Luxmoore. 1988. Infiltration, macroporosity, and mesoporosity distributions on two forested watersheds. *Soil Sci. Soc. Am. J.* 52:329-335.

Wilson, G. V., P. M. Jardine, and J. P. Gwo. 1992. Modeling the hydraulic properties of a multiregion soil. *Soil Sci. Soc. Am. J.* 56:1731-1737.

Wilson, G. V., P. M. Jardine, J. D. O'Dell, and M. Collineau. 1993. Field-scale transport from a buried line source in variably saturated soil. *J. Hydrol.* 145:83-109.

Yeh, G. T., J. R. Chang, J. P. Gwo, H. C. Lin, and D. R. Richards. 1994. 3DSALT: A three-dimensional finite element model of density-dependent flow and transport through saturated-unsaturated media. Instruction report HL-94-1. U.S. Army Corps of Engineers.

Zimmerman, R. W., G. Chen, T. Hadgu, and G. S. Bodvarsson. 1993. A numerical dual-porosity model with semianalytical treatment of fracture/matrix flow. *Water Resour. Res.* 29(7):2127-2137.

## APPENDIX A. DERIVATION OF MULTIREGION SUBSURFACE SOLUTE TRANSPORT EQUATIONS

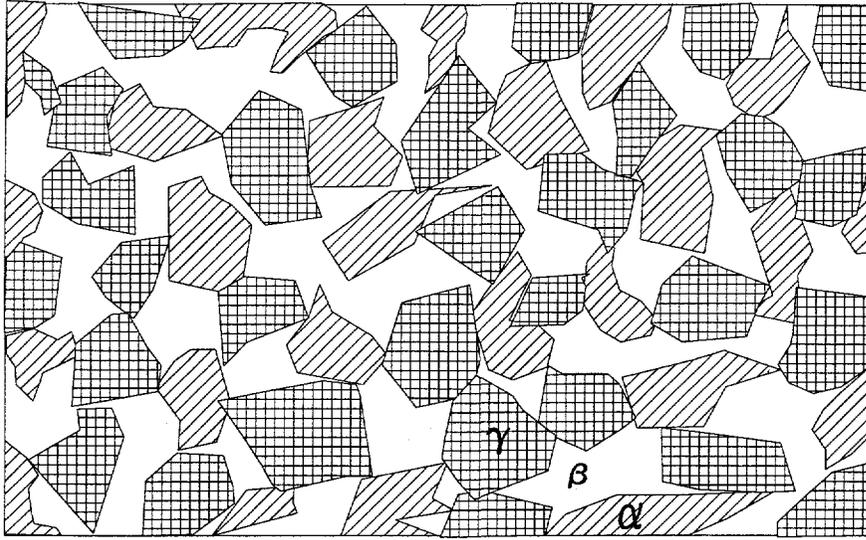
### A.1 THEORETICAL BACKGROUND

Theoretical development of multiregion subsurface solute transport models owes its background to experimental work in the laboratory and field (Coats and Smith 1964; Seyfried and Rao 1987; Jardine et al. 1993; Wilson et al. 1992; Wilson and Luxmoore 1988; Jardine et al. 1990; Wilson et al. 1993). These studies revealed the need to describe solute transport through fast and slow flow regions in undisturbed structured soils as two interacting mass transport domains (Coats and Smith 1964; Jardine et al. 1990). Dead-end-pore and mobile-immobile models devised by Coats and Smith (1964) and van Genuchten and Wierenga (1976), respectively, represent the first attempt by soil scientists to incorporate both high permeability in the inter-aggregate flowing region and high porosity in the intra-aggregate stagnant region. Two-region models, with mobile water in the intra-aggregate region, were also suggested by previous investigators (e.g., Skopp et al. 1981; Skopp and Gardner 1992; Jarvis et al. 1991a; Jarvis et al. 1991b; Sciortino et al. 1994). Near-surface soils in the C horizon at the Oak Ridge National Laboratory are largely composed of partially weathered shale/limestone sequences, or saprolite, which is very rich in microfractures that act as a second secondary pore region and result in fast solute transport under partially saturated conditions (Wilson and Luxmoore 1988). Multiple-pore-region conceptual models (Wilson and Luxmoore 1988) and numerical models (Gwo et al. 1995; Gwo et al. 1994b) are therefore proposed and devised to interpret field and laboratory observations (Gwo et al. 1994a; Gwo et al. 1995).

Because at least two pore regions, or two continua, are associated with a mathematical point, these methods also owe their theoretical background to the theory of homogenization (Arbogast et al. 1989). Pores within a unique distribution of pore sizes usually possess a discernible flow field that can be characterized by a unique set of hydraulic parameters. These pores, distributed over a representative elementary volume (REV), can be collectively represented by a medium so that the flow field can be described by macroscopic balance equations. The interactions between flow fields, as will become obvious later, can be characterized by advective and diffusive mass transfer.

### A.2 DERIVING MULTIREGION SOLUTE TRANSPORT EQUATIONS BASED ON VOLUME-AVERAGE OVER HOMOGENIZED MEDIA

To derive the multiregion solute transport equations, let us consider a structured soil with microscopic features that can be divided into three pore domains, namely pore regions  $\alpha$ ,  $\beta$ , and  $\gamma$  (Fig. A.1). The pore space occupied by a pore region is the union of a large number of inter-connected subregions within which the fluid and solid phases form two immiscible continua. We then homogenize the soil structures into three pore regions and describe the flow fields of the pore regions by microscopic balance equations, for example, momentum and mass balance equations (Hassanizadeh and Gray 1979).



**Fig. A. 1. A hypothetical three-region system with each region occupying a portion of the voids of the porous medium. Mass conservation is maintained within each region and across pore-region boundaries.**

By applying volume-average procedures (Hassanizadeh and Gray 1979) to a single-species multiregion system, one can obtain the following mass balance equation for the aqueous phase (Gwo 1992):

$$\frac{\partial \rho_{\alpha}^f \theta_{\alpha} c_{\alpha}}{\partial t} + \nabla \cdot (\rho_{\alpha}^f \theta_{\alpha} V_{\alpha} c_{\alpha}) - \nabla \cdot \rho_{\alpha}^f J_{\alpha} = \rho_{\alpha}^f \theta_{\alpha} (f_{\alpha} + R_{\alpha} + e_{\alpha r}^i + e_{\alpha p}^i + J_{\alpha r}^i + J_{\alpha p}^i) \quad (\text{A.1})$$

$$\alpha = 1, 2, \dots, N,$$

where, for pore region  $\alpha$ ,

$\theta_{\alpha}$  = water content ( $L^3/L^3$ ),

$\rho_{\alpha}^f$  = fluid phase density ( $M/L^3$ ),

$t$  = time (T),

$V_{\alpha}$  = velocity of the fluid phase (L/T),

$c_{\alpha}$  = solute concentration ( $M/L^3$ ),

$f_{\alpha}$  = external source/sink rates of solute ( $M/L^3T$ ),

$R_{\alpha}$  = internal physical-chemical reaction rates of solute in the fluid phase ( $M/L^3T$ ),

- $e_{\alpha r}^i$  = interregion mass transfer rate due to diffusion (M/L<sup>3</sup>T),  
 $J_{\alpha r}^i$  = interregion mass transfer rate due to mechanical interactions (M/L<sup>3</sup>T),  
 $e_{\alpha p}^i$  = production rates of solute in the fluid phase due to phase change (M/L<sup>3</sup>T),  
 $J_{\alpha p}^i$  = production rates of solute in the fluid phase due to inter-phase mechanical interactions (M/L<sup>3</sup>T),  
 $N$  = number of pore regions.

The surface flux term  $J_{\alpha}$  has been postulated to be proportional to the gradient of solute concentration  $c_{\alpha}$  (Nguyen et al. 1982) as follows:

$$J_{\alpha} = -\theta_{\alpha} D_{\alpha} \cdot \nabla c_{\alpha}, \quad (\text{A.2})$$

$$D_{\alpha} = a_{T\alpha} |V_{\alpha}| \delta + (a_{L\alpha} - a_{T\alpha}) V_{\alpha} V_{\alpha} / |V_{\alpha}| + a_m \tau_{\alpha} \delta, \quad (\text{A.3})$$

where, for pore region  $\alpha$ ,

- $a_{T\alpha}$  = transverse dispersivity (L),  
 $a_{L\alpha}$  = longitudinal dispersivity (L),  
 $D_{\alpha}$  = hydrodynamic dispersion coefficient tensor (L<sup>2</sup>/T),  
 $\delta$  = Kronecker delta tensor,  
 $|V_{\alpha}|$  = the magnitude of the darcy velocity  $V_{\alpha}$  (L/T),  
 $a_m$  = molecular diffusion coefficient (L<sup>2</sup>/T),  
 $\tau_{\alpha}$  = tortuosity.

For the sorbed phase, one can obtain the following mass balance equation for a multiregion system (Gwo 1992):

$$\frac{\partial \rho_{\alpha}^s s_{\alpha}}{\partial t} = \rho_{\alpha}^s \Lambda_{\alpha} + \rho_{\alpha}^s (e_{\alpha p}^s + J_{\alpha p}^s), \quad (\text{A.4})$$

where, for pore region  $\alpha$ ,

- $\rho_{\alpha}^s$  = density of the solid phase (M/L<sup>3</sup>),  
 $s_{\alpha}$  = solute concentration in the sorbed phase (M/L<sup>3</sup>),  
 $\Lambda_{\alpha}$  = production rates of solute in the sorbed phase due to internal chemical-physical reactions (M/L<sup>3</sup>T),  
 $e_{\alpha p}^s$  = the production rates of solute in the sorbed phase due to phase change (M/L<sup>3</sup>T),  
 $J_{\alpha p}^s$  = the production rates of solute in the sorbed phase due to inter-phase mechanical interactions (M/L<sup>3</sup>T).

To arrive at (A.4), we assume that there is no external source/sink entering or exiting the system in the form of the sorbed phase and that interregion mass transfer of species is only through the fluid phase. Note that we need to relate Eqs. (A.1) and (A.4) to form a closed mathematical system. In this report, we adopt equilibrium adsorption/desorption relations, including linear, Freundlich, and Langmuir isotherms, such that the sorbed phase concentration can be expressed analytically as functions of aqueous phase concentration as follows:

$$s_{\alpha} = K_d c_{\alpha} \quad \text{for linear isotherms,} \quad (\text{A.5a})$$

$$s_{\alpha} = K c_{\alpha}^n \quad \text{for Freundlich isotherms,} \quad (\text{A.5b})$$

$$s_{\alpha} = \frac{s_{\alpha,m} K c_{\alpha}}{1 + K c_{\alpha}} \quad \text{for Langmuir isotherms,} \quad (\text{A.5c})$$

where  $K_d$  is the distribution coefficient ( $L^3/M$ ),  $s_{\alpha,m}$  is the maximum concentration allowed on the solid surface ( $M/L^3$ ),  $n$  is the order of Freundlich isotherms, and  $K$  is a functional coefficient for Freundlich and Langmuir isotherms. Internal chemical-physical reactions considered in this report include first-order decay and biodegradation in fluid and solid phases. The internal reaction terms in Eqs. (A.1) and (A.4) can be expressed, respectively, as follows:

$$R_{\alpha} = -\lambda_{\alpha} c_{\alpha} - K_{w\alpha} c_{\alpha}, \quad (\text{A.6})$$

$$\Lambda_{\alpha} = -\lambda_{\alpha} s_{\alpha} - K_{s\alpha} s_{\alpha}, \quad (\text{A.7})$$

where, for pore region  $\alpha$ ,

$\lambda_{\alpha}$  = first-order physical-chemical decay rate ( $T^{-1}$ ),

$K_{w\alpha}$  = first-order biodegradation rate in the fluid phase ( $T^{-1}$ ),

$K_{s\alpha}$  = first-order biodegradation rate in the sorbed phase ( $T^{-1}$ ).

Expanding the left hand side of Eq. (A.1), neglecting the second order term  $J_{\alpha} \cdot \nabla \rho_{\alpha}^f$ , and using relations (A.2) and (A.6), we have the mass balance equation for the aqueous phase:

$$\begin{aligned} \rho_{\alpha}^f \theta_{\alpha} \frac{\partial c_{\alpha}}{\partial t} + \rho_{\alpha}^f \theta_{\alpha} V_{\alpha} \cdot \nabla c_{\alpha} &= \nabla \cdot (\rho_{\alpha}^f \theta_{\alpha} D_{\alpha} \cdot \nabla c_{\alpha}) - \rho_{\alpha}^f \theta_{\alpha} (\lambda_{\alpha} + K_{w\alpha}) c_{\alpha} \\ + \rho_{\alpha}^f q_{\alpha} c_{\alpha}^* - c_{\alpha} \left( \frac{\partial \rho_{\alpha}^f \theta_{\alpha}}{\partial t} + \nabla \cdot \rho_{\alpha}^f \theta_{\alpha} V_{\alpha} \right) &+ \rho_{\alpha}^f \theta_{\alpha} (e'_{\alpha r} + e'_{\alpha p} + J'_{\alpha r} + J'_{\alpha p}) \end{aligned} \quad (\text{A.8})$$

$$\alpha = 1, 2, \dots, N,$$

in which the external source/sink term  $f_\alpha$  is represented by  $\rho_\alpha^f q_\alpha c_{q\alpha}^*$ , where  $q_\alpha$  is external source/sink rates ( $M^3/M^3T$ ) and  $c_{q\alpha}^*$  is solute concentration in the incoming fluid,  $c_{in,\alpha}$ , if  $q_\alpha$  is a source or the resident solute concentration,  $c_\alpha$ , if  $q_\alpha$  is a sink ( $M/L^3$ ). Assuming that the sorbed phase is incompressible, expressing the sorbed phase concentration as mass per unit mass of the porous medium, and utilizing relation (A.7), one arrives at the following equation for the mass balance of the sorbed phase:

$$\rho_{b\alpha} \frac{\partial s_\alpha}{\partial t} = -\rho_{b\alpha} s_\alpha (\lambda_\alpha + K_{s\alpha}) + \rho_{b\alpha} (e_{\alpha p}^s + J_{\alpha p}^s) \quad \alpha = 1, 2, \dots, N, \quad (A.9)$$

where  $\rho_{b\alpha}$  is the bulk density of the porous media allocated to pore region  $\alpha$  ( $M/L^3$ ).

Based on the volume-average theorem (Hassanizadeh and Gray 1979) and the theory of homogenization, Gwo et al. (1994b) developed a variably saturated multiregion subsurface flow system for slightly compressible porous media as follows:

$$\frac{\partial \rho_\alpha^f \theta_\alpha}{\partial t} = -\nabla \cdot \rho_\alpha^f \theta_\alpha V_\alpha + \rho_\alpha^f \theta_\alpha (q_\alpha + e_{\alpha r}^f) \quad \alpha = 1, 2, \dots, N, \quad (A.10)$$

where for pore region  $\alpha$ ,

$n_{e\alpha}$  = effective porosity,

$h_\alpha$  = pressure head (L),

$e_{\alpha r}^f$  = production of fluid mass due to the combined effect of phase change and interface mechanical interactions through the interface.

Substituting Eq. (A.10) into (A.8), expanding the dispersion term, neglecting the second-order term  $\nabla \rho_\alpha^f \nabla \cdot \theta_\alpha D_\alpha \cdot \nabla c_\alpha$ , and dividing the resultant equation by  $\rho_\alpha^f$ , we arrive at:

$$\begin{aligned} \theta_\alpha \frac{\partial c_\alpha}{\partial t} + \theta_\alpha V_\alpha \cdot \nabla c_\alpha = \nabla \cdot (\theta_\alpha D_\alpha \cdot \nabla c_\alpha) - \theta_\alpha (\lambda_\alpha + K_{w\alpha}) c_\alpha + q_\alpha c_{q\alpha}^* \\ - c_\alpha \theta_\alpha (q_\alpha + e_{\alpha r}^f) + \theta_\alpha (e_{\alpha r}^t + e_{\alpha p}^t + J_{\alpha r}^t + J_{\alpha p}^t) \quad \alpha = 1, 2, \dots, N, \end{aligned} \quad (A.11)$$

Note that in Eq. (A.11), mass transfer between pore regions is characterized by a diffusion driven term,  $\theta_\alpha e_{\alpha r}^t$ , and two mechanical interaction, or advection, terms,  $\theta_\alpha e_{\alpha r}^f$  and  $\theta_\alpha J_{\alpha r}^t$ . The first advective mass transfer term is inherited from the fluid mass balance or the flow equation and is mechanistically analogous to the external source/sink term  $\theta_\alpha q_\alpha$ . The solute concentration associated with these movements of fluid mass is  $c_\alpha$ . The second advective mass transfer term is native to the multiregion solute transport equation

and is mechanistically analogous to the advection term on the left-hand side of Eq. (A.11). The solute concentration associated with this movement of fluid mass is that of the upstream pore region. The external source/sink terms,  $q_\alpha c_{q\alpha}^*$  and  $q_\alpha c_\alpha$ , should cancel each other if  $q_\alpha$  is a sink. Similarly, if a pore region is a sink relative to another, the advective mass transfer terms should cancel each other also. For example, using a linear first-order fluid mass transfer equation (Gwo et al. 1994b),

$$\theta_\alpha e_{\alpha r}^f = - \sum_{\substack{j=1 \\ j \neq \alpha}}^N \varepsilon_{\alpha j}^f (h_\alpha - h_j), \quad (\text{A.12})$$

where  $\varepsilon_{\alpha j}^f$  is the fluid mass transfer coefficient between pore regions  $\alpha$  and  $j$ , we have the two advective mass transfer terms:

$$c_\alpha \theta_\alpha e_{\alpha r}^f = - \sum_{\substack{j=1 \\ j \neq \alpha}}^N \varepsilon_{\alpha j}^f (h_\alpha - h_j) c_\alpha \quad (\text{A.13})$$

and

$$\theta_\alpha J_{\alpha r}^f = - \sum_{\substack{j=1 \\ j \neq \alpha}}^N \varepsilon_{\alpha j}^f (h_\alpha - h_j) c_{r\alpha}^*, \quad (\text{A.14})$$

where  $c_{r\alpha}^* = c_\alpha$  if  $h_\alpha > h_j$  or pore region  $\alpha$  is a sink pore region with respect to pore region  $j$ , in which case Eq. (A.13) and (A.14) cancel each other in (A.11), and  $c_{r\alpha}^* = c_j$  if  $h_\alpha < h_j$  or pore region  $\alpha$  is a source region or being "recharged" by pore region  $j$ .

We propose that the intra-region interphase mass transfer is instantaneously equilibrium such that

$$\theta_\alpha (e_{\alpha p}^f + J_{\alpha p}^f) + \rho_{b\alpha} (e_{\alpha p}^s + J_{\alpha p}^s) = 0. \quad (\text{A.15})$$

Adding Eqs. (A.10) and (A.11) and utilizing Eq. (A.15), we have the mass balance equation for a single-species multiregion solute transport system in subsurface media as follows:

$$\begin{aligned} \theta_\alpha \frac{\partial c_\alpha}{\partial t} + \rho_{b\alpha} \frac{\partial s_\alpha}{\partial t} + \theta_\alpha V_\alpha \cdot \nabla c_\alpha &= \nabla \cdot (\theta_\alpha D_\alpha \cdot \nabla c_\alpha) - \theta_\alpha (\lambda_\alpha + K_{w\alpha}) c_\alpha \\ - \rho_{b\alpha} s_\alpha (\lambda_\alpha + K_{s\alpha}) + q_\alpha c_{q\alpha}^* - c_\alpha \theta_\alpha (q_\alpha + e_{\alpha f}^f) & \\ + \theta_\alpha (e_{\alpha r}^f + J_{\alpha r}^f) \quad \alpha = 1, 2, \dots, N, & \end{aligned} \quad (\text{A.16})$$

with the relation between aqueous and solid phase solute concentrations defined by Eq. (A.5a), (A.5b), or (A.5c). To obtain the working multiregion solute transport equation, one needs to determine the appropriate mass transfer relations in Eq. (A.16), which should be based on factors such as the time-scale of the application, the soils of interest, and the complexity of the problem domain. The finite element implementation of Eq. (A.16) with linear first-order mass transfer relations (see Sect. 2) is presented in Appendix B.

### A.3 REFERENCES

Arbogast, T., J. Douglas, Jr., and U. Hornung. 1989. Derivation of the double porosity model of single phase flow via homogenization theory. *SIAM J. Math. Anal.* 21(4):823-836.

Coats, K. H. and B. D. Smith. 1964. Dead-end pore volume and dispersion in porous media. *Soc. Petr. Eng. J.* 4:73-84.

Gwo, J. P. 1992. Multi-region flow and transport modeling in subsurface media. Ph.D. dissertation. Pennsylvania State University, University Park.

Gwo, J. P., P. M. Jardine, G. V. Wilson, and G. T. Yeh. 1994a. Modeling small-scale physical non-equilibrium and large-scale preferential fluid and solute transport in a structured soil. pp. 465-472. In A. Peters, G. Wittum, B. Herrling, U. Meissner, C. A. Brebbia, W. G. Gray, and G. F. Pinder (eds.), *Computational Methods in Water Resources X*, Vol. 1. Kluwer Academic, Dordrecht.

Gwo, J. P., P. M. Jardine, G. T. Yeh, and G. V. Wilson. 1994b. MURF user's guide: A finite element model of multiple-pore-region flow through variably saturated subsurface media. ORNL/GWPO-011. Oak Ridge National Laboratory, Tenn.

Gwo, J. P., P. M. Jardine, G. V. Wilson, and G. T. Yeh. 1995. A multiple-pore-region concept to modeling mass transfer in subsurface media. *J. Hydrol.* 164:217-237.

Hassanizadeh, M., and W. G. Gray. 1979. General conservation equations for multi-phase systems: 1. Averaging procedure. *Adv. in Water Resour.* 2:131-144.

Jardine, P. M., G. V. Wilson, and R. J. Luxmoore. 1990. Unsaturated solute transport through a forested soil during rain storm events. *Geoderma* 46:103-118.

Jardine, P. M., G. K. Jacobs, and G. V. Wilson. 1993. Unsaturated transport processes in undisturbed heterogeneous porous media: I. Inorganic contaminants. *Soil Sci. Soc. Am. J.* 57:945-953.

Jarvis, N. J., P. E. Jansson, P. E. Dik, and I. Messing. 1991a. Modeling water and solute transport in macroporous soil. I. Model description and sensitivity analysis. *J. Soil Sci.* 42:59-70.

Jarvis, N. J., L. Bergstrom, and P. E. Dik. 1991b. Modelling water and solute transport in macroporous soil. II. Chloride breakthrough under non-steady flow. *J. Soil Sci.* 42:71-81.

Nguyen, V. V., W. G. Gray, G. F. Pinder, J. F. Botha, and D. A. Crerar. 1982. A theoretical investigation on the transport of chemicals in reactive porous media. *Water Resour. Res.* 18(4):1149-1156.

Sciortino, A., P.-H Tseng, and M. Th. van Genuchten. 1994. A partitioned solution procedure for water flow and solute transport in a dual-porosity medium. *EOS* 75(44):235.

Seyfried, M. S. and P. S. C. Rao. 1987. Solute transport in undisturbed columns of an aggregated tropical soil: Preferential flow effects. *Soil Sci. Soc. Am. J.* 51:1434-1444.

Skopp, J., W. R. Gardner, and E. J. Tyler. 1981. Solute movement in structured soils: Two-region model with small interaction. *Soil Sci. Soc. Am. J.* 45:837-842.

Skopp, J. and W. R. Gardner. 1992. Miscible displacement: An interacting flow region model. *Soil Sci. Soc. Am. J.* 56:1680-1686.

Solomon, D. K., G. K. Moore, L. E. Toran, R. B. Dreier, and W. M. McMaster. 1992. Status report: A hydrologic framework for the Oak Ridge reservation. ORNL/TM-12026. Oak Ridge National Laboratory, Tenn.

van Genuchten, M. Th. and P. J. Wierenga. 1976. Mass transfer studies in sorbing porous media. I. Analytical solutions. *Soil Sci. Soc. Am. J.* 40:473-480.

Wilson, G. V. and R. J. Luxmoore. 1988. Infiltration, macroporosity, and mesoporosity distributions on two forested watersheds. *Soil Sci. Soc. Am. J.* 52:329-335.

Wilson, G. V., P. M. Jardine, and J. P. Gwo. 1992. Modeling the hydraulic properties of a multiregion soil. *Soil Sci. Soc. Am. J.* 56:1731-1737.

Wilson, G. V., P. M. Jardine, J. D. O'Dell and M. Collineau. 1993. Field-scale transport from a buried line source in variably saturated soil. *J. Hydrol.* 145:83-109.

## APPENDIX B. NUMERICAL IMPLEMENTATION OF MULTIREGION SOLUTE TRANSPORT EQUATIONS

### B.1 OVERVIEW

The multiregion subsurface solute transport equations derived in Appendix A are a set of simultaneous nonlinear differential equations, if the Freundlich or Langmuir isotherm is selected. Together with the boundary and initial conditions given in Sect. 2, the system of equations is too complex to be solved with available analytical methods. Numerical methods such as finite element and finite difference methods are therefore needed to solve the equations. Because of the flexibility of finite element methods in dealing with anisotropy, irregular boundaries, media of irregular shapes, and directness and simplicity in incorporating boundary conditions, we choose finite element over finite difference methods. Additionally, to reduce numerical dispersion caused by upstream weighting methods, a hybrid Lagrangian-Eulerian approach with backward particle tracking (Baptista 1987) is also implemented in MURT, the multiregion solute transport code presented in this report.

A basic difference between multiregion and one-region subsurface solute transport models is that the former models have  $N$  degrees of freedom, or  $N$  concentration entries for  $N$  pore regions, at a locale, while the latter only have one. A multiregion equation can be solved iteratively from pore region to pore region or simultaneously by explicitly coupling pore-region solute transport equations. The latter approach is adopted because the characterization of mass transfer processes is largely the emphasis of this research and pore-region coupling may dominate the convergence of the solution of the nonlinear equations. The mass transfer terms may increase the weight of the mass matrix and may result in a more diagonally dominant matrix for the linearized matrix equations.

This section presents a hybrid Lagrangian-Eulerian finite element implementation of the multiregion solute transport equations. The equations are derived in Appendix A and are not repeated here. The implementation of the governing equation and various types of boundary conditions are discussed in the next section. Given the concentration field, the mass flux equation is solved by the Galerkin finite element method to obtain pore-region mass fluxes. A brief description of the matrix solvers used in MURT is also presented.

### B.2 NUMERICAL APPROXIMATION OF MULTIREGION SOLUTE TRANSPORT EQUATIONS

#### B.2.1 Finite Element Approximation of Solute Transport Equations

To systematically implement a finite element method, one typically divides the area of interest into  $m$  elements and  $n$  nodes. An element matrix can be calculated analytically or numerically for each of the elements according to Eqs. (2.9), (2.10), and (2.31). Finally, the element matrices are assembled over the entire domain to obtain the global matrix

equation. Before calculating the element matrices, one needs to determine an interpolation scheme to approximate local solute concentrations. An approximated concentration, in general, can be represented by the interpolation function

$$\hat{c}_{e\alpha}(x, z, t) = \sum_{l=1}^{ns} c_{l\alpha}(t) B_l(x, z) \quad ns = 3 \text{ or } 4, \quad (\text{B.1})$$

where

- $B_l$  = base function associated with local node  $l$ ,
- $c_{l\alpha}$  = solute concentration at local node  $l$  of element  $e$  in region  $\alpha$  (L),
- $\hat{c}_{e\alpha}$  = interpolated concentration within the area of element  $e$  in region  $\alpha$  (L),
- $t$  = time,
- $x$  and  $z$  = spatial coordinates,
- $ns = 3$  for triangular elements and  $4$  for quadrilateral elements.

To approximate concentrations at locales other than the finite element nodes, one needs to choose an appropriate set of base functions. The implementation of base functions in finite element codes is usually accomplished by expressing the functions in local coordinates. For example, a quadrilateral element is mapped into a square whose corners are located at  $\xi = \pm 1$  and  $\zeta = \pm 1$  (Fig. B.1). A triangular element, on the other hand, is mapped into local natural coordinates (Fig. B.1). The base functions for triangular elements can be expressed in terms of global coordinates as

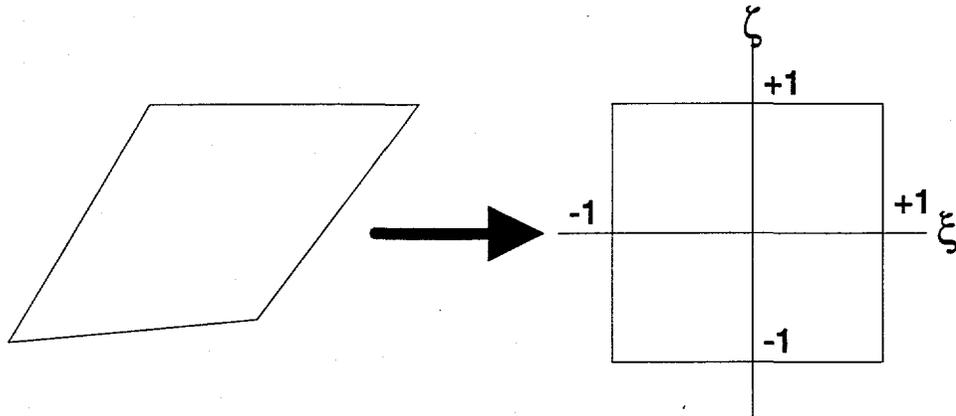
$$\begin{bmatrix} B_1 \\ B_2 \\ B_3 \end{bmatrix} = \frac{1}{2A} \begin{bmatrix} x_2 z_3 - x_3 z_2 & z_2 - z_3 & x_3 - x_2 \\ x_3 z_1 - x_1 z_3 & z_3 - z_1 & x_1 - x_3 \\ x_1 z_2 - x_2 z_1 & z_1 - z_2 & x_2 - x_1 \end{bmatrix} \begin{bmatrix} 1 \\ x \\ z \end{bmatrix}, \quad (\text{B.2a})$$

where  $A$  is the area of the triangle and  $(x_i, z_i)$ ,  $i = 1, 2, 3$  are the global nodal coordinates of the vertices. For quadrilateral elements, the four base functions are obtained by taking the tensor product of the two base functions for linear line elements, in terms of local coordinate:

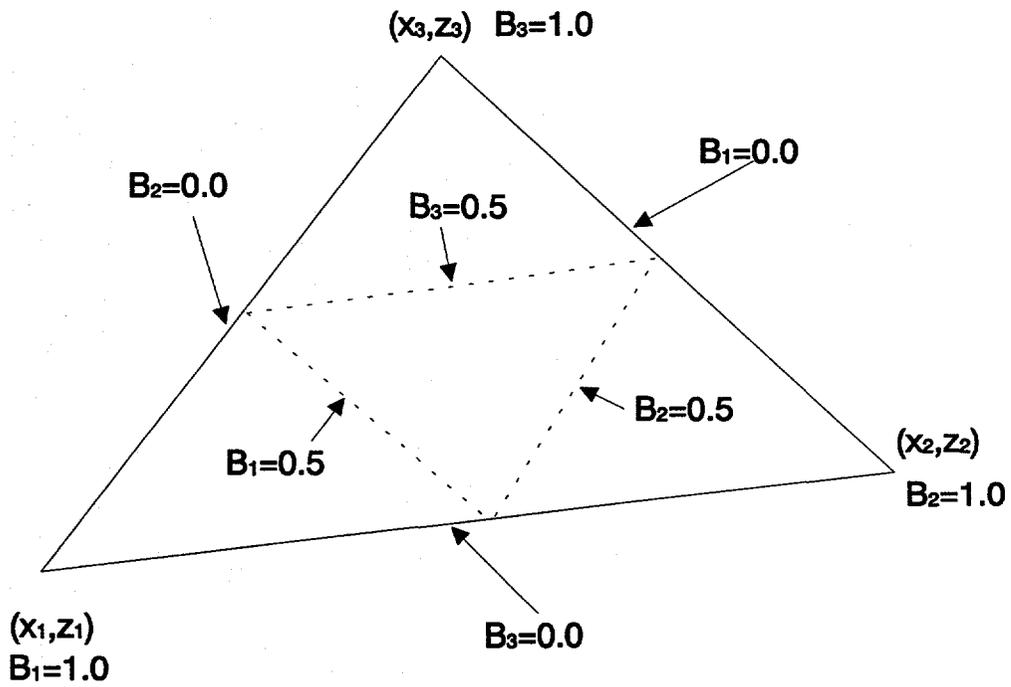
$$B_l(\xi, \zeta) = \frac{1}{4} (1 + \xi \xi_l) (1 + \zeta \zeta_l) \quad l = 1, \dots, 4, \quad (\text{B.2b})$$

where  $(\xi_l, \zeta_l)$  represents the local coordinates of the corner nodes. Substituting Eq. (B.1) into (2.31), we may define the residue for pore region  $\alpha$  as follows:

(a) quadrilateral elements



(b) triangular elements



**Fig. B.1. Local coordinate systems used for quadrilateral and triangular elements. Quadrilateral elements are mapped to a square, and a natural coordinate system is used for triangular elements.**

$$\begin{aligned}
R_{\epsilon\alpha} = & \left( \theta_\alpha + \rho_{b\alpha} \frac{d\hat{s}_\alpha}{d\hat{c}_\alpha} \right) \frac{D\hat{c}_\alpha}{Dt} - \theta_\alpha \nabla \cdot (D_\alpha \cdot \nabla \hat{c}_\alpha) + \lambda_\alpha \left( \theta_\alpha + \rho_{b\alpha} \frac{d\hat{s}_\alpha}{d\hat{c}_\alpha} \right) \hat{c}_\alpha \\
& + \left( K_{w\alpha} + K_{s\alpha} \rho_{b\alpha} \frac{d\hat{s}_\alpha}{d\hat{c}_\alpha} \right) \hat{c}_\alpha - \theta_\alpha q_\alpha \hat{c}_{q\alpha}^* + \hat{c}_\alpha \theta_\alpha q_\alpha + \sum_{\substack{j=1 \\ j \neq \alpha}}^N \epsilon_{\alpha j}^f (h_\alpha - h_j) \hat{c}_\alpha^* \\
& + \sum_{\substack{j=1 \\ j \neq \alpha}}^N \epsilon_{\alpha j}^i (\hat{c}_\alpha - \hat{c}_j) - \sum_{\substack{j=1 \\ j \neq \alpha}}^N \epsilon_{\alpha j}^f (h_\alpha - h_j) \hat{c}_\alpha - \rho_{b\alpha} (\lambda_\alpha + K_{s\alpha}) \left( \hat{s}_{\alpha,0} - \frac{d\hat{s}_\alpha}{d\hat{c}_\alpha} \hat{c}_{\alpha,0} \right),
\end{aligned}$$

(B.3)

where

$\hat{s}_\alpha = f_1(\hat{c}_\alpha)$  is the approximated solute concentration in the solid phase (M/M),

$\hat{c}_{q\alpha}^* = \hat{c}_\alpha$  if  $q_\alpha < 0$ ;  $\hat{c}_{q\alpha}^* = c_{in,\alpha}$ , solute concentration in the incoming fluid,  
if  $q_\alpha > 0$ ,

$\hat{c}_\alpha^* = \hat{c}_\alpha$  if  $h_\alpha > h_j$ ;  $\hat{c}_\alpha^* = \hat{c}_j$  otherwise.

For the Galerkin finite element method, the weighting functions are taken as the same as the base functions. Applying the principle of variation and forcing the residue to zero, we assemble the elements over the entire area of interest and arrive at the following equation, for Lagrangian-Eulerian approach:

$$\begin{aligned}
& \sum_{e=1}^m \int_{R_e} B_k \left[ \left( \theta_\alpha + \rho_{b\alpha} \sum_{l=1}^{ns} B_l \frac{ds_\alpha}{dc_\alpha} \Big|_{c_\alpha=c_{l\alpha}} \right) \frac{D}{Dt} \left( \sum_{l=1}^{ns} B_l c_{l\alpha} \right) - \theta_\alpha \nabla \cdot \left( D_\alpha \cdot \nabla \left( \sum_{l=1}^{ns} B_l c_{l\alpha} \right) \right) \right. \\
& + \lambda_\alpha \left( \theta_\alpha + \rho_{b\alpha} \sum_{l=1}^{ns} B_l \frac{ds_\alpha}{dc_\alpha} \Big|_{c_\alpha=c_{l\alpha}} \right) \left( \sum_{l=1}^{ns} B_l c_{l\alpha} \right) + \left( K_{w\alpha} + K_{s\alpha} \rho_{b\alpha} \sum_{l=1}^{ns} B_l \frac{ds_\alpha}{dc_\alpha} \Big|_{c_\alpha=c_{l\alpha}} \right) \left( \sum_{l=1}^{ns} B_l c_{l\alpha} \right) \\
& - \theta_\alpha q_\alpha \hat{c}_{q\alpha}^* + \theta_\alpha q_\alpha \hat{c}_\alpha + \sum_{\substack{j=1 \\ j \neq \alpha}}^N \epsilon_{\alpha j}^f \left( \sum_{l=1}^{ns} B_l (h_{l\alpha} - h_{lj}) c_{l\alpha}^* \right) \\
& + \sum_{\substack{j=1 \\ j \neq \alpha}}^N \epsilon_{\alpha j}^i \left( \sum_{l=1}^{ns} B_l (c_{l\alpha} - c_{lj}) \right) - \sum_{\substack{j=1 \\ j \neq \alpha}}^N \epsilon_{\alpha j}^f \left( \sum_{l=1}^{ns} B_l (h_{l\alpha} - h_{lj}) c_{l\alpha} \right) \\
& \left. + \rho_{b\alpha} (\lambda_\alpha + K_{s\alpha}) \left( \sum_{l=1}^{ns} B_l s_{\alpha,0} - \sum_{l=1}^{ns} B_l \frac{ds_\alpha}{dc_\alpha} \Big|_{c_\alpha=c_{\alpha,0}} c_{\alpha,0} \right) \right] dR = 0
\end{aligned}$$

$k = 1, \dots, 4$  and  $ns = 3$  or  $4$ .

(B.4)

The second external source/sink term in Eq. (B.4) is implemented as

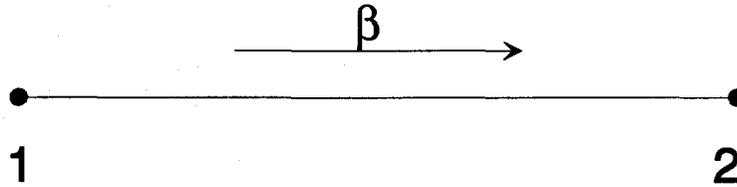
$$\theta_{\alpha} q_{\alpha} \hat{c}_{\alpha} = \theta_{\alpha} q_{\alpha} \sum_{l=1}^{ns} B_l c_{l\alpha} \text{ if non-point sources are specified by the modeler. For the}$$

upstream finite element approach, one needs to use a weighting function with orders higher than the base function. In this report, an  $N + 1$  upstream weighting function is used. An appropriate  $N + 1$  weighting function (Yeh et al. 1994) for a line element (Fig. B.2) may be defined as follows:

$$F_1(\xi, \beta) = B_1(\xi) - \beta \frac{3}{4}(1 + \xi)(1 - \xi) , \quad (\text{B.5a})$$

$$F_2(\xi, \beta) = B_2(\xi) - \beta \frac{3}{4}(1 + \xi)(1 - \xi) , \quad (\text{B.5b})$$

where  $\beta$  is the weighting factor along the line from node 1 to node 2 (Fig. B.2).



**Fig. B.2. Weighting factor along a line element.**

The weighting functions for a quadrilateral element are obtained by an appropriate tensor product as follows:

$$W_1 = F_1(\xi, \beta_1) F_1(\zeta, \gamma_1) , \quad (\text{B.6a})$$

$$W_2 = F_2(\xi, \beta_2) F_1(\zeta, \gamma_1) , \quad (\text{B.6b})$$

$$W_3 = F_1(\xi, \beta_1) F_2(\zeta, \gamma_2) , \quad (\text{B.6c})$$

$$W_4 = F_2(\xi, \beta_2) F_2(\zeta, \gamma_2) , \quad (\text{B.6d})$$

where the  $\beta$ s and  $\gamma$ s are the weighting factors along the element sides, as shown in Fig. B.3.

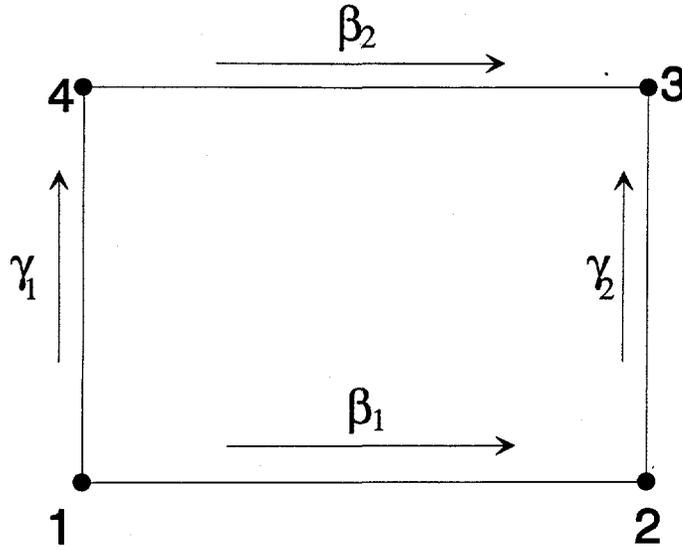


Fig. B.3. Upstream weighting factors along the sides of a quadrilateral element.

Applying the upstream weighting functions to the advection term and using the base functions as weighting functions for the others, we arrive at the following equation for upstream finite element method:

$$\begin{aligned}
 & \sum_{e=1}^m \int_{R_e} \left\{ W_k \theta_\alpha v_\alpha \cdot \nabla \left( \sum_{l=1}^{ns} B_l c_{l\alpha} \right) + B_k \left[ \left( \theta_\alpha + \rho_{b\alpha} \sum_{l=1}^{ns} B_l \frac{ds_\alpha}{dc_\alpha} \Big|_{c_\alpha=c_{l\alpha}} \right) \frac{d}{dt} \left( \sum_{l=1}^{ns} B_l c_{l\alpha} \right) \right. \right. \\
 & - \theta_\alpha \nabla \cdot \left( D_\alpha \cdot \nabla \left( \sum_{l=1}^{ns} B_l c_{l\alpha} \right) \right) + \lambda_\alpha \left( \theta_\alpha + \rho_{b\alpha} \sum_{l=1}^{ns} B_l \frac{ds_\alpha}{dc_\alpha} \Big|_{c_\alpha=c_{l\alpha}} \right) \left( \sum_{l=1}^{ns} B_l c_{l\alpha} \right) \\
 & + \left( K_{w\alpha} + K_{s\alpha} \rho_{b\alpha} \sum_{l=1}^{ns} B_l \frac{ds_\alpha}{dc_\alpha} \Big|_{c_\alpha=c_{l\alpha}} \right) \left( \sum_{l=1}^{ns} B_l c_{l\alpha} \right) - \theta_\alpha q_\alpha \hat{c}_{q\alpha}^* + \theta_\alpha q_\alpha \hat{c}_\alpha \\
 & + \sum_{\substack{j=1 \\ j \neq \alpha}}^N \varepsilon_{\alpha j}^f \left( \sum_{l=1}^{ns} B_l (h_{l\alpha} - h_{lj}) c_{l\alpha}^* \right) + \sum_{\substack{j=1 \\ j \neq \alpha}}^N \varepsilon_{\alpha j}^t \left( \sum_{l=1}^{ns} B_l (c_{l\alpha} - c_{lj}) \right) - \sum_{\substack{j=1 \\ j \neq \alpha}}^N \varepsilon_{\alpha j}^f \left( \sum_{l=1}^{ns} B_l (h_{l\alpha} - h_{lj}) c_{l\alpha} \right) \\
 & \left. + \rho_{b\alpha} \left( \lambda_\alpha + K_{s\alpha} \right) \left( \sum_{l=1}^{ns} B_l s_{\alpha,0} - \sum_{l=1}^{ns} B_l \frac{ds_\alpha}{dc_\alpha} \Big|_{c_\alpha=c_{\alpha,0}} c_{\alpha,0} \right) \right\} dR = 0,
 \end{aligned}$$

$$k = 1, \dots, 4 \text{ and } ns = 3 \text{ or } 4$$

(B.7)

Applying boundary conditions, we can rewrite (B.4) in matrix form as the following:

$$\begin{aligned}
[M_\alpha] \left\{ \frac{Dc_\alpha}{Dt} \right\} + ([D_\alpha] + [K_\alpha]) \{c_\alpha\} + \sum_{\substack{j=1 \\ j \neq \alpha}}^N [E'_{\alpha j}] \{c_\alpha\} - \sum_{\substack{j=1 \\ j \neq \alpha}}^N [E'_{\alpha j}] \{c_j\} \\
+ \sum_{\substack{j=1 \\ j \neq \alpha}}^N [E^f_{\alpha j}] \{c_\alpha^*\} - \sum_{\substack{j=1 \\ j \neq \alpha}}^N [E^f_{\alpha j}] \{c_\alpha\} = \{Q_\alpha\} + \{L_\alpha\},
\end{aligned} \tag{B.8}$$

where, for pore region  $\alpha$ ,  $[M_\alpha]$  is the global mass matrix from the storage term;  $[D_\alpha]$  is the global stiffness matrix from the dispersion term;  $[K_\alpha]$  is the global stiffness matrix from the reaction terms;  $[E'_{\alpha j}]$  is the global mass matrix resulting from the diffusive mass transfer coefficients  $\epsilon'_{\alpha j}$ ;  $[E^f_{\alpha j}]$  is the global mass matrix resulting from the product of advective mass transfer coefficients and pressure head difference;  $\{Dc_\alpha/Dt\}$ ,  $\{c_\alpha\}$ , and  $\{c_j\}$  are the global vectors containing values of  $Dc_\alpha/Dt$ ,  $c_\alpha$ , and solute concentration at pore region  $j$ ,  $c_j$ , respectively; and  $\{Q_\alpha\}$  and  $\{L_\alpha\}$  are the global load vectors resulting from external source/sink and boundary conditions, respectively. Expanding Eq. (B.5) and applying integration by parts to the dispersion term, we can further express the coefficient matrices and load vectors as follows:

$$M_{\alpha,ij} = \sum_{e \in M_e} \int_{R_e} B_k^e \left( \theta_\alpha + \rho_{b\alpha} \sum_{l=1}^{ns} B_l \frac{ds_\alpha}{dc_\alpha} \Big|_{c_\alpha=c_{l\alpha}} \right) B_l^e dR, \tag{B.9}$$

$$D_{\alpha,ij} = \sum_{e \in M_e} \int_{R_e} \nabla B_k^e \cdot \theta_\alpha D_\alpha \cdot \nabla B_l^e dR, \tag{B.10}$$

$$\begin{aligned}
K_{\alpha,ij} = \sum_{e \in M_e} \int_{R_e} B_k^e \left[ \lambda_\alpha \left( \theta_\alpha + \rho_{b\alpha} \sum_{l=1}^{ns} B_l \frac{ds_\alpha}{dc_\alpha} \Big|_{c_\alpha=c_{l\alpha}} \right) + \theta_\alpha q_\alpha \right. \\
\left. + \left( K_{w\alpha} + K_{s\alpha} \rho_{b\alpha} \sum_{l=1}^{ns} B_l \frac{ds_\alpha}{dc_\alpha} \Big|_{c_\alpha=c_{l\alpha}} \right) \right] B_l^e dR,
\end{aligned} \tag{B.11}$$

$$E'_{\alpha\beta,ij} = \sum_{e \in M_e} \int_{R_e} B_k^e \epsilon'_{\alpha\beta} B_l^e dR, \tag{B.12}$$

$$E^f_{\alpha\beta,ij} = \sum_{e \in M_e} \int_{R_e} B_k^e \epsilon^f_{\alpha\beta} (h_{l\alpha} - h_{l\beta}) B_l^e dR, \tag{B.13}$$

$$\begin{aligned}
Q_{\alpha,i} = & \sum_{e \in M_e} \int_{R_e} B_k^e \left[ -\rho_{b\alpha} K_{s\alpha} \left( \sum_{l=1}^{ns} B_l s_{\alpha,0} - \sum_{l=1}^{ns} B_l \frac{ds_{\alpha}}{dc_{\alpha}} \Big|_{c_{\alpha}=c_{\alpha,0}} c_{\alpha,0} \right) \right. \\
& \left. + \theta_{\alpha} q_{\alpha} c_{in,\alpha} - \rho_{b\alpha} \lambda_{\alpha} \left( \sum_{l=1}^{ns} B_l s_{\alpha,0} - \sum_{l=1}^{ns} B_l \frac{ds_{\alpha}}{dc_{\alpha}} \Big|_{c_{\alpha}=c_{\alpha,0}} c_{\alpha,0} \right) \right] dR
\end{aligned} \tag{B.14}$$

$$L_{\alpha,i} = - \sum_{e \in N_{se}} \int_{L_e} B_k^e \mathbf{n} \cdot (-\theta_{\alpha} \mathbf{D}_{\alpha} \cdot \nabla c_{\alpha}) dL, \tag{B.15}$$

where

$R_e$  = region of element  $e$ ,

$M_e$  = elements that have a local side  $k-l$  coinciding with the global side  $i-j$ ,

$B_k^e$  =  $k$ th local base function,

$N_{se}$  = boundary segments that have a local node  $k$  coinciding with the global node  $i$ ,

$L_e$  = length of boundary segment  $e$ .

Similarly, if one chooses to use upstream weighting methods, one can obtain the following matrix equation:

$$\begin{aligned}
[M_{\alpha}] \left\{ \frac{dc_{\alpha}}{dt} \right\} + ([A_{\alpha}] + [D_{\alpha}] + [K_{\alpha}]) \{c_{\alpha}\} + \sum_{\substack{j=1 \\ j \neq \alpha}}^N [E_{\alpha j}^t] \{c_{\alpha}\} - \sum_{\substack{j=1 \\ j \neq \alpha}}^N [E_{\alpha j}^t] \{c_j\} \\
+ \sum_{\substack{j=1 \\ j \neq \alpha}}^N [E_{\alpha j}^f] \{c_{\alpha}^*\} - \sum_{\substack{j=1 \\ j \neq \alpha}}^N [E_{\alpha j}^f] \{c_{\alpha}\} = \{Q_{\alpha}\} + \{L_{\alpha}\},
\end{aligned} \tag{B.16}$$

where

$$A_{\alpha,ij} = \sum_{e \in M_e} \int_{R_e} B_k^e v_{\alpha} \cdot \nabla B_l^e dR. \tag{B.17}$$

To ensure that mass transfer is confined locally at a finite element node and between those pore regions sharing the same node, the mass matrix representing mass transfer is always lumped in MURT. The lumping of the mass matrix representing the storage term is optional and will be described later.

## B.2.2 Numerical Implementation of Mass Flux Equation

After the concentration fields are obtained, the Galerkin finite element method is similarly applied to the mass flux equation

$$F_{\alpha} = -v_{\alpha}c_{\alpha} + \theta_{\alpha}D_{\alpha} \cdot \nabla c_{\alpha} , \quad (\text{B.18})$$

at each direction, such that

$$[T_{\alpha}]\{F_{\alpha,x_i}\} = \{D_{\alpha,x_i}\} \quad x_i = x, z , \quad (\text{B.19})$$

where  $F_{\alpha,x_i}$  is the mass flux in  $x_i$  direction, and the mass matrix  $[T_{\alpha}]$  and load vectors  $\{D_{\alpha,x_i}\}$  can be expressed as

$$T_{ij} = \sum_{e \in M_e} \int_{R_e} B_k^e B_i^e dR , \quad (\text{B.20})$$

$$D_{\alpha,x_i} = - \sum_{e \in M_e} \int_{R_e} B_k^e n_i \cdot (v_{\alpha}c_{\alpha} - \theta_{\alpha}D_{\alpha} \cdot \nabla c_{\alpha}) dR , \quad (\text{B.21})$$

where  $n_i$  is a unit vector along a coordinate axis. A 2 x 2 point Gaussian quadrature is used to numerically integrate Eqs. (B.9)-(B.15), (B.17), (B.20) through (B.21) locally for each element and boundary sides. To integrate locally, global coordinates must be transformed to local coordinates. For MURT, an isoparametric transformation, in which the global coordinates are interpolated using the same set of base functions as those in Eqs. (B.9) through B(15), is used. For details of the isoparametric finite element method, interested readers are referred to Lapidus and Pinder (1982).

## B.2.3 Finite Difference Approximation of Time Derivative Term

Finite difference is typically used to approximate the time derivative term in Eq. (B.8) and (B.16) so that the equation can be expressed as a matrix equation that can be solved by a matrix solver. Equation (B.8) therefore becomes

$$\begin{aligned}
& \frac{[M_\alpha]}{\Delta t} (\{c_\alpha\}_{t+\Delta t} - \{c_{\alpha,L}\}_t) + ([D_\alpha] + [K_\alpha]) [\omega \{c_\alpha\}_{t+\Delta t} + (1-\omega) \{c_\alpha\}_t] \\
& + \sum_{\substack{j=1 \\ j \neq \alpha}}^N [E_{\alpha j}^t] [\omega \{c_\alpha\}_{t+\Delta t} + (1-\omega) \{c_\alpha\}_t] - \sum_{\substack{j=1 \\ j \neq \alpha}}^N [E_{\alpha j}^t] [\omega \{c_j\}_{t+\Delta t} + (1-\omega) \{c_j\}_t] \\
& + \sum_{\substack{j=1 \\ j \neq \alpha}}^N [E_{\alpha j}^f] [\omega \{c_\alpha^*\}_{t+\Delta t} + (1-\omega) \{c_\alpha^*\}_t] - \sum_{\substack{j=1 \\ j \neq \alpha}}^N [E_{\alpha j}^f] [\omega \{c_\alpha\}_{t+\Delta t} + (1-\omega) \{c_\alpha\}_t] \\
& = \{Q_\alpha\} + \{L_\alpha\},
\end{aligned}$$

(B.22)

where  $\{c_{\alpha,L}\}_t$  is the Lagrangian concentration vector, and the mass matrices, stiff matrix, and the local vectors are evaluated at  $t + \omega\Delta t$ . Similar procedure can be applied to Eq. (B.16) and is not detailed here. The coefficient  $\omega$  is a weighting factor and may be varied from 0.0 to 1.0;  $\omega = 0.0$  results in an explicit forward-difference scheme,  $\omega = 0.5$  results in a Crank-Nicholson central-difference scheme, and  $\omega = 1.0$  results in an implicit backward-difference scheme. The Picard iteration method used in MURT in fact allows the entire range of  $[0.0,1.0]$  for weighting factor  $\omega$ . Users should be aware of the oscillatory instability with  $\omega$  near 0.5 and the conditional stability with  $\omega$  near 0.0 for nonlinear problems. Implementation of the load vectors is discussed in the next section.

#### B.2.4 Numerical Implementation of Boundary Conditions

There are four types of boundary conditions that need to be implemented. Among them, Cauchy, Neumann, and variable boundaries are flux boundaries, and Dirichlet boundaries are concentration boundaries. For Cauchy boundaries, the boundary condition in Eq. (2.14) is reproduced here for reference:

$$\theta_\alpha \mathbf{n} \cdot \mathbf{v}_\alpha c_\alpha - \theta_\alpha \mathbf{n} \cdot \mathbf{D}_\alpha \nabla c_\alpha = q_{c\alpha}, \quad (\text{B.23})$$

where  $q_{c\alpha}$  is the prescribed Cauchy boundary flux. Substituting Eq. (B.23) into (B.15), we have

$$L_{c\alpha,i} = - \sum_{e \in N_{ce}} \int_{L_e} B_k^e (q_{c\alpha} - \theta_\alpha \mathbf{n} \cdot \mathbf{v}_\alpha c_\alpha) dL, \quad (\text{B.24})$$

where  $L_{c\alpha,i}$  is the entry of boundary load vector  $\{L_\alpha\}$  associated with Cauchy boundary sides and  $N_{ce}$  are the Cauchy boundary segments. The Cauchy flux is therefore a normal flux to an element side. Similarly, the Neumann boundary flux is reproduced as follows:

$$-n \cdot \theta_\alpha D_\alpha \cdot \nabla c_\alpha = q_{n\alpha}, \quad (\text{B.25})$$

where  $q_{n\alpha}$  is the prescribed Neumann boundary flux. Substituting Eq. (B.25) into (B.15), we have

$$L_{n\alpha,i} = - \sum_{e \in N_{ne}} \int_{L_e} B_k^e q_{n\alpha} dL, \quad (\text{B.26})$$

where  $L_{n\alpha,i}$  is the entry of boundary load vector  $\{L_\alpha\}$  associated with Neumann boundary sides and  $N_{ne}$  are the Neumann boundary segments. The Neumann flux is therefore a normal flux to an element side that is independent of solute concentration. Segments on which variable boundary conditions are prescribed are flow-through boundaries with unknown flow direction a priori. When the flow is directed into a problem domain, the boundary is the same as a Cauchy boundary and the implementation is therefore the same; that is, when  $n \cdot \theta_\alpha v_\alpha \leq 0$ ,

$$L_{v\alpha,i} = - \sum_{e \in N_{ve}} \int_{L_e} B_k^e (q_{v\alpha} - \theta_\alpha n \cdot v_\alpha c_\alpha) dL, \quad (\text{B.27})$$

where  $L_{v\alpha,i}$  is the entry of boundary load vector  $\{L_\alpha\}$  associated with variable boundary sides and  $N_{ve}$  are the variable boundary segments. When the flow is directed out of a problem domain,  $L_{v\alpha,i}$  is set to zero and there is no contribution of the boundary to the global boundary load vector  $\{L_\alpha\}$ . Assembling the flux boundary segments, we arrive at the global boundary vector:

$$\{L_\alpha\} = \{L_{c\alpha}\} + \{L_{n\alpha}\} + \{L_{v\alpha}\}. \quad (\text{B.28})$$

Substituting Eq. (B.28) into (B.22), rearranging the equation so that known vectors are moved to the right hand side and the unknown with the coefficient matrix remain on the left hand side, we have

$$\begin{aligned} [C_\alpha] \{c_\alpha\}_{t+\Delta t} - \sum_{\substack{\beta=1 \\ \beta \neq \alpha}}^N [E_{\alpha\beta}^t] \omega \{c_\beta\}_{t+\Delta t} + \sum_{\substack{\beta=1 \\ \beta \neq \alpha \\ h_\alpha \leq h_\beta}}^N [E_{\alpha\beta}^f] \omega \{c_\beta\}_{t+\Delta t} = \{R_\alpha\} \\ + \sum_{\substack{\beta=1 \\ \beta \neq \alpha}}^N [E_{\alpha\beta}^t] (1-\omega) \{c_\beta\}_t - \sum_{\substack{\beta=1 \\ \beta \neq \alpha \\ h_\alpha \leq h_\beta}}^N [E_{\alpha\beta}^f] (1-\omega) \{c_\beta\}_t, \end{aligned} \quad (\text{B.29})$$

where

$$[C_\alpha] = \frac{[M_\alpha]}{\Delta t} + \omega([D_\alpha] + [K_\alpha]) + \sum_{\substack{\beta=1 \\ \beta \neq \alpha}}^N \omega[E_{\alpha\beta}^t] - \sum_{\substack{\beta=1 \\ \beta \neq \alpha \\ h_\alpha \leq h_\beta}}^N \omega[E_{\alpha\beta}^f] \quad \text{for Lagrangian-} \\ \text{Eulerian approaches ,} \quad (\text{B.30})$$

$$[C_\alpha] = \frac{[M_\alpha]}{\Delta t} + \omega([A_\alpha] + [D_\alpha] + [K_\alpha]) + \sum_{\substack{\beta=1 \\ \beta \neq \alpha}}^N \omega[E_{\alpha\beta}^t] - \sum_{\substack{\beta=1 \\ \beta \neq \alpha \\ h_\alpha \leq h_\beta}}^N \omega[E_{\alpha\beta}^f] \\ \text{for upstream methods,} \quad (\text{B.31})$$

$$\{R_\alpha\} = \frac{[M_\alpha]}{\Delta t} \{c_{\alpha,L}\}_t - (1-\omega)([D_\alpha] + [K_\alpha]) \{c_\alpha\}_t - \sum_{\substack{\beta=1 \\ \beta \neq \alpha}}^N (1-\omega)[E_{\alpha\beta}^t] \{c_\alpha\}_t \\ + \sum_{\substack{\beta=1 \\ \beta \neq \alpha \\ h_\alpha \leq h_\beta}}^N (1-\omega)[E_{\alpha\beta}^f] \{c_\alpha\}_t + \{Q_\alpha\} + \{L_\alpha\} \quad \text{for Lagrangian-Eulerian} \\ \text{approaches ,} \quad (\text{B.32})$$

$$\{R_\alpha\} = \frac{[M_\alpha]}{\Delta t} \{c_\alpha\}_t - (1-\omega)([A_\alpha] + [D_\alpha] + [K_\alpha]) \{c_\alpha\}_t - \sum_{\substack{\beta=1 \\ \beta \neq \alpha}}^N (1-\omega)[E_{\alpha\beta}^t] \{c_\alpha\}_t \\ + \sum_{\substack{\beta=1 \\ \beta \neq \alpha \\ h_\alpha \leq h_\beta}}^N (1-\omega)[E_{\alpha\beta}^f] \{c_\alpha\}_t + \{Q_\alpha\} + \{L_\alpha\} \quad \text{for upstream methods ,} \quad (\text{B.33})$$

The second and third terms, or the summation terms, in Eq. (B.29) are the coupling terms that assemble individual pore-region solute transport equations into a global multiregion matrix equation, as follows:

$$[C]\{c\}_{t+\Delta t} = \{R\} , \quad (\text{B.34})$$

where

$$[C] = \sum_{\alpha=1}^N \left( [C_{\alpha}] - \sum_{\substack{\beta=1 \\ \beta \neq \alpha}}^N \omega [E_{\alpha\beta}^i] + \sum_{\substack{\beta=1 \\ \beta \neq \alpha \\ h_{\alpha} \leq h_{\beta}}}^N \omega [E_{\alpha\beta}^f] \right), \quad (\text{B.35})$$

$$\{R\} = \sum_{\alpha=1}^n \left( \{R_{\alpha}\} + \sum_{\substack{\beta=1 \\ \beta \neq \alpha}}^N (1-\omega) [E_{\alpha\beta}^i] \{c_{\beta}\}_t - \sum_{\substack{\beta=1 \\ \beta \neq \alpha \\ h_{\alpha} \leq h_{\beta}}}^N (1-\omega) [E_{\alpha\beta}^f] \{c_{\beta}\}_t \right), \quad (\text{B.36})$$

For concentration boundary nodes, an identity equation is generated for each of the nodes, with the corresponding entries of the right-hand side vector  $\{R\}$  set to the specified concentration and the associated columns of the coefficient matrix  $[C]$  modified. Matrix Eq. (B.34) is the linearized version of Eq. (2.31) and is ready to be solved by a matrix equation solver. Three matrix solvers are incorporated in MURT, which, together with other modeling options available in MURT, are described in the next two sections.

### B.2.5 Mass Lumping

Mass lumping usually results in a better solution because the diagonal entries of the resultant mass matrix become larger and may help the convergence of iterative solution schemes. One may recall that the mass matrix is an identity matrix if finite difference methods instead of finite element methods are used. By proper scaling, one can reduce the mass matrix to the equivalent of finite difference (Clough 1971). This option is provided in MURT for the mass matrix representing the storage term only. The lumping of the mass matrix  $[M_{\alpha}]$  can be described as follows:

$$M_{\alpha,ij} = \sum_{e \in M_{\alpha}} \left( \sum_{k=1}^{ns} \int_{R_e} B_k^e \left( \theta_{\alpha} + \rho_{b\alpha} \sum_{l=1}^{ns} B_l \frac{ds_{\alpha}}{dc_{\alpha}} \Big|_{c_{\alpha}=c_{ia}} \right) B_l^e dR \right) \quad (\text{B.37})$$

$$ns = 3 \text{ or } 4 \text{ and } i = j$$

and

$$M_{\alpha,ij} = 0, \quad i \neq j. \quad (\text{B.38})$$

## B.2.6 Matrix Solvers for Linear System

Equation (B.34) is a linearized equation of the nonlinear system (2.31) and can be solved either directly or iteratively. To solve (B.34) directly, one applies Gaussian elimination type solution methods. Because the coefficient matrix  $[C]$  is a band sparse matrix, variations of Gaussian methods (e.g., MA28 used in MURT) usually are used to improve computational and computer storage efficiencies. Because the coefficient matrix is first decomposed into a lower and an upper triangle matrix and then solved by back substitution, only one pass is needed. In contrast, iterative methods need a few passes of a solution loop to arrive at a converged solution. The iteration of solution loops is stopped when a user specified convergence criterion is met. Two iterative methods are available in MURT, namely, pointwise iteration and polynomial preconditioned conjugate gradient methods.

The advantage and disadvantage of both solution methods and the size of a user's problem should dictate the choice of matrix solvers. A direct solver, because only one pass is required, usually incurs round-off error. However, because only one pass is needed, direct solvers are also more efficient for midsize to small problems. Large problems are very demanding in terms of computer storage if direct solution methods are to be used. For iterative solution methods, the coefficient matrix is usually compressed and the storage requirement is eased. However, because obtaining an iterative solution involves an initial guess and successively improving the solution until a convergence criterion is met, several passes of solution loops are needed. Therefore, if the problem size is not big enough to have the benefit of saving computer storage, iterative methods are generally not as efficient as direct methods. Nonetheless, round-off error is usually self-corrected in iterative solvers. In general, if computer storage is not a concern or the problem size is relatively small, the user should use direct solvers. For large problems, iterative methods are usually more feasible than direct methods. Users should usually consider the trade-offs between computer storage and computer time to determine the more favorable solution methods.

Equation (2.31) is a system of nonlinear equations and needs to be solved iteratively. A Picard method is used in MURT, where the linearized Eq. (B.34) is solved once during each sweep of a solution loop. After the initial guess or a solution from a previous time step is determined, the solution loop, as follows, is repeated until a convergence criterion is met:

- obtain water contents, fluid velocities, and dispersion coefficients for all of the finite element nodes in the problem domain;
- assemble element matrix and apply boundary conditions;
- solve the linearized equations;
- test convergence; if converged, stop or go to next time step;
- update nodal concentrations as follows:

$$\{c^{k+1}\} = \omega\{c\} + (1-\omega)\{c^k\}, \quad (\text{B.39})$$

where  $\{c^{k+1}\}$  is the new estimate,  $\{c\}$  is the solution obtained during the current iterate, and  $\{c^k\}$  is the previous estimate. The values of weighting factor  $\omega$  can be in the range of 0.0 to 2.0. For  $\omega$  less than 1.0, the solution is underrelaxed and is appropriate for slow convergence due to fluctuation of solutions. For  $\omega$  greater than 1.0, the solution is overrelaxed and is appropriate for slow convergence due to decrease of convergence rates.

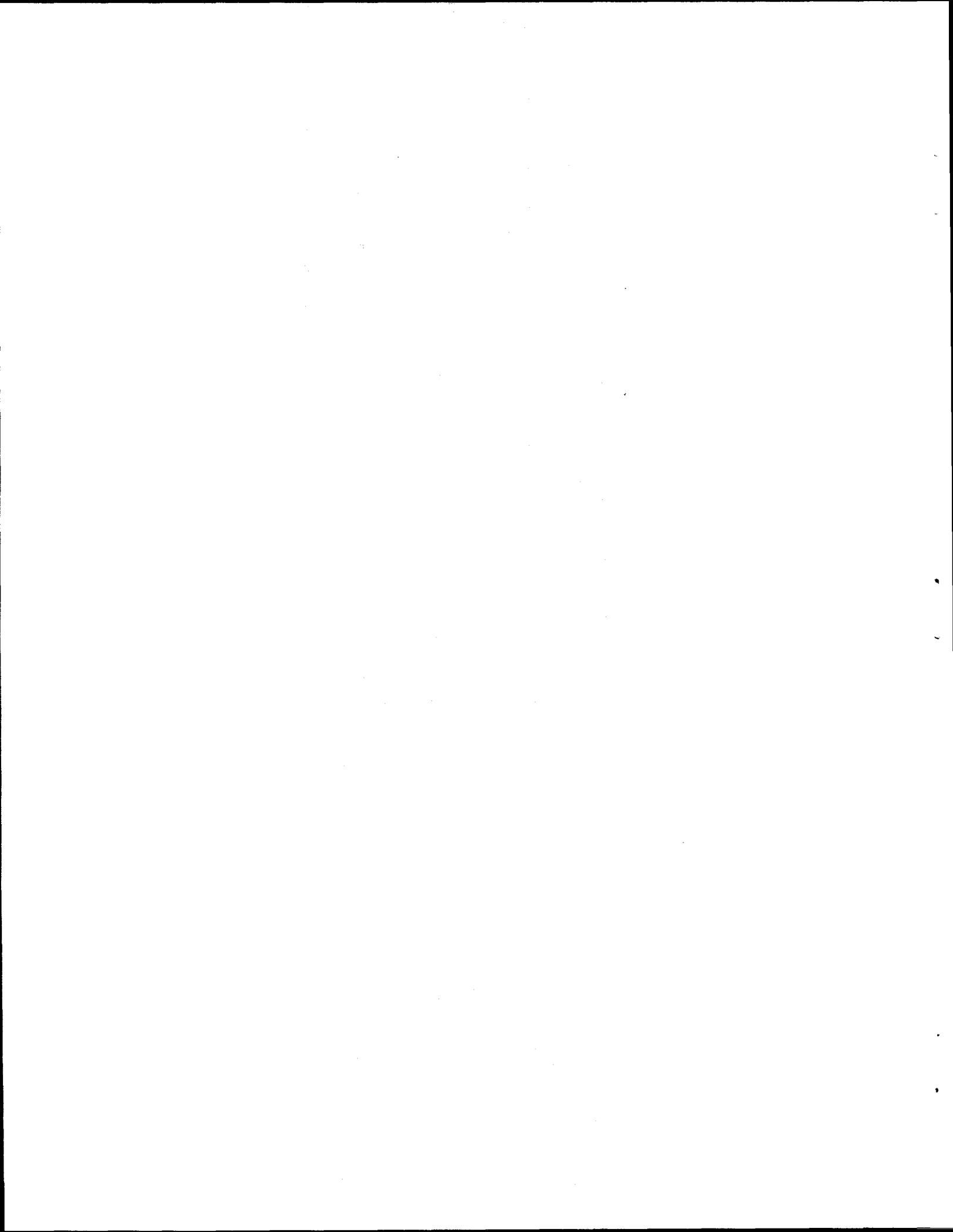
### B.3 REFERENCES

Baptista, A. M. 1987. Solution of advection-dominated transport by Eulerian-Lagrangian methods using the backwards methods of characteristics. Ph.D. Dissertation. M.I.T., Cambridge.

Clough, R. W. 1971. Analysis of structural vibrations and dynamic response. In R. H. Gallagher, Y. Yamada, and J. T. Oden (eds.), Recent Advances in Matrix Methods of Structural Analysis and Design. Proceedings, United States-Japan Seminar on Matrix Methods of Structural Analysis and Design. University of Alabama Press, Huntsville.

Lapidus, L. and G. F. Pinder. 1982. Numerical Solution of Partial Differential Equations in Science and Engineering. Wiley, New York.

Yeh, G. T., J. R. Chang, J. P. Gwo, H. C. Lin, and D. R. Richards. 1994. 3DSALT: A three-dimensional finite element model of density-dependent flow and transport through saturated-unsaturated media. Instruction report HL-94-1. U.S. Army Corps of Engineers.



## APPENDIX C. DATA INPUT GUIDES FOR MURT

To use the computer code MURT, modelers must prepare six input files, in addition to a file that lists the paths of the input files. The input files list contains the paths of the following six data files: (1) array dimension, (2) I/O control, (3) geometry of problem domain and material type corrections, (4) material properties and modeling control and option parameters, (5) initial and boundary conditions, and (6) hydraulic data including pressure heads, water contents, and specific discharge, or darcy velocities. These six input files are defined as:

1. **Array dimension:** This is a self-explanatory template. This file is needed by MURT to allocate memory for dimensional arrays (e.g., Fig. 4.1).
2. **I/O control:** This is also a self-explanatory template. Users need to change the contents only when the default values are not appropriate (e.g., Fig. 4.2).
3. **Geometry of problem domain and material type corrections:** This file contains three data sets, namely, the coordinates and connectivity of the finite element nodes in the problem domain, and material type corrections. Material types are assigned following the pattern prescribed during connectivity autogeneration. Users may find the material types thus generated inappropriate and correction, therefore, needed. This data file is independent of pore regions.
4. **Material properties and modeling control and option parameters:** This file contains specifications of material properties and the control and option parameters that dictate the flow of computation.
5. **Initial and boundary conditions:** This file defines the boundary conditions as specified by the array dimension file. An initial condition is needed for transient simulations. For steady-state simulations, the input initial condition is taken as an initial guess for the solution.
6. **Hydraulic parameters:** Pressure heads, water contents, and specific discharge, or darcy velocities need to be provided through this file if  $KVI = -1$  or  $KVI = 0$ .

Users are referred to Sect. 4 for discussions on preparation of files 1 and 2. The following sections discuss the contents of data files 3 through 5. Examples of these input data files can be found in Sect. 4.1.

### C.1 DATA INPUT GUIDE FOR FILE 3: GEOMETRY OF PROBLEM DOMAIN AND MATERIAL TYPE CORRECTIONS

This file consists of three data sets: nodal coordinates, element connectivity, and material type corrections. The first two data sets are necessary to define the areas of interest within which solute concentrations are to be calculated. Material types can be automatically generated and corrected later, if necessary.

**Data set 1:** Nodal coordinates, free format. In addition to the data set description line, three groups of records are needed for the  $x$  and  $z$  coordinates of each finite element node. Note that this data set is required if and only if  $IGEOM = 0$  (i.e., geometry data are to be

read through a formatted FORTRAN file) and KSTR = 0 (i.e., not a restart job).

1. Data set description line.
2. Data group I: Type of coordinate system and orientation of coordinate axes.  
The default coordinate system is Cartesian and the default horizontal-vertical axes are aligned with the Cartesian coordinate axes. Two parameters are needed:
  - ICYLIND = Integer indicating if cylindrical coordinate system is to be used; 0 = no, 1 = yes.
  - FE = Angle between coordinate axes and the horizontal-vertical axes, in degrees.
3. Data group II:  $x$  coordinates of finite element nodes. In general, NNP records of the following variables are required. However, if the nodal coordinates follow a pattern, they can be automatically generated in MURT. A similar group of records is used to input  $z$  coordinates.
  - NI = Node number of the first node in the sequence.
  - NSEQ = The number of subsequent nodes to be automatically generated.
  - NAD = Increment of node number for each of the NSEQ subsequent nodes.
  - XNI =  $x$  coordinate of node NI (L).
  - XAD = Increment of  $x$ -coordinate for each of the NSEQ subsequent nodes (L).
  - XRD = Percentage of the increase of the increment over its preceding increment (decimal point). If XRD = 0, all increment XADs are the same. If XRD > 0, the first increment is  $XAD(1 + XRD)$ ; the second increment is  $XAD(1 + XRD)^2$ ; etc.

**Note:** A set of three integer 0s and three real 0s must be added to signal the end of this data group.
4. Data group III:  $z$  coordinates of the finite element nodes. In general, NNP records of the following variables are required. However, if the nodal coordinates follow a pattern, they can be automatically generated in MURT.
  - NI = Node number of the first node in the sequence.
  - NSEQ = The number of subsequent nodes to be automatically generated.
  - NAD = Increment of node number for each of the NSEQ subsequent nodes.
  - ZNI =  $z$  coordinate of node NI (L).
  - ZAD = Increment of  $z$  coordinate for each of the NSEQ subsequent nodes (L).
  - ZRD = Percentage of the increase of the increment over its preceding increment (decimal point). If ZRD = 0, all increment ZADs are the same. If ZRD > 0, the first increment is  $ZAD(1 + ZRD)$ ; the second increment is  $ZAD(1 + ZRD)^2$ ; etc.

**Note:** A set of three integer 0s and three real 0s must be added to signal the end of this data group.

**Data set 2:** Element incidence and node connectivity, free format. In addition to the data description line, one data group is needed. Element connectivity can be generated automatically by giving the patterns of node numbers associated with elements. Note that

this data set is required if and only if IGEOM = 0 (i.e., geometry data are to be read through a formatted FORTRAN file) and KSTR = 0 (i.e., not a restart job).

1. Data set description line.
2. Data group I: Element connectivity. Eight parameters are needed to generate incidence of the elements automatically. They are defined as
  - MI = Global element number.
  - IE(MI,1) = Global node number of the first node of element MI.
  - IE(MI,2) = Global node number of the second node of element MI.
  - IE(MI,3) = Global node number of the third node of element MI.
  - IE(MI,4) = Global node number of the fourth node of element MI. Input zero for triangular elements.
  - IE(MI,5) = Material type to be applied to element MI.
  - MODL = Number of elements in the direction with node numbers increasing most rapidly.
  - NLAY = Number of elements in the direction with node numbers increasing least rapidly.

**Note:** IE(MI,1) through IE(MI,4) are numbered beginning with the lower left corner and progressing around the element in a counterclockwise direction. For a rectangular block of elements, it is only necessary to specify the first element, the width MODL, and the length NLAY, where MODL and NLAY are measured in elements. For example, the following problem domain is a regular finite element mesh that can be automatically generated by this method:

	8				24
4	<u>3</u> 7				<u>15</u> 23
3	<u>2</u> 6				<u>14</u> 22
2	<u>1</u> 5				<u>13</u> 21
1					

1 : element number

1 : node number

The object is rectangular with width MODL = 3 on two opposite sides and length NLAY = 5 on the other two opposite sides. To generate definitions of elements 2 through 15 automatically, including both the incidence and material type, only one card is necessary, as follows,

IE(MI,1)	IE(MI,2)	IE(MI,3)	IE(MI,4)	IE(MI,5)	MODL	NLAY
1	5	6	2	1	3	5

Although all elements of this example will be assumed to contain the same material type, IE(MI,5) = 1, this situation can easily be changed by material correction described in the next data set.

**Data set 3:** Material type correction, free format. Note that this data set is required only if NCM (number of material type corrections) > 0. One data group, in addition to the data set description line, is required.

1. Data set description line.
2. Data group I: In general, NCM records of the following variables are required. However, they can also be automatically generated by specifying the pattern of material correction. The variables are defined as
  - MI = Global element number of the first element in the sequence.
  - NSEQ = NSEQ subsequent elements that will be generated automatically.
  - MAD = Increment of element number for each of the NSEQ subsequent elements.
  - MITYP = Type of material correction for element MI.
  - MTYPAD = Increment of MITYP for each of the NSEQ subsequent elements.

**Note:** A set of five integer 0s must be added to signal the end of this data group.

## **C.2 DATA INPUT GUIDE FOR FILE 4: CONTROL AND OPTION PARAMETERS AND MATERIAL PROPERTIES**

This input file contains the following four data sets: (1) problem identification, (2) universal control parameters, (3) control parameters for simulations and output to disk files, and (4) material properties.

**Data set 1:** Problem identification. Two lines are required to define a problem set.

Line 1: Problem set number, free format.

Line 2: Problem set description, must be less than 80 characters, with FORTRAN format = A80.

**Data set 2:** Universal control parameters. Two lines are needed, free format.

Line 1: Data set description.

Line 2: Four universal control parameters defined as

- IBUG = Is diagnostic information desired? 0 = no, 1 = yes.
- ILUMP = Is mass lumping to be used? 0 = no, 1 = yes.
- IWET = Weighting function control; 0 = Galerkin weighting; 1 = upstream weighting.
- IOPTIM = Optimization factor indicator; 1 = optimization factor is to be computed, 0 = optimization factor to be set to -1.0, 0.0, or 1.0.

**Data set 3:** Control parameters for simulations and output to disk files. In addition to the data set description line, six groups of data (free format, unless otherwise specified) are needed.

1. Data set description line.
2. Data group I: Simulation control parameters. Four control parameters are

needed and are defined as

- KSS = Steady-state control; 0 = steady-state solution desired, 1 = transient-state solution desired.
  - NITER = Maximum number of iterations allowed for solving the nonlinear equations.
  - IMSOLV = Matrix solver option parameter; 1 = pointwise, 2 = polynomial preconditioned conjugate gradient, and 3 = sparse matrix solver MA28.
  - NPITER = Maximum number of iterations allowed for solving the linear equations if pointwise or polynomial preconditioned conjugate gradient matrix solver is used.
3. Data group II: Time-advancing parameters. Five time-step and iteration parameters defined as follows are needed. These parameters are disregarded if steady-state solutions are to be computed.
- DELT = Initial time-step size (T).
  - CHNG = Percentage of change in time-step size in each of the subsequent time increments (dimensionless in decimal point).
  - DELMAX = Maximum value of time-step size allowed (T).
  - TMAX = Maximum simulation time (T).
  - W = Time derivative weighting factor; 0.5 = Crank-Nicholson central or mid-difference, 1.0 = backward difference, 0.0 = forward difference.
4. Data group III: Control parameters for solving nonlinear and linear equations. Four parameters are needed and are defined as
- OME = Iteration parameter for solving the nonlinear equation; 0.0 - 1.0 = under-relaxation, 1.0 = exact relaxation, 1.0 - 2.0 = over-relaxation.
  - OMI = Iteration parameter for solving the linear matrix equation by pointwise iteration method; 0.0 - 1.0 = underrelaxation, 1.0 = exact relaxation, 1.0 - 2.0 = overrelaxation.
  - TOLA = Steady-state convergence criteria (relative concentration).
  - TOLB = Transient-state convergence criteria (relative concentration).
5. Data group IV: Printer control parameters, FORTRAN format = 80I1, each defined as
- KPR0 = Printer control for steady-state and initial condition.
  - KPR(I) = Printer control for the Ith (I = 1, 2, ..., NTI) time step similar to KPR0. Computational results are printed according to the following rule: KPR(I) = 0, print nothing; KPR(I) = 1, print FLOW, FRATE, and TFLOW; KPR(I) = 2, print KPR(I) = 1 plus solute concentrations; KPR(I) = 3, print KPR(I) = 2 plus material fluxes.
6. Data group V: Storage control parameters, FORTRAN format = 80 I1, each defined as
- KDSK0 = Auxiliary storage control for steady-state and initial condition; 0 = no storage, 1 = store on the designated logical units.
  - KDSK(I) = Auxiliary storage control for the Ith (I = 1, 2, ..., NTI) time step similar to KDSK0.
7. Data group VI: Times to change time-step sizes. TDTCH(I), I = 1, 2, ..., NDTCHG are times at which time-step size is to be reset to DELT. NDTCHG

is the number of times to reset time-step size, which is specified in the array dimension file.

**Data set 4:** Material properties, free format. In addition to the data set description line, two data groups are needed to define the porous media within the problem domain.

1. Data set description line.
2. Data group I: Adsorption isotherm indicator; 1 = linear, 2 = Freundlich, 3 = Langmuir.
3. Data group II: The number of material properties (NMPPM) that a modeler can specify could be up to 12. If one put all of these in a line, the total number of lines is therefore  $NMAT \cdot NDF$ , with the index  $IDF = 1, \dots, NDF$  varying before  $I = 1, \dots, NMAT$ . In other words, material properties of all of the pore regions in a medium must be defined before those of the next medium are given.  $NMAT$  and  $NDF$  are the number of soil types and the number of pore regions, respectively.
  - $PROP(1, IDF, I) = K_d$ , distribution coefficient for linear isotherms ( $L^3/M$ ), or  $K$ , the functional coefficient of Freundlich and Langmuir isotherms. See Sect. 2 for descriptions of adsorption isotherms.
  - $PROP(2, IDF, I) =$  bulk density of the porous medium ( $M/L^3$ ).
  - $PROP(3, IDF, I) =$  longitudinal dispersivity ( $L$ ).
  - $PROP(4, IDF, I) =$  transverse dispersivity ( $L$ ).
  - $PROP(5, IDF, I) =$  first-order radioactive decay constant in the fluid and solid phases ( $T^{-1}$ ).
  - $PROP(6, IDF, I) =$  molecular diffusion coefficient of the solute in the fluid ( $L^2/T$ ).
  - $PROP(7, IDF, I) =$  tortuosity of the porous medium (dimensionless).
  - $PROP(8, IDF, I) =$  first-order biodegradation coefficient in the fluid phase ( $T^{-1}$ ).
  - $PROP(9, IDF, I) =$  first-order biodegradation coefficient in the solid phase ( $T^{-1}$ ).
  - $PROP(10, IDF, I) =$  order of the Freundlich isotherm (dimensionless).
  - $PROP(11, IDF, I) =$  maximum concentration allowed on the solid surface for Langmuir isotherms ( $M/L^3$ ).
  - $PROP(12, IDF, I) =$  decay constant for the first-order mass transfer coefficient ( $T^{-1}$ ).

**Data set 5:** Advective and diffusive mass transfer coefficients, free format. Two data groups, in addition to a data set description line, are needed to specify the mass transfer coefficients.

1. Data set description line.
2. Data group I: Advective mass transfer coefficients between pore regions for a material type must occupy one single line. The order of input is to vary  $KDF = 1, \dots, NDF$  before  $IDF = 1, \dots, NDF$ . The mass transfer coefficient  $EXCF(KDF, IDF, I)$  stands for mass transfer coefficient in soil type  $I$ , from pore region  $KDF$  to pore region  $IDF$ . The user must make sure that the mass

transfer coefficients between every two pore regions are symmetric to guarantee mass conservation across internal pore region boundaries; that is,  $EXCF(KDF, IDF, I) = EXCF(IDF, KDF, I)$ . The mass transfer coefficients therefore form a symmetric matrix that needs to be repetitively input NMAT times, where NMAT = the number of material types.

3. Data group II: Diffusive mass transfer coefficients between pore regions for a material type must occupy one single line. The order of input is to vary KDF = 1, ..., NDF before IDF = 1, ..., NDF. The mass transfer coefficient  $EXCT(KDF, IDF, I)$  stands for mass transfer coefficient in soil type I, from pore region KDF to pore region IDF. The user must make sure that the mass transfer coefficients between every two pore regions are symmetric to guarantee mass conservation across internal pore region boundaries, i.e.,  $EXCT(KDF, IDF, I) = EXCT(IDF, KDF, I)$ . The mass transfer coefficients therefore form a symmetric matrix that needs to be repetitively input NMAT times, where NMAT = the number of material types.

### C.3 DATA INPUT GUIDE FOR FILE 5: INITIAL CONDITION, EXTERNAL SOURCES/SINKS, AND BOUNDARY CONDITIONS

**Data set 1:** Initial condition, free format. Besides the data set description line, one data group is needed. An automatic generation procedure is available in MURT. By specifying the number of nodal concentrations to be generated and giving the pattern of the concentrations, users can generate the initial condition of the entire problem domain in a few lines.

1. Data set description line.
2. Data group I: This data group must be repeated NDF (the number of pore regions) times. Six parameters are needed to generate one or more concentrations:
  - NI = Global node number of the first node in the sequence.
  - NSEQ = NSEQ subsequent nodes will be generated automatically.
  - NAD = Increment of node number for each of the NSEQ nodes.
  - HNI = Initial or pre-initial concentration of node NI ( $ML^{-3}$ ).
  - HAD = Unit increment of initial or pre-initial concentration for each of the NSEQ nodes ( $ML^{-3}$ ).
  - HRD = Percentage (in decimal point) of increment for each of the NSEQ nodes.

**Note:** A set of three integer 0s and three real 0s must be added to signal the end of this data group.

**Data set 2:** Non-point or distributed sources/sinks, free format. A non-point source/sink is defined by assigning source/sink profiles to elements, which occupy conceptually an area rather than a discrete point in space. This data set is needed only when the number of non-point source/sink elements as specified in the array dimension file is greater than zero. If the number of non-point sources/sinks is greater than zero, users need to specify

source/sink profiles and source/sink types associated with each of the non-point source/sink elements. Four data groups plus a data set description line are needed.

1. Data set description line.
2. Data group I: Source profile indicator; 0 = tabulated, 1 = analytical. Users need to supply source profiles in subroutine *essfct* and give parameters of the analytical profiles in the next data group for the latter choice.
3. Data group II: Source/sink profiles. For analytical profiles, only functional parameters are required (free format). For tabulated profiles, at least two data points of (time, flow strength, solute concentration) must be given to completely define a profile. The user must give exactly the same numbers of profiles and data points as those specified in the array dimension file. The (time, flow strength, solute concentration) = (TSOSF(i,j), SOSFL(i,j), SOSFC(i,j)) data point is defined as
  - TSOSF(i,j) = Time of the *i*th data point in the *j*th profile (T),
  - SOSFL(i,j) = Source/sink flow strength of the *i*th data point in the *j*th profile ( $L^3/T/L^2/L$ ).
  - SOSFC(i,j) = Source/sink solute concentration of the *i*th data point in the *j*th profile ( $ML^{-3}$ ).
4. Data group III: Global element numbers of source/sink elements. The number of elements given here must be consistent with that specified in the array dimension file.
5. Data group IV: Source/sink profile types of source/sink elements. An automatic generation mechanism is available in MURT. Since an element consists of NDF pore regions, this data group must be repeated NDF times. One can assign source/sink types to the source/sink elements by giving the patterns of the profile types as
  - MI = Global element number of the first element in the sequence.
  - NSEQ = Number of elements to be generated automatically.
  - MAD = Increment of element number for each of the NSEQ elements.
  - MITYP = Source type of element MI.
  - MTYPAD = Increment of MITYP for each of the NSEQ elements.

**Note:** A set of five integer 0s must be added to signal the end of this data group.

**Data set 3:** Point/well sources/sinks, free format. A point source/sink is defined by assigning source/sink profiles to a nodal point, which occupy conceptually a discrete point in space. This data set is needed only when the number of point/well source/sink nodes as specified in the array dimension file is greater than zero. If the number of point sources/sinks is greater than zero, users need to specify source/sink profiles and source/sink types associated with each of the point source/sink nodes. Four data groups, in addition to the data description line, are needed.

1. Data set description line.
2. Data group I: Source profile indicator; 0 = tabulated, 1 = analytical. Users need to supply source profiles in subroutine *wssfct* and give parameters of the analytical profiles in the next data group for the latter choice.

3. Data group II: Source/sink profiles. For analytical profiles, only functional parameters are required (free format). For tabulated profiles, at least two data points of (time, flow strength, solute concentration) must be given to completely define a profile. The user must give exactly the same numbers of profiles and data points as those specified in the array dimension file. The (time, flow strength, solute concentration) = (TWSSF(i,j), WSSFL(i,j), WSSFC(i,j)) data point is defined as
  - TWSSF(i,j) = Time of the *i*th data point in the *j*th profile (T).
  - WSSFL(i,j) = Source/sink flow strength of the *i*th data point in the *j*th profile ( $L^3/T/L^2/L$ ).
  - WSSFC(i,j) = Source/sink solute concentration of the *i*th data point in the *j*th profile ( $ML^{-3}$ ).
4. Data group III: Global node numbers of source/sink nodes. The number of nodes given here must be consistent with that specified in the array dimension file.
5. Data group IV: Source/sink profile types of source/sink nodes. An automatic generation mechanism is available in MURT. Since a node consists of NDF pore regions, this data group must be repeated NDF times. One can assign source/sink types to the source/sink nodes by giving the patterns of the profile types as
  - NI = Global node number of the first node in the sequence.
  - NSEQ = Number of nodes to be generated automatically.
  - NAD = Increment of node number for each of the NSEQ nodes.
  - NITYP = Source type of node NI.
  - NTYPAD = Increment of NITYP for each of the NSEQ nodes.

**Note:** A set of five integer 0s must be added to signal the end of this data group.

**Data set 4:** Run-in/seep-out or variable boundary (VB) conditions, free format. Variable boundary is defined by assigning run-in concentrations at boundary segments where direction of flow is not known a priori. This data set is needed only when the number of VB element sides is greater than zero, as defined in the array dimension file. In addition to the data description line, five data groups are needed.

1. Data set description line.
2. Data group I: Boundary profile indicator; 0 = tabulated, 1 = analytical. Either tabulated or functional run-in concentration profiles can be used in MURT. If the number 1 is specified, the user needs to supply the profiles in subroutine *vbvft* and give parameters of the analytical profiles in the next data group. If the number 0 is specified, tabulated run-in concentration profiles must be given below.
3. Data group II: Run-in concentration profiles. For analytical profiles, only functional parameters are required (free format). For tabulated profiles, at least two data points of (time, concentration) must be given to completely define a profile. The user must give exactly the same numbers of profiles and data points as those specified in the array dimension file. The (time, concentration)

- = (TRF(i,j), RF(i,j)) data point is defined as
- TRF(i,j) = Time of the *i*th data point in the *j*th profile (T).
  - RF(i,j) = Run-in concentration of the *i*th data point in the *j*th profile (M/L<sup>-3</sup>).
4. Data group III: Global node number of VB nodes. An automatic generation mechanism is available to generate the node numbers. Given the first nodes, the number of nodes to be generated, and the pattern of the nodes, the mechanism generates an array of VB nodes using the following five parameters:
- NI = Compressed VB node number of the first node in the sequence.
  - NSEQ = Number of nodes to be generated automatically.
  - NIAD = Increment of NI for each of the NSEQ nodes.
  - NODE = Global node number of node NI.
  - NODAD = Increment of NODE for each of the NSEQ nodes.
- Note:** A set of five integer 0s must be added to signal the end of this data group.
5. Data group IV: Run-in concentration profile types of VB elements. Since each VB element consists of NDF degrees of freedom and profile types of pore regions are allowed to be different, this group of data must be repeated for each of the pore regions. An automatic generation mechanism is available in MURT. One can assign run-in concentration profile types to the VB elements by giving the patterns of the profile types as
- NI = Compressed element number of the first element in the sequence.
  - NSEQ = Number of elements to be generated automatically.
  - NAD = Increment of element number for each of the NSEQ elements.
  - NITYP = Run-in concentration profile type of element NI.
  - NTYPAD = Increment of NITYP for each of the NSEQ elements.
- Note:** A set of five integer 0s must be added to signal the end of this data group.
6. Data group V: Run-in/seep-out element sides. A mechanism is available in MURT to automatically generate the element sides by specifying the following parameters:
- MI = Compressed VB element side number of the first element side in a sequence.
  - NSEQ = Number of subsequent VB element sides to be generated automatically.
  - M = Global element number to which *M*th element side belongs.
  - IS1 = Compressed VB node number of the first node of element side MI.
  - IS2 = Compressed VB node number of the second node of element side MI.
  - MIAD = Increment of MI for each of the NSEQ subsequent VB element sides.
  - MAD = Increment of M for each of the NSEQ subsequent VB element sides.
  - IS1AD = Increment of IS1 for each of the NSEQ subsequent VB element sides.

- IS2AD = Increment of IS2 for each of the NSEQ subsequent VB element sides.

**Note:** A set of nine integer 0s must be added to signal the end of this data group.

**Data set 5:** Dirichlet boundary condition, free format. A Dirichlet node, however, can be either internal or external. Internal Dirichlet nodes can be used to define injection wells maintained at specified concentrations, while external Dirichlet nodes are usually used to define intersection with surface sources. This data set is needed only when the number of Dirichlet nodes, as specified in the array dimension file, is greater than zero. This data set consists of the following four groups of data and a data set description line:

1. Data set description line.
2. Data group I: Boundary profile indicator; 0 = tabulated, 1 = analytical. Either tabulated or functional Dirichlet profiles can be used in MURT. If the number 1 is specified, the user needs to supply the profiles in subroutine *dbvfct* and give parameters of the analytical profiles in the next data group. If the number 0 is specified, tabulated Dirichlet profiles must be given below.
3. Data group II: Dirichlet concentration profiles. For analytical profiles, only functional parameters are required (free format). For tabulated profiles, at least two data points of (time, concentration) must be given to completely define a profile. The user must give exactly the same numbers of profiles and data points as those specified in the array dimension file. The (time, concentration) = (TCDBF(i,j), CDBF(i,j)) components are defined as follows:
  - TCDBF(i,j) = Time of the *i*th data point in the *j*th profile (T).
  - CDBF(i,j) = Concentration of the *i*th data point in the *j*th profile (ML<sup>-3</sup>).
4. Data group III: Global node numbers of Dirichlet nodes. The number of Dirichlet nodes listed here must be consistent with that specified in the array dimension file.
5. Data group IV: Dirichlet concentration profile types of Dirichlet nodes. Since each Dirichlet node consists of NDF degrees of freedom and profile types of pore regions are allowed to be different, this group of data must be repeated for each of the pore regions. An automatic generation mechanism is available in MURT. One can assign Dirichlet profile types to Dirichlet nodes by giving the patterns of the profile types as
  - NI = Compressed node number of the first node in the sequence.
  - NSEQ = Number of nodes to be generated automatically.
  - NAD = Increment of node number for each of the NSEQ nodes.
  - NITYP = Dirichlet profile type of node NI.
  - NTYPAD = Increment of NITYP for each of the NSEQ nodes.

**Note:** A set of five integer 0s must be added to signal the end of this data group.

**Data set 6:** Cauchy boundary condition, free format. A Cauchy boundary is defined by assigning mass flux to a boundary segment. This data set is needed only when the number of Cauchy elements, as specified in the array dimension file, is greater than zero.

This data set contains the following five groups of data plus a data set description line:

1. Data set description line.
2. Data group I: Boundary profile indicator; 0 = tabulated, 1 = analytical. Either tabulated or functional Cauchy profiles can be used in MURT. If the number 1 is specified, the user needs to supply the profiles in subroutine *cbvfct* and give parameters of the analytical profiles in the next data group. If the number 0 is specified, tabulated Cauchy profiles must be given below.
3. Data group II: Cauchy flux profiles. For analytical profiles, only functional parameters are required (free format). For tabulated profiles, at least two data points of (time, flux) must be given to completely define a profile. The user must give exactly the same number of profiles and data points as those specified in the array dimension file. The (time, flux) = (TQCBF(i,j), QCBF(i,j)) data point components are defined as follows:
  - TQCBF(i,j) = Time of the *i*th data point in the *j*th profile (T).
  - QCBF(i,j) = Cauchy flux of the *i*th data point in the *j*th profile ( $ML^{-3}/TL^2$ ).
4. Data group III: Global node numbers of Cauchy nodes. The number of Cauchy nodes listed here must be consistent with that specified in the array dimension file.
5. Data group IV: Cauchy boundary flux profile types of Cauchy nodes. Since each Cauchy node consists of NDF degrees of freedom and profile types of pore regions are allowed to be different, this group of data must be repeated for each of the pore regions. An automatic generation mechanism is available in MURT. One can assign Cauchy profile types to the Cauchy nodes by giving the patterns of the profile types as
  - NI = Compressed element number of the first element in the sequence.
  - NSEQ = The number of nodes to be generated automatically.
  - NAD = Increment of node number for each of the NSEQ nodes.
  - NITYP = Cauchy flux type of node NI.
  - NTYPAD = Increment of NITYP for each of the NSEQ nodes.

**Note:** A set of five integer 0s must be added to signal the end of this data group.
6. Data group V: Cauchy boundary element sides. A mechanism is available in MURT to automatically generate the element sides by specifying the following parameters:
  - MI = Compressed Cauchy element side number of the first element side in a sequence.
  - NSEQ = Number of subsequent Cauchy element sides to be generated automatically.
  - M = Global element number to which *M*th element side belong to.
  - IS1 = Compressed Cauchy node number of the first node of element side MI.
  - IS2 = Compressed Cauchy node number of the second node of element side MI.
  - MIAD = Increment of MI for each of the NSEQ subsequent Cauchy element sides.

- MAD = Increment of M for each of the NSEQ subsequent Cauchy element sides.
- IS1AD = Increment of IS1 for each of the NSEQ subsequent Cauchy element sides.
- IS2AD = Increment of IS2 for each of the NSEQ subsequent Cauchy element sides.

**Note:** A set of nine integer 0s must be added to signal the end of this data group.

**Data set 7:** Neumann boundary condition, free format. A Neumann boundary is specified by assigning mass flux to boundary segments where flux is contributed mostly by concentration gradient. This data set is needed only when the number of Neumann elements, as specified in the array dimension file, is greater than zero. This data set contains the following five groups of data plus a data set description line:

1. Data set description line.
2. Data group I: Boundary profile indicator; 0 = tabulated, 1 = analytical. Either tabulated or functional Neumann profiles can be used in MURT. If the number 1 is specified, the user needs to supply the profiles in subroutine *nbvft* and give parameters of the analytical profiles in the next data group. If the number 0 is specified, tabulated Neumann profiles must be given below.
3. Data group II: Neumann flux profiles. For analytical profiles, only functional parameters are required (free format). For tabulated profiles, at least two data points of (time, flux) must be given to completely define a profile. The user must give exactly the same number of profiles and data points as those specified in the array dimension file. The (time, flux) = (TQNB*F*(i,j), QNB*F*(i,j)) data point components are defined as
  - TQNB*F*(i,j) = Time of the *i*th data point in the *j*th profile (T).
  - QNB*F*(i,j) = Neumann flux of the *i*th data point in the *j*th profile (ML<sup>-3</sup>/TL<sup>2</sup>)
4. Data group III: Global node numbers of Neumann nodes. The number of Neumann nodes listed here must be consistent with that specified in the array dimension file.
5. Data group IV: Neumann boundary flux profile types of Neumann nodes. Since each Neumann node consists of NDF degrees of freedom and profile types of pore regions are allowed to be different, this group of data must be repeated for each of the pore regions. An automatic generation mechanism is available in MURT. One can assign Neumann profile types to the Neumann nodes by giving the patterns of the profile types as follows:
  - NI = Global node number of the first node in the sequence.
  - NSEQ = Number of nodes to be generated automatically.
  - NAD = Increment of node number for each of the NSEQ nodes.
  - NITYP = Neumann flux type of node NI.
  - NTYPAD = Increment of NITYP for each of the NSEQ nodes.

**Note:** A set of five integer 0s must be added to signal the end of this data group.

6. Data group V: Neumann boundary element sides. A mechanism is available in MURT to automatically generate the element sides by specifying the following parameters:

- MI = Compressed Neumann element side number of the first element side in a sequence.
- NSEQ = Number of subsequent Neumann element sides to be generated automatically.
- M = Global element number to which *MI*th element side belong to.
- IS1 = Compressed Neumann node number of the first node of element side MI.
- IS2 = Compressed Neumann node number of the second node of element side MI.
- MIAD = Increment of MI for each of the NSEQ subsequent Neumann element sides.
- MAD = Increment of M for each of the NSEQ subsequent Neumann element sides.
- IS1AD = Increment of IS1 for each of the NSEQ subsequent Neumann element sides.
- IS2AD = Increment of IS2 for each of the NSEQ subsequent Neumann element sides.

**Note:** A set of nine integer 0s must be added to signal the end of this data group.

#### C.4 DATA INPUT GUIDE FOR FILE 6: HYDRAULIC PARAMETERS

This file contains four data sets that describe hydraulic parameter input methods and hydraulic parameters including pore-region pressure heads, water contents, and darcy velocities. These parameters can be provided by this file or by a FORTRAN unformatted file created by the subsurface multiregion flow code MURF (Gwo et al. 1994). Note that if the modeler chooses to use formatted hydraulic parameters in this file for transient simulations, data sets 2 to 4, together as a single unit, must be repeated  $NTI + 1$  times, including the initial time step.

**Data set 1:** This data set consists of a single line that determines the value of the input flag KVI as follows:

1. KVI = -1; only pore-region darcy velocities and water contents are provided in this file. The flow fields are under steady state.
2. KVI = 0; pore-region pressure heads, darcy velocities, and water contents are specified through this file. The flow fields are under transient state.
3. KVI = 1; pore region pressure heads, darcy velocities, and water contents are given in a FORTRAN unformatted file provided by modelers or created by the subsurface multiregion flow code MURF. The flow fields are under steady state.
4. KVI = 2; pore region pressure heads, darcy velocities, and water contents are

given in a FORTRAN unformatted file provided by modelers or created by the subsurface multiregion flow code MURF. The flow fields are under transient state.

**Data set 2:** Pore-region pressure heads, free format. This data set is needed only when  $KVI = 0$ . An automatic generation procedure is available in MURF. By specifying the number of nodal pressure heads to be generated and giving the patterns of the pressure heads, the users can generate pore-region pressure heads of the entire problem domain in a few lines. This data set contains a data set description line and a data group that specifies the patterns of autogeneration.

1. Data set description line.
2. Data group I: This data group must be repeated NDF (the number of pore regions) times. Six parameters are needed to generate one or more pressure heads:

- NI = Global node number of the first node in the sequence.
- NSEQ = NSEQ subsequent nodes will be generated automatically.
- NAD = Increment of node number for each of the NSEQ nodes.
- HNI = Pressure head of node NI (L).
- HAD = Unit increment of pressure head for each of the NSEQ nodes (L).
- HRD = Percentage (in decimal point) of increment for each of the NSEQ nodes.

**Note:** A set of three integer 0s and three real 0s must be added to signal the end of this data group.

**Data set 3:** Pore-region darcy velocities, free format. This data set is needed only when  $KVI = 0$  or  $KVI = -1$ . An automatic generation procedure is available in MURF. By specifying the number of nodal darcy velocities to be generated and giving the patterns of the velocities, users can generate velocity fields of the entire problem domain in a few lines. This data set contains a data set description line and a data group that specifies the patterns of autogeneration.

1. Data set description line.
2. Data group I: This data group must be repeated NDF (the number of pore regions) times. Seven parameters are needed to generate one or more darcy velocities:

- NI = Global node number of the first node in the sequence.
- NSEQ = NSEQ subsequent nodes will be generated automatically.
- NAD = Increment of node number for each of the NSEQ nodes.
- VXNI = Darcy velocity of node NI in  $x$  direction (L/T).
- VZNI = Darcy velocity of node NI in  $z$  direction (L/T).
- VXAD = Increment of darcy velocity in  $x$  direction for each of the subsequent NSEQ nodes (L/T).
- VZAD = Increment of darcy velocity in  $z$  direction for each of the subsequent NSEQ nodes (L/T).

**Note:** A set of three integer 0s and four real 0s must be added to signal the end of this data group.

**Data set 4:** Pore-region water contents, free format. This data set is needed only when  $KVI = 0$  or  $KVI = -1$ . An automatic generation procedure is available in MURF. By specifying the number of elemental water contents to be generated and giving the patterns of the water contents, users can generate pore-region water contents of the entire problem domain in a few lines. This data set contains a data set description line and a data group that specifies the patterns of autogeneration.

1. Data set description line.
2. Data group I: This data group must be repeated NDF times. Five parameters are needed to generate one or more water contents:
  - NI = Global element number of the first element in the sequence.
  - NSEQ = NSEQ subsequent elements will be generated automatically.
  - NAD = Increment of element number for each of the NSEQ nodes.
  - THNI = Water content of element NI (dimensionless).
  - THNIAD = Increment of water content for each of the subsequent NSEQ elements (L/T).

**Note:** A set of three integer 0s and four real 0s must be added to signal the end of this data group.

## C.5 REFERENCES

Gwo, J. P., P. M. Jardine, G. T. Yeh, and G. V. Wilson. 1994. MURF user's guide: a finite element model of multiple-pore-region flow through variably saturated subsurface media. ORNL/GWPO-011. Oak Ridge National Laboratory, Tenn.

## INTERNAL DISTRIBUTION

- |        |                  |        |                             |
|--------|------------------|--------|-----------------------------|
| 1.     | L. D. Bates      | 28.    | R. H. Ketelle               |
| 2.     | C. A. Bednarz    | 29.    | K. L. Kliewer               |
| 3.     | B. A. Berven     | 30.    | A. J. Kuhaida               |
| 4.     | H. L. Boston     | 31.    | J. M. Loar                  |
| 5.     | W. D. Brickeen   | 32.    | G. Mahithakumar             |
| 6.     | R. B. Clapp      | 33.    | G. R. Moline                |
| 7.     | J. H. Cushman    | 34.    | G. K. Moore                 |
| 8.     | V. H. Dale       | 35.    | D. E. Reichle               |
| 9.     | E. F. D'Azevedo  | 36.    | C. T. Rightmire             |
| 10.    | T. O. Early      | 37.    | C. H. Romine                |
| 11.    | D. E. Fowler     | 38.    | M. E. Sale                  |
| 12.    | D. L. Ganett     | 39.    | W. E. Sanford               |
| 13.    | S. B. Garland    | 40.    | F. E. Sharples              |
| 14.    | R. L. Graham     | 41.    | S. H. Stow                  |
| 15.    | D. C. Griffith   | 42.    | L. E. Toran                 |
| 16-20. | J. P. Gwo        | 43.    | S. D. VanHoesen             |
| 21.    | S. G. Hildebrand | 44.    | M. W. Yambert               |
| 22.    | D. D. Huff       | 45-59. | Central Research Library    |
| 23.    | G. K. Jacobs     | 60-61. | ESD Library                 |
| 24.    | B. R. James      | 62.    | Laboratory Records Dept.    |
| 25.    | P. M. Jardine    | 63.    | Laboratory Records, ORNL-RC |
| 26.    | P. Kanciruk      | 64.    | ORNL Patent Section         |
| 27.    | C. Kendrick      | 65.    | ORNL Y-12 Technical Library |

## EXTERNAL DISTRIBUTION

66. M. A. Ahsan, U.S. Bureau of Reclamation, P.O. Box 25007, D-5711, DFC, Building 67, Denver, CO 80255-0007
67. L. R. Ahuja, USDA-ARS, 301 S. Howes St. POB E, Grt Plns Sys. Res. Unit, Fort Collins, CO 80522
68. S. H. Anderson, Professor, University of Missouri, 144 Mumford, Department of Soil Science, School-Nat. Resources, Columbia, MO 65211
69. D. A. Barry, Professor, University of Western Australia, Centre for Water Res., Nedlands, WA 6009, Australia
70. P. Baveye, Professor, Cornell University, Department of Soil, Crop, and Atmosph. Science, Bradfield Hall, Ithaca, NY 14853
71. M. E. Best, Geological Survey of Canada, Pacific Geoscience Centre, P.O. Box 6000, Sidney, BC, V8L 4B2, Canada
72. J. V. Bellini, Baker Engineers, 4301 Dutch Ridge Road, Beaver, PA 15009
73. J. Benegar, Geotrans, Inc., 46050 Manekin Plaza Sutie 100, Sterling, VA 22170

74. P. Y. Biver, Liege University LGIH, University Sart Tilman B19, 4000 Liege, Belgium
75. C. W. Boast, Professor, University of Illinois, S-212 Turner, 1102 South Goodwin Ave., Urbana, IL 61801
76. G. Bodenstine, U.S. Department of Energy Oak Ridge Operations, Federal Building, Oak Ridge, TN 37880
77. W. J. Bond, CSIRO Division of Soils, G.P.O Box 639 ACT 2601, Canberra City, Australia
78. WillemJan P. Bosma, Agricultural University, P.O. Box 8005, 6700 EC, Wageningen, Netherlands
79. R. S. Bowman, Professor, Department of Geosciences, New Mexico Technology, Socorro, NM 87801
80. J. Bouma, Professor, Department of Agriculture, Soil Science and Geology University, Box 37 6700AJ, Wageningen, 670-OAB, Netherlands
81. G. Brett Doerr, CH2M Hill, P.O. Box 4400, Reston, VA 22090-1483
82. M. Broido, Acting Director, Environmental Sciences Division, Department of Energy, 19901 Germantown Road, Germantown, MD 20874
83. J. J. Bronswijk, Winand Staring Ctr., P.O. Box 125, Wageningen, 670-OAC, Netherlands
84. M. L. Brusseau, Professor, University of Arizona, Department of Soil and Water Science, 429 Schantz, Building 38, Tucson, AZ 85721
85. R. G. Burau, Professor, University of California, LAWR, Davis, CA 95616
86. A. C. Campbell, U.S. Nuclear Regulatory Commission, Mail Stop 5E4-OWFN, Washington, DC 20555
87. D. K. Cassel, Professor, North Carolina St. University, Department of Soil Science Box 7619, Raleigh, NC 27695-7619
88. J. Cawfield, Professor, 129 McNutt Hall, University Missouri, Rolla, MO 65401
89. C. Chen, Professor, Hawaii Institute of Geophysics, University of Hawaii, 2525 Correa Rd., Honolulu, HI 96822
90. I. G. Choi, Westinghouse Savannah River Co., Savannah River Technology Center, Aiken, SC 29808
91. B. E. Clothier, New Zealand, DSIR, Plant Physiology Div., Palmerson North, New Zealand
92. G. Crosta, University of Milan, Department of Info Science, 9, V Moretto da B, Milan, Italy 20133
93. J. H. Cushman, Professor, Purdue University, Agronomy Department, West Lafayette, IN 47907
94. P. A. deGroot, SAIC Engineering Inc., 10 Riverside Drive, Lakeville, MA 02347
95. J. H. Dane, Professor, Auburn University, Department of Agronomy and Soils, Auburn, AL 36849-5412
96. Eeltje de Jong, Professor, University of Saskatchewan, Department of Soil Science, Saskatoon, SK S7N0W0, Canada
97. F. A. Donath, Director, Institute for Environmental Education, Geological Society of America, Las Posas, San Clemente, CA 92673

98. W. Durner, Professor, University of Bayreuth, Department of Hydrology, PB 101251, Bayreuth, D8580, Germany
99. W. M. Edwards, USDA-ARS, P.O. Box 478, Coshocton, OH 43812
100. D. E. Elrick, Professor, University of Guelph, Department of Land Res. Science, Guelph, ON, N1G-2W1, Canada
101. R. N. Farvolden, Professor, Department of Earth Sciences, University of Waterloo, Waterloo, Ontario N2L 3G1 Canada
102. Husheng "Terry" Feng, Bechtel Corporation, 50 Beale St., P.O. Box 193965, San Francisco, CA 94119
103. S. A. Flores, Comision Federal De Electricidad, Oklahgma 85 3rd piso, Mexico D. F. 03810
104. H. Fluhler, Soil Physics, Fed. Institute of Tech, IWHF, ETH-Zentrum, Zurich, CH8-092, Switzerland
105. D. W. Freckman, Director, College of Natural Resources, 101 Natural Resources Building, Colorado State University, Fort Collins, CO 80523
106. T. Ginn, Battelle Laboratories, 2031 Howell Ave., Richland, WA 99352
107. D. S. Goldman, G.W.L., Inc., 8021 Phillips Highway, Suite 12, Jacksonville, FL 32256
108. R. E. Green, Professor, University of Hawaii, Agronomy and Soils, 1910 E.W. Rd., Honolulu, HI 96822
109. S. C. Gupta, Professor, University of Minnesota, Department of Soil Science, 1911 Upper Buford, St. Paul, MN 55108
110. J. M. Hendricks, New Mexico Institute Min and Technology, Department of Geoscience, Socorro, NM 87801
111. P. K. Hildbrandt, Chevron Research and Technonogy Company, 1003 W. Cutting Blvd., Room 202, Richmond, CA 94804
112. D. Hillel, Professor, University of Massachusetts, Department of Plant and Soil Sciences, Amherst, MA 01003
113. J. W. Hopmans, Professor, University of California-Davis, Department of LAWR, Viehmeyer Hall, Davis, CA 95616
114. G. Hornberger, Professor, University of Virginia, Department of Environmental Science, Charlottesville, VA 22903
115. A. G. Hornsby, Professor, University of Florida, 2171 McCarty Hall, Soil Science Department, Gainesville, FL 32601-0151
116. N. J. Jarvis, Professor, University of Agriculture, Frodingsgatan 16 754 21, Uppsala, Sweden
117. D. B. Jaynes, USDA-ARS-NSTL, 2150 Pammel Dr., Ames IA 50011
118. P. D. Jenssen, Ctr. for Soil Environmental Res., P.O. Box 9, AAS-NLH, 1432 Norway
119. G. Y. Jordy, Director, Office of Program Analysis, Office of Energy Research, ER-30, G-226, U.S. Department of Energy, Washington, DC 20545
120. W. A. Jury, Professor, University of California, Riverside, Department of Soil and Environmental Science, Riverside, CA 92521
121. R. S. Kanwar, Professor, Iowa State University, 214B Davidson Hall, Ames, IA 50011

122. J. B. Kool, HydroGeologic, Inc., 1165 Herndon Pkwy, Suite 900, Herndon, VA 22070
123. P. M. Kroopnick, Groundwater Technology Inc., 4057 Port Chicago Hwy., Concord, CA 94520
124. King-Jau S. Kung, Professor, Unif. of Wisconsin, Rm 74 Soils, 1525 Observatory, Madison, WI 53706
125. F. J. Leij, Professor, University of California, US Salinity Laboratory, 4500 Glenwood Dr., Riverside, CA 92501
126. Hsin-Ehi J. Lin, U.S. Army Crop Engineers, WES, 3909 Hall Ferry Road/WESJE-E, Vicksburg, MS 39180-6199
127. K. Loague, Professor, Department of Soil Science, 108 Hilgard Hall, University of California, Berkeley, CA 94720
128. D. C. Marozas, U.S. Bureau of Mines, TCRC, 5629 Minneahaha Avenue South, Minneapolis, MN 55417-3099
129. R. S. Mansell, Professor, University of Florida, 4215 NW 18th Place, Gainesville, FL 32611
130. L. McKay, Professor, Department of Geological Science, University of Tennessee, Knoxville, TN 37996
131. S. Mochizuki, Wacos Japan Co., Ltd., 6-11 Shoei-Cho, Numazu-Shi, Japan 00410
132. C. D. Montemagno, Argonne National Laboratory, ER/203, 9700 S. Cass Avenue, Argonne, IL 60439
133. J. R. Nimmo, USGS, Mail Stop 421, 345 Middlefield Rd., Menlo Park, CA 94025
134. N. E. Olaque, Sandia National Laboratory, P.O. Box 5800, Division 6416, Albuquerque, NM 87185
135. J. L. Olson, Westinghouse Env. and Geo., Suite 700, 3980 Dekalb Tech. Pkwy., Atlanta, GA 30340
136. C. K. Owens, Exxon Production Resrh., Mail Drop ST-767, P.O. Box 2189, Houston, TX 77252-2189
137. Jean-Yves Parlange, Professor, Cornell University, Department of Agric. Eng., Riley-Robb, Ithaca, NY 14853
138. J. Parker, ES&T Inc., 2608 Sheffield Dr., Blacksburg, VA 24060-8270
139. A. Patrinos, Associate Director, Office of Health and Environmental Research, Department of Energy, G-165, Washington, DC 20585
140. P. A. Raats, DLO-Institute for Soil Fert. Res., Postbus 30003, Haren, 975-ORA, Netherlands
141. P. S. Rao, Professor, University of Florida, 2171 McCarty Hall Soil Science, Gainesville, FL 32611-0151
142. J. I. Restrepo, South Florida Water Maint., 240 Sussex Circle, Jupiter, FL 33458
143. D. Reynolds, Land Resource Res. Ctr., Cen. Expt. Farm Agric., Ottawa, ON K1A-0C6, Canada
144. C. J. Ritsema, Staring Ctr., P.O. Box 125, Institute Integrtd Land, Soil/Water, Wageningen, 6700, Netherlands

145. P. J. Ross, CSIRO, Div. of Soils, PMB Aitkenvale QLD 4814, Townsville, Australia
146. K. Roth, Professor, Soil Science and Pl. Envir.; University of Hohenheim, P.O. 700562, 7000 Stuttgart 70, Germany
147. D. Russo, Israel Institute Soil and Water, ARO, The Volcani Center, Bet Dagan, Israel
148. G. S. Sayler, Professor, 10515 Research Drive, Suite 100, The University of Tennessee, Knoxville, TN 37932-2567
149. B. R. Scanlon, Professor, Bureau of Econ. Geology, University of Texas-Austin, U. Stn. Box X, Austin, TX 78713
150. R. Schulín, Swiss Fed. Institute of Technology, Heliosstr, 11, Zurich, CH8-032, Switzerland
151. J. R. Schuring, New Jersey Institute of Technology, University Heights, Newark, NJ 07102
152. H. M. Selim, Professor, Louisiana State Univ., Agronomy Department, Baton Rouge, LA 70803
153. S. Sharp-Hansen, Aqua Terra Consultants, 2672 Bayshore Parkway, #1001, Mountain View, CA 94043
154. B. D. Smith, Geotech, Inc., P.O. Box 14000, Grand Junction, CO 81502
155. Kip Solomon, Professor, Department of Geology and Geophys., School of Mines and Earth Science, University of Utah, 717 WBB, Salt Lake City, UT 84112-1183
156. P. J. Shouse, USDA-ARS, U.S. Salinity Laboratory, 4500 Glenwood Dr., Riverside, CA 92501
157. B. Sisson, P.O. Box 1625, MS-2107, Idaho Falls, ID 83415
158. J. M. Skopp, Professor, University of Nebraska, Department of Agronomy, Lincoln, NE 68583
159. K. R. Smettem, CSIRO, Div. of Soils, Private Bag #2, Glen Osmond, Adelaide, Australia
160. V. A. Snyder, Professor, University of Puerto Rico, Department of Agronomy and Soils, Mayaguez, PR 00708
161. G. Sposito, Professor, Department of Environmental Science, 108 Hilgard Hall, University of California #3110, Berkeley, CA 94720-3110
162. T. S. Steenhuis, Professor, Cornell University, Department of Ag. Engineering, Ithaca, NY 14853
163. D. B. Stephens, Daniel B. Stephens and Assoc. Inc., 6020 Academy Rd. NE, Albuquerque, NM 87109
164. J. M. A. Streng, Delft Hydraulics, P.O. Box 117, Delft, The Netherlands 2600MH
165. M. Seyfried, USDA-ARS, 800 Park Blvd, Plaza 4, Boise, ID 83712-7716
166. J. Sweeney, U.S. Department of Energy Oak Ridge Operations, Federal Building, Oak Ridge, TN 37880
167. J. Szecsody, Battele, P.O. Box 999, MS-K6-77, Pacific Northwest Laboratory, Richland, WA 99352
168. D. J. Timlin, USDA-ARS, System Res. Laboratory, Building 007, Beltsville, MD 20705

169. A. F. B. Tompson, Lawrence Livermore National Laboratory, P.O. Box 808, Livermore, CA 94551
170. S. W. Tyler, Professor, University of Nevada, Desert Research Institute, P.O. Box 60220, Reno, NV 89506
171. G. Vachaud, Institute de Macanique, BP68-38402-St. Martin, d'Herès Cedex, France
172. A. Valocchi, Department of Civil Engineering, University of Illinois, 208 N. Romine St., Urbana, IL 61801
173. Rienk van der Ploeg, Professor, Institute Soil Science, University Hanover, Herrenhauser Str. 2, Hanover, 300-021, Germany
174. S. E. van Der Zee, Professor, Agric. University, Van Doesburglaan 10, Wageningen, 670-8MC, Netherlands
175. H. M. Van Es, Professor, Cornell University, Department of Agronomy, Ithaca, NY 14853
176. Rein van Genuchten, U.S. Salinity Laboratory, 4500 Glenwood Dr., Riverside, CA 92501
177. M. M. Vauclin, Institute de Macanique, BP 68 38402-St. Martin, d'Herès, France
178. K. Veluswamy, WACOS Japan Co. Ltd., 6-11 Shoei-Cho, Numazu-Shi 410, Japan
179. R. J. Wagenet, Professor, Cornell University, Department of Agronomy, 235 Emerson Hall, Ithaca, NY 14853
180. A. W. Warrick, Professor, University of Arizona, Soil and Water Science Department, 429 Shantz Building #38, Tucson, AZ 85721
181. M. F. Wheeler, Professor, Department of Computational and Applied Mathematics, Rice University, Houston, TX 77251-1892
182. P. J. Wierenga, Professor, Soil and Water Science Department, 429 Shantz Building #38, University of Arizona, Tucson, AZ 85721
183. G. V. Wilson, Professor, Department of Plant and Soil Science, The University of Tennessee, Knoxville, TN 37901
184. F. J. Wobber, Environmental Sciences Division, Office of Health and Environmental Research, ER-74, U.S. Department of Energy, Washington, DC 20585
185. D. R. Wuthrich, Geomatrix Consultants, 100 Pine St., 10th Floor, San Francisco, CA 94111
186. E. K. Yanful, Noranda Technology Centre, 240 Hymus Blvd., Pointe-Claire, Quebec, H9R 1G5, Canada
187. Fei-Fan Yeh, EBASCO Services Inc., 3000 W. Mac Arthur Blvd., Santa Ana, CA 92704
188. G. T. Yeh, Professor, Department of Civil and Environmental Engineering, Pennsylvania State University, 212 Sackett Building, University Park, PA 16802
189. Hun-Der Yeh, Professor, Department of Environmental Engineering, National Chiao-Tung University, Hsin Chu, Taiwan
190. T. C. Jim Yeh, Professor, Hydrology and Water Resources Department, University of Arizona, Tucson, AZ 85721

191. W. W.-G. Yeh, Professor, Department of Civil Engineering, UCLA, Los Angeles, CA 90024
192. J. Zachara, Pacific Northwest Laboratory, Battelle Boulevard, Richland, WA 99352
193. R. E. Zartman, Professor, Texas Tech University, Agronomy Department, Lubbock, TX 79409-2122
194. You-Kuan Zhang, Professor, Department of Geology, University of Iowa, Iowa City, IA 52242-1379
195. R. W. Zimmerman, Earth Sciences Division, Lawrence Berkeley Laboratory, University of California, Berkeley, CA 94720
196. Office of Assistant Manager for Energy Research and Development, U.S. Department of Energy Oak Ridge Operations, P.O. Box 2001, Oak Ridge, TN 37831-8600
- 197-198. Office of Scientific and Technical Information, P.O. Box 62, Oak Ridge, TN 37831