[Ph.D. Thesis]

UNIVERSITY OF NEWCASTLE UPON TYNE

DEPARTMENT OF CIVIL ENGINEERING

Development of a Generalized Compositional Multiphase Model for Flow and Transport in Porous Media

by

Ki Young Lee

For the Degree of Ph.D.

in Water Resources Engineering

April 1997

ACKNOWLEDGEMENTS

It is a very exciting moment to write acknowledgements, because it means that a precious work has finished. Four and half years have passed since I came here in Newcastle for a Ph.D. degree. However, what makes me sad is that it is time to leave this beautiful place. Every morning I used to go to a park for exercise and fresh air. I dare say that the most striking memory is all about the park in the morning. It has been more than a park to me.

Actually I have produced a lot in many areas during the life in this University. The consideration of my supervisor, Professor Rae Mackay, has been essential for me to produce many valuable things. This thesis must not have been finished without his advice and guidance. I really thank him.

I can not help mentioning of Dr. Allan Herbert and Michael Riley who examined my thesis carefully. I feel that my thesis has improved greatly with their proper advice. Besides the work that is concerned with the thesis, I personally like the big smile of my internal examiner.

The Robinson library has been a valuable place too, because I have spent most of time there. I have met so many good people there and I have got so many ideas there. I hope that all the people in the library would be happy.

Anyway I have got a very precious experience during my Ph.D. course. I am such a lucky person. Finally, remembering the first day I arrived at the Newcastle airport, I thank Professor P. E. O'connell who introduced Newcastle University to me.

TABLE OF CONTENTS

CHAPTER I
INTRODUCTION

1.1 Research purpose

Fresh water is one of the most important natural resources. However, like other natural resources, the usable water is limited while the demand for water increases as industrialization proceeds and the population grows. What makes matters worse is that water resources are being reduced by pollution.

Groundwater is an important water resource. However, in many countries, it has not been fully developed yet, either because of sufficient surface water sources, technical problems, or geographical conditions. Generally groundwater is relatively clean and is better protected from pollutants than surface water. Thus groundwater is an important subject for water engineers and scientists who have focused on its development and protection. In both cases, research into the movement of pollutants plays an important role in the effective exploitation of groundwater.

Recently hydrologists concerned with groundwater pollution have studied multiphase flows in the subsurface because many pollution problems are characterized by multiphase contamination. The simplest multiphase pollution problem is solute transport in the unsaturated zone. More complex multiphase pollution problems involve organic matter such as petroleum products discharged to use oil. Since many of organic products are essential to our normal life and industry, the potential for groundwater pollution by them is significant unless they are controlled properly. In multiphase problems, the organic compounds may form their own flows that are distinct from the subsurface water flow but partly dissolve with the water phase and cause low concentration long term pollution of the water phase.

There have been many efforts dedicated to predicting the movement of pollutants. A lot of mathematical and numerical models have been developed with the aid of laboratory and field works. However almost all models have been developed to solve a few restricted scenarios. Model users are obliged to invest considerable time in understanding the various models; their numerical accuracy and coding.

The purpose of this study is to categorize the pollution patterns in the subsurface and to develop a numerical model that can be applicable to a wide variety of subsurface contamination. The general primary variables and generalizing procedures are employed to make the numerical model applicable to various pollution patterns. Many kinds of tracers can be used to know the behaviors of fluid phases in the subsurface. Because the model is able to describe partitioning of mass of a component among fluid phases, tracer problems also can be simulated by the model.

1.2 Literature review

1.2.1 Basic definitions and pollution patterns

Beneath the surface of the ground, the solid matrix consists of soil, sand, gravel, and rock. Usually there exists void space in the solid matrix, through which fluids may migrate. This kind of solid matrix is defined as *porous medium*. Through the void space, there may be flows of gas, water, or oil. If more than one fluid is found in a porous medium and they can be characterized as distinct bodies separating each other by distinct physical boundaries, and each being identifiable by distinct quantities in space and time, they are referred to as *phases*. If void space is filled with only one fluid phase, then the term, *single phase flow*, is used to characterize the movement of the fluid; otherwise, *multiphase flow*. A phase may consist of several chemical species defined as *components*.

The pollution patterns in the subsurface are subjected to properties of solid matrix, fluid phases in void space, and interactions among fluid phases. Bear and Buchlin(1991) discuss the behaviors of water, gas, and oil, according to the property of the solid matrix. If the solid matrix consists of *hydrophilic material*, water is the wetting phase. The affinity of oil to the solid matrix is somewhere between those of water and gas. On the other hand, if the solid matrix consists of *oliophilic material*, the behaviors of water and oil are exchanged. The fluid phases in a porous medium contribute to determining the pollution patterns. This study considers the four

combinations of the fluid phases: water-gas, organic-gas, water-organic, and water-organic-gas. The interphase mass transfer that can be caused by dissolution and evaporation is also an important factor that determines the pollution patterns. This study categorizes the pollution patterns that can be simulated by a numerical model.

1.2.2 Mathematical model

The physical phenomena associated with multiphase fluid flows in porous media can be expressed analytically to determine the basic thermodynamic quantities such as mass density, motion, and temperature. The governing equations that describe movements of these quantities are usually based on the conservation laws of mass, momenta, energy and entrophy.

Grouse(1966), Soo(1967), and Butterworth and Hewitt(1977) have relied on somewhat intuitive or empirical concepts to derive the conservation equations. So these models are generally restricted in application to particular multiphase systems.

The continuum theory of mixtures are employed by Eringen and Ingram(1965), and Muller(1968) to obtain governing equations for multiphase systems. In this method, phases are viewed as overlapping continua, which simultaneously exist everywhere and occupy the whole space.

The previous two approaches are focused on particular assumptions such as incompressible, steady state, or one-dimensional flow. To derive flexible general conservation equations for multiphase systems, the local volume averaging technique is adopted by some scholars.

In the approach of local volume averaging, the system is considered to be composed of interpenetrating continua. The thermodynamic quantities of a phase are assumed to be continuous for the phase but discontinuous over the entire space, because each phase occupies part of space and is separated by highly irregular interfaces. The classical balance laws of continuum mechanics may be applied to the system. However because the description of the configuration of pore space is an overwhelming task, the governing equations obtained at the microscopic scale should be averaged over representative local volume.

Various scholars emphasize the value of the local volume averaging technique in obtaining equations applicable to multiphase systems. Whitaker(1966) employs local area averaging to develop Darcy's law. However, this technique proves too complicated from the standpoint of the notational conventions. The difficulties of area averaging are overcome when Whitaker(1967) and Slattery(1967) adopt the volume averaging method. Slattery(1967) and Whitaker(1969) develop a theorem which relates the average of a spatial derivative of a function to the spatial derivative of the average of the function. Bachmat(1972) applies a column averaging technique to continuum equations that contain spatial derivatives and time derivatives. However, it seems that one of the most useful researches has been conducted by Gray and Lee(1977). They present the theorems of local volume averaging which relate averages of derivatives to derivatives of averages. Given its simple and clear generalization, it is very useful for deriving macroscopic mass balance equations of multiphase fluids in porous media.

1.2.3 Numerical model

Some authors have developed analytical solutions for multiphase flows in porous media. However most of them are focused on special cases which are extremely simplified. It is very difficult to solve highly nonlinear partial differential equations analytically that describe fluids transport in porous media. Consequently many modelers have turned to numerical techniques for their solution. The two most popular numerical techniques are the finite difference method and the finite element method. Each has advantages and disadvantages and it is difficult to say which one is better.

For a long time, the finite difference technique has been very popular among hydrologists. Nolen(1972) develops a reservoir simulator to solve both water-coning and gas-percolation problems in an oil reservoir. The reservoir simulator is based on the nonlinear form of the semi-implicit finite difference equations.

Abriola and Pinder(1985) present a mathematical model for three-phase flows(water, gas, and organic compounds). Because they consider interphase transfer from organic compounds to water and gas phases, the transport of a chemical contaminant is described as a nonaqueous phase, as a soluble component of an aqueous phase, and as

a mobile fraction of a gas phase. The contaminant is assumed to be composed of , at most, two distinct components, one of which is volatile and water soluble and the other of which is non-volatile and insoluble in water. In addition, the effects of matrix and fluid compressibilities, gravity, capillarity, diffusion and dispersion are all considered. The resulting mathematical model is solved using the finite difference technique. However, this model is restricted to only water dominated situation, because it assumes that water phase is in contact with gas phase.

Faust(1985) produces a numerical model that describes the simultaneous flow of water, a second immiscible fluid and gas. The two-dimensional equations for flow in a vertical plane are approximated by the finite difference scheme. No mass transfer between the phases is considered.

Corapcioglu and Baehr(1987a) develop a mathematical model describing the fate of hydrocarbon constituents of petroleum products. Three-phase(water, gas and oil) flows are expressed analytically, allowing mass transfer of reactive constituents such as benzene, toluene, and xylene found in refined petroleum products like gasoline. In addition adsorption is considered and microbial degradation of petroleum products is also discussed, focusing on the mass conservation of oxygen which plays an important role in the metabolism of hydrocarbons. This solution is obtained by using a finite difference method and a method of forward projection to evaluate the nonlinear coefficients. However, unlike the model of Abriola and Pinda(1985a,b), it is not suitable for treating the water dominated system.

Sleep and Sykes(1993) develop a compositional simulator for analyzing simultaneous flow of three fluid phases(water, gas, and organic). The model which uses a block-centered finite difference discretization can simulate interphase partitioning and transport of an arbitrary number of organic and inorganic components. However, its application is restricted to only the case that the water phase is the most wetting and the gas phase is the least wetting, and that in a three-phase system only organic-water and organic-gas interfaces are formed. Thus the compositional model is not general because of the fixed assumption.

Baehr(1987) presents a system of partial differential equations defining radially symmetric transport of solutes and vapours from a multiconstituent immiscible contaminant at residual saturations. The finite difference method is used for the

numerical solution of the equations. His model predicts long-term groundwater pollution from vapours and solutes emanating from organic liquids at residual saturations. He considers the mass transfer of organic phases to water and gas phases.
A numerical model to predict the migration of organic contaminants in the subsurface is developed by Kia(1991). In a three-phase fluid system of gas, water and contaminant, simultaneous flow of the water and contaminant phases is formulated by applying mass conservation principles to each of the phases under the condition of no interphase mass exchange. The complex formulations are solved numerically using an implicit finite difference scheme.

The finite element technique is more suitable for complex geometry and can track sharp fronts more accurately. However, when applying them to highly nonlinear, immiscible flow problems, several numerical difficulties can be encountered. Mercer and Faust(1977) discuss these difficulties and suggest several techniques to overcome them.
Voss(1984) produces the numerical model, SUTRA, employing a finite element technique. SUTRA solves solute transport problems in the unsaturated and saturated zone. Many authors have made considerable progress in the solution of solute transport problems. In this thesis, they are considered as a subject of the more general multiphase problem.
Langsrud(1976) presents a finite element model for two-phase(oil and gas) flows in porous media. Two-phase flow of compressible fluids in a porous medium is characterized by two coupled, non-linear equations for oil and gas.
Osborne and Sykes(1986) develop a two-dimensional mathematical model for two-phase flows in porous media. The numerical model is based on a generalized method of weighted residuals in conjunction with the finite element method and linear quadrilateral isoparametric elements. Incompressible fluid and porous medium are assumed.
Lin(1987) develops a model for the flow of two incompressible and immiscible fluids in incompressible porous media. He does not consider mass transfer, diffusion and dispersion. Numerical analysis, based on the finite element method, is presented. To overcome nonlinear systems of equations, the Newton-Raphson's method is employed.

Kuppusamy et al(1987) develops a finite element model for multiphase flow through porous media involving three immiscible fluids(gas, water and a nonaqueous phase). A variational method is employed for the finite element formulation. No mass transfer is considered among phases. Simplified flow equation is used because an incompressible porous medium is assumed with constant fluid densities and viscosities.

As previously shown in this section, the existing numerical models have very narrow validity in spite of the fact that there could be many cases of flow patterns in the subsurface especially for the multiphase flows. Thus it is necessary to develop a compositional multiphase model which is applicable to a wide range of multiphase flow problems in porous media. This study is dedicated to meet the purpose, which may make the scientists and engineers freer in applications and assumptions.

1.3 Research content

Chapter II derives the general macroscopic balance equation that describes the movements of quantities such as mass, momentum, energy, or temperature. It discusses the general physical characteristics of subsurface flows and introduces the volume averaging method. The boundary conditions are considered to complete the mathematical model.

Chapter III discusses the physical phenomena for multiphase systems in the subsurface, focusing on interfacial tension, wettability, capillary pressure curve, velocity of a fluid phase, dispersion and diffusion. Employing the general primary variables, the three governing equations are determined by applying the mass balance law to components and phases of concern. This chapter discusses the scope of application that can be dealt with the three governing equations.

Chapter IV introduces the finite element technique used to discretize the governing equations derived in chapter III. It explains the numerical techniques needed to overcome difficulties that are caused by the attempt of dealing with various pollution types. The variations of the general primary variables and the spatial and temporal derivatives in the governing equations are categorized according to the pollution patterns.

Chapter V describes the verification of the numerical model, COMPO (COMPO1D and COMPO2D), applying it to four examples. The accuracy of the code, COMPO, is conveyed by the comparisons of simulation results obtained by COMPO and other numerical models. The convergence properties of COMPO are also shown by grid and time step refinement. Finally the extended use of the code is shown by applying it to a tracer problem.

CHAPTER II
TRANSPORT PHENOMENA IN POROUS MEDIA

The purpose of this chapter is to provide the general balance equation for a quantity such as mass, momentum, energy, or temperature. The fundamental knowledges of the continuum mechanics are introduced in section 2.1. The microscopic balance equation is derived in section 2.2. Section 2.3 presents the averaging rules which are required for deriving the macroscopic balance equation. Integrating the microscopic balance equation over REV(Representative Elementary Volume), the macroscopic balance equation is obtained in section 2.4 using the averaging rules. To obtain specific solutions from the macroscopic balance equations, boundary conditions are needed. Section 2.5 derives microscopic and macroscopic boundary conditions.

2.1 Basics of continuum

Phases may not be continuous throughout the domain of concern due to the geological complexity. However, all phases are assumed to be continua over the domain to construct an mathematical model. In this section, the basic concept of kinematics of continuous materials is introduced.

2.1.1 Coordinates

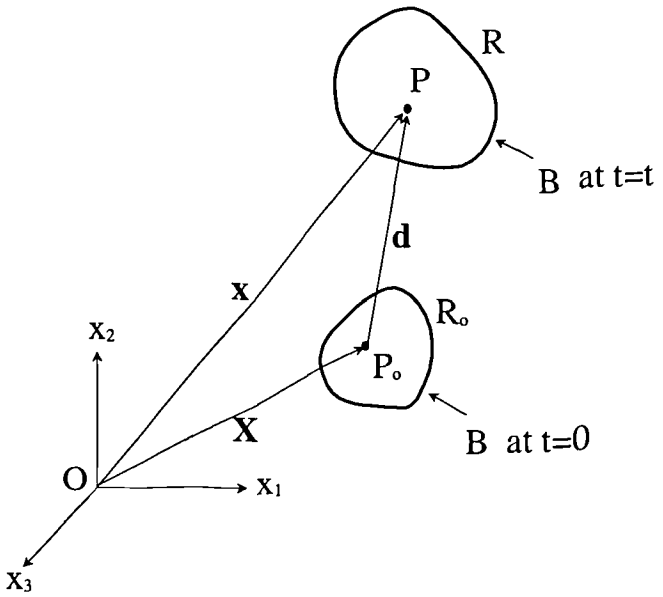


Figure 2.1.1 Coordinate description

There are two ways to describe the position of a particle of a phase, depending on the position of an observer. To start with, consider a small body, B, of the phase at time, t=0, that is occupied by continuously distributed matter of the phase(see figure 2.1.1). R_o is supposed to be the region which is occupied by B. The position of a typical point P_o within R_o can be presented by using the spatial(Eulerian) coordinate system where the position of the observer is fixed. Every point in R_o can be described in this way. Let **X** be the position vector of P_o. Then the components X_R of **X**, in the chosen coordinate system, are the coordinates of the position, P_o, occupied by a particle of B at t=0. Each point of the region R_o corresponds to a particle of the body B, and B is the assemblage of all such particles.

If the body which occupies the region R_o at t=0 moves by any external forces, it will occupy a new continuous region R at time t. Although the configuration of the body B at t=0 changes with time, the same particles will be distributed continuously but in different space. So the individual particles of the body B can be identified at time t. Let us define a point of R as P which is occupied by the same particle that occupied P_o at t=0. As shown in figure 2.1.1, the position vector of P is **x** in the spatial coordinate system.

Another coordinate system is the material(Lagrangian) coordinate system where the observer moves along with the particle of B that occupies P_o at t=0 and P at t=t. At the

reference time t=0, the particle can be identified by the position vector **X**. Regardless of time elapse, the particle is defined by **X** according to the material coordinate system. For practical purposes, it is required to establish the relationship between two coordinate systems. The position of the particle of B, at t=t, can be denoted as **x** in the spatial coordinate system. So the position vector **x** can be represented as the function of **X** and t, because **X** reflects the origin of the particle and time is another independent factor which decides the position of the particle:

$$\mathbf{x} = \mathbf{x}(\mathbf{X}, t), \quad \text{or} \quad x_i = x_i(X_R, t) \quad (i, R = 1,2,3) \tag{2.1.1}$$

where x_i : components of **x**

X_R : components of **X**

For physically realizable motions it is possible in principle to solve (2.1.1) for **X** in terms of **x** and t, which gives equations of the form:

$$\mathbf{X} = \mathbf{X}(\mathbf{x}, t), \quad \text{or} \quad X_R = X_R(x_i, t) \quad (i, R = 1,2,3) \tag{2.1.2}$$

Problems in continuum mechanics may be formulated either with the material coordinates X_R as independent variables, or with the spatial coordinates x_i as independent variables. In the material description attention is focused on what is happening at, or in the neighbourhood of, a particular material particle. In the spatial description attention is focused on events at, or near to, a particular point in space. In principle it is possible to transform a problem from the material to the spatial description or vice versa by using (2.1.1) or (2.1.2).

2.1.2 Displacement and velocity of a particle

The displacement vector, **d**, of the particle of B can be expressed as follows(see figure 2.1.1):

$$\mathbf{d} = \mathbf{x} - \mathbf{X} \tag{2.1.3}$$

Adopting the material coordinate system, it can be rewritten as:

$$\mathbf{d}(\mathbf{X},t)=\mathbf{x}(\mathbf{X},t)-\mathbf{X} \tag{2.1.4}$$

On the other hand, equation(2.1.3) is able to be expressed in terms of the spatial description as follows:

$$\mathbf{d}(\mathbf{x},t)=\mathbf{x}-\mathbf{X}(\mathbf{x},t) \tag{2.1.5}$$

In the material coordinate system, the particle of B is always identified as the position vector, **X**, regardless of time elapse. Thus the velocity vector of the particle can be expressed in the material coordinate system as follows:

$$\mathbf{v}(\mathbf{X},t)=\frac{\partial \mathbf{d}(\mathbf{X},t)}{\partial t}=\frac{\partial \mathbf{x}(\mathbf{X},t)}{\partial t} \tag{2.1.6}$$

where the differentiations are performed with **X** held constant. In terms of the components v_i of **v**, `equation(2.1.6) can be written as:

$$v_i(X_R,t)=\frac{\partial x_i(X_R,t)}{\partial t} \tag{2.1.7}$$

The result of performing the differentiation (2.1.6) or (2.1.7) is to express the velocity components as functions of X_R and t; that is, they give the velocity at time t of the particle which was at **X** at t=0. Frequently it needs to employ the spatial description. To do so, it is necessary to express v_i in terms of x_i by using the relations (2.1.2).

2.1.3 Material derivative

Assuming that a general quantity, G, varies throughout the domain of interest with time and space, G is the function of time and space. As mentioned previously, it can be expressed by using either the material coordinate system or the spatial coordinate system:

$$G=G(X_R,t)=G(x_i,t) \tag{2.1.8}$$

The mathematical formulation of general physical laws and the description of the properties of particular materials is often most easily accomplished in the material description, but for the solution of particular problems it is frequently preferable to use the spatial description. It is therefore necessary to employ both descriptions, and to relate them to each other. The time derivative of G in terms of the material description is called *the material derivative*. It can be expressed as:

$$\frac{DG}{Dt}=\frac{\partial G(X_R,t)}{\partial t} \tag{2.1.9}$$

Adopting equation(2.1.1), equation(2.1.8) can be rewritten as follows:

$$G= G\{x_i(X_R,t),t\}= G\{x_1(X_R,t), x_2(X_R,t), x_3(X_R,t),t\} \tag{2.1.10}$$

Then, the following equation is obtained by differentiating equation (2.1.10) with respect to t with X_R constant:

$$\begin{aligned}\frac{DG}{Dt}=&\frac{\partial G(x_i,t)}{\partial x_1}\frac{\partial x_1(X_R,t)}{\partial t}+\frac{\partial G(x_i,t)}{\partial x_2}\frac{\partial x_2(X_R,t)}{\partial t}\\&+\frac{\partial G(x_i,t)}{\partial x_3}\frac{\partial x_3(X_R,t)}{\partial t}+\frac{\partial G(x_i,t)}{\partial t}\end{aligned} \tag{2.1.11}$$

By using the summation convention, this is rewritten concisely as:

$$\frac{DG}{Dt} = \frac{\partial G(x_i,t)}{\partial x_j}\frac{\partial x_j(X_R,t)}{\partial t} + \frac{\partial G(x_i,t)}{\partial t} \tag{2.1.12}$$

Equation (2.1.12) is rewritten in the simpler form by employing equation (2.1.7) as follows:

$$\frac{DG}{Dt} = v_j\frac{\partial G(x_i,t)}{\partial x_j} + \frac{\partial G(x_i,t)}{\partial t} \tag{2.1.13}$$

The final form of the derivative *DG/Dt* is called the material derivative or the convective derivative that represents the changing rate of a general variable, G, of a particle with time in terms of the spatial coordinates.

2.2 Microscopic balance equation

This section derives the microscopic balance equation by the two approaching methods: the spatial and material approach. In both cases, it is assumed that there is a single phase continuum in the domain of interest.

2.2.1 Spatial approach

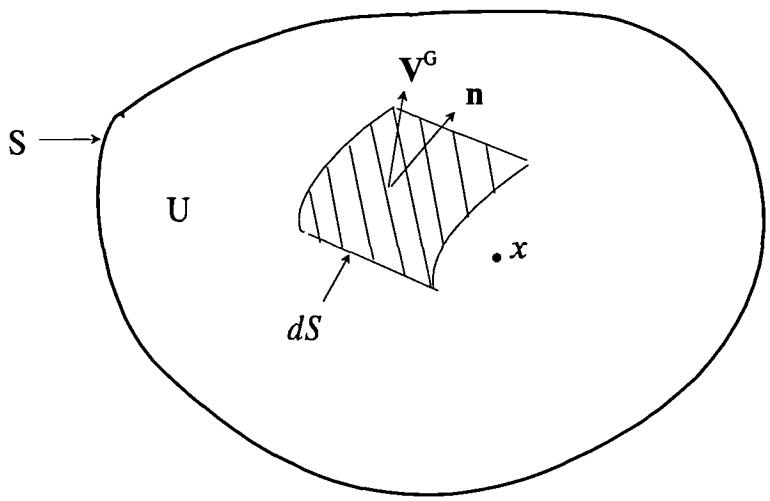


Figure 2.2.1 Control volume for the spatial approach

To start with, an arbitrary point is chosen within the phase continuum. Its position vector is defined as **x**. Consider a general quantity, G, in a volume, U, around the point. The amount of the quantity within U is $\int_u g d\mathrm{U}$, where g is the density of G. The fixed finite domain, u, is referred to as the control domain of the volume U of arbitrary shape, bounded by a closed surface, S. Figure (2.2.1) shows such a control domain, with U and S denoting its volume and the area of the surface bounding it, respectively.

The instantaneous accumulation rate of the general quantity, G, within u can be caused by influx or efflux crossing the boundary, self-production within the domain, and external supply. Thus, the following relation can be established verbally:

$$\begin{pmatrix}\text{Rate of}\\ \text{accumulation of}\\ \text{G within U}\end{pmatrix} = \begin{pmatrix}\text{Net influx of}\\ \text{G into U}\\ \text{through S}\end{pmatrix} + \begin{pmatrix}\text{Net rate of}\\ \text{production of}\\ \text{G within U}\end{pmatrix} + \begin{pmatrix}\text{Rate of}\\ \text{external supply}\\ \text{into U}\end{pmatrix}$$

(1) (2) (3) (4)

It can be rewritten in a mathematical form as follows:

(1) The rate of accumulation of G in U is expressed by

$$\frac{\partial}{\partial t}\int_u g d\mathrm{U} = \int_u \frac{\partial g}{\partial t} d\mathrm{U}$$

where the exchange of integration and differentiation is permitted in view of the fact that the boundary of the domain u is fixed : the domain does not change in the course of time.

(2) The net influx (= total influx minus total efflux) of G into u through S, is represented as:

$$-\int_S g\mathbf{V}^{\mathrm{G}} \cdot \mathbf{n} d\mathrm{S}$$

where $\mathbf{V}^{\mathrm{G}}$: velocity vector of the G continuum

$g\mathbf{V}^{\mathrm{G}}$: flux vector of the G continuum

Because **n** is defined as the outward normal unit vector on the elemental area dS, the integral has the minus sign.

(3) The net rate of production of G from sources in U, is expressed by:

$$\int_u \rho \Gamma^G dU$$

where Γ^G: rate of internal production of G, per unit mass of the phase.

ρ : mass density of the phase

(4) The rate of external supply of G can be defined as follows:

$$\int_u \rho f^G dU$$

where f^G : rate of external supply of G, per unit mass of the phase.

In summary, the balance of the G continuum in U is expressed by:

$$\int_u \frac{\partial g}{\partial t} dU = -\int_s g\mathbf{V}^G \cdot \mathbf{n} dS + \int_u \rho \Gamma^G dU + \int_u \rho f^G dU \qquad (2.2.1)$$

Assuming that $\rho \mathbf{V}^G$ is differentiable within U, Gauss' theorem(refer to Bear and Bachmat, 1990) can be applied to the first term on the right of (2.2.1). The Gauss theorem is represented for the general quantity, **G**, as follows:

$$\int_u \nabla \cdot \mathbf{G} dU = \int_s \mathbf{G} \cdot \mathbf{n} dS \qquad (2.2.2)$$

Employing equation(2.2.2), equation (2.2.1) can be rewritten as:

$$\int_u \left(\frac{\partial g}{\partial t} + \nabla \cdot g\mathbf{V}^G - \rho \Gamma^G - \rho f^G \right) dU = 0 \qquad (2.2.3)$$

By shrinking the volume U to zero around an arbitrary point, we obtain:

$$\frac{\partial g}{\partial t} + \nabla \cdot g\mathbf{V}^{G} - \rho\Gamma^{G} - \rho f^{G} = 0 \tag{2.2.4}$$

where all terms refer to the point. Equation (2.2.4) is the general microscopic balance equation in a continuum.

2.2.2 Material Approach

Here, the observer follows a general quantity, G, within a domain, $\mathcal{U}^{G}$, enclosed by a material surface, S^{G}. Unlike the spatial approach, $\mathcal{U}^{G}$ and S^{G} change in the course of time. The amount of G is defined within $\mathcal{U}^{G}(t)$ at time t as follows:

$$G = \int_{\mathcal{U}^{G}(t)} g dU \tag{2.2.5}$$

Since the observer follows the general quantity, G, within U, no amount of G can cross the boundary, S^{G}. However, there is a possibility of the growth of G within the material volume and the external supply of G in the course of time. Hence, the following equation can be established as:

$$\frac{D_{G}}{Dt}\int_{\mathcal{U}^{G}(t)} g dU = \int_{\mathcal{U}^{G}(t)} \rho\Gamma^{G} dU + \int_{\mathcal{U}^{G}(t)} \rho f^{G} dU \tag{2.2.6}$$

The first term of equation (2.2.6) can be rewritten by applying Reynold's transport theorem(Bear and Bachmat,1990):

$$\frac{D_{G}}{Dt}\int_{\mathcal{U}^{G}(t)} g dU = \int_{\mathcal{U}^{G}(t)} \frac{\partial g}{\partial t} dU + \int_{S^{G}(t)} g\mathbf{V}^{G} \cdot \mathbf{n} dS \tag{2.2.7}$$

As in the spatial approach, the Gauss' theorem, equation(2.2.2) is applied in order to transform the surface integral in equation(2.2.7) into a volume integral. Then, equation(2.2.7) can be rewritten as:

$$\frac{D_G}{Dt}\int_{u^G(t)} g d\mathrm{U} = \int_{u^G(t)} \frac{\partial g}{\partial t} d\mathrm{U} + \int_{u^G(t)} \nabla \cdot g\mathbf{V}^G d\mathrm{U} \tag{2.2.8}$$

Substituting equation(2.2.8) into (2.2.6), the following equation can be obtained:

$$\int_{u^G(t)} \left(\frac{\partial g}{\partial t} + \nabla \cdot g\mathbf{V}^G - \rho\Gamma^G - \rho f^G \right) d\mathrm{U} = 0 \tag{2.2.9}$$

Since the domain $u^G(t)$ is arbitrary, the integrand itself must vanish everywhere. Thus the same balance equation is derived as follows:

$$\frac{\partial g}{\partial t} + \nabla \cdot g\mathbf{V}^G - \rho\Gamma^G - \rho f^G = 0 \tag{2.2.10}$$

2.3 Averaging rules

The necessity of the REV and the basic definitions for averaging are discussed in subsection 2.3.1. The useful averaging theorems for deriving the macroscopic balance equation are derived in subsection 2.3.2.

2.3.1 REV and the basic definitions for averaging

In the previous section, the microscopic balance equation has been derived for single phase. However, there arise two big problems: one is how to trace out the complex boundaries at this level and the other is that proof that the solution of the mathematical equation is right can not be obtained, because of limitations of experimental methods.

Even if it is possible to circumvent these problems, it is impractical to spend sufficient money and time to solve the problem at the microscopic scale.

Hence, another approach is required which describes the transport phenomena as accurate as possible, but overcomes these problems. To simplify the microscopic behavior, averaging over a suitable volume which contains all characteristics under consideration can be adopted. Introducing this simplifying tool, it is feasible to define a point within a phase to represent a discrete volume, the REV(Representative Elementary Volume).

Since the REV is the basic element for the macroscopic analysis, its selection demands certain conditions. The REV should contain all characteristics of the system around a point of interest. If it is too small, there can be a severe discontinuity problem. For case that it is too large, the transport phenomena in the porous media can not be analyzed exactly. Whittaker(1969) has shown that the conditions of REV can be met when the characteristic length of the averaging volume is much greater than the pore diameter in the medium but much less than the characteristic length of the medium. Hassanizadeh and Gray(1979) also discuss the conditions for the REV. Additionally the shape, size, and orientation of the averaging volume are required to be independent of space and time.

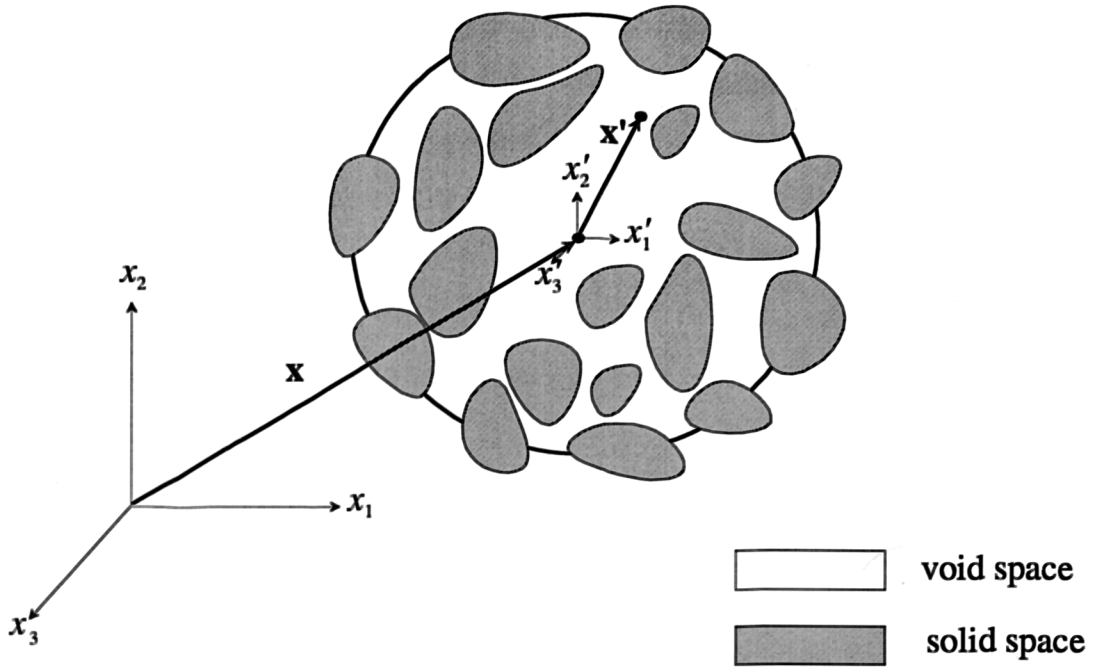


Figure 2.3.1 Representative Elementary Volume

Fig 2.3.1 shows a planar cross-section across a REV. For the purpose of averaging, it is convenient to define a local co-ordinate system, x_1', x_2', x_3', which has axes parallel with the x_1, x_2, x_3, system but whose origin is located at position **x**, center of the REV. The location of the averaging volume with respect to the **x′** co-ordinate system is independent of **x**. For instance, the averaging volume may be defined such that its centroid always coincides with the origin of the **x′** system. Let U define the volume of the REV. From figure 2.3.1, the following relations are established.

$$0 < U_s(\mathbf{x},t) < U \qquad U_s(\mathbf{x},t) + U_v(\mathbf{x},t) = U \tag{2.3.1}$$

where $U_s(\mathbf{x},t)$: volume of solid phase

$U_v(\mathbf{x},t)$: volume of void space

where the volume U is independent of space and time. However, U_s and U_v may depend on **x** and t if the medium is deformable or the system is in motion. The *porosity* is defined at the center point of REV as:

$$\varepsilon(\mathbf{x},t) = \frac{U_v(\mathbf{x},t)}{U} \tag{2.3.2}$$

$U_v(\mathbf{x},t)$ may be occupied by fluid phases. Letting the subscript α one fluid phase in the void space and the subscript β all other phases in the REV, the following relations can be established as:

$$\varepsilon_\alpha(\mathbf{x},t) = \frac{U_\alpha(\mathbf{x},t)}{U} \quad , \qquad U = U_\alpha(\mathbf{x},t) + U_\beta(\mathbf{x},t) \tag{2.3.3}$$

$$S_\alpha(\mathbf{x},t) = \frac{U_\alpha(\mathbf{x},t)}{U_v(\mathbf{x},t)} = \frac{\varepsilon_\alpha(\mathbf{x},t)}{\varepsilon(\mathbf{x},t)} \tag{2.3.4}$$

where $\varepsilon_\alpha(\mathbf{x},t)$: volumetric fraction of the α-phase

$S_\alpha(\mathbf{x},t)$: saturation of the α-phase

$U_\alpha(\mathbf{x},t)$: volume of the α-phase

$U_\beta(\mathbf{x},t)$: volume of the β-phase

The amount of the α-phase in the porous medium is able to be given by using $\varepsilon_\alpha(\mathbf{x},t)$ or $S_\alpha(\mathbf{x},t)$. The distribution function $\gamma_\alpha(\mathbf{x}+\mathbf{x}')$ is defined as follows :

$$\gamma_\alpha(\mathbf{x}+\mathbf{x}') = \begin{cases} 1 & \text{within } U_\alpha \\ 0 & \text{outside } U_\alpha \end{cases} \tag{2.3.5}$$

The phase average, $\overline{G_\alpha}$, of a general quantity G is defined as:

$$\overline{G_\alpha}(\mathbf{x},t) = \frac{1}{U}\int_u G(\mathbf{x}+\mathbf{x}',t)\gamma_\alpha(\mathbf{x}+\mathbf{x}',t)dU \tag{2.3.6}$$

where the volume of integration, $U = U_\alpha + U_\beta$, is independent of space and time. However U_α and U_β may depend on $\mathbf{x}$ and t. Physically, the phase average is a

property of the α-phase only averaged over the entire volume occupied by the α- and β-phases in the averaging volume(e.g. specific discharge is the phase average of the fluid velocity). Because γ_α is zero in the β-phase, equation(2.3.6) can be rewritten as:

$$\overline{G_\alpha}(\mathbf{x},t)=\frac{1}{U}\int_{U_\alpha(x,t)} G(\mathbf{x}+\mathbf{x}',t)dU \tag{2.3.7}$$

where the limits of integration depend on spatial location and time if the medium deforms or the system is in motion. The intrinsic phase average, $\overline{G_\alpha}^{\alpha}$, of a general quantity G is defined as:

$$\overline{G_\alpha}^{\alpha}(\mathbf{x},t)=\frac{1}{U_\alpha(\mathbf{x},t)}\int_{U_\alpha(x,t)} G(\mathbf{x}+\mathbf{x}',t)dU \tag{2.3.8}$$

This type of average describes a property of the α-phase averaged over that phase only(e.g. the fluid velocity obtained by averaging the point fluid velocities over the volume occupied by the fluid is an intrinsic phase average). Employing equation(2.3.3), comparison of equation(2.3.7) and (2.3.8) indicates that:

$$\overline{G_\alpha}(\mathbf{x},t)=\varepsilon_\alpha(\mathbf{x},t)\overline{G_\alpha}^{\alpha}(\mathbf{x},t) \tag{2.3.9}$$

The deviation of G_α at a point $\mathbf{x}+\mathbf{x}'$ within an REV can be defined as:

$$G'_\alpha(\mathbf{x}+\mathbf{x}',t)=G_\alpha(\mathbf{x}+\mathbf{x}',t)-\overline{G_\alpha}(\mathbf{x},t) \tag{2.3.10}$$

By taking the volumetric phase average, the following relation can be established as:

$$\overline{G'_\alpha}=0 \tag{2.3.11}$$

2.3.2 Averaging theorems

Average of a sum

Let $G_1(\mathbf{x}+\mathbf{x}',t)$ and $G_2(\mathbf{x}+\mathbf{x}',t)$ be two quantities in a phase, and $\overline{G_1}$ and $\overline{G_2}$ be their corresponding averages. Then, the next averaging rule is obtained:

$$\begin{aligned}&\frac{1}{U}\int_{u(\mathbf{x},t)}\{G_1(\mathbf{x}+\mathbf{x}',t)+G_2(\mathbf{x}+\mathbf{x}',t)\}dU\\&=\frac{1}{U}\int_{u(\mathbf{x},t)}G_1(\mathbf{x}+\mathbf{x}',t)dU+\frac{1}{U}\int_{u(\mathbf{x},t)}G_2(\mathbf{x}+\mathbf{x}',t)dU\end{aligned}$$

from which it follows that:

$$\overline{G_1+G_2}=\overline{G_1}+\overline{G_2} \tag{2.3.12}$$

Average of product

$G_1(\mathbf{x}+\mathbf{x}',t)$ and $G_2(\mathbf{x}+\mathbf{x}',t)$ can be rewritten as follows:

$$G_1(\mathbf{x}+\mathbf{x}',t)=\overline{G_1}(\mathbf{x},t)+G_1'(\mathbf{x}+\mathbf{x}',t),\quad G_2(\mathbf{x}+\mathbf{x}',t)=\overline{G_2}(\mathbf{x},t)+G_2'(\mathbf{x}+\mathbf{x}',t)$$

The average of a product can be expanded as:

$$\begin{aligned}\overline{G_1G_2}&=\frac{1}{U}\int_{u(\mathbf{x})}G_1(\mathbf{x}+\mathbf{x}',t)G_2(\mathbf{x}+\mathbf{x}',t)dU\\&=\frac{1}{U}\int_{u(\mathbf{x})}\{\overline{G_1}(\mathbf{x},t)+G_1'(\mathbf{x}+\mathbf{x}',t)\}\{\overline{G_2}(\mathbf{x},t)+G_2'(\mathbf{x}+\mathbf{x}',t)\}dU\\&=\frac{1}{U}\{\overline{G_1}(\mathbf{x},t)\overline{G_2}(\mathbf{x},t)\}\int_{u(\mathbf{x})}dU\\&\quad+\overline{G_1}(\mathbf{x},t)\frac{1}{U}\int_{u(\mathbf{x})}G_2'(\mathbf{x}+\mathbf{x}',t)dU\\&\quad+\overline{G_2}(\mathbf{x},t)\frac{1}{U}\int_{u(\mathbf{x})}G_1'(\mathbf{x}+\mathbf{x}',t)dU\\&\quad+\frac{1}{U}\int_{u(\mathbf{x})}\{G_1'(\mathbf{x}+\mathbf{x}',t)\}\{G_2'(\mathbf{x}+\mathbf{x}',t)\}dU\end{aligned} \tag{2.3.13}$$

Employing equation(2.3.11), the average of a product reduces to the following form:

$$\overline{G_1 G_2} = \overline{G_1}\ \overline{G_2} + \overline{G_1' G_2'} \tag{2.3.14}$$

Spatial average of time derivative

The spatial average of a time derivative can be related to the time derivative of a spatial average. This theorem is used by Whitaker(1973) who considers it to be the transport theorem associated with a point fixed in space. From equation(2.3.6), the phase average of a time derivative becomes:

$$\overline{\frac{\partial G_\alpha}{\partial t}}(\mathbf{x},t) = \frac{1}{U}\int_u \frac{\partial G}{\partial t}(\mathbf{x}+\mathbf{x}',t)\gamma_\alpha(\mathbf{x}+\mathbf{x}',t)dU \tag{2.3.15}$$

Application of the chain rule to the right of equation(2.3.15) yields:

$$\begin{aligned}\overline{\frac{\partial G_\alpha}{\partial t}} = \frac{1}{U}\int_u \frac{\partial}{\partial t}\left[G(\mathbf{x}+\mathbf{x}',t)\gamma_\alpha(\mathbf{x}+\mathbf{x}',t)\right]dU \\ -\frac{1}{U}\int_u G(\mathbf{x}+\mathbf{x}',t)\frac{\partial \gamma_\alpha}{\partial t}(\mathbf{x}+\mathbf{x}',t)dU\end{aligned} \tag{2.3.16}$$

Because U is independent of time, the order of differentiation and integration in the first term on the right side can be changed. Thus, employing equation(2.3.6), equation(2.3.16) can be rewritten as:

$$\overline{\frac{\partial G_\alpha}{\partial t}} = \frac{\partial \overline{G_\alpha}}{\partial t} - \frac{1}{U}\int_u G(\mathbf{x}+\mathbf{x}',t)\frac{\partial \gamma_\alpha}{\partial t}(\mathbf{x}+\mathbf{x}',t)dU \tag{2.3.17}$$

If the α-phase is deforming, γ_α is a function of time and the last term in equation(2.3.17) is non-zero. The total derivative of γ_α with respect to time is:

$$\frac{d\gamma_\alpha}{dt}=\frac{\partial\gamma_\alpha}{\partial t}+\frac{dx_1}{dt}\frac{\partial\gamma_\alpha}{\partial x_1}+\frac{dx_2}{dt}\frac{\partial\gamma_\alpha}{\partial x_2}+\frac{dx_3}{dt}\frac{\partial\gamma_\alpha}{\partial x_3} \tag{2.3.18}$$

$(\partial\gamma_\alpha/\partial x_1),(\partial\gamma_\alpha/\partial x_2),\text{and}(\partial\gamma_\alpha/\partial x_3)$ are non-zero only on the $S_{\alpha\beta}$ interface. If (dx_1/dt), (dx_2/dt), and (dx_3/dt) are chosen to be the velocity components of the interface, the total derivative becomes a material derivative that moves with the interface. Because an observer riding on the interfacial boundary sees no change, this derivative is zero:

$$\frac{d\gamma_\alpha}{dt}=\frac{D\gamma_\alpha}{Dt}=0=\frac{\partial\gamma_\alpha}{\partial t}+\mathbf{u}\cdot\nabla\gamma_\alpha \tag{2.3.19}$$

where **u** is the velocity of the interface. Thus, the following relation can be established as:

$$\frac{\partial\gamma_\alpha}{\partial t}=-\mathbf{u}\cdot\nabla\gamma_\alpha \tag{2.3.20}$$

Substitution of equation(2.3.20) into (2.3.17) yields:

$$\overline{\frac{\partial G_\alpha}{\partial t}}=\frac{\partial\overline{G_\alpha}}{\partial t}+\frac{1}{U}\int_u G(\mathbf{x}+\mathbf{x}',t)\mathbf{u}(\mathbf{x}+\mathbf{x}',t)\cdot\nabla\gamma_\alpha(\mathbf{x}+\mathbf{x}',t)dU \tag{2.3.21}$$

Gray and Lee(1979) have related the spatial derivative of the distribution function to Dirac function as follows:

$$\nabla\gamma_\alpha(\mathbf{x})=-\mathbf{n}_\alpha\delta(\mathbf{x}-\mathbf{x}_{\alpha\beta}) \tag{2.3.22}$$

where $\delta(\mathbf{x}-\mathbf{x}_{\alpha\beta})=\begin{cases}0, & \mathbf{x}\neq\mathbf{x}_{\alpha\beta}\\ \infty, & \mathbf{x}=\mathbf{x}_{\alpha\beta}\end{cases}$ ← Dirac function.

Substituting equation(2.3.22) into the last term of equation(2.3.21), the following equation is obtained:

$$\frac{1}{U}\int_u G(\mathbf{x}+\mathbf{x}',t)\mathbf{u}(\mathbf{x}+\mathbf{x}',t)\cdot\nabla\gamma_\alpha(\mathbf{x}+\mathbf{x}',t)dU = -\frac{1}{U}\int_u G(\mathbf{x}+\mathbf{x}',t)\mathbf{u}(\mathbf{x}+\mathbf{x}',t)\cdot\mathbf{n}_\alpha\delta(\mathbf{x}+\mathbf{x}'-\mathbf{x}_{\alpha\beta},t)dU \tag{2.3.23}$$

The right of equation(2.3.23) involves the delta function which is zero everywhere except at the α-β phase interphase. The value of an integral whose integrand is a δ-function multiplied by some other quantity is just that quantity evaluated at the singular points of the δ-function. Therefore,

$$-\frac{1}{U}\int_u G(\mathbf{x}+\mathbf{x}',t)\mathbf{u}(\mathbf{x}+\mathbf{x}',t)\cdot\mathbf{n}_\alpha\delta(\mathbf{x}+\mathbf{x}'-\mathbf{x}_{\alpha\beta},t)dU = -\frac{1}{U}\int_{S_{\alpha\beta}(\mathbf{x},t)} G_\alpha(\mathbf{x}+\mathbf{x}',t)\mathbf{u}(\mathbf{x}+\mathbf{x}',t)\cdot\mathbf{n}_\alpha dS \tag{2.3.24}$$

Substituting equation(2.3.24) into (2.3.21), the final form can be obtained as follows:

$$\overline{\frac{\partial G_\alpha}{\partial t}} = \frac{\partial \overline{G_\alpha}}{\partial t} - \frac{1}{U}\int_{S_{\alpha\beta}} G_\alpha\mathbf{u}\cdot\mathbf{n}_\alpha dS \tag{2.3.25}$$

where **u** is velocity of interface. Equation(2.3.25) is a relationship between the spatial average of a time derivative and the time derivative of a spatial average.

Spatial average of spatial derivative

The last theorem of interest relates the average of a gradient to the gradient of an average and was developed by Slattery(1967) and Whitaker(1967). The average of the spatial derivative of G within the α-phase can be represented by employing equation(2.3.6):

$$\overline{\nabla G_\alpha} = \frac{1}{U}\int_u [\nabla G(\mathbf{x}+\mathbf{x}',t)]\gamma_\alpha(\mathbf{x}+\mathbf{x}',t)dS \tag{2.3.26}$$

Application of the chain rule to the terms on the right of equation(2.3.26) yields:

$$\overline{\nabla G_\alpha} = \frac{1}{U}\int_u \nabla[G(\mathbf{x}+\mathbf{x}',t)\gamma_\alpha(\mathbf{x}+\mathbf{x}',t)]dU - \frac{1}{U}\int_u G(\mathbf{x}+\mathbf{x}',t)[\nabla\gamma_\alpha(\mathbf{x}+\mathbf{x}',t)]dU \tag{2.3.27}$$

Substitution of equation(2.3.22) into equation(2.3.27) yields:

$$\begin{aligned}\overline{\nabla G_\alpha} = &\frac{1}{U}\int_u \nabla[G(\mathbf{x}+\mathbf{x}',t)\gamma_\alpha(\mathbf{x}+\mathbf{x}',t)]dU \\ &+\frac{1}{U}\int_u G(\mathbf{x}+\mathbf{x}',t)\mathbf{n}_\alpha\delta_\alpha(\mathbf{x}+\mathbf{x}'-\mathbf{x}_{\alpha\beta},t)dU\end{aligned} \tag{2.3.28}$$

According to the definition of δ-function, the last integral of equation(2.3.28) can be rewritten as follows:

$$\frac{1}{U}\int_u G(\mathbf{x}+\mathbf{x}',t)\mathbf{n}_\alpha\delta_\alpha(\mathbf{x}+\mathbf{x}'-\mathbf{x}_{\alpha\beta},t)dU = \frac{1}{U}\int_{S_{\alpha\beta(x,t)}} G_\alpha(\mathbf{x}+\mathbf{x}',t)\mathbf{n}_\alpha dS \tag{2.3.29}$$

Then, equation(2.3.28) becomes:

$$\overline{\nabla G_\alpha} = \frac{1}{U}\int_u \nabla[G(\mathbf{x}+\mathbf{x}',t)\gamma_\alpha(\mathbf{x}+\mathbf{x}',t)]dU + \frac{1}{U}\int_{S_{\alpha\beta(x,t)}} G_\alpha\mathbf{n}_\alpha dS \tag{2.3.30}$$

If ∇ on the right of equation(2.3.30) is considered to be ∇_x, it may be removed from the integral sign because the volume of integration has been specified to be independent of **x**. Thus,

$$\overline{\nabla G_\alpha} = \nabla_x\left[\frac{1}{U}\int_u G(\mathbf{x}+\mathbf{x}',t)\gamma_\alpha(\mathbf{x}+\mathbf{x}',t)dU\right] + \frac{1}{U}\int_{S_{\alpha\beta(x,t)}} G_\alpha\mathbf{n}_\alpha dS \tag{2.3.31}$$

Adopting equation(2.3.6), equation(2.3.31) can be rewritten as:

$$\overline{\nabla G_\alpha} = \nabla \overline{G_\alpha} + \frac{1}{U}\int_{S_{\alpha\beta}} G_\alpha \mathbf{n}_\alpha dS \tag{2.3.32}$$

Finally the averaging theorem for spatial derivatives has been derived.

2.4 Macroscopic mass balance equation

The averaging rules have been derived in the previous section. The macroscopic balance equation is able to be obtained by integrating the microscopic balance equation over the REV with the help of the averaging rules. A general quantity in the α-phase is considered, letting all other phases the β-phase. To begin with, the microscopic balance equation(2.2.10) is rewritten as:

$$\frac{\partial g_\alpha}{\partial t} = -\nabla \cdot (g_\alpha \mathbf{V}^\alpha + \mathbf{J}_\alpha^{GU}) + \rho^\alpha \Gamma_\alpha^G + \rho^\alpha f_\alpha^G \tag{2.4.1}$$

where the total flux of G is decomposed into an advective flux and a diffusive flux $\mathbf{J}_\alpha^{GU}$, $\mathbf{J}_\alpha^{GU} = g_\alpha(\mathbf{V}_\alpha^G - \mathbf{V}^\alpha)$, which represents the net influx of G per unit volume of the phase per unit time.

By integrating equation (2.4.1) over the α-phase present within u, the domain of the REV, and dividing the result by the REV, U, the following equation can be obtained:

$$\frac{1}{U}\int_{u_\alpha} \frac{\partial g_\alpha}{\partial t} dU = -\frac{1}{U}\int_{u_\alpha} \nabla \cdot (g_\alpha \mathbf{V}^\alpha + \mathbf{J}_\alpha^{GU}) dU + \frac{1}{U}\int_{u_\alpha} \rho^\alpha \Gamma_\alpha^G dU + \frac{1}{U}\int_{u_\alpha} \rho^\alpha f_\alpha^G dU \tag{2.4.2}$$

Employing equation(2.3.7), equation (2.4.2) takes the form:

$$\overline{\frac{\partial g_\alpha}{\partial t}} = -\overline{\nabla \cdot (g_\alpha \mathbf{V}^\alpha + \mathbf{J}_\alpha^{GU})} + \overline{\rho^\alpha \Gamma_\alpha^G} + \overline{\rho^\alpha f_\alpha^G} \tag{2.4.3}$$

Adopting equation(2.3.8) and (2.3.3), equation(2.4.3) can be rewritten as:

$$\varepsilon_\alpha \overline{\frac{\partial g_\alpha}{\partial t}}^{\alpha} = -\varepsilon_\alpha \overline{\nabla \cdot (g_\alpha \mathbf{V}^\alpha + \mathbf{J}_\alpha^{GU})}^{\alpha} + \varepsilon_\alpha \overline{\rho^\alpha \Gamma_\alpha^G}^{\alpha} + \varepsilon_\alpha \overline{\rho^\alpha f_\alpha^G}^{\alpha} \tag{2.4.4}$$

By employing equation(2.3.25) and (2.3.32), equation(2.4.4) can be rewritten as follows:

$$\begin{aligned} &\frac{\partial}{\partial t}\varepsilon_\alpha \overline{g_\alpha}^{\alpha} - \frac{1}{U}\int_{S_{\alpha\beta}} g_\alpha \mathbf{u}\cdot\mathbf{n}dS \\ &= -\nabla\cdot\varepsilon_\alpha \overline{(g_\alpha \mathbf{V}^\alpha + \mathbf{J}_\alpha^{GU})}^{\alpha} - \frac{1}{U}\int_{S_{\alpha\beta}} (g_\alpha \mathbf{V}^\alpha + \mathbf{J}_\alpha^{GU})\cdot\mathbf{n}dS + \varepsilon_\alpha \overline{\rho^\alpha \Gamma_\alpha^G}^{\alpha} + \varepsilon_\alpha \overline{\rho^\alpha f_\alpha^G}^{\alpha} \end{aligned} \tag{2.4.5}$$

or

$$\begin{aligned} &\frac{\partial \varepsilon_\alpha \overline{g_\alpha}^{\alpha}}{\partial t} = -\nabla\cdot\varepsilon_\alpha \overline{(g_\alpha \mathbf{V}^\alpha + \mathbf{J}_\alpha^{GU})}^{\alpha} - \frac{1}{U}\int_{S_{\alpha\beta}} g_\alpha (\mathbf{V}^\alpha - \mathbf{u})\cdot\mathbf{n}dS \\ &\quad (1) \qquad\qquad (2) \qquad\qquad (3) \\ &- \frac{1}{U}\int_{S_{\alpha\beta}} \mathbf{J}_\alpha^{GU}\cdot\mathbf{n}dS + \varepsilon_\alpha \overline{\rho^\alpha \Gamma_\alpha^G}^{\alpha} + \varepsilon_\alpha \overline{\rho^\alpha f_\alpha^G}^{\alpha} \\ &\quad (4) \qquad\qquad (5) \qquad\qquad (7) \end{aligned} \tag{2.4.6}$$

where

(1) Rate of increase of G(in the α-phase), per unit volume of porous medium.

(2) Net influx of G by averaged advection and diffusion, per unit volume of porous medium.

(3) Amount of G entering the phase, through the interface surface, $S_{\alpha\beta}$, of the phase within U, per unit volume of porous medium and per unit time, by advection with respect to the (possibly moving) $S_{\alpha\beta}$-surface.

(4) Same as (3), but by diffusion through $S_{\alpha\beta}$.

(5) Amount of G generated by sources of G within U, per unit volume of porous medium and per unit time.

(6) Amount of G generated by external supply, per unit volume of porous medium and per unit time.

By equation(2.3.14), the averaged advective flux, $\overline{g^\alpha \mathbf{V}^\alpha}^\alpha$, may be decomposed into two fluxes: a flux $\overline{g'\mathbf{V}'}^\alpha$ and a macroscopic advective flux $\overline{g^\alpha}^\alpha \overline{\mathbf{V}^\alpha}^\alpha$. With these fluxes, (2.4.6) is rewritten in the form:

$$\frac{\partial \varepsilon_\alpha \overline{g^\alpha}^\alpha}{\partial t} = -\nabla \cdot \varepsilon_\alpha (\overline{g^\alpha}^\alpha \overline{\mathbf{V}^\alpha}^\alpha + \overline{g'\mathbf{V}'}^\alpha + \overline{\mathbf{J}_\alpha^{GU}}^\alpha) \\ -\frac{1}{U}\int_{S_{\alpha\beta}} \left\{ g^\alpha (\mathbf{V}^\alpha - \mathbf{u}) + \mathbf{J}_\alpha^{GU} \right\} \cdot \mathbf{n} dS + \varepsilon_\alpha \overline{\rho^\alpha \Gamma_\alpha^G}^\alpha + \varepsilon_\alpha \overline{\rho^\alpha f_\alpha^G}^\alpha \tag{2.4.7}$$

Equation (2.4.7) is the general macroscopic differential balance equation of a general quantity, G, of a phase. Comparing (2.4.7) with the microscopic mass balance equation (2.2.10), it can be observed that the macroscopic equation contains two additional terms, introduced as a result of the averaging process. The first term is $\overline{g'\mathbf{V}'}^\alpha$ which is the flux of G in excess of the average advection of G by the phase. The other is $-\frac{1}{U}\int_{S_{\alpha\beta}} \left\{ g^\alpha (\mathbf{V}^\alpha - \mathbf{u}) + \mathbf{J}_\alpha^{GU} \right\} \cdot \mathbf{n} dS$ which expresses the influx of G across the $S_{\alpha\beta}$-surface, which separates the considered phase from all other phases within U, by advection relative to the possibly moving $S_{\alpha\beta}$-surface and by diffusion.

2.5 Boundary conditions

To obtain the solutions from the closed set of balance equations, it is necessary to provide supplementary information such as initial conditions and boundary conditions. In the microscopic analysis, the boundary simply indicates the contact area between two phases. However, the macroscopic boundary is based on hypothetical assumption.

2.5.1 Microscopic boundary condition

The microscopic differential balance equation for the general quantity, G, is developed in section 2.2. If the quantity is energy or heat, it can be transported across the microscopic(physical) boundary between two phases. On the other hand, if the quantity can not be transported across the boundary, the boundary is defined as the *material boundary*. For example, a fluid-solid boundary is material to fluid mass, but not to energy. In the absence of sources and sinks of a general quantity on the boundary, the amount of the quantity should be conserved as it is being transported across the boundary.

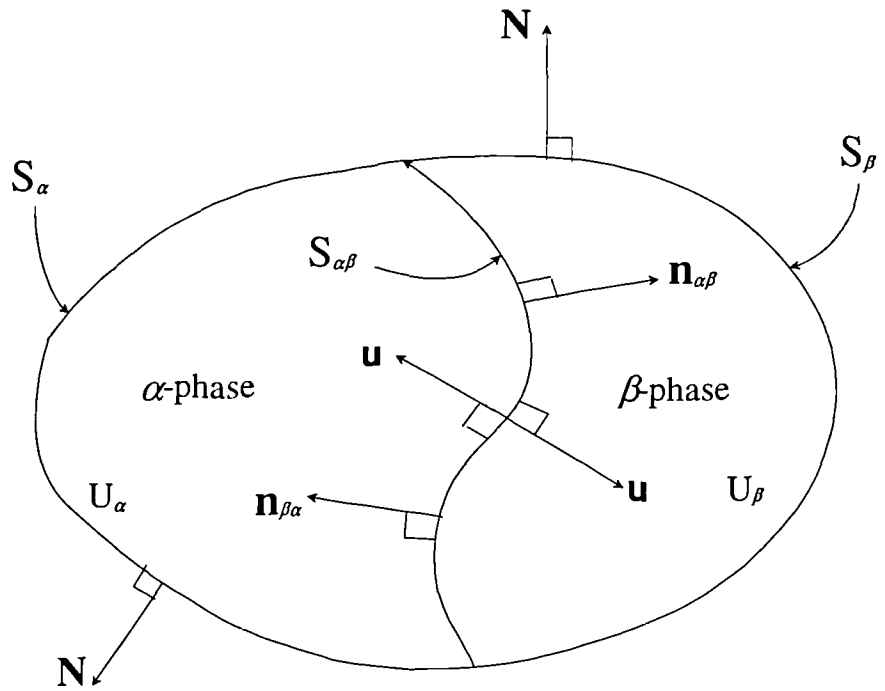


Figure 2.5.1 Microscopic boundary

An arbitrary portion of the system which contains the α-and β-phases is chosen to derive the general microscopic boundary equation(see figure 2.5.1). Here, the surface surrounding U_α and U_β is not the material surface with respect to the quantity G. Assuming that the general quantity exists throughout the whole volume($U_\alpha + U_\beta$), the microscopic mass balance equation can be applied to the concerned volume.

For the purpose of deriving the microscopic boundary condition, the material approach is adopted as in subsection 2.2.2. Equation(2.2.9) is derived assuming that the observer follows the material boundary that no amount of the general quantity, G, can cross. However, to derive the microscopic boundary condition, the observer must follow the selected boundary that is not material. The balance equations of the general quantity, G, can be given for the given α- and β-phase portions respectively, replacing $\mathbf{V}^{G}$, the velocity of G, in equation(2.2.9) with $\mathbf{u}$ the velocity of the boundary.

For theα-phase in figure 2.5.1, U_{α}, the balance equation can be established as:

$$\begin{aligned}&\int_{U_{\alpha}(t)}\left(\frac{\partial g}{\partial t}+\nabla\cdot g\mathbf{V}^{G}-\rho\Gamma^{G}-\rho f^{G}\right)dU\\&\quad=\int_{S_{\alpha\beta}(t)}g(\mathbf{V}^{G}-\mathbf{u})\cdot\mathbf{n}_{\alpha\beta}dS+\int_{S_{\alpha}(t)}g(\mathbf{V}^{G}-\mathbf{u})\cdot\mathbf{N}dS\end{aligned}\tag{2.5.1}$$

For the β-phase in figure 2.5.1, U_{β}, the balance equation can be established as:

$$\begin{aligned}&\int_{U_{\beta}(t)}\left(\frac{\partial g}{\partial t}+\nabla\cdot g\mathbf{V}^{G}-\rho\Gamma^{G}-\rho f^{G}\right)dU\\&\quad=\int_{S_{\alpha\beta}(t)}g(\mathbf{V}^{G}-\mathbf{u})\cdot\mathbf{n}_{\beta\alpha}dS+\int_{S_{\beta}(t)}g(\mathbf{V}^{G}-\mathbf{u})\cdot\mathbf{N}dS\end{aligned}\tag{2.5.2}$$

Then, the whole boundary surface surrounding the α-and β-phases should be considered to derive the boundary condition. Considering the sources, Γ^{SG}, of G on $S_{\alpha\beta}$, the following balance equation of G over $U_{\alpha}+U_{\beta}$ can be obtained:

$$\begin{aligned}&\int_{U_{\alpha}(t)+U_{\beta}(t)}\left(\frac{\partial g}{\partial t}+\nabla\cdot g\mathbf{V}^{G}-\rho\Gamma^{G}-\rho f^{G}\right)dU\\&\quad=-\int_{S_{\alpha\beta}(t)}\Gamma^{SG}dS+\int_{S_{\alpha}(t)+S_{\beta}(t)}g(\mathbf{V}^{G}-\mathbf{u})\cdot\mathbf{N}dS\end{aligned}\tag{2.5.3}$$

By adding equation (2.5.1) and (2.5.2), and comparing the sum with equation (2.5.3) the following equation can be established as:

$$\int_{S_{\alpha\beta}(t)} \left\{ \left[g(\mathbf{V}^G - \mathbf{u}) \right]_{\alpha,\beta} \cdot \mathbf{n} + \Gamma^{SG} \right\} dS = 0 \tag{2.5.4}$$

where $[\ ]_{\alpha,\beta} = [\ ]_{\alpha} - [\ ]_{\beta}$

$[\ ]_{\alpha}$: the quantity in the bracket is from the α-phase side

$\mathbf{n} = \mathbf{n}_{\alpha\beta} = -\mathbf{n}_{\beta\alpha}$

Since equation (2.5.4) is valid for any size of $S_{\alpha\beta}$, the integrand must be zero at every point of $S_{\alpha\beta}$. Thus the boundary condition should be as follows:

$$\left[g(\mathbf{V}^G - \mathbf{u}) \right]_{\alpha,\beta} \cdot \mathbf{n} + \Gamma^{SG} = 0 \tag{2.5.5}$$

2.5.2 Macroscopic boundary condition

Two typical examples of the macroscopic boundary are the environment of the concerned domain and two adjacent porous medium subdomains of different solid matrix porosities. For case of the discontinuity of porosities, a macroscopic boundary can be shown for the single fluid phase(see figure 2.5.2).

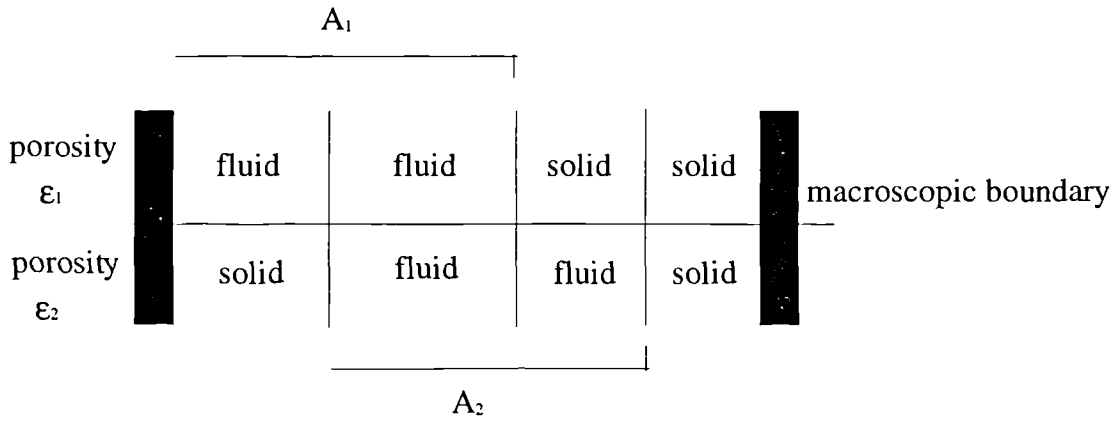


Figure 2.5.2 Macroscopic boundary(single fluid)

where $A_1 \neq A_2$. It can be observed that there must be fluid-solid, solid-solid and fluid-fluid contact areas at the macroscopic boundary across which there is discontinuity in porosity. The same idea can be extended to multiple fluid phases in the void space.

Unlike the microscopic boundary, an idealized boundary is chosen at the macroscopic level, which assumes an abrupt change in porosity across the boundary(figure 2.5.2). To derive a general macroscopic boundary condition for the general quantity, G, an arbitrarily chosen(mathematical) boundary should be considered within a domain on the basis of the concept of the idealized macroscopic boundary. The general quantity, G, is assumed to exist and transfer through all phases.

Following the same procedures used in deriving the microscopic balance equation, a macroscopic boundary condition is obtained as follows(see Bear and Bachmat, 1990):

$$\left[\overline{g(\mathbf{V}^G - \mathbf{u}))}\right]_{1,2} \cdot \mathbf{n} + \overline{\Gamma^{SG}} = 0 \tag{2.5.6}$$

where subscript 1 and 2 indicate each side of the boundary. Thus it shows that the difference in total flux between two sides, 1 and 2 is balanced by the possible source of G on the boundary. Equation (2.5.6) can be rewritten in the form:

$$\sum_{(\alpha)}\left[\varepsilon_\alpha \overline{g_\alpha (\mathbf{V}_\alpha^G - \mathbf{u})}^\alpha\right]_{1,2} \cdot \mathbf{n} + \sum_{(\alpha)} \varepsilon_\alpha \overline{\Gamma_\alpha^{SG}}^\alpha = 0 \tag{2.5.7}$$

or

$$\sum_{(\alpha)}\left[\varepsilon_\alpha \left\{\overline{g_\alpha}^\alpha \left(\overline{\mathbf{V}_\alpha}^\alpha - \mathbf{u}\right) + \mathbf{J}_\alpha^{*G} + \overline{\mathbf{J}_\alpha^{GU}}^\alpha\right\}\right]_{1,2} \cdot \mathbf{n} + \sum_{(\alpha)} \varepsilon_\alpha \overline{\Gamma_\alpha^{SG}}^\alpha = 0 \tag{2.5.8}$$

where $\sum_{(\alpha)} \varepsilon_\alpha = 1$, $\mathbf{J}_\alpha^{*G} (= \overline{g' \mathbf{V}'}^\alpha)$ denotes the dispersive flux of G and $\overline{\mathbf{J}_\alpha^{GU}}^\alpha$ denotes the averaged diffusive flux of G. Equation (2.5.8) represents the general macroscopic boundary condition for any general quantity, G, in a porous medium, and expresses the notion that G does not accumulate on the boundary. In the absence of sources on the boundary, (i.e., $\Gamma^{SG} = 0$), equation (2.5.8) reduces to the continuity of the total flux across the boundary

$$\sum_{(\alpha)}\left[\varepsilon_\alpha\left\{\overline{g_\alpha}^{\alpha}\left(\overline{\mathbf{V}_\alpha}^{\alpha}-\mathbf{u}\right)+\mathbf{J}_\alpha^{*\,G}+\overline{\mathbf{J}_\alpha^{GU}}^{\alpha}\right\}\right]_{1,2}\cdot\mathbf{n}=0 \tag{2.5.9}$$

However, practically the boundary conditions take the form of specification of the values of state variables or of fluxes on the boundary. In other words, assuming that side 1 of a boundary is the considered porous medium domain and side 2 is the external environment, the information related to the external side must be given by one of the two ways.

Firstly, if the general quantity, G(**x**, t), is specified on the external side of a boundary segment, the boundary condition can be simply represented as follows:

$$G(\mathbf{x},t)=f_1(\mathbf{x},t) \tag{2.5.10}$$

where $f_1(\mathbf{x},t)$ is known at all points of the boundary. The condition is referred as a boundary condition of the first kind, or a Dirichlet condition.

Secondly, if the total flux, $f_2(\mathbf{x},t)$, is known on the external side of a boundary, the macroscopic boundary condition can be obtained from equation(2.5.9) as:

$$\sum_{\alpha}\left[\varepsilon_\alpha\left\{\overline{g_\alpha}^{\alpha}\left(\overline{\mathbf{V}_\alpha}^{\alpha}-\mathbf{u}\right)+\mathbf{J}_\alpha^{*\,G}+\overline{\mathbf{J}_\alpha^{GU}}^{\alpha}\right\}\right]_1=f_2(\mathbf{x},t) \tag{2.5.11}$$

where $f_2(\mathbf{x},t)$ is a known function which is not affected by any processes that take place in the concerned domain. This boundary condition is known as the boundary condition of the second type, or a Neumann condition.

CHAPTER III
MULTIPHASE FLOWS IN POROUS MEDIA

In this chapter, the physical and mathematical descriptions of multiphase flow are developed. Section 3.1 is dedicated to describing the physical phenomena for multiphase flows in porous media. Then the governing equations that can be used for the numerical model are developed in section 3.2.

3.1 General description of multiphase flow systems

The term, *immiscible fluids*, is used to indicate fluid phases which at the microscopic level maintain a distinct surface that separates them. A component of a fluid phase may cross this interface and dissolve and diffuse in a fluid phase present on the other side of the surface of separation. Similarly, a volatile component of a liquid may evaporate by crossing the gas-liquid boundary. However, under certain conditions, in spite of these transfer phenomena, the two adjacent fluid phases continue to keep a distinctive interface between them. This study assumes that at least two immiscible fluid phases flow through the solid matrix. However, since four immiscible fluid phases rarely flow simultaneously in porous media, three-phase (water, gas and organic phase) flows are assumed to be the maximum number considered in this study. Unlike in single phase flows, we must consider the physical phenomena describing the interactions between fluid phases. In subsection 3.1.2, the concept of capillary pressure is introduced, relating it to the interfacial tension. Capillary pressure curves are introduced in subsection 3.1.3 which represents the relationships between capillary pressures and saturations that can be obtained by experimental work. Darcy's law is introduced in subsection 3.1.4. Finally subsection 3.1.5 deals with dispersion and diffusion phenomena.

3.1.1 Interfacial tension and wettability

A liquid molecule in a porous medium is influenced by two attraction forces : cohesion and adhesion. Cohesion acts in the interior of the liquid, while adhesion acts between two liquids, liquid and gas, or liquid and solid matrix. The interfacial tension is caused by the difference of the cohesive and adhesive forces on the interface. Assume that water phase is in contact with the gas phase. A water molecule is attracted equally in all directions within the water phase, because it is surrounded by the same kind of molecules that have the same cohesive force. It applies to the gas phase as well. However, a water molecule on the interface is not attracted equally by the surrounding molecules because the adhesive force between water and gas molecules is weaker than the cohesive force between water molecules. Thus, it is natural that the water molecule moves towards the interior of the water phase. As a consequence of the pull towards the water phase interior, the surface of the water phase always tends to contract as much as possible to its interior. It results in the formation of a thin film at the interface between the two phases which is capable of sustaining tension. This property is known as *interfacial tension.* The interfacial tension between a liquid and its vapour is known as *surface tension.* The same phenomenon takes place at the interface between a liquid and a gas, or any two immiscible liquids.

At the molecular level, no sharp surface of separation exists, instead, a gradual transition takes place, across a relatively narrow zone from the phase occupied only by one kind of molecules to that occupied only by molecules of the other kind. Because of the different behaviour of the molecules in this transition zone, this zone is conceptually replaced, as an approximation, by an interface that is assumed to separate the two phases. In multiphase flows, transport of mass, or mass of a component may take place through this interface, which is a curved two-dimensional surface.

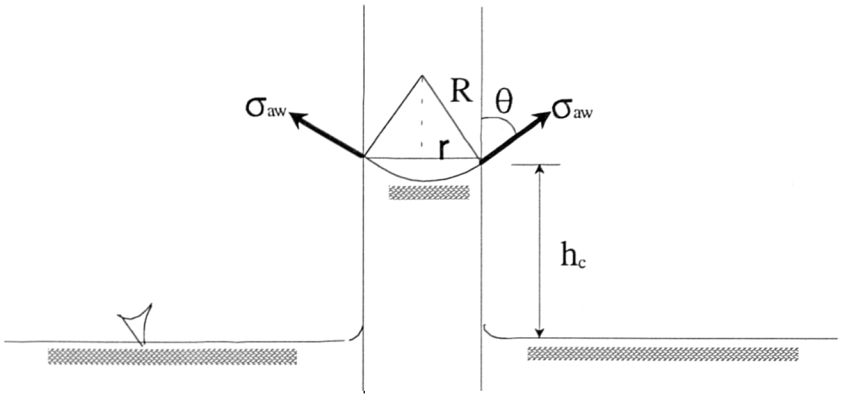


Figure 3.1.1 Capillary rise through tube

One of the simplest measurements of surface tension is by means of capillary rise. As shown in figure 3.1.1, water rises through a tube of circular cross section of radius r, because the affinity of water for the tube is greater than that of air. For sufficiently small diameter tubes, it is reasonable to assume a spherical interface, so that the principal radius of curvature becomes $R = r/\cos\theta$. An equilibrium may be established along vertical direction, yielding:

$$\sigma_{aw} = \frac{h_c \rho_w g r}{2\cos\theta} \tag{3.1.1}$$

where σ_{aw} : surface tension

g : gravity

ρ_w : water density

The shape of the interface depends on the geometry, say the configuration of the solid matrix, and relationships among phases. But it is a rule that, as observed in a small falling drop of a liquid, the interface between two immiscible phases tends to contract, making as small a contacting area as possible.

For a more general description of interfacial tension and the shape of interface between fluid phases, assume two immiscible fluid phases(a liquid and a gas phase) that are in contact with a plane solid surface(see figure 3.1.2).

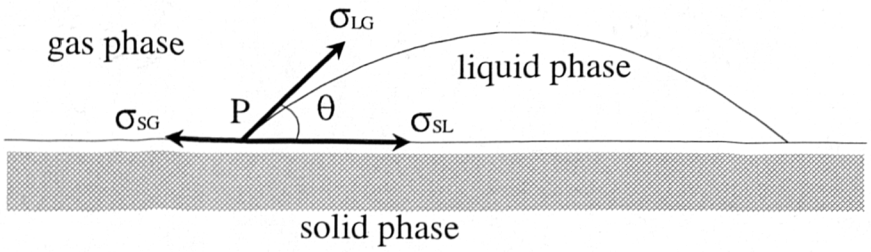


Figure 3.1.2 Interfacial tensions and contact angle

Three interfacial tensions can be observed at point P. If σ_{ij} indicates the interfacial tension between i phase and j phase, i and j may be S, L, or G to specify solid, liquid, or gas respectively. The angle θ, called contact angle, or wetting angle, indicates the angle between the solid surface and the liquid-gas, or liquid-liquid interface, measured through the liquid that tends to be more adhesive to the solid plane.

Equilibrium along the horizontal plane yields:

$$\sigma_{SG} = \sigma_{LG}\cos\theta + \sigma_{SL}, \quad \text{or} \quad \cos\theta = \frac{\sigma_{SG} - \sigma_{SL}}{\sigma_{LG}} \tag{3.1.2}$$

The equation, called Young's equation, relates the wetting angle to the interface tensions of the three phases. In the case that $\sigma_{SG} > \sigma_{LG} + \sigma_{SL}$, the state of equilibrium does not exit. The liquid spreads indefinitely over the solid surface, wetting the solid perfectly. According to the contact angle, θ, it can be decided which fluid is wetting the solid phase more preferentially. If θ is less than 90 , it is said that the liquid wets the solid and is called the wetting phase, whereas the gas is called the nonwetting one. On the other hand, if θ is more than 90 , the liquid is said to be the nonwetting phase, while the gas phase is the wetting phase. But practically the latter rarely takes place. If two liquids are in contact with the solid phase, θ plays a more important role to decide the wettability. As mentioned before, this study deals with water, organic, and gas phase. Normally water is the most adhesive phase to the solid phase among the three

phases. If the solid matrix is composed of hydrophilic material, water is the wetting phase and gas is the nonwetting phase. The wettability of organic phase lies between water and gas. However, if the solid matrix is oliophilic, the behavior of water and organic phases are interchanged.

3.1.2 Capillary pressure

For multiphase flows in porous media, there is a discontinuity in pressure across a curved interface that separates any two immiscible fluids. The shape of the interface between two fluids in a porous medium depends on the configuration of porous media and pressures of the two fluids on the interface. Let us assume an infinitesimal element of a curved interface between water and air phases (see figure 3.1.3).

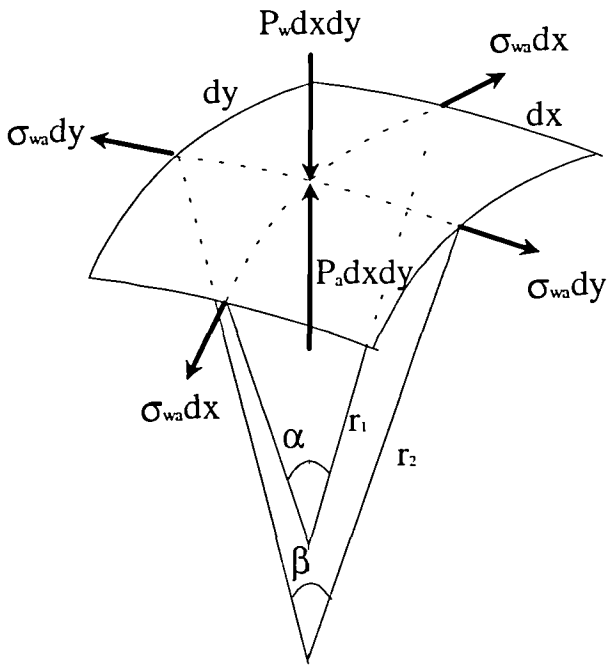


Figure 3.1.3 Infinitesimal element of a curved interface

Assuming the interfacial tension, σ_{wa} , to be constant, a balance equation of force component normal to this element can be established as follows:

$$P_a - P_w = \sigma_{wa}\left(\frac{1}{r'} + \frac{1}{r''}\right) \tag{3.1.3}$$

where r', r'' : two principal radii of curvature

σ_{wa} : 72.75 dyne/cm at 20 C

Equation(3.1.3) is known as the Laplace formula. The left hand side of the equation(3.1.3) represents the discontinuity of the two adjacent pressures, while the right hand side consists of interfacial tension and curvature of the interface. Assuming the interfacial tension to be constant, equation(3.1.3) relates the pressure difference to the shape of interface. The difference in pressure of the two contacting fluids is known as the *capillary pressure*, P_{nw} :

$$P_{nw} = P_n - P_w \tag{3.1.4}$$

where P_n : pressure of nonwetting phase

P_w : pressure of wetting phase

Let us introduce the mean curvature κ_m:

$$\kappa_m = \frac{1}{r_m} = \frac{1}{2}\left(\frac{1}{r'} + \frac{1}{r''}\right)$$

where r_m : harmonic mean of the principal radius of curvature

So equation(3.1.3) can be written as follows:

$$P_{aw} = \frac{2\sigma_{wa}}{r_m} \tag{3.1.5}$$

where water is the wetting phase and air is the nonwetting phase. Thus it can be said that the interface curvature at a point on the interface is related to the capillary pressure at that point.

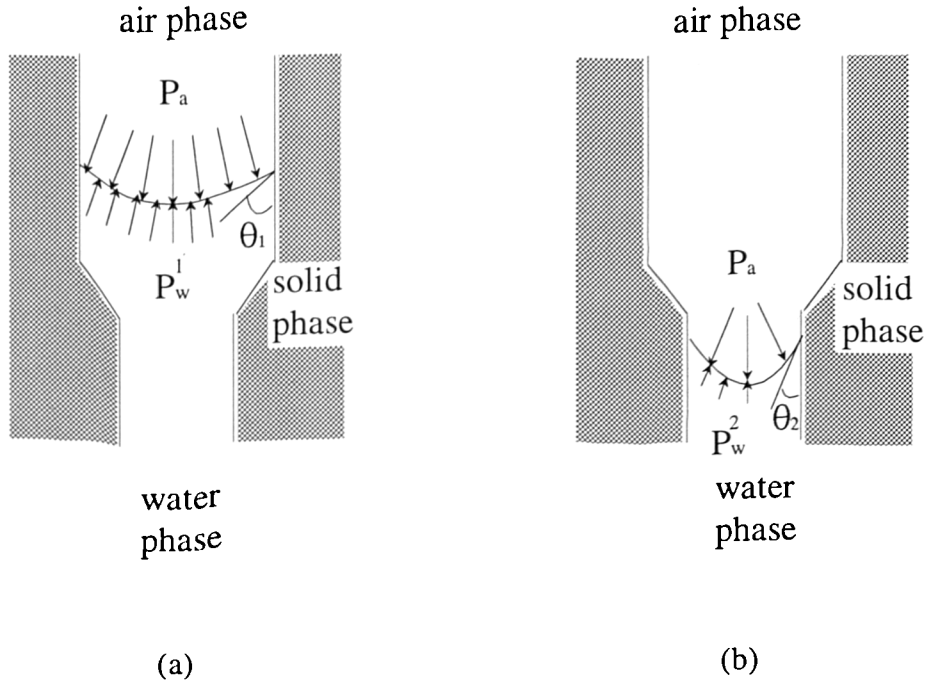


Figure 3.1.4 Movement of interface caused by the change of capillary pressure

Figure 3.1.4 shows the plane views of two cases, where the water and air phases form an interface in a solid column that has wide passage in upper part and narrow passage in lower part. In figure 3.1.4(a), the two fluids are in equilibrium between capillary pressure , $P_a - P_w^1$, and interfacial tension, making curvature whose radius is r_1. However, if P_w changes from P_w^1 to P_w^2 by some reasons such as water table drop, the balance does not hold any more. Thus another balance may be established as in figure 3.1.4(b). To sustain the bigger capillary pressure, the radius, r_2 , must be smaller than r_1 . As a consequence, it can be said that the change of capillary pressure causes the movement of multiple fluids in porous media. If P_w becomes P_w^1 again, water rises. However, water may not rise to the same point as it was, because the shape of porous media or the properties of solid matrix may influence the system when the air-water

interface is advancing or receding on the solid surface. This phenomenon is called hysteresis. It will be referred again in the next subsection.

3.1.3 Capillary pressure curve

Even if equation(3.1.5) is valid mathematically, it is not suitable to apply practically, because there is no way to measure the principal radii of curvature of interface. Thus another expression that is practical and holds the concept of equation (3.1.5) is required.

For a water-air fluid system in a porous medium, the smaller radius means that the capillary pressure is bigger on the interface between the two fluid phases. Thus the largest curvature (smallest radius of curvature) can be found in the narrowest pores, where the capillary pressure is the biggest. The biggest capillary pressure indicates that water exists only in the narrowest pores, whereas water in larger pores is withdrawn until water meets the narrowest pores that can resist the corresponding capillary pressure. Generally the larger pores empty at low capillary pressures, while the narrow pores, supporting interfaces of larger curvatures, empty at higher capillary pressures. Because pores have different sizes and shapes even at the same height, some of them may contain water, if they are narrow enough to resist the corresponding capillary pressure, while others are empty.

Therefore, it can be said that the quantity of water within an REV is related to the capillary pressure. Another expression for water quantity is water saturation. As a consequence, the capillary pressure can be related to the saturation of a fluid. A capillary pressure curve, $P_{aw} = P_{aw}(S_w)$, or a capillary pressure head curve, $h_c = h_c(S_w)$, can be obtained by experiments for a certain porous medium which contains water and air phases. In soil sciences, these curves are called retention curves, as they show how much water is retained in the soil by capillary forces against gravity. The capillary pressure curves can be applicable to any two phase flows in porous media, for instance, water-organic or organic-gas.

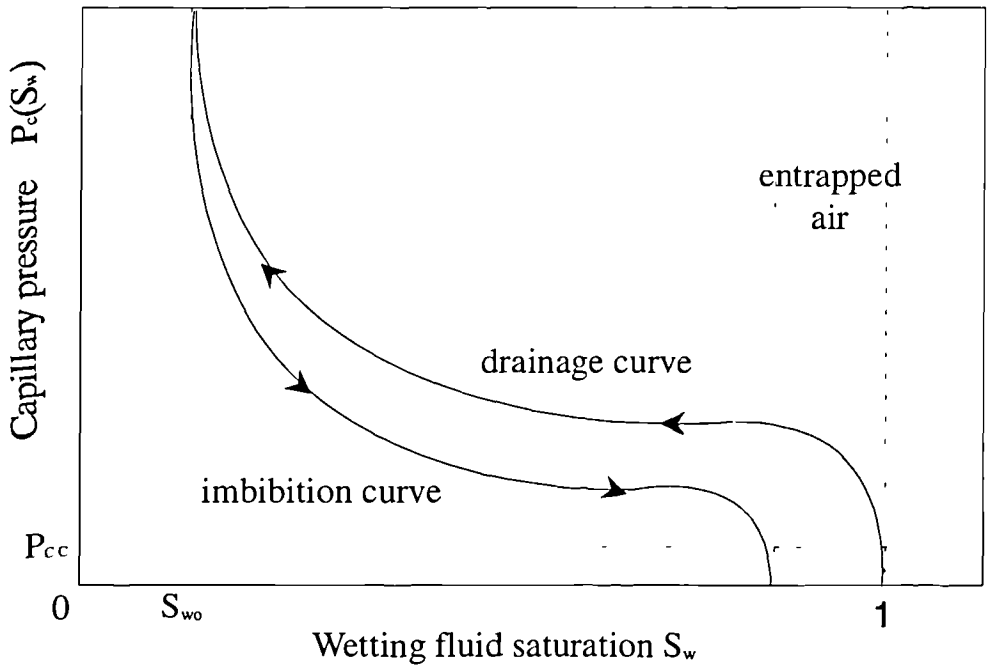


Figure 3.1.5 Capillary pressure curve

A typical form of capillary pressure curve is shown in figure 3.1.5. Figure 3.1.5 shows that the capillary pressure curves may be different, depending on whether the wetting phase, water, is replaced by the air phase or the nonwetting phase, air, is replaced by the water phase. So let us call the process desaturation, dewetting, or drainage that the wetting phase is being displaced by a nonwetting phase. Conversely, if the nonwetting phase is displaced by a wetting phase, it is known as wetting or imbibition.

The two different capillary pressure curves for the same fluids and solid matrix are due to hysteresis. Generally the complex configuration of pore space causes hysteresis. This study introduces four causes for it.

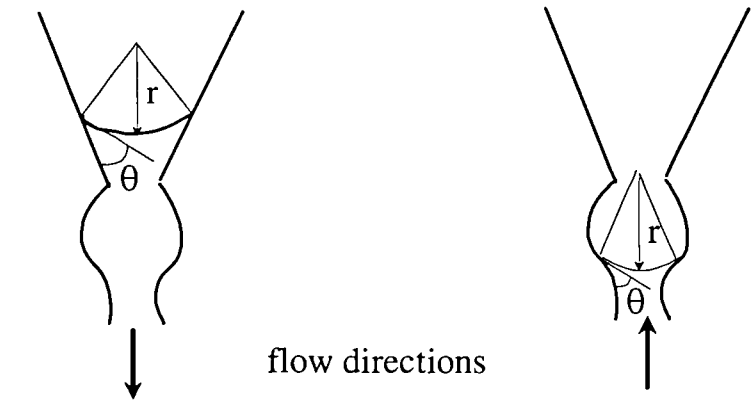


Figure 3.1.6 Ink bottle effect

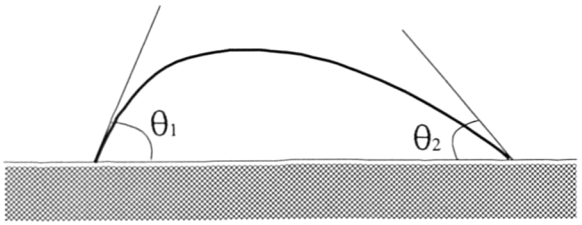


Figure 3.1.7 Rain drop effect

Firstly the ink-bottle effect may cause hysteresis. As shown in figure 3.1.6, the shape of pore space with interchanging narrow and wide passages makes the ink-bottle effect. During drainage and rewetting, interfaces having the same curvature occur at different elevations, thus yielding different wetting fluid saturations for the same capillary pressure. The second one is the raindrop effect. Figure 3.1.7 shows that the contact angle at the advancing trace of an interface on a solid, differs from that at the receding one, because of impurities in the minerals that compose the surface and the roughness. The third reason is due to the entrapment of nonwetting fluid, as an initially saturated sample is drained and then rewet. Finally, consolidation, swelling and shrinkage of the solid matrix, as it is dried and wet, may cause hysteresis in the capillary pressure curve.

Figure 3.1.5 is given by starting from a fully water-saturated sample. Even if a small quantity of wetting fluid is drained at first stage, nonwetting fluid will not enter the sample, until a certain capillary pressure is reached as shown in figure 3.1.5. The corresponding pressure P_{cc} , is called the critical pressure, threshold pressure, or bubbling pressure. If the critical pressure is reached, the air begins to occupy the larger pores, reducing water saturation S_w.

As shown in figure 3.1.5, the drainage curve stops at S_{wo}, residual water saturation, or irreducible water saturation. Because soil particles may absorb water and water may remain in very tiny pores, complete drainage can not be achieved. Likewise, imbibition curve can not reach the stage of complete wetting because of entrapped air, denoted by S_{no}. Since residual fluid saturations depend on solid matrix and pair of fluids, they should be decided by experiments.

In two phase flows, there is only one capillary pressure, which makes it possible to decide the capillary pressure curve by experiments. Brooks and Corey(1964), Corey(1986), Scheidegger(1974), and Su and Brooks(1980) contribute to obtaining the capillary pressure curves for the two-phase systems. However, unlike the two-phase systems, it is not easy to establish the relationship experimentally between saturations and capillary pressures for a three-phase system. On top of this, the concept of the capillary pressure curve itself is based on the two-phase systems. Thus some scholars adopt assumptions linking two phase data to obtain the three-phase data. For reasonable assumptions, it is important to decide the degree of affinity to the solid matrix among the three fluid phases, say, water, organic and gas. If the porous matrix consists of hydrophilic material, water is the wetting phase. The affinity of organic to the solid matrix is somewhere between those of water and gas. So it is defined as an intermediate wetting phase. Gas is the nonwetting phase with respect to other phases. Occasionally organic phase may be the wetting phase for a certain solid phase and water becomes the intermediate wetting phase. This study considers both hydrophilic and oliophilic solid matrix.

Various authors, using two-phase data, have tried to establish the relationships between saturations and capillary pressures in three-phase flow systems. For hydrophilic solid matrix, Leverett(1941) proposes that the total liquid saturation in the three-phase fluid system is the function of gas-organic capillary pressure: $S_l = f(P_{og})$.

Aziz and Settari(1979) assume that water saturation in a water-gas-organic system is the function of water-organic capillary pressure: $S_w = f(P_{ow})$. Thus, all the three fluid saturations can be determined by simple manipulations: $S_o = S_l - S_w$, $S_g = 1 - S_l$.

Lenhard and Parker(1988) develop an experimental apparatus to directly measure functional relationships between fluid saturations and capillary pressures of three-phase(gas-oil-water) or two-phase fluid systems in consolidated porous media. The experiment proves that the two-phase gas-oil data can be used to predict total liquid content in the gas-oil-water systems and that the two-phase oil-water data can be used to predict water saturations in the three-phase systems for rigid water wet porous media subject to monotonic drainage saturation paths.

On the other hand, Abriola and Pinder(1985a,b) adopts the two phase water-gas data to predict the gas saturation in the three-phase systems for hydrophilic solid matrix: $S_w = f(P_{ow})$, $S_g = f(P_{wg})$, $S_o = 1 - S_w - S_g$.

To describe flow in the subsurface correctly, it is be very important to decide which method should be used for a certain problem. This research has been made to deal with various assumptions and experiments. However, most of numerical models for analyzing multiphase flows in porous media adopt fixed assumptions and experiments.

3.1.4 Fluid velocity in porous media

Darcy's law can be applied to even multiphase flows for deciding the velocity of a fluid phase. Unlike single phase flows, the influence of the presence of other fluid phases should be considered. However the pressure gradient and gravity are still driving forces which make a fluid phase move. The general form of Darcy's law which is commonly used to describe flow in porous media is presented for the α-phase in a multiphase flow:

$$\mathbf{V}^{\alpha} = -\left(\frac{\mathbf{k}k_{r\alpha}}{\varepsilon S_{\alpha}\mu_{\alpha}}\right)(\nabla P_{\alpha} - \rho_{\alpha}\mathbf{g}) \qquad (3.1.7)$$

where

$\mathbf{V}^{\alpha}$: average velocity of α-phase

$\mathbf{k}$: solid matrix permeability

$k_{r\alpha}$: relative permeability to α-phase flow

$\mathbf{g}$: gravity acceleration

The solid matrix permeability, **k**, reflects the pure characteristics of the solid matrix such as pore configuration, or roughness of the surface. It does not depend on the properties of the fluid flowing in it. On the other hand, it is not easy to decide the relative permeability $\kappa_{r\alpha}$ of the α-phase, because it is influenced by other fluids. For single phase flows, the relative permeability $\kappa_{r\alpha}$, surely becomes unity. However in case of more than one fluid phase, it is subjected to the saturations of all fluid phases in a porous media. For instance, the relative permeability of the α-phase increases but less than 1, if the saturation of the α-phase increases. For two phase flows, many scholars have conducted experimental works and developed mathematical expressions to relate the relative permeability to the capillary pressure between two fluids.

Van Genuchten(1980) derives a mathematical expression, based on Mualem's(1976) theory, which is

$$\kappa_{r\alpha} = \frac{(1-(ah_c)^{n-1}(1+(ah_c)^n)^{-m})^2}{(1+(ah_c)^n)^{m/2}} \tag{3.1.8}$$

where a : parameter to be fit

h_c : capillary head between the wetting and nonwetting fluid based on the water density, $h_c = \dfrac{-P_{nw}}{\rho_w g}$

n : parameter to be fit

m : $1-1/n$

However, for three phase flows, the relative permeabilities of fluid phases can not be measured directly. Stone(1970) assumes that in a three-phase system, water relative

permeability and water saturation are functions of capillary pressure P_{ow}, resulting that they are identical to the two-phase system relations. Similarly, in both cases of two-phase flows and three-phase flows, gas phase relative permeability and gas saturation are subject to gas-organic capillary pressure. Stone's assumption is based on the wettability order, water-organic-gas, to the solid matrix : no direct contact between water and gas. However, the relative permeability of the organic phase κ_{ro} , can not be obtained directly from the two-phase data, whereas the saturation of the organic phase S_o , can be easily obtained by the constraint: $S_0 = 1 - S_w - S_g$. Thus Stone suggested the following relations:

$$S_o^* = \frac{S_o - S_{om}}{1 - S_{wir} - S_{om}} \quad , \quad for\ S_o \geq S_{om}$$

$$S_w^* = \frac{S_w - S_{wir}}{1 - S_{wir} - S_{om}} \quad , \quad for\ S_w \geq S_{wir}$$

$$S_g^* = \frac{S_g}{1 - S_{wir} - S_{om}}$$

where S_{wir} : residual water saturation

S_{om} : minimum value of the residual organic saturation

Then the organic relative permeability is given as follows:

$$\kappa_{ro} = S_o^* \Omega_w \Omega_g \tag{3.1.9}$$

where $\Omega_w = \frac{\kappa_{row}}{1 - S_w^*}$, $\Omega_g = \frac{\kappa_{rog}}{1 - S_g^*}$

κ_{row} : relative permeability of organic to water in a two phase system

κ_{rog} : relative permeability of organic to gas in a two phase system

In the region where $S_w \leq S_{wir}$, κ_{ro} is assumed to be a function of gas saturation only. For $S_o \leq S_{om}$, κ_{ro} is set to zero. Stone(1973) produces another predictive equation for organic relative permeability.

$$\kappa_{ro} = (\kappa_{row} + \kappa_{rw})(\kappa_{rog} + \kappa_{rg}) - (\kappa_{rw} + \kappa_{rg}) \tag{3.1.10}$$

3.1.5 Dispersion and Diffusion

Dispersion is caused by the difference between the average velocity calculated from Darcy's law and the actual velocity of a fluid. The deviations from an average advective flux of solute mass must be considered properly.

Regardless of velocity difference, a molecule of a species moves through another species in a fluid phase. The phenomenon is known as *diffusion*. Bear and Bachmat(1967) develop the following equation to consider dispersion and diffusion for α-phase in a multiphase system:

$$D_{ij}^{\alpha} = D_{ij}^{m\alpha} + a_{ijkm}^{\alpha} \frac{\mathbf{V}_k^{\alpha} \mathbf{V}_m^{\alpha}}{\mathbf{V}^{\alpha}} \tag{3.1.11}$$

where $\mathbf{V}^{\alpha}$: average velocity of α-phase

$\mathbf{V}_k^{\alpha}$, $\mathbf{V}_m^{\alpha}$: velocity components in the coordinate directions

$D_{ij}^{m\alpha}$: molecular diffusion tensor

a_{ijkm}^{α} : dispersivity tensor

Equation (3.1.11) is also derived by Nikolaevski(1959) using a statistical approach and analogy with the theory of turbulence.

Bear(1979) introduces the longitudinal dispersivity, a_L^{α}, and the transversal dispersity, a_T^{α}, which can be given by experiments. The longitudinal dispersivity, a_L^{α}, is related to a diffusion coefficient which causes dispersion forward and backward along the local

direction of fluid flow. The transversal dispersity, a_T^α, is related to a diffusion coefficient which causes dispersion evenly in the directions perpendicular to the local flow direction. The dispersivity tensor is expressed as follows:

$$a_{ijkm}^\alpha = a_T^\alpha \delta_{ij}\delta_{km} + \frac{a_L^\alpha - a_T^\alpha}{2}\left(\delta_{ik}\delta_{jm} + \delta_{im}\delta_{jk}\right) \tag{3.1.12}$$

where δ_{ij} : Kronecker delta

Employing equation(3.1.12), equation(3.1.11) can be rewritten for two dimensions as follows:

$$D_{ij}^\alpha = \begin{bmatrix} D_{xx}^{m\alpha} + (a_T^\alpha \mathbf{V}_y^{\alpha\,2} + a_L^\alpha \mathbf{V}_x^{\alpha\,2})/\mathbf{V}^\alpha & D_{xy}^{m\alpha} + (a_L^\alpha - a_T^\alpha)\mathbf{V}_x^\alpha \mathbf{V}_y^\alpha / \mathbf{V}^\alpha \\ D_{yx}^{m\alpha} (a_L^\alpha - a_T^\alpha)\mathbf{V}_x^\alpha \mathbf{V}_y^\alpha / \mathbf{V}^\alpha & D_{yy}^{m\alpha} + (a_T^\alpha \mathbf{V}_x^{\alpha\,2} + a_L^\alpha \mathbf{V}_y^{\alpha\,2})/\mathbf{V}^\alpha \end{bmatrix} \tag{3.1.13}$$

3.2 Governing Equations

The mathematical expressions which govern the multiphase flows in porous media are derived in this section by employing the concept of mass balance. As a first step, in subsection 3.2.1, the mass balance for a general component present in the multiphase system is obtained from the general balance equation in section 2.4. Subsection 3.2.2 shows components and phases under consideration in this study. Subsection 3.2.3 expands the mass balance equation in subsection 3.2.1 adopting the general primary variables.

In subsection 3.2.4, the mass balance equation for the general component is applied to all components in this system. Subsection 3.2.5 derives the water phase equation by combining two component equations in the water phase. Finally subsection 3.2.6 is set to mention the application range of the mathematical model derived in this chapter.

3.2.1 General mass balance equation

In chapter II, the balance equation of the general quantity, G, was developed for a multiphase flow in the subsurface. Mass transport has been a major concern for analyzing the multiphase flows. In order to obtain the general mass balance equation, g_α, the density of the general quantity, G, in equation (2.4.7) is replaced by ρ_c^α, the density of mass of a component, c, in the α-phase:

$$\frac{\partial \varepsilon_\alpha \overline{\rho_c^\alpha}^\alpha}{\partial t} = -\nabla \cdot \varepsilon_\alpha (\overline{\rho_c^\alpha}^\alpha \overline{\mathbf{V}^\alpha}^\alpha + \overline{\rho_c' \mathbf{V}'}^\alpha + \overline{\mathbf{J}_\alpha^{M_c U}}^\alpha) - \frac{1}{U}\int_{S_{\alpha\beta}} \left\{\rho_c^\alpha (\mathbf{V}^\alpha - \mathbf{u}) + \mathbf{J}_\alpha^{M_c U}\right\} \cdot \mathbf{n} dS + \varepsilon_\alpha \overline{\rho_c^\alpha \Gamma_\alpha^{M_c}}^\alpha + \varepsilon_\alpha \overline{\rho_c^\alpha f_\alpha^{M_c}}^\alpha \tag{3.2.1}$$

where M_c : mass of component c

$\rho_c^\alpha = M_c / U_{o\alpha}$, density of component c in α-phase

Equation (3.2.1) contains microscopic values such as $\mathbf{V}^\alpha, \mathbf{u}, \text{and } \mathbf{V}'$. Thus proper constraints and assumptions are required to get rid of microscopic values.

Since this study allows the interphase mass exchange through dissolution between two contacting liquid phases and evaporation between liquid and gas phase, a component can be found in all fluid phases. Thus the equation 3.2.1 is not suitable for describing the movement of the component, because it is restricted to only one fluid phase. The sum of the component mass balance equations over all fluid phases is required to trace the movements of the component c. So the complete form of the general mass balance equation for the component is given as follows:

$$\sum_\alpha \left\{ \frac{\partial \varepsilon_\alpha \overline{\rho_c^\alpha}^\alpha}{\partial t} + \nabla \cdot \varepsilon_\alpha (\overline{\rho_c^\alpha}^\alpha \overline{\mathbf{V}^\alpha}^\alpha + \overline{\rho_c' \mathbf{V}'}^\alpha + \overline{\mathbf{J}_\alpha^{M_c U}}^\alpha) + \frac{1}{U}\int_{S_{\alpha\beta}} \left\{\rho_c^\alpha (\mathbf{V}^\alpha - \mathbf{u}) + \mathbf{J}_\alpha^{M_c U}\right\} \cdot \mathbf{n} dS - \varepsilon_\alpha \overline{\rho_c^\alpha \Gamma_\alpha^{M_c}}^\alpha - \varepsilon_\alpha \overline{\rho_c^\alpha f_\alpha^{M_c}}^\alpha \right\} = 0 \tag{3.2.2}$$

The following constrains can be adopted for equation(3.2.2).

(1) The sum of all components' densities over the α-phase is equal to the density of the α-phase.

$$\sum_{c} \rho_c^{\alpha} = \rho^{\alpha} \tag{3.2.3}$$

(2) The third term of equation(3.2.2) represents the flux of a component to (or) from the α-phase. So the sum of the term over all components existing in the α-phase is equal to the mass gained(or lost) by that phase.

$$-\sum_{c} \frac{1}{U} \int_{S_{\alpha\beta}} \left\{ \rho_c^{\alpha} (\mathbf{V} - \mathbf{u}) + \mathbf{j}^{M_c U} \right\} \cdot \mathbf{n} dS = -\frac{1}{U} \int_{S_{\alpha\beta}} \left\{ \rho^{\alpha} (\mathbf{V} - \mathbf{u}) \right\} \cdot \mathbf{n} dS \tag{3.2.4}$$

(3) The mass of the total system is conserved. So the following constraint is given as:

$$-\sum_{\alpha} \frac{1}{U} \int_{S_{\alpha\beta}} \left\{ \rho^{\alpha} (\mathbf{V}^{\alpha} - \mathbf{u}) \right\} \cdot \mathbf{n} dS = 0 \tag{3.2.5}$$

(4) The mass of each component is conserved over the entire system.

$$-\sum_{\alpha} \frac{1}{U} \int_{S_{\alpha\beta}} \left\{ \rho_c^{\alpha} (\mathbf{V}^{\alpha} - \mathbf{u}) + \mathbf{J}_{\alpha}^{M_c U} \right\} \cdot \mathbf{n} dS = 0 \tag{3.2.6}$$

Employing constraint (4) to get rid of the third term of the left hand side of (3.2.2), the mass balance equation of the component, c, in a system is given as follows.

$$\sum_{\alpha}\left\{\frac{\partial \varepsilon_{\alpha}\overline{\rho^{\alpha}W_{c}^{\alpha}}^{\alpha}}{\partial t}+\nabla\cdot\varepsilon_{\alpha}(\overline{\rho^{\alpha}W_{c}^{\alpha}}^{\alpha}\overline{\mathbf{V}^{\alpha}}^{\alpha}+\overline{\rho_{c}'\mathbf{V}'}^{\alpha}+\overline{\mathbf{J}_{\alpha}^{M_{c}U}}^{\alpha})\right.$$
$$\left.-\varepsilon_{\alpha}\overline{\rho^{\alpha}\Gamma_{\alpha}^{M_{c}}}^{\alpha}-\varepsilon_{\alpha}\overline{\rho^{\alpha}f_{\alpha}^{M_{c}}}^{\alpha}\right\}=0 \qquad (3.2.7)$$

where $W_{c}^{\alpha}=\dfrac{\rho_{c}^{\alpha}}{\rho^{\alpha}}$, mass fraction of component c

But it still contains microscopic values in the flux terms, $\mathbf{V}'$, and ρ_{c}'. They are concerned with diffusive and dispersive flux. They can be written as the macroscopically non-advective flux of the component, c :

$$\overline{\mathbf{D}_{c}^{\alpha}}^{\alpha}=\varepsilon_{\alpha}(\overline{\rho_{c}'\mathbf{V}'}^{\alpha}+\overline{\mathbf{J}_{\alpha}^{M_{c}U}}^{\alpha}) \qquad (3.2.8)$$

Equation(3.2.8) will be rewritten in terms of macroscopic values in Chapter IV. Let us remove the averaging symbol ($\overline{\quad}^{\alpha}$) for simplicity. So, otherwise stated, all variables represent macroscopic values from now on. Equation(3.2.7) can be reexpressed as:

$$\sum_{\alpha}\left\{\frac{\partial}{\partial t}(\varepsilon_{\alpha}\rho^{\alpha}W_{c}^{\alpha})+\nabla\cdot(\varepsilon_{\alpha}\rho^{\alpha}W_{c}^{\alpha}\mathbf{V}^{\alpha})+\nabla\cdot\mathbf{D}_{c}^{\alpha}-\varepsilon_{\alpha}\rho^{\alpha}\Gamma_{\alpha}^{M_{c}}-\varepsilon_{\alpha}\rho^{\alpha}f_{\alpha}^{M_{c}}\right\}=0 \qquad (3.2.9)$$

where $\overline{\rho^{\alpha}W_{c}^{\alpha}}\equiv\overline{\rho^{\alpha}}\,\overline{W_{c}^{\alpha}}$

In this study, internal production and external supply terms will not be dealt with. Thus equation (3.2.9) can be simplified as follows:

$$\sum_{\alpha}\left\{\frac{\partial}{\partial t}(\varepsilon_{\alpha}\rho^{\alpha}W_{c}^{\alpha})+\nabla\cdot(\varepsilon_{\alpha}\rho^{\alpha}W_{c}^{\alpha}\mathbf{V}^{\alpha})+\nabla\cdot\mathbf{D}_{c}^{\alpha}\right\}=0 \qquad (3.2.10)$$

3.2.2 Components and phases under consideration

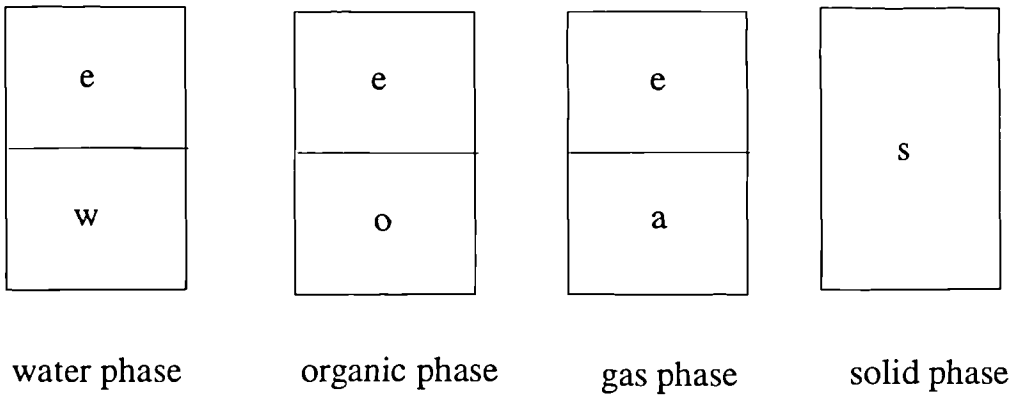
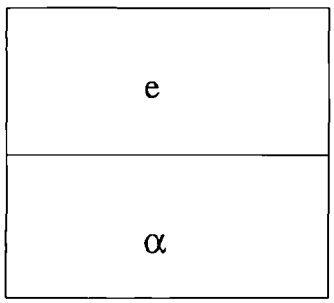


Figure 3.2.1 Components of phases under consideration

Water, organic, gas, and solid phases are assumed to occupy the entire domain in this study. Each phase is composed of at best two components as shown in figure 3.2.1. Unlike other phases, the solid phase has only one component, since this study does not consider any possibilities of adsorption of a fluid component to the soil phase. An extensive component, e, is introduced in this thesis. The e component in the water phase may be a solute(water-gas system) or a component of the organic phase which is able to dissolve in water. On the other hand, the e component in the organic and gas phase is always the component of the organic phase that is soluble and volatile. Whereas the o component in the organic phase is assumed to be non-soluble and non-volatile.

Let us designate soil , gas, water and organic phase as S, G, W and O subscript respectively and soil, air, water, organic, extensive component as s, a, w, o and e respectively.

3.2.3 General mass balance equation for component c



α-phase

Figure 3.2.2 Components of α-phase

Figure 3.2.2 shows a general phase which consists of two component e and α. Here the α-phase may be one of the three fluid phases-water organic, or gas phase. To derive the mass balance equation for the general component c, equation(3.2.10) is employed again:

$$\sum_{\alpha}\left\{\frac{\partial}{\partial t}(\varepsilon_{\alpha}\rho^{\alpha}W_{c}^{\alpha})+\nabla\cdot(\varepsilon_{\alpha}\rho^{\alpha}W_{c}^{\alpha}\mathbf{V}^{\alpha})+\nabla\cdot\mathbf{D}_{c}^{\alpha}\right\}=0 \qquad (3.2.11)$$

where $\varepsilon_{\alpha}=\varepsilon S_{\alpha}$

c : water(w), organic(o), air(a), or e component

The first term can be recognized as the total change in c component's mass of the α-phase in void space in the course of time. The second term represents contributions to c component's mass change due to excess of fluid inflows over outflows at a point. The third term is for diffusion and dispersion of the c component in the α-phase. In this thesis, the general primary variables are adopted for solving various pollution patterns in the subsurface. The three general primary variables are defined as G_1 , G_2 , and G_3 . The primary variables can be capillary pressures or mass fractions. They are

determined by the flow patterns(see table 4.3.1). For three phase flows, G_1 and G_2 are capillary pressures and G_3 is mass fraction, W_e^o. Equation(3.2.11) are expanded for the three-fluid phase system in this section. The first term of equation.(3.2.11) can be expanded as follows:

$$\frac{\partial}{\partial t}(\varepsilon S_\alpha \rho^\alpha W_c^\alpha) = S_\alpha \rho^\alpha W_c^\alpha \frac{\partial \varepsilon}{\partial t} + \varepsilon \rho^\alpha W_c^\alpha \frac{\partial S_\alpha}{\partial t} + \varepsilon S_\alpha W_c^\alpha \frac{\partial \rho^\alpha}{\partial t} + \varepsilon S_\alpha \rho^\alpha \frac{\partial W_c^\alpha}{\partial t} \tag{3.2.12}$$

Time derivatives of ε, S_α, and ρ^α must be rewritten in terms of the general primary variables.

To express $\frac{\partial \varepsilon}{\partial t}$ in terms of the general primary variables, the coefficient of compressibility for porous solid matrix is employed as:

$$\beta_s = -\frac{1}{Vol}\frac{d(Vol)}{d\sigma'} \tag{3.2.13}$$

where Vol : bulk volume of solid matrix

σ' : intergranular stress

If the total stress is nearly constant, any change in intergranular(effective) stress will be due to an equal and opposite change in the fluid pore pressure. Assuming that individual solid grains are incompressible, equation(3.2.13) can be rewritten as:

$$\beta_s = -\frac{1}{\varepsilon_s}\frac{d\varepsilon_s}{d P_{ave}} \tag{3.2.14}$$

where $\varepsilon_s = \frac{Vol_s}{Vol}$

Vol_s : volume of solid grains, constant

P_{ave} : average soil fluid pressure

Thus, the derivative of porosity, $\frac{\partial \varepsilon}{\partial t}$ is expressed as follows:

$$\frac{\partial \varepsilon}{\partial t} = -\frac{\partial \varepsilon_s}{\partial t} = -\frac{\partial \varepsilon_s}{\partial P_{av}} \frac{\partial P_{av}}{\partial t} \tag{3.2.15}$$

$$\frac{\partial \varepsilon}{\partial t} = (1-\varepsilon)\beta_s \frac{\partial P_{av}}{\partial t} \tag{3.2.16}$$

where P_{av} is the average pressure of fluids in void space that is function of water and organic pressures, P_w and P_o. This study employs Abriola's(1985) assumption to define P_{av}:

$$P_{av} = \kappa P_o + (1-\kappa) P_w \tag{3.2.17}$$

where $\kappa = \frac{S_o}{S_o + S_w}$: weighting factor

P_w , P_o : functions of the general primary variables(see table 4.3.3)

Secondly, because saturation is totally dependent on capillary pressures, derivative of saturation, $\frac{\partial S_\alpha}{\partial t}$ can be expanded as follows:

$$\frac{\partial S_\alpha}{\partial t} = \frac{\partial S_\alpha}{\partial G_1} \frac{\partial G_1}{\partial t} + \frac{\partial S_\alpha}{\partial G_2} \frac{\partial G_2}{\partial t} \tag{3.2.18}$$

The density of the α-phase changes according to the pressure of the α-phase and mass fraction. So the compressibility of the α-phase can be defined as:

$$\beta^{\alpha} = \frac{1}{\rho_{\alpha}} \frac{\partial \rho^{\alpha}}{\partial P_{\alpha}}\bigg|_{W_c^{\alpha}} + \frac{1}{\rho_{\alpha}} \frac{\partial \rho^{\alpha}}{\partial W_c^{\alpha}}\bigg|_{P_{\alpha}} \qquad (3.2.19)$$
$$= \beta_P^{\alpha} + \beta_w^{\alpha}$$

where β^{α} = compressibility of α-phase

β_P^{α} = compressibility of α-phase with respect to pressure

β_w^{α} = compressibility of α-phase with respect to mass fraction

Thus,

$$\frac{\partial \rho^{\alpha}}{\partial t} = \rho^{\alpha}\beta_P^{\alpha} \frac{\partial P_{\alpha}}{\partial t} + \rho^{\alpha}\beta_w^{\alpha} \frac{\partial W_c^{\alpha}}{\partial t} \qquad (3.2.20)$$

So time derivative term can be summarized as follows:

$$\begin{aligned} \frac{\partial}{\partial t}\left(\varepsilon S_{\alpha}\rho^{\alpha}W_c^{\alpha}\right) &= S_{\alpha}\rho^{\alpha}W_c^{\alpha}(1-\varepsilon)\beta_s \frac{\partial P_{a_1}}{\partial t} \\ &+ \varepsilon\rho^{\alpha}W_c^{\alpha}\left(\frac{\partial S_{\alpha}}{\partial G_1}\frac{\partial G_1}{\partial t} + \frac{\partial S_{\alpha}}{\partial G_2}\frac{\partial G_2}{\partial t}\right) \\ &+ \varepsilon S_{\alpha}W_c^{\alpha}\rho^{\alpha}\left(\beta_P^{\alpha}\frac{\partial P_{\alpha}}{\partial t} + \beta_w^{\alpha}\frac{\partial W_c^{\alpha}}{\partial t}\right) \\ &+ \varepsilon S_{\alpha}\rho^{\alpha}\frac{\partial W_c^{\alpha}}{\partial t} \end{aligned} \qquad (3.2.21)$$

The second term of equation (3.2.11) can be expanded as follows:

$$\nabla\cdot(\varepsilon S_{\alpha}\rho^{\alpha}W_c^{\alpha}\mathbf{V}^{\alpha}) = \rho^{\alpha}\nabla\cdot(\varepsilon S_{\alpha}W_c^{\alpha}\mathbf{V}^{\alpha}) + \varepsilon S_{\alpha}W_c^{\alpha}\mathbf{V}^{\alpha}\cdot\nabla\rho^{\alpha} \qquad (3.2.22)$$

The first term on the right side of equation(3.2.22) can be written by employing equation (3.1.7):

$$\rho^{\alpha}\nabla\cdot(\varepsilon S_{\alpha}W_c^{\alpha}\mathbf{V}^{\alpha}) = -\rho^{\alpha}\nabla\cdot\left\{W_c^{\alpha}\frac{\mathbf{k}k_{r\alpha}}{\mu_{\alpha}}\left(\nabla P_{\alpha} - \rho^{\alpha}\mathbf{g}\right)\right\} \qquad (3.2.23)$$

The second term on the right side of equation (3.2.22) is expanded by adopting the compressibility coefficients:

$$\varepsilon S_\alpha W_c^\alpha \mathbf{V}^\alpha \cdot \nabla \rho^\alpha = -W_c^\alpha \left\{ \frac{\mathbf{k} k_{r\alpha}}{\mu_\alpha} \left(\nabla P_\alpha - \rho^\alpha \mathbf{g} \right) \right\} \cdot \left(\rho^\alpha \beta_P^\alpha \nabla P_\alpha + \rho^\alpha \beta_w^\alpha \nabla W_c^\alpha \right) \tag{3.2.24}$$

So equation(3.2.22) becomes:

$$\begin{aligned} \nabla \cdot (\varepsilon S_\alpha \rho^\alpha W_c^\alpha \mathbf{V}^\alpha) = & -\rho^\alpha \nabla \cdot \left\{ W_c^\alpha \frac{\mathbf{k} k_{r\alpha}}{\mu_\alpha} \left(\nabla P_\alpha - \rho^\alpha \mathbf{g} \right) \right\} \\ & - W_c^\alpha \left\{ \frac{\mathbf{k} k_{r\alpha}}{\mu_\alpha} \left(\nabla P_\alpha - \rho^\alpha \mathbf{g} \right) \right\} \cdot \left(\rho^\alpha \beta_P^\alpha \nabla P_\alpha + \rho^\alpha \beta_w^\alpha \nabla W_c^\alpha \right) \end{aligned} \tag{3.2.25}$$

Finally the mass balance equation for c component has been derived in terms of the general primary variables:

$$\begin{aligned} \sum_\alpha \Bigg[& S_\alpha \rho^\alpha W_c^\alpha (1-\varepsilon) \beta_s \frac{\partial P_{a\nu}}{\partial t} + \varepsilon \rho^\alpha W_c^\alpha \left(\frac{\partial S_\alpha}{\partial G_1} \frac{\partial G_1}{\partial t} + \frac{\partial S_\alpha}{\partial G_2} \frac{\partial G_2}{\partial t} \right) \\ & + \varepsilon S_\alpha W_c^\alpha \rho^\alpha \left(\beta_P^\alpha \frac{\partial P_\alpha}{\partial t} + \beta_w^\alpha \frac{\partial W_c^\alpha}{\partial t} \right) + \varepsilon S_\alpha \rho^\alpha \frac{\partial W_c^\alpha}{\partial t} - \rho^\alpha \nabla \cdot \left\{ W_c^\alpha \frac{\mathbf{k} k_{r\alpha}}{\mu_\alpha} \left(\nabla P_\alpha - \rho^\alpha \mathbf{g} \right) \right\} \\ & - W_c^\alpha \left\{ \frac{\mathbf{k} k_{r\alpha}}{\mu_\alpha} \left(\nabla P_\alpha - \rho^\alpha \mathbf{g} \right) \right\} \cdot \left(\rho^\alpha \beta_P^\alpha \nabla P_\alpha + \rho^\alpha \beta_w^\alpha \nabla W_c^\alpha \right) + \nabla \cdot \mathbf{D}_c^\alpha \Bigg] = 0 \end{aligned} \tag{3.2.26}$$

3.2.4 Mass balance equations for components

Since equation (3.2.26) is the general mass balance equation, it can be applied to all components : water, organic, air, and e component. From equation (3.2.26), it can be observed that $P_w, P_o, P_g, W_e^w, W_e^o$, and W_e^g are the primary unknowns for the three phase flows. Thus, 6 conditions are required to solve the three-phase systems. The

assumption of a constant gas phase pressure reduces the unknowns to 5. In the case that the organic phase is the source of e component, the relationships between W_e^w, W_e^o, W_e^g can be established by experiments, producing two relevant equations. Consequently three conditions must be obtained from equation (3.2.26). As a first step, let us apply equation (3.2.26) to three components, water, organic, and e component.

Water component equation

Since water component exists only in the water phase, the following equation can be derived by applying equation (3.2.26) to the water component:

$$
\begin{aligned}
&S_w \rho^w W_w^w (1-\varepsilon)\beta_s \frac{\partial P_{av}}{\partial t} + \varepsilon \rho^w W_w^w \left(\frac{\partial S_w}{\partial G_1}\frac{\partial G_1}{\partial t} + \frac{\partial S_w}{\partial G_2}\frac{\partial G_2}{\partial t} \right) \\
&+\varepsilon S_w W_w^w \rho^w \left(\beta_P^w \frac{\partial P_w}{\partial t} + \beta_w^w \frac{\partial W_w^w}{\partial t} \right) + \varepsilon S_w \rho^w \frac{\partial W_w^w}{\partial t} - \rho^w \nabla \cdot \left\{ W_w^w \frac{\mathbf{k} k_{rw}}{\mu_w} \left(\nabla P_w - \rho^w \mathbf{g} \right) \right\} \\
&- W_w^w \left\{ \frac{\mathbf{k} k_{rw}}{\mu_w} \left(\nabla P_w - \rho^w \mathbf{g} \right) \right\} \cdot \left(\rho^w \beta_P^w \nabla P_w + \rho^w \beta_w^w \nabla W_w^w \right) + \nabla \cdot \mathbf{D}_w^w = 0
\end{aligned}
\tag{3.2.27}
$$

Organic component equation

Like the water component, the organic component which is insoluble and non-volatile is assumed to be found only in the organic phase. So the following equation can be obtained as:

$$
\begin{aligned}
&S_o W_o^o (1-\varepsilon)\beta_s \frac{\partial P_{av}}{\partial t} + \varepsilon W_o^o \left(\frac{\partial S_o}{\partial G_1}\frac{\partial G_1}{\partial t} + \frac{\partial S_o}{\partial G_2}\frac{\partial G_2}{\partial t} \right) \\
&+\varepsilon S_o W_o^o \left(\beta_P^o \frac{\partial P_o}{\partial t} + \beta_w^o \frac{\partial W_o^o}{\partial t} \right) + \varepsilon S_o \frac{\partial W_o^o}{\partial t} - \nabla \cdot \left\{ W_o^o \frac{\mathbf{k} k_{ro}}{\mu_o} \left(\nabla P_o - \rho^o \mathbf{g} \right) \right\} \\
&- W_o^o \left\{ \frac{\mathbf{k} k_{ro}}{\mu_o} \left(\nabla P_o - \rho^o \mathbf{g} \right) \right\} \cdot \left(\beta_P^o \nabla P_o + \beta_w^o \nabla W_o^o \right) + \frac{1}{\rho^o} \nabla \cdot \mathbf{D}_o^o = 0
\end{aligned}
$$

(3.2.28)

e component equation

The e component is considered to be observed in all three phases, water, organic and gas. The importance of the gaseous migration of pollutants has been demonstrated by the field and laboratory investigations of various authors and supported by the work of Fried, et al (1979). Pelikan, et al (1978) uses soil-air sampling and analysis to determine the extent of groundwater contamination in case studies. Experimental work with the volatilization of organics from solutions and soils (Dilling(1977), Kilzer, et al (1979)) suggests that the flux of volatiles is controlled by macroscopic diffusion. In accordance with these findings, a Fickian-type of diffusive flux vector will be postulated for $\mathbf{D}_e^g$. The convective movement of the gas phase is neglected in this study. So the following equation can be obtained as follows:

$$
\begin{aligned}
&S_w \rho^w W_e^w (1-\varepsilon)\beta_s \frac{\partial P_{av}}{\partial t} + \varepsilon \rho^w W_e^w \left(\frac{\partial S_w}{\partial G_1}\frac{\partial G_1}{\partial t} + \frac{\partial S_w}{\partial G_2}\frac{\partial G_2}{\partial t} \right) \\
&+\varepsilon S_w W_e^w \rho^w \left(\beta_P^w \frac{\partial P_w}{\partial t} + \beta_w^w \frac{\partial W_w^w}{\partial t} \right) + \varepsilon S_w \rho^w \frac{\partial W_e^w}{\partial t} - \rho^w \nabla \cdot \left\{ W_e^w \frac{\mathbf{k} k_{rw}}{\mu_w} \left(\nabla P_w - \rho^w \mathbf{g} \right) \right\} \\
&-W_e^w \left\{ \frac{\mathbf{k} k_{rw}}{\mu_w} \left(\nabla P_w - \rho^w \mathbf{g} \right) \right\} \cdot \left(\rho^w \beta_P^w \nabla P_w + \rho^w \beta_w^w \nabla W_e^w \right) + \nabla \cdot \mathbf{D}_e^w \\
&+S_o \rho^o W_e^o (1-\varepsilon)\beta_s \frac{\partial P_{av}}{\partial t} + \varepsilon \rho^o W_e^o \left(\frac{\partial S_o}{\partial G_1}\frac{\partial G_1}{\partial t} + \frac{\partial S_o}{\partial G_2}\frac{\partial G_2}{\partial t} \right) \\
&+\varepsilon S_o W_e^o \rho^o \left(\beta_P^o \frac{\partial P_o}{\partial t} + \beta_w^o \frac{\partial W_o^o}{\partial t} \right) + \varepsilon S_o \rho^o \frac{\partial W_e^o}{\partial t} - \rho^o \nabla \cdot \left\{ W_e^o \frac{\mathbf{k} k_{ro}}{\mu_o} \left(\nabla P_o - \rho^o \mathbf{g} \right) \right\} \\
&-W_e^o \left\{ \frac{\mathbf{k} k_{ro}}{\mu_o} \left(\nabla P_o - \rho^o \mathbf{g} \right) \right\} \cdot \left(\rho^o \beta_P^o \nabla P_o + \rho^o \beta_w^o \nabla W_e^o \right) + \nabla \cdot \mathbf{D}_e^o \\
&+ S_g \rho^g W_e^g (1-\varepsilon)\beta_s \frac{\partial P_{av}}{\partial t} + \varepsilon \rho^g W_e^g \left(\frac{\partial S_g}{\partial G_1}\frac{\partial G_1}{\partial t} + \frac{\partial S_g}{\partial G_2}\frac{\partial G_2}{\partial t} \right) \\
&+\varepsilon S_g W_e^g \rho^g \beta_w^g \frac{\partial W_e^g}{\partial t} + \varepsilon S_g \rho^g \frac{\partial W_e^g}{\partial t} + \nabla \cdot \mathbf{D}_e^g = 0
\end{aligned}
$$

(3.2.29)

3.2.5 Water phase equation

In subsection 3.2.4, the three governing equations are derived, which are focused on the mass of components. However it is better to use simpler forms of the governing equations, if possible. Assuming that contaminants in the water phase are slightly soluble, any changes in water properties such as density or viscosity due to the presence of solute are small enough to be neglected. Thus this study assumes that ρ^w and μ_w are not affected by the presence of the e component, and, therefore, β_w^w becomes zero. From figure 3.2.2, it can be observed that $W_w^w + W_e^w = 1$. Hence summing the water component equation and the part of e component equation that is related to the water phase yields a water phase mass balance equation with a simpler form than equation (3.2.27):

$$\begin{aligned} & S_w(1-\varepsilon)\beta_s \frac{\partial P_{av}}{\partial t} + \varepsilon\left(\frac{\partial S_w}{\partial G_1}\frac{\partial G_1}{\partial t} + \frac{\partial S_w}{\partial G_2}\frac{\partial G_2}{\partial t}\right) \\ & + \varepsilon S_w \beta_p^w \frac{\partial P_w}{\partial t} - \nabla\cdot\left\{\frac{\mathbf{k}k_{rw}}{\mu_w}\left(\nabla P_w - \rho^w \mathbf{g}\right)\right\} \\ & - \left\{\frac{\mathbf{k}k_{rw}}{\mu_w}\left(\nabla P_w - \rho^w \mathbf{g}\right)\right\}\beta_p^w \cdot \nabla P_w = 0 \end{aligned} \tag{3.2.30}$$

3.2.6 Application range

As a consequence, equation (3.2.28), (3.2.29), and (3.2.30) are selected as the governing equations for the analysis of the multiphase flows in the subsurface. This model is basically designed for treating various pollution patterns that are subjected to fluid phases in a porous medium, interphase mass exchange, and solid matrix properties.

If polluted water flows through the unsaturated zone, it is considered as two-phase(water and gas phase) flows. It can be solved by using the water phase equation and e component equation, assuming that W_e^o and W_e^g go to zero. It requires only two governing equations.

Pore space may be filled with water and organic phases. In the case that organic matter does not dissolve in water, two governing equations(water phase equation and organic component equation) are needed to trace the movements of the fluid flows. However, if there is an organic component that dissolves in water, all three governing equations are required to analyze the system. Notice that W_e^g should be zero in the e component equation.

If the organic phase infiltrates through a porous medium in which no water is found, the two phase(organic and gas phase) flow system needs only one(organic component) or two(organic and e component) equations for simulation, depending on whether the organic phase contains e component or not.

Even in the three(water, organic, and gas) phase flows, we don't have to use all the three governing equations, if there is no e component in the organic phase. However, all the three governing equations are required, if e component exists in all the fluid phases.

Table 3.2.1 shows the pollution patterns in the subsurface that are able to be solved by the model developed in this study. It also discusses which governing equations are used for the pollution patterns. According to whether the solid matrix is hydrophilic or oliophilic, four more pollution patterns can be added.

	Phase	Component	Water Phase Equation	Organic Component Equation	e Component Equation
Pollution pattern(I)	**W and G**	**w and a**	**Yes**	**No**	**Yes**
Pollution pattern(II)	**W and O**	**w and o**	**Yes**	**Yes**	**No**
Pollution pattern(III)	**W and O**	**w, o and e**	**Yes**	**Yes**	**Yes**
Pollution pattern(IV)	**O and G**	**o and a**	**No**	**Yes**	**No**
Pollution pattern(V)	**O and G**	**o, a and e**	**No**	**Yes**	**Yes**
Pollution pattern(VI)	**W, O and G**	**w, o, and a**	**Yes**	**Yes**	**No**
Pollution pattern(VII)	**W, O and G**	**w, o, a, and e**	**Yes**	**Yes**	**Yes**

Where W : water phase

O : organic phase

G : gas phase

w : water component

o : organic component

a : air component

e : e component

Table 3.2.1 Summary of application range

CHAPTER IV
NUMERICAL APPROACH

The governing equations derived in chapter III can not be solved analytically. The numerical methods have been used to solve these non-linear partial differential equations. Section 4.1 refers to the general finite element method. The governing equations are discretized in section 4.2 using a finite element method. Section 4.3 shows how the general primary variables and the derivatives in the governing equations change according to the pollution patterns.

4.1 Finite element Analysis

The finite difference methods and finite element methods are two approximate methods for solving linear and non-linear partial differential equations. Both methods have been extensively used and developed for numerical modelling of groundwater problems. The choice depends on the complexity of the problem and the user's familiarity with the method. However, the finite element method is more flexible for problems in which the boundaries are irregular or in which the medium is heterogeneous or anisotropic. The basic principle of the finite element method is introduced in this section. The discussion focuses on Galerkin's method which is one of the most popular finite element techniques. Basic definitions and numerical techniques are introduced for the numerical model, COMPO.

4.1.1 Galerkin's method

Considering a 1-dimensional domain which has N node points and $N-1$ elements, the true solution, $u(x)$, for a partial differential equation may be defined continuously throughout the domain.

However, it is very difficult to solve the partial differential equations analytically which describe the subsurface flows. Thus, many scholars make use of numerical methods. The finite element method begins with assuming an approximate solution. Denoting the approximate solution as $\hat{u}(x)$, it can be expressed in terms of the nodal values and interpolation functions:

$$\hat{u}(x) = \sum_{i=1}^{N} u_i \phi_i(x) \tag{4.1.1}$$

where u_i : an approximate value for a primary unknown at node i

$\phi_i(x)$: basis function or interpolation function

N : number of nodes

This study adopts the simplest basis function, $\phi_i(x)$, as shown in figure 4.1.1.

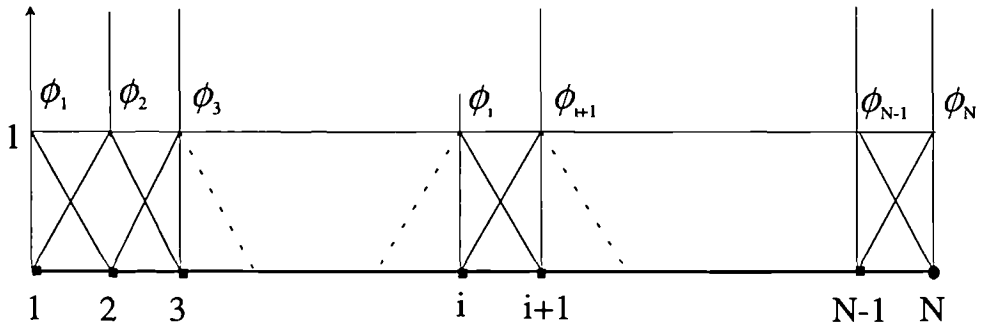


Figure 4.1.1 Basis functions

To prove that an approximate solution is acceptable, an admissibility condition is required. Substituting the approximate solution, $\hat{u}(x)$, into the partial differential equation, $L(u)$, a residual value can be defined as r(x, t): $L(\hat{u}) = r(x,t)$. The problem is how to minimise the residual caused by wild guess.

The residual $r(x,t)$ is caused by employing the approximate solution that contains N approximate values. The residual at each node is a measure of the degree to which the approximate value does not satisfy the partial differential equation. If the residual

$r(x,t)$ is too big, the approximate values must be adjusted. Thus it can be said that the finite element technique is the process of finding suitable approximate values by iteration. Since N unknowns are assumed, N conditions are required to solve. The method of weighted residuals is to make error of approximation unimportant for certain sub-domains of concern. The Galerkin method is one of the weighted residual methods. It adopts the basis function, $\phi_i(x)$, as the weighting function. Then the partial differential equation weighted by the basis function is integrated over the entire domain of the problem. N conditions can be given by making the integration zero for N weighting functions centered on nodes:

$$\int_x L(\hat{u})\phi_i(x)dx = 0 \tag{4.1.2}$$

where i=1,2, ... , N

4.1.2 Basis function in a local coordinate system

The local element is employed, because it allows efficient organization of the integration. For one-dimensional analysis, a local coordinate and basis functions are shown in figure 4.1.2.

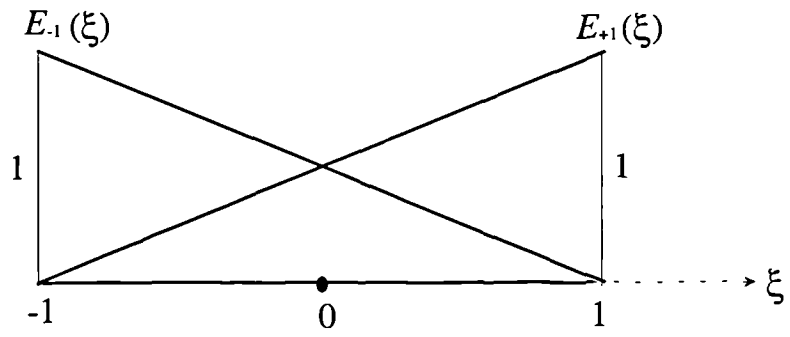


Figure 4.1.2 Basis functions in the local coordinate system

The one dimensional basis functions are described over the local element as follows:

$$E_{-1}(\xi) = \frac{1}{2}(1-\xi)$$
$$E_{+1}(\xi) = \frac{1}{2}(1+\xi) \qquad (4.1.3)$$

where $-1 \le \xi \le 1$

The linear basis functions are continuous in the ξ coordinate and have either a value of zero or one depending on whether ξ is +1 or -1. Derivatives of the basis functions are given as follows:

$$\frac{\partial E_{-1}}{\partial \xi} = -\frac{1}{2}, \quad \frac{\partial E_{+1}}{\partial \xi} = \frac{1}{2} \qquad (4.1.4)$$

For the two dimensional problems, this study adopts a basic element that consists of four node points. The element is a square in the local coordinates, ξ and η, as shown in figure 4.1.3.

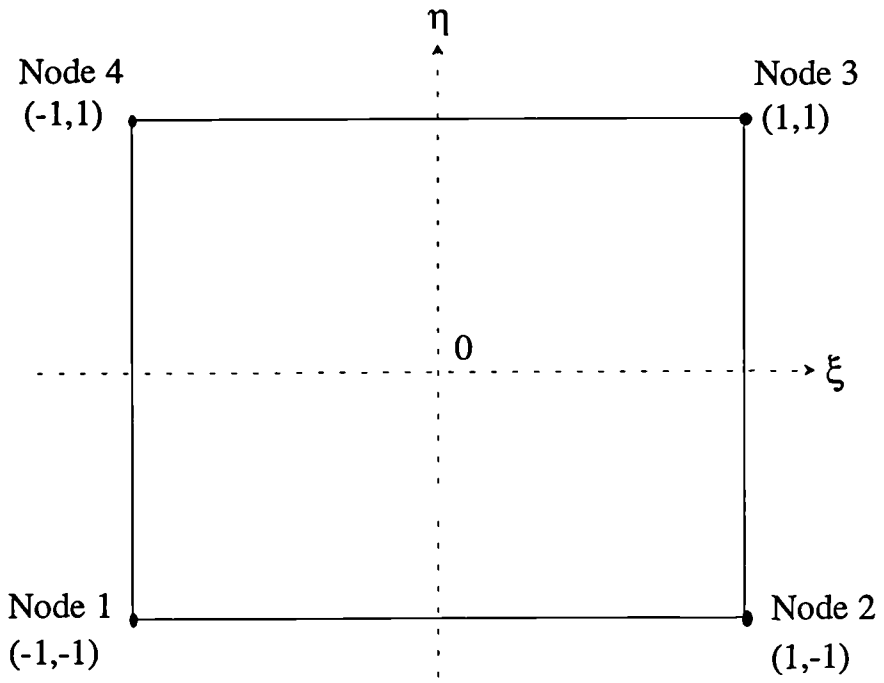


Figure 4.1.3 Finite element in local coordinate system

Figure 4.1.3 shows that the local nodes are numbered counterclockwise from node 1 whose local coordinates are $(\xi,\eta)=(-1,-1)$. The basis functions for two-dimension are defined in terms of one-dimensional basis functions:

$$\begin{aligned} E_{-1}(\xi) &= \frac{1}{2}(1-\xi) \\ E_{+1}(\xi) &= \frac{1}{2}(1+\xi) \\ H_{-1}(\eta) &= \frac{1}{2}(1-\eta) \\ H_{+1}(\eta) &= \frac{1}{2}(1+\eta) \end{aligned} \tag{4.1.5}$$

The above equations are combined to create the bi-linear basis functions for two dimension:

$$\begin{aligned} \Omega_1(\xi,\eta) &= E_{-1}H_{-1} \\ \Omega_2(\xi,\eta) &= E_{+1}H_{-1} \\ \Omega_3(\xi,\eta) &= E_{+1}H_{+1} \\ \Omega_4(\xi,\eta) &= E_{-1}H_{+1} \end{aligned} \tag{4.1.6}$$

The basis functions are defined for each node, having a value of one at the node and zero at the other three nodes. The surface of $\Omega_i(\xi,\eta)$, where i=1,2,3,4, over the local element is curved due to the product of ξ and η. The derivatives of the bi-linear basis functions are represented as:

$$\begin{aligned} \frac{\partial \Omega_1}{\partial \xi} &= -\frac{1}{2}H_{-1} & \frac{\partial \Omega_1}{\partial \eta} &= -\frac{1}{2}E_{-1} \\ \frac{\partial \Omega_2}{\partial \xi} &= +\frac{1}{2}H_{-1} & \frac{\partial \Omega_2}{\partial \eta} &= -\frac{1}{2}E_{+1} \\ \frac{\partial \Omega_3}{\partial \xi} &= +\frac{1}{2}H_{+1} & \frac{\partial \Omega_3}{\partial \eta} &= +\frac{1}{2}E_{+1} \\ \frac{\partial \Omega_4}{\partial \xi} &= -\frac{1}{2}H_{+1} & \frac{\partial \Omega_4}{\partial \eta} &= +\frac{1}{2}E_{-1} \end{aligned} \tag{4.1.7}$$

4.1.3 Coordinate transformation

Figure 4.1.4 shows the global basis functions for an 1-D arbitrary element. For the main calculations for the finite element mesh to be executed in the local coordinate systems, the information should be transferred between the local coordinate system and the global coordinate system.

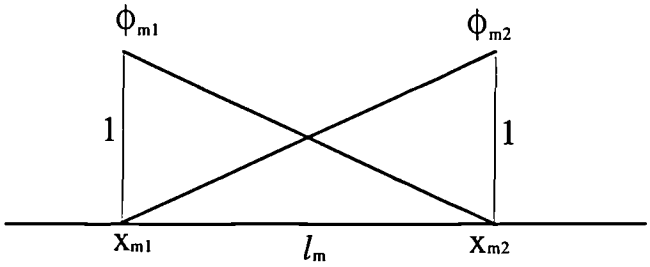


Figure 4.1.4 Global basis function for an arbitrary element

where $\phi_1 = \frac{x_2 - x}{l_m}$, $\phi_2 = \frac{x - x_1}{l_m}$, $\leftarrow x_1 \le x \le x_2$

For one dimensional domain, x can be expressed as the function of ξ:

$$x = x(\xi) = E_{-1}(\xi)x_1 + E_{+1}(\xi)x_2 \tag{4.1.8}$$

Derivatives of both sides in equation(4.1.8) give the following relations:

$$\begin{aligned} dx &= \frac{1}{2}(x_2 - x_1)d\xi = \frac{1}{2}l_m d\xi \\ \frac{dx}{d\xi} &= \frac{1}{2}l_m \end{aligned} \tag{4.1.9}$$

Applying the chain rule, the derivative of basis function can be transformed from the global to the local coordinate system and the reverse:

$$\frac{\partial E_i}{\partial \xi} = \frac{\partial \phi_i}{\partial x}\frac{\partial x}{\partial \xi}$$
$$\frac{\partial \phi_i}{\partial x} = \frac{\partial E_i}{\partial \xi}\frac{\partial \xi}{\partial x} \qquad (4.1.10)$$

where i=1,2

Transformation between the two coordinate systems can be accomplished in a 1-D domain by using equation(4.1.8), (4.1.9) and (4.1.10).

Following the same steps with the one dimensional coordinate system, transformations between the local and global coordinate systems for the two dimensional problems can be conducted. The global coordinates x, y are represented by using the local basis functions and the corner values in the global coordinate (see figure 4.1.5):

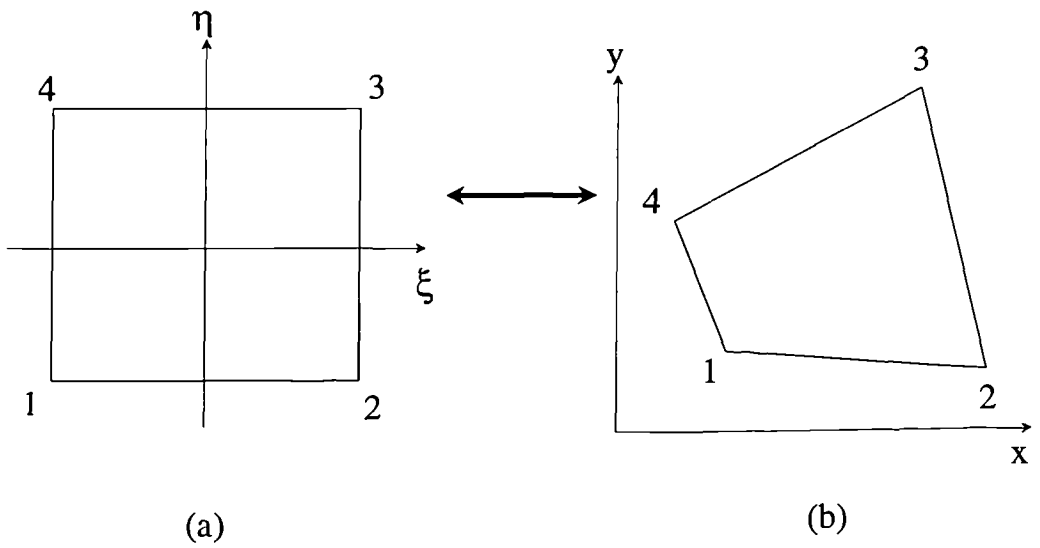


Figure 4.1.5 (a) Square element in the local coordinate system
(b) Quadrilateral element in the global coordinate system

$$x = x(\xi,\eta) = \sum_{i=1}^{4} \Omega_i(\xi,\eta) x_i$$
$$y = y(\xi,\eta) = \sum_{i=1}^{4} \Omega_i(\xi,\eta) y_i \tag{4.1.11}$$

The Jacobian matrix of the transformation from (ξ,η) to (x, y) coordinates is defined as follows:

$$[J] = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} \tag{4.1.12}$$

where

$$\begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{4} \frac{\partial \Omega_i}{\partial \xi} x_i & \sum_{i=1}^{4} \frac{\partial \Omega_i}{\partial \xi} y_i \\ \sum_{i=1}^{4} \frac{\partial \Omega_i}{\partial \eta} x_i & \sum_{i=1}^{4} \frac{\partial \Omega_i}{\partial \eta} y_i \end{bmatrix}$$

As in the one dimensional system, the chain rule can be applied to transform the derivatives of basis functions between the two coordinate systems:

$$\begin{bmatrix} \frac{\partial \Omega_j}{\partial \xi} \\ \frac{\partial \Omega_j}{\partial \eta} \end{bmatrix} = [J] \begin{bmatrix} \frac{\partial \phi_j}{\partial x} \\ \frac{\partial \phi_j}{\partial y} \end{bmatrix} \tag{4.1.13}$$

$$\begin{bmatrix} \frac{\partial \phi_j}{\partial x} \\ \frac{\partial \phi_j}{\partial y} \end{bmatrix} = [J^{-1}] \begin{bmatrix} \frac{\partial \Omega_j}{\partial \xi} \\ \frac{\partial \Omega_j}{\partial \eta} \end{bmatrix} \tag{4.1.14}$$

where j=1,2,3,4

ϕ_j : global basis function

$\left[J^{-1}\right]$: inverse Jacobian matrix

The differential area *dxdy* in the quadrilateral element is transformed as follows:

$$dA = dxdy = |J| d\xi\, d\eta \tag{4.1.15}$$

where $|J| = \frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial y}{\partial \xi}\frac{\partial x}{\partial \eta}$

Since the governing equation is derived in the global coordinate system, it is necessary that the derivative, $\frac{\partial \phi_i}{\partial x}$, should be rewritten in the local coordinate system. Using equation (4.1.14), the following relation can be established as:

$$\frac{\partial \phi_i}{\partial x} = \frac{\frac{\partial \Omega_i}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial y}{\partial \xi}\frac{\partial \Omega_i}{\partial \eta}}{\frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial y}{\partial \xi}\frac{\partial x}{\partial \eta}} \tag{4.1.16}$$

where $\Omega_i = \frac{1}{4}(1+\xi_i\xi)(1+\eta_i\eta)$, $\quad i = 1,2,3,4$

$(\xi_1, \xi_2, \xi_3, \xi_4) = (-1,1,1,-1)$

$(\eta_1, \eta_2, \eta_3, \eta_4) = (-1,-1,1,1)$

Using equation (4.1.11), the derivatives on the right hand side of equation (4.1.16) are expanded as follows:

$$\begin{aligned}
\frac{\partial x}{\partial \xi} &= \sum_{i=1}^{4} x_i \frac{1}{4}\xi_i(1+\eta_i\eta) \\
\frac{\partial x}{\partial \eta} &= \sum_{i=1}^{4} x_i \frac{1}{4}\eta_i(1+\xi_i\xi) \\
\frac{\partial y}{\partial \xi} &= \sum_{i=1}^{4} y_i \frac{1}{4}\xi_i(1+\eta_i\eta) \\
\frac{\partial y}{\partial \eta} &= \sum_{i=1}^{4} y_i \frac{1}{4}\eta_i(1+\xi_i\xi)
\end{aligned} \tag{4.1.17}$$

So the final form of $\frac{\partial \phi_i}{\partial x}$ and $\frac{\partial \phi_i}{\partial y}$ becomes:

$$\begin{aligned}
\frac{\partial \phi_i}{\partial x} &= \frac{(a_1+b\xi)(\xi_i+\rho_i\eta)-(a_2+b\eta)(\eta_i+\rho_i\xi)}{(c_1+d\eta)(a_1+b\xi)-(c_2+d\xi)(a_2+b\eta)} \\
\frac{\partial \phi_i}{\partial y} &= \frac{(c_1+d\eta)(\eta_i+\rho_i\xi)-(c_2+d\xi)(\xi_i+\rho_i\eta)}{(c_1+d\eta)(a_1+b\xi)-(c_2+d\xi)(a_2+b\eta)}
\end{aligned} \tag{4.1.18}$$

where

$$a_1 = \sum_{j\ 1}^{4} y_j\eta_j = -y_1 - y_2 + y_3 + y_4$$

$$a_2 = \sum_{j\ 1}^{4} y_j\xi_j = -y_1 + y_2 + y_3 - y_4$$

$$b = \sum_{j-1}^{4} y_j\xi_j\eta_j$$

$$c_1 = \sum_{j=1}^{4} x_j\xi_j$$

$$c_2 = \sum_{j=1}^{4} x_j\eta_j$$

$$d = \sum_{j=1}^{4} x_j\xi_j\eta_j$$

$$\rho_j = \xi_j\eta_j$$

$$|J| = \frac{1}{16}\left[(c_1+d\eta)(a_1+b\xi)-(c_2+d\xi)(a_2+b\eta)\right]$$

4.2 Development of Numerical Model

This section deals with the procedure of numerical simulation, applying the Galerkin's method to the partial differential equations which govern the multiphase flows in porous media. In subsection 4.2.1, the Galerkin's method is applied to the governing equations derived in chapter III and numerical techniques are introduced. The Newton Raphson method is employed in subsection 4.2.2 to solve the discretized equation system which is nonlinear. Lastly, in subsection 4.2.3, the relations are established among e components that may be found in all fluid phases, and the densities and compressibilities of fluid phases that are coded in the model are presented.

4.2.1 Application of the Galerkin's method

For the general primary variables, G_1, G_2 and G_3, the approximate solutions can be written as follows:

$$\begin{aligned}
\hat{G}_1(x,y) &= \sum_{i=1}^{N} G_{1_i}(t)\phi_i(x,y) \\
\hat{G}_2(x,y) &= \sum_{i=1}^{N} G_{2_i}(t)\phi_i(x,y) \\
\hat{G}_3(x,y) &= \sum_{i=1}^{N} G_{3_i}(t)\phi_i(x,y)
\end{aligned} \tag{4.2.1}$$

where N : number of nodes

$\phi_i(x,y)$: basis function in the global coordinate system

The approximate solutions can be applied to the organic component equation, equation(3.2.30).

$$S_o\hat{W}_o^o(1-\varepsilon)\beta_s\frac{\partial\hat{P}_{av}}{\partial t}+\varepsilon\hat{W}_o^o\left(\frac{\partial S_o}{\partial\hat{G}_1}\frac{\partial\hat{G}_1}{\partial t}+\frac{\partial S_o}{\partial\hat{G}_2}\frac{\partial\hat{G}_2}{\partial t}\right)$$
$$+\varepsilon S_o\hat{W}_o^o\left(\beta_P^o\frac{\partial\hat{P}_o}{\partial t}+\beta_w^o\frac{\partial\hat{W}_o^o}{\partial t}\right)+\varepsilon S_o\frac{\partial\hat{W}_o^o}{\partial t}-\nabla\cdot\left\{\hat{W}_o^o\frac{\mathbf{k}k_r}{\mu_o}\left(\nabla\hat{P}_o-\rho^o\mathbf{g}\right)\right\}$$
$$-\hat{W}_o^o\left\{\frac{\mathbf{k}k_r}{\mu_o}\left(\nabla\hat{P}_o-\rho^o\mathbf{g}\right)\right\}\cdot\left(\beta_P^o\nabla\hat{P}_o+\beta_w^o\nabla\hat{W}_o^o\right)+\frac{1}{\rho^o}\nabla\cdot\mathbf{D}_o^o=r\left(\hat{G}_1,\hat{G}_2,\hat{G}_3\right) \tag{4.2.2}$$

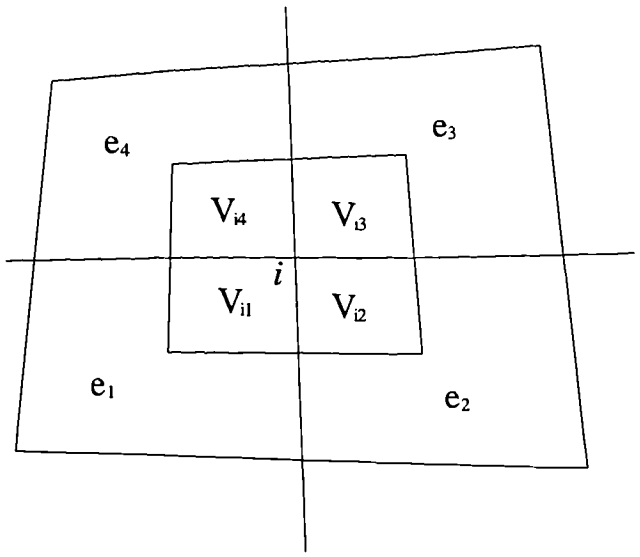
where $r(\hat{G}_1,\hat{G}_2,\hat{G}_3)$ is residual. The following equation can be given by using equation(4.1.2):

$$\int_v r(\hat{G}_1,\hat{G}_2,\hat{G}_3)\phi_i(x,y)dV=0 \tag{4.2.3}$$

Although cellwise discretization(see appendix A) may have a little problem in accuracy because of mass lumping, this research adopts cellwise discretization for time derivative terms in equation (4.2.3) to save computing time. Thus the time derivative terms can be rewritten as follows:

$$\left[S_o\hat{W}_o^o(1-\varepsilon)\beta_s\frac{\partial\hat{P}_{av}}{\partial t}+\varepsilon\hat{W}_o^o\left(\frac{\partial S_o}{\partial\hat{G}_1}\frac{\partial\hat{G}_1}{\partial t}+\frac{\partial S_o}{\partial\hat{G}_2}\frac{\partial\hat{G}_2}{\partial t}\right)\right.$$
$$\left.+\varepsilon S_o\hat{W}_o^o\left(\beta_P^o\frac{\partial\hat{P}_o}{\partial t}+\beta_w^o\frac{\partial\hat{W}_o^o}{\partial t}\right)+\varepsilon S_o\frac{\partial\hat{W}_o^o}{\partial t}\right]\cdot V_i \tag{4.2.4}$$

where $V_i=\iint_A\phi_i(x,y)dxdy\int_0^1 dz$. Unit width is considered for the z direction which is perpendicular to the x-y plane.



$V_i = V_{i1} + V_{i2} + V_{i3} + V_{i4}$

Figure 4.2.1 Volume for cellwise discretization at node i

Figure 4.2.1 shows that node i is surrounded by e_1, e_2, e_3, and e_4 elements. In the numerical model, V_{i1}, V_{i2}, V_{i3} and V_{i4} are calculated in the local coordinate system as follows(see Appendix D):

$$
\begin{aligned}
V_{i1} &= \iint \frac{1}{4}(1+\xi)(1+\eta)|J|_1 \, d\xi \, d\eta \\
V_{i2} &= \iint \frac{1}{4}(1-\xi)(1+\eta)|J|_2 \, d\xi \, d\eta \\
V_{i3} &= \iint \frac{1}{4}(1-\xi)(1-\eta)|J|_3 \, d\xi \, d\eta \\
V_{i4} &= \iint \frac{1}{4}(1+\xi)(1-\eta)|J|_4 \, d\xi \, d\eta
\end{aligned}
\qquad (4.2.5)
$$

where $|J|_1$, $|J|_2$, $|J|_3$, and $|J|_4$: Jacobian matrix of e_1, e_2, e_3, and e_4

The time derivatives in equation (4.2.4) are assumed to be approximated by finite differences. Therefore,

$$\frac{\partial \hat{G}_{1i}}{\partial t} = \frac{G_{1i}^{n+1} - G_{1i}^{n}}{t^{n+1} - t^{n}}$$
$$\frac{\partial \hat{G}_{2i}}{\partial t} = \frac{G_{2i}^{n+1} - G_{2i}^{n}}{t^{n+1} - t^{n}} \qquad (4.2.6)$$

where t^{n+1} : current time

t^{n} : previous time

G_{1i}^{n+1}, G_{2i}^{n+1} : capillary pressures at time t^{n+1}

G_{1i}^{n}, G_{2i}^{n} : capillary pressures at time t^{n}

Using the same approach, the two derivatives, $\frac{\partial S_o}{\partial G_1}$ and $\frac{\partial S_o}{\partial G_2}$, also should be expressed as follows:

$$\frac{\partial S_o}{\partial G_1} = \frac{S_o' - S_o^n}{G_1^{n+1} - G_1^n} \quad , \quad \frac{\partial S_o}{\partial G_2} = \frac{S_o'' - S_o^n}{G_2^{n+1} - G_2^n} \qquad (4.2.7)$$

where

$$S_o' = S_o(G_1^{n+1}, G_2^n)$$
$$S_o'' = S_o(G_1^n, G_2^{n+1})$$

The time derivative of the last term of equation (4.2.4) can be rewritten as follows:

$$\frac{\partial \hat{P}_o}{\partial t} = \frac{P_{oi}^{n+1} - P_{oi}^{n}}{t^{n+1} - t^{n}} \qquad (4.2.8)$$

where P_o is represented as the function of G_1 and G_2 (see table 4.3.3). So, using equation(4.2.6), (4.2.7) and (4.2.8), equation (4.2.4) becomes:

$$\left[S_o \hat{W}_o^o (1-\varepsilon)\beta_s \frac{P_{avi}^{n+1} - P_{avi}^{n}}{\Delta t} + \varepsilon \hat{W}_o^o \left(\frac{\partial S_o}{\partial \hat{G}_1} \frac{G_{1i}^{n+1} - G_{1i}^{n}}{\Delta t} + \frac{\partial S_o}{\partial \hat{G}_2} \frac{G_{2i}^{n+1} - G_{2i}^{n}}{\Delta t} \right) \right.$$
$$\left. + \varepsilon S_o \hat{W}_o^o \left(\beta_P^o \frac{P_{oi}^{n+1} - P_{oi}^{n}}{\Delta t} + \beta_w^o \frac{W_{oi}^{on+1} - W_{oi}^{on}}{\Delta t} \right) + \varepsilon S_o \frac{W_{oi}^{on+1} - W_{oi}^{on}}{\Delta t} \right] \cdot V_i \tag{4.2.9}$$

The fifth term in equation (4.2.2) involving the divergence of fluid flux is rewritten by using Green's Theorem.

$$-\int_V \left[\nabla \cdot \left\{ \hat{W}_o^o \frac{\mathbf{k} k_{ro}}{\mu_w} \left(\nabla \hat{P}_o - \rho^o \mathbf{g} \right) \right\} \right] \cdot \phi_i(x,y) dV$$
$$= -\int_\Gamma \left[\hat{W}_o^o \frac{\mathbf{k} k_{ro}}{\mu_o} \left(\nabla \hat{P}_o - \rho^o \mathbf{g} \right) \right] \cdot \mathbf{n} \phi_i(x,y) d\Gamma \tag{4.2.10}$$
$$+ \int_V \left[\hat{W}_o^o \frac{\mathbf{k} k_{ro}}{\mu_o} \left(\nabla \hat{P}_o - \rho^o \mathbf{g} \right) \right] \cdot \nabla \phi_i(x,y) dV$$

where $\mathbf{n}$: unit outward vector normal to the three dimensional surface

Γ : surface of the region

The first term on the right of equation(4.2.10) is the fluid mass flux across the region's boundary at node *i*.

$$q_{ow}(t)_i = -\int_\Gamma \left[\hat{W}_o^o \frac{\mathbf{k} k_{ro}}{\mu_o} \left(\nabla \hat{P}_o - \rho^o \mathbf{g} \right) \right] \cdot \mathbf{n} \phi_i(x,y) d\Gamma \tag{4.2.11}$$

To make matters simple, it is assumed that the principle axes of the intrinsic permeability tensor, **k**, coincide with the coordinate directions. This is a reasonable assumption for groundwater flow problems if the coordinate axes are taken perpendicular and parallel to the bedding planes.

$$\mathbf{k} = \begin{bmatrix} k_x & 0 \\ 0 & k_y \end{bmatrix} \tag{4.2.12}$$

The second term on the right of equation(4.2.10), adopting equation(4.2.1), can be approximated as:

$$\begin{aligned}
&\sum_{k=1}^{N} W_{o\,k}^{o\,n+1} \sum_{j=1}^{N} P_{o\,j}^{\,n+1} \int_x \int_y \left(\phi_k \tau_{ox} \frac{\partial \phi_j}{\partial x} \right) \frac{\partial \phi_i}{\partial x} dydx \\
&\qquad - \sum_{k=1}^{N} W_{o\,k}^{o\,n+1} \int_x \int_y \phi_k \tau_{ox} \left(\rho^o g \cos \lambda_x \right) \frac{\partial \phi_i}{\partial x} dydx \\
&+ \sum_{k=1}^{N} W_{o\,k}^{o\,n+1} \sum_{j=1}^{N} P_{o\,j}^{\,n+1} \int_x \int_y \left(\phi_k \tau_{oy} \frac{\partial \phi_j}{\partial y} \right) \frac{\partial \phi_i}{\partial y} dydx \\
&\qquad - \sum_{k=1}^{N} W_{o\,k}^{o\,n+1} \int_x \int_y \phi_k \tau_{oy} \left(\rho^o g \cos \lambda_y \right) \frac{\partial \phi_i}{\partial y} dydx
\end{aligned} \tag{4.2.13}$$

where $\tau_{ox} = \dfrac{k_x k_{ro}}{\mu_o}$, $\tau_{oy} = \dfrac{k_y k_{ro}}{\mu_o}$

λ_x, λ_y : angles between the gravity acceleration vector and x or y direction

τ_{ox} and τ_{oy} are mobilities that depend on capillary pressures and locations. They are discretized nodewise at current time step, n+1. Thus, equation (4.2.13) becomes

$$\begin{aligned}
&\sum_{l-1}^{N} W_{o\,l}^{o\,n+1} \sum_{k=1}^{N} \tau_{\mathrm{ox}\,k}^{\,n+1} \sum_{j=1}^{N} P_{o\,j}^{\,n+1} \int_x \int_y \phi_l \phi_k \frac{\partial \phi_j}{\partial x} \frac{\partial \phi_i}{\partial x} dydx \\
&- \sum_{l=1}^{N} W_{o\,l}^{o\,n+1} \sum_{k=1}^{N} \tau_{\mathrm{ox}\,k}^{\,n+1} \sum_{j=1}^{N} \left(\rho^o g \right)_j^n \cos \lambda_x \int_x \int_y \phi_l \phi_k \phi_j \frac{\partial \phi_i}{\partial x} dydx \\
&+ \sum_{l=1}^{N} W_{o\,l}^{o\,n+1} \sum_{k=1}^{N} \tau_{\mathrm{oy}\,k}^{\,n+1} \sum_{j=1}^{N} P_{o\,j}^{\,n+1} \int_x \int_y \phi_l \phi_k \frac{\partial \phi_j}{\partial y} \frac{\partial \phi_i}{\partial y} dydx \\
&- \sum_{l=1}^{N} W_{o\,l}^{o\,n+1} \sum_{k=1}^{N} \tau_{\mathrm{oy}\,k}^{\,n+1} \sum_{j=1}^{N} \left(\rho^o g \right)_j^n \cos \lambda_y \int_x \int_y \phi_l \phi_k \phi_j \frac{\partial \phi_i}{\partial y} dydx
\end{aligned} \tag{4.2.14}$$

The sixth term of equation(4.2.2) is integrated as follows:

$$
\begin{aligned}
&-\int_V \hat{W}_o^o \left\{ \frac{\mathbf{k}k_r}{\mu_o} \left(\nabla \hat{P}_o - \rho^o \mathbf{g} \right) \right\} \cdot \left(\beta_P^o \nabla \hat{P}_o + \beta_w^o \nabla \hat{W}_o^o \right) \phi_i dV \\
&= -\sum_{n=1}^{N} W_{o\,n}^{o\,n+1} \sum_{m=1}^{N} \tau_{\mathrm{ox}\,m}^{\,n+1} \sum_{l=1}^{N} P_{\mathrm{o}\,l}^{\,n+1} \sum_{k=1}^{N} \beta_{p\,k}^{o\,n} \sum_{j=1}^{N} P_{\mathrm{o}\,j}^{\,n+1} \int_x \int_y \phi_n \phi_m \frac{\partial \phi_l}{\partial x} \phi_k \frac{\partial \phi_j}{\partial x} \phi_i dydx \\
&\quad + \sum_{n=1}^{N} W_{o\,n}^{o\,n+1} \sum_{m=1}^{N} \tau_{\mathrm{ox}\,m}^{\,n+1} \sum_{l=1}^{N} \left(\rho^o g \right)_l^n \cos\lambda_x \sum_{k=1}^{N} \beta_{p\,k}^{o\,n} \sum_{j=1}^{N} P_{\mathrm{o}\,j}^{\,n+1} \int_x \int_y \phi_n \phi_m \phi_l \phi_k \frac{\partial \phi_j}{\partial x} \phi_i dydx \\
&\quad - \sum_{n-1}^{N} W_{o\,n}^{o\,n+1} \sum_{m=1}^{N} \tau_{\mathrm{oy}\,m}^{\,n+1} \sum_{l=1}^{N} P_{\mathrm{o}\,l}^{\,n+1} \sum_{k=1}^{N} \beta_{p\,k}^{o\,n} \sum_{j=1}^{N} P_{\mathrm{o}\,j}^{\,n+1} \int_x \int_y \phi_n \phi_m \frac{\partial \phi_l}{\partial y} \phi_k \frac{\partial \phi_j}{\partial y} \phi_i dydx \\
&\quad + \sum_{n-1}^{N} W_{o\,n}^{o\,n+1} \sum_{m-1}^{N} \tau_{\mathrm{oy}\,m}^{\,n+1} \sum_{l-1}^{N} \left(\rho^o g \right)_l^n \cos\lambda_y \sum_{k=1}^{N} \beta_{p\,k}^{o\,n} \sum_{j=1}^{N} P_{\mathrm{o}\,j}^{\,n+1} \int_x \int_y \phi_n \phi_m \phi_l \phi_k \frac{\partial \phi_j}{\partial y} \phi_i dydx
\end{aligned}
\tag{4.2.15}
$$

The last term of equation (4.2.2) is related to the non-advective flux. At the macroscopic level, a Fickian-type of the non-advective flux vector, $\mathbf{D}_o^o$, is adopted as follows (Abriola and Pinder, 1985a):

$$
\mathbf{D}_o^o = -\rho^o \varepsilon S_o D_{ij}^o \cdot \nabla W_o^o \tag{4.2.16}
$$

where the macroscopic second order tensor, D_{ij}^o, can be obtained from equation (3.1.13). Velocity components $\mathbf{V}_x^o$ and $\mathbf{V}_y^o$ for D_{ij}^o are calculated explicitly by a finite difference discretization of Darcy's law. It is a reasonable assumption, because velocities of fluids do not change significantly for a small time step. The Green's theorem is applied to the last term, assuming that ρ^o is constant over the elements of concern. So the last term can be expanded as follows:

$$
\begin{aligned}
&-\int_v \left\{ \nabla \cdot \left(\varepsilon S_o D_{ij}^o \cdot \nabla W_o^o \right) \right\} \phi_i(x,y) dV \\
&\quad = -\int_\Gamma \left(\varepsilon S_o D_{ij}^o \cdot \nabla W_o^o \right) \cdot \mathbf{n} \phi_i(x,y) d\Gamma \\
&\quad + \int_V \left(\varepsilon S_o D_{ij}^o \cdot \nabla W_o^o \right) \cdot \nabla \phi_i(x,y) dV
\end{aligned}
\tag{4.2.17}
$$

The first term on the right of equation (4.2.17) represents the diffusive and dispersive flux of the organic component across the boundary in the region of node *I*. This term is denoted as follows:

$$\psi_{out}(t)_i = -\int_\Gamma \left(\varepsilon S_o D_{ij}^o \cdot \nabla W_o^o\right) \cdot \mathbf{n}\phi_i(x,y)d\Gamma \tag{4.2.18}$$

The second term on the right of equation (4.2.7) can be rewritten as follows:

$$\begin{aligned}
\varepsilon \sum_{l=1}^{N} S_{ol}^{n+1} \Bigg\{ & \sum_{k=1}^{N} D_{xxk}^{o\,n} \sum_{j=1}^{N} W_{oj}^{o\,n+1} \int_x \int_y \phi_l \phi_k \frac{\partial \phi_j}{\partial x} \frac{\partial \phi_i}{\partial x} dydx \\
& + \sum_{k=1}^{N} D_{xyk}^{o\,n} \sum_{j=1}^{N} W_{oj}^{o\,n+1} \int_x \int_y \phi_l \phi_k \frac{\partial \phi_j}{\partial y} \frac{\partial \phi_i}{\partial x} dydx \\
& + \sum_{k=1}^{N} D_{yxk}^{o\,n} \sum_{j=1}^{N} W_{oj}^{o\,n+1} \int_x \int_y \phi_l \phi_k \frac{\partial \phi_j}{\partial x} \frac{\partial \phi_i}{\partial y} dydx \\
& + \sum_{k=1}^{N} D_{yyk}^{o\,n} \sum_{j=1}^{N} W_{oj}^{o\,n+1} \int_x \int_y \phi_l \phi_k \frac{\partial \phi_j}{\partial y} \frac{\partial \phi_i}{\partial y} dydx \Bigg\}
\end{aligned} \tag{4.2.19}$$

Finally, by using equation (4.2.9), (4.2.11), (4.2.14), (4.2.15), (4.2.18), and (4.2.19), equation(4.2.3) becomes:

$$\begin{aligned}
& \Bigg[S_o \hat{W}_o^o (1-\varepsilon)\beta_s \frac{P_{avi}^{n+1} - P_{avi}^{n}}{\Delta t} + \varepsilon \hat{W}_o^o \left(\frac{\partial S_o}{\partial \hat{G}_1} \frac{G_{1i}^{n+1} - G_{1i}^{n}}{\Delta t} + \frac{\partial S_o}{\partial \hat{G}_2} \frac{G_{2i}^{n+1} - G_{2i}^{n}}{\Delta t} \right) \\
& \quad + \varepsilon S_o \hat{W}_o^o \left(\beta_P^o \frac{P_{oi}^{n+1} - P_{oi}^{n}}{\Delta t} + \beta_w^o \frac{W_{oi}^{o\,n+1} - W_{oi}^{o\,n}}{\Delta t} \right) + \varepsilon S_o \frac{W_{oi}^{o\,n+1} - W_{oi}^{o\,n}}{\Delta t} \Bigg] \cdot V_i \\
& + \sum_{l=1}^{N} W_{ol}^{o\,n+1} \sum_{k-1}^{N} \tau_{oxk}^{n+1} \sum_{j=1}^{N} P_{oj}^{n+1} \int_x \int_y \phi_l \phi_k \frac{\partial \phi_j}{\partial x} \frac{\partial \phi_i}{\partial x} dydx \\
& - \sum_{l=1}^{N} W_{ol}^{o\,n+1} \sum_{k=1}^{N} \tau_{oxk}^{n+1} \sum_{j=1}^{N} \left(\rho^o g\right)_j^n \cos\lambda_x \int_x \int_y \phi_l \phi_k \phi_j \frac{\partial \phi_i}{\partial x} dydx \\
& + \sum_{l=1}^{N} W_{ol}^{o\,n+1} \sum_{k=1}^{N} \tau_{oyk}^{n+1} \sum_{j=1}^{N} P_{oj}^{n+1} \int_x \int_y \phi_l \phi_k \frac{\partial \phi_j}{\partial y} \frac{\partial \phi_i}{\partial y} dydx \\
& - \sum_{l=1}^{N} W_{ol}^{o\,n+1} \sum_{k=1}^{N} \tau_{oyk}^{n+1} \sum_{j=1}^{N} \left(\rho^o g\right)_j^n \cos\lambda_y \int_x \int_y \phi_l \phi_k \phi_j \frac{\partial \phi_i}{\partial y} dydx
\end{aligned}$$

$$
\begin{aligned}
&-\sum_{n-1}^{N} W_{o\,n}^{o\,n+1} \sum_{m=1}^{N} \tau_{ox\,m}^{n+1} \sum_{l=1}^{N} P_{o\,l}^{n+1} \sum_{k=1}^{N} \beta_{p\,k}^{o\,n} \sum_{j=1}^{N} P_{o\,j}^{n+1} \int_x \int_y \phi_n \phi_m \frac{\partial \phi_l}{\partial x} \phi_k \frac{\partial \phi_j}{\partial x} \phi_i dydx \\
&+\sum_{n-1}^{N} W_{o\,n}^{o\,n+1} \sum_{m=1}^{N} \tau_{ox\,m}^{n+1} \sum_{l=1}^{N} \left(\rho^o g\right)_l^n \cos\lambda_x \sum_{k=1}^{N} \beta_{p\,k}^{o\,n} \sum_{j=1}^{N} P_{o\,j}^{n+1} \int_x \int_y \phi_n \phi_m \phi_l \phi_k \frac{\partial \phi_j}{\partial x} \phi_i dydx \\
&-\sum_{n=1}^{N} W_{o\,n}^{o\,n+1} \sum_{m=1}^{N} \tau_{oy\,m}^{n+1} \sum_{l=1}^{N} P_{o\,l}^{n+1} \sum_{k=1}^{N} \beta_{p\,k}^{o\,n} \sum_{j=1}^{N} P_{o\,j}^{n+1} \int_x \int_y \phi_n \phi_m \frac{\partial \phi_l}{\partial y} \phi_k \frac{\partial \phi_j}{\partial y} \phi_i dydx \\
&+\sum_{n-1}^{N} W_{o\,n}^{o\,n+1} \sum_{m=1}^{N} \tau_{oy\,m}^{n+1} \sum_{l=1}^{N} \left(\rho^o g\right)_l^n \cos\lambda_y \sum_{k=1}^{N} \beta_{p\,k}^{o\,n} \sum_{j=1}^{N} P_{o\,j}^{n+1} \int_x \int_y \phi_n \phi_m \phi_l \phi_k \frac{\partial \phi_j}{\partial y} \phi_i dydx \\
&+\varepsilon \sum_{l-1}^{N} S_{o\,l}^{n+1} \left\{ \sum_{k=1}^{N} D_{xx\,k}^{o\,n} \sum_{j=1}^{N} W_{o\,j}^{o\,n+1} \int_x \int_y \phi_l \phi_k \frac{\partial \phi_j}{\partial x} \frac{\partial \phi_i}{\partial x} dydx \right. \\
&\qquad + \sum_{k=1}^{N} D_{xy\,k}^{o\,n} \sum_{j=1}^{N} W_{o\,j}^{o\,n+1} \int_x \int_y \phi_l \phi_k \frac{\partial \phi_j}{\partial y} \frac{\partial \phi_i}{\partial x} dydx \\
&\qquad + \sum_{k=1}^{N} D_{yx\,k}^{o\,n} \sum_{j=1}^{N} W_{o\,j}^{o\,n+1} \int_x \int_y \phi_l \phi_k \frac{\partial \phi_j}{\partial x} \frac{\partial \phi_i}{\partial y} dydx \\
&\qquad \left. + \sum_{k\ 1}^{N} D_{yy\,k}^{o\,n} \sum_{j=1}^{N} W_{o\,j}^{o\,n+1} \int_x \int_y \phi_l \phi_k \frac{\partial \phi_j}{\partial y} \frac{\partial \phi_i}{\partial y} dydx \right\} \\
&+q_{out}(t)_i + \psi_{out}(t)_i = 0
\end{aligned}
\tag{4.2.20}
$$

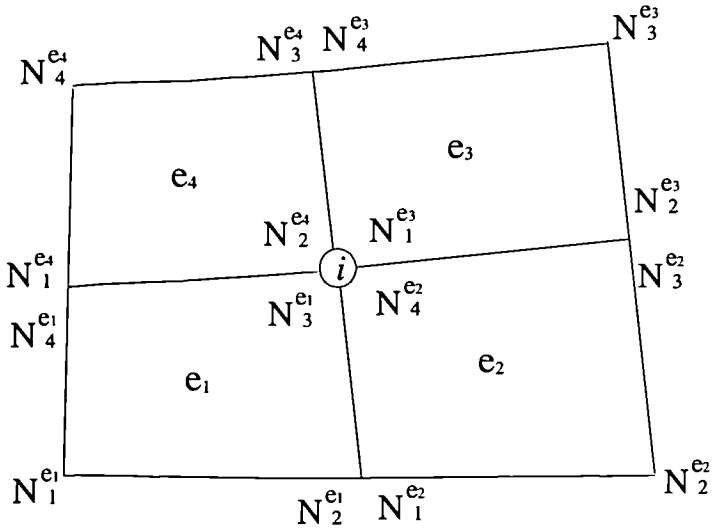


Figure 4.2.2 Elements which are concerned with calculation for node *i*

Equation (4.2.20) can be rewritten for practical application, considering elements around node *i* (see figure 4.2.2):

$$\left[S_o \hat{W}_o^o (1-\varepsilon)\beta_s \frac{P_{avi}^{n+1} - P_{avi}^{n}}{\Delta t} + \varepsilon \hat{W}_o^o \left(\frac{\partial S_o}{\partial \hat{G}_1} \frac{G_{1i}^{n+1} - G_{1i}^{n}}{\Delta t} + \frac{\partial S_o}{\partial \hat{G}_2} \frac{G_{2i}^{n+1} - G_{2i}^{n}}{\Delta t} \right) \right.$$

$$\left. + \varepsilon S_o \hat{W}_o^o \left(\beta_P^o \frac{P_{oi}^{n+1} - P_{oi}^{n}}{\Delta t} + \beta_w^o \frac{W_{oi}^{on+1} - W_{oi}^{on}}{\Delta t} \right) + \varepsilon S_o \frac{W_{oi}^{on+1} - W_{oi}^{on}}{\Delta t} \right] \cdot V_i$$

$$+ \sum_{e=e_1}^{e_4} \left[\sum_{l=N_1^e}^{N_4^e} W_{ol}^{on+1} \sum_{k-N_1^e}^{N_4^e} \tau_{oxk}^{n+1} \sum_{j=N_1^e}^{N_4^e} P_{oj}^{n+1} \int_x \int_y \phi_l \phi_k \frac{\partial \phi_j}{\partial x} \frac{\partial \phi_i}{\partial x} dydx \right.$$

$$- \sum_{l=N_1^e}^{N_4^e} W_{ol}^{on+1} \sum_{k=N_1^e}^{N_4^e} \tau_{oxk}^{n+1} \sum_{j=N_1^e}^{N_4^e} \left(\rho^o g \right)_j^n \cos\lambda_x \int_x \int_y \phi_l \phi_k \phi_j \frac{\partial \phi_i}{\partial x} dydx$$

$$+ \sum_{l-N_1^e}^{N_4^e} W_{ol}^{on+1} \sum_{k=N_1^e}^{N_4^e} \tau_{oyk}^{n+1} \sum_{j=N_1^e}^{N_4^e} P_{oj}^{n+1} \int_x \int_y \phi_l \phi_k \frac{\partial \phi_j}{\partial y} \frac{\partial \phi_i}{\partial y} dydx$$

$$\left. - \sum_{l=N_1^e}^{N_4^e} W_{ol}^{on+1} \sum_{k=N_1^e}^{N_4^e} \tau_{oyk}^{n+1} \sum_{j=N_1^e}^{N_4^e} \left(\rho^o g \right)_j^n \cos\lambda_y \int_x \int_y \phi_l \phi_k \phi_j \frac{\partial \phi_i}{\partial y} dydx \right]$$

$$- \sum_{n-N_1^e}^{N_4^e} W_{on}^{on+1} \sum_{m=N_1^e}^{N_4^e} \tau_{oxm}^{n+1} \sum_{l=N_1^e}^{N_4^e} P_{ol}^{n+1} \sum_{k=N_1^e}^{N_4^e} \beta_{pk}^{on} \sum_{j=N_1^e}^{N_4^e} P_{oj}^{n+1} \int_x \int_y \phi_n \phi_m \frac{\partial \phi_l}{\partial x} \phi_k \frac{\partial \phi_j}{\partial x} \phi_i dydx$$

$$+ \sum_{n-N_1^e}^{N_4^e} W_{on}^{on+1} \sum_{m=N_1^e}^{N_4^e} \tau_{oxm}^{n+1} \sum_{l=N_1^e}^{N_4^e} \left(\rho^o g \right)_l^n \cos\lambda_x \sum_{k=N_1^e}^{N_4^e} \beta_{pk}^{on} \sum_{j=N_1^e}^{N_4^e} P_{oj}^{n+1} \int_x \int_y \phi_n \phi_m \phi_l \phi_k \frac{\partial \phi_j}{\partial x} \phi_i dydx$$

$$- \sum_{n-N_1^e}^{N_4^e} W_{on}^{on+1} \sum_{m=N_1^e}^{N_4^e} \tau_{oym}^{n+1} \sum_{l=N_1^e}^{N_4^e} P_{ol}^{n+1} \sum_{k=N_1^e}^{N_4^e} \beta_{pk}^{on} \sum_{j=N_1^e}^{N_4^e} P_{oj}^{n+1} \int_x \int_y \phi_n \phi_m \frac{\partial \phi_l}{\partial y} \phi_k \frac{\partial \phi_j}{\partial y} \phi_i dydx$$

$$+ \sum_{n-N_1^e}^{N_4^e} W_{on}^{on+1} \sum_{m-N_1^e}^{N_4^e} \tau_{oym}^{n+1} \sum_{l=N_1^e}^{N_4^e} \left(\rho^o g \right)_l^n \cos\lambda_y \sum_{k=N_1^e}^{N_4^e} \beta_{pk}^{on} \sum_{j=N_1^e}^{N_4^e} P_{oj}^{n+1} \int_x \int_y \phi_n \phi_m \phi_l \phi_k \frac{\partial \phi_j}{\partial y} \phi_i dydx$$

$$+ \varepsilon \sum_{l-N_1^e}^{N_4^e} S_{ol}^{n+1} \left\{ \sum_{k-N_1^e}^{N_4^e} D_{xxk}^{on} \sum_{j=N_1^e}^{N_4^e} W_{oj}^{on+1} \int_x \int_y \phi_l \phi_k \frac{\partial \phi_j}{\partial x} \frac{\partial \phi_i}{\partial x} dydx \right.$$

$$+ \sum_{k-N_1^e}^{N_4^e} D_{xyk}^{on} \sum_{j=N_1^e}^{N_4^e} W_{oj}^{on+1} \int_x \int_y \phi_l \phi_k \frac{\partial \phi_j}{\partial y} \frac{\partial \phi_i}{\partial x} dydx$$

$$+ \sum_{k=N_1^e}^{N_4^e} D_{yxk}^{on} \sum_{j=N_1^e}^{N_4^e} W_{oj}^{on+1} \int_x \int_y \phi_l \phi_k \frac{\partial \phi_j}{\partial x} \frac{\partial \phi_i}{\partial y} dydx$$

$$\left. \left. + \sum_{k=N_1^e}^{N_4^e} D_{yyk}^{on} \sum_{j=N_1^e}^{N_4^e} W_{oj}^{on+1} \int_x \int_y \phi_l \phi_k \frac{\partial \phi_j}{\partial y} \frac{\partial \phi_i}{\partial y} dydx \right\} \right]$$

$$+ q_{out}(t)_i + \Psi_{out}(t)_i = 0 \qquad (4.2.21)$$

The same procedures can be applied to the water phase and e component equations to transform the mathematical expressions into the numerical expressions. Although the integrations in equation(4.2.21) may be solved analytically, numerical integration by Gaussian quadrature is simpler to code into a computer program when the integrands are polynomials in x and y. As shown previously, the integrands are expressed in the local coordinate system by using equation (4.1.11) and (4.1.18). Furthermore, the shape of the local element matches with that of Gaussian quadrature exactly. This study employs the following numerical integration which uses four Gauss points(see figure 4.2.3).

$$\int_{-1}^{1}\int_{-1}^{1} A(\xi,\eta)d\xi\, d\eta = \sum_{l=1}^{L} A(\xi_l,\eta_l)W_l \qquad (4.2.22)$$

where l : order of integration

ξ_l,η_l : location of quadrature point

W_l : weighting factor

$A(\xi,\eta)$: polynomial in the local coordinate

$L = 4$

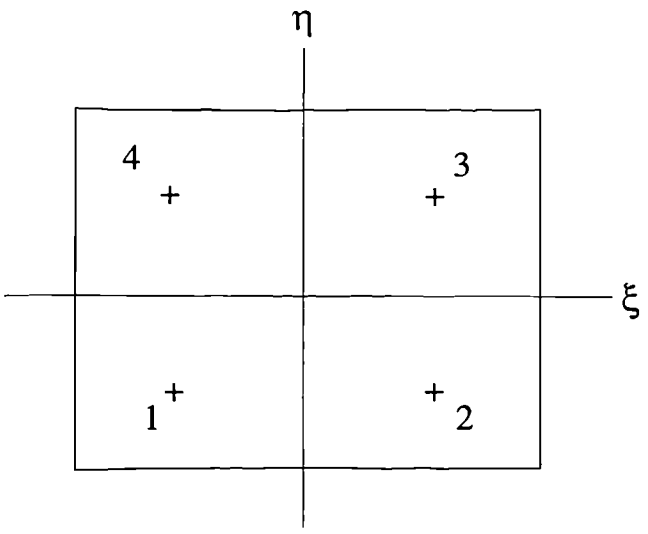


Figure 4.2.3 Gauss points for integration

where

$$(\xi_1,\eta_1)=\left(-\frac{1}{\sqrt{3}},-\frac{1}{\sqrt{3}}\right)$$
$$(\xi_2,\eta_2)=\left(\frac{1}{\sqrt{3}},-\frac{1}{\sqrt{3}}\right)$$
$$(\xi_3,\eta_3)=\left(\frac{1}{\sqrt{3}},\frac{1}{\sqrt{3}}\right)$$
$$(\xi_4,\eta_4)=\left(-\frac{1}{\sqrt{3}},\frac{1}{\sqrt{3}}\right)$$

Appendix E shows the subprogram that is coded into the numerical model, COMPO. The numerical integrations are carried out by the subprogram.

4.2.2 Newton Raphson Method

Because three governing equations are given and three unknowns are defined at all node points, a $3N*3N$ matrix is expected if the system of equations is written in a matrix form. However, since the discretized equations are nonlinear for the primary unknowns G_1, G_2, and G_3, the Newton-Raphson iteration method is employed to solve a system of nonlinear algebraic equations. Denoting $f_i(G_1,G_2,G_3)$ as the water phase equation, a Taylor's series expansion of $f_i(G_1,G_2,G_3)$ can be written about initial guesses $G_{1_i}^{(0)}$, $G_{2_i}^{(0)}$, and $G_{3_i}^{(0)}$:

$$\begin{aligned} f_i(G_1,G_2,G_3)=0=f_i(G_1^{(0)},G_2^{(0)},G_3^{(0)})+\delta G_{1i}^{(0)}\frac{\partial f_i(G_1,G_2,G_3)}{\partial G_1^{(0)}} \\ +\delta G_{2_i}^{(0)}\frac{\partial f_i(G_1,G_2,G_3)}{\partial G_2^{(0)}}+\delta G_{3i}^{(0)}\frac{\partial f_i(G_1,G_2,G_3)}{\partial G_3^{(0)}}+\cdots \end{aligned} \tag{4.2.23}$$

where

$$\begin{aligned} \delta G_{1i}^{(0)} &= G_{1i}-G_{1i}^{(0)} \\ \delta G_{2i}^{(0)} &= G_{2i}-G_{2i}^{(0)} \\ \delta G_{3i}^{(0)} &= G_{3i}-G_{3i}^{(0)} \end{aligned} \tag{4.2.24}$$

$\delta G_{1_i}^{(0)}$, $\delta G_{2i}^{(0)}$ and $\delta G_{3_i}^{(0)}$ can be solved, neglecting terms in δ greater than order 1 and establishing two more relevant conditions that correspond to the mass balance equations for the organic component and e-component: $g_i(G_1, G_2, G_3)$ and $h_i(G_1, G_2, G_3)$. So new trial values for primary variables at node i can be given as follows:

$$\begin{aligned} G_{1i}^{(1)} &= G_{1i}^{(0)} + \delta G_{1i}^{(0)} \\ G_{2i}^{(1)} &= G_{2i}^{(0)} + \delta G_{2i}^{(0)} \\ G_{3i}^{(1)} &= G_{3i}^{(0)} + \delta G_{3i}^{(0)} \end{aligned} \tag{4.2.25}$$

The procedure can be summarized as follows:

$$\begin{aligned} \sum_{j=1}^{N} \delta G_{1j}^{(v)} \frac{\partial f_i}{\partial G_{1j}^{(v)}} + \sum_{j=1}^{N} \delta G_{2j}^{(v)} \frac{\partial f_i}{\partial G_{2j}^{(v)}} + \sum_{j=1}^{N} \delta G_{3j}^{(v)} \frac{\partial f_i}{\partial G_{3j}^{(v)}} &= -f_i(G_1^{(v)}, G_2^{(v)}, G_3^{(v)}) \\ \sum_{j=1}^{N} \delta G_{1j}^{(v)} \frac{\partial g_i}{\partial G_{1j}^{(v)}} + \sum_{j=1}^{N} \delta G_{2j}^{(v)} \frac{\partial g_i}{\partial G_{2j}^{(v)}} + \sum_{j=1}^{N} \delta G_{3j}^{(v)} \frac{\partial g_i}{\partial G_{3j}^{(v)}} &= -g_i(G_1^{(v)}, G_2^{(v)}, G_3^{(v)}) \\ \sum_{j=1}^{N} \delta G_{1j}^{(v)} \frac{\partial h_i}{\partial G_{1j}^{(v)}} + \sum_{j=1}^{N} \delta G_{2j}^{(v)} \frac{\partial h_i}{\partial G_{2j}^{(v)}} + \sum_{j=1}^{N} \delta G_{3j}^{(v)} \frac{\partial h_i}{\partial G_{3j}^{(v)}} &= -h_i(G_1^{(v)}, G_2^{(v)}, G_3^{(v)}) \end{aligned}$$

(4.2.26)

$$\begin{aligned} G_{1i}^{(v+1)} &= G_{1i}^{(v)} + \delta G_{1i}^{(v)} \\ G_{2i}^{(v+1)} &= G_{2i}^{(v)} + \delta G_{2i}^{(v)} \\ G_{3i}^{(v+1)} &= G_{3i}^{(v)} + \delta G_{3i}^{(v)} \end{aligned} \tag{4.2.27}$$

where i=1,2,3, ····· ,N

ν=0,1,2, ····· ← ν th try

By repeating equation (4.2.26) and (4.2.27) until $\delta G_{1_i}^{(v)}$, $\delta G_{2i}^{(v)}$ and $\delta G_{3_i}^{(v)}$ become small enough to satisfy convergence criteria, the primary variables may have acceptable

values. The following matrix equation is the expanded form of equation (4.2.22) for the domain that has N node points:

$$\begin{bmatrix} F_{1,1} & F_{1,2} & F_{1,3} & F_{1,4} & F_{1,5} & F_{1,6} & \bullet\bullet\bullet & F_{1,3N-2} & F_{1,3N-1} & F_{1,3N} \\ F_{2,1} & F_{2,2} & F_{2,3} & F_{2,4} & F_{2,5} & F_{2,6} & \bullet\bullet\bullet & F_{2,3N-2} & F_{2,3N-1} & F_{2,3N} \\ F_{3,1} & F_{3,2} & F_{3,3} & F_{3,4} & F_{3,5} & F_{3,6} & \bullet\bullet\bullet & F_{3,3N-2} & F_{3,3N-1} & F_{3,3N} \\ & & \bullet & \bullet & \bullet & & & \bullet & \bullet & \bullet \\ & & \bullet & \bullet & \bullet & & & \bullet & \bullet & \bullet \\ & & \bullet & \bullet & \bullet & & & \bullet & \bullet & \bullet \\ F_{3i-2,1} & F_{3i-2,2} & F_{3i-2,3} & F_{3i-2,4} & F_{3i-2,5} & F_{3i-2,6} & \bullet\bullet\bullet & F_{3i-2,3N-2} & F_{3i-2,3N-1} & F_{3i-2,3N} \\ F_{3i-1,1} & F_{3i-1,2} & F_{3i-1,3} & F_{3i-1,4} & F_{3i-1,5} & F_{3i-1,6} & \bullet\bullet\bullet & F_{3i-1,3N-2} & F_{3i-1,3N-1} & F_{3i-1,3N} \\ F_{3i,1} & F_{3i,2} & F_{3i,3} & F_{3i,4} & F_{3i,5} & F_{3i,6} & \bullet\bullet\bullet & F_{3i,3N-2} & F_{3i,3N-1} & F_{3i,3N} \\ & & \bullet & \bullet & \bullet & & & \bullet & \bullet & \bullet \\ & & \bullet & \bullet & \bullet & & & \bullet & \bullet & \bullet \\ & & \bullet & \bullet & \bullet & & & \bullet & \bullet & \bullet \\ F_{3N-2,1} & F_{3N-2,2} & F_{3N-2,3} & F_{3N-2,4} & F_{3N-2,5} & F_{3N-2,6} & \bullet\bullet\bullet & F_{3N-2,3N-2} & F_{3N-2,3N-1} & F_{3N-2,3N} \\ F_{3N-1,1} & F_{3N-1,2} & F_{3N-1,3} & F_{3N-1,4} & F_{3N-1,5} & F_{3N-1,6} & \bullet\bullet\bullet & F_{3N-1,3N-2} & F_{3N-1,3N-1} & F_{3N-1,3N} \\ F_{3N,1} & F_{3N,2} & F_{3N,3} & F_{3N,4} & F_{3N,5} & F_{3N,6} & \bullet\bullet\bullet & F_{3N,3N-2} & F_{3N,3N-1} & F_{3N,3N} \end{bmatrix} \begin{bmatrix} \delta G_{1} \\ \delta G_{2} \\ \delta G_{3} \\ \bullet \\ \bullet \\ \bullet \\ \delta G_{1i} \\ \delta G_{2i} \\ \delta G_{3i} \\ \bullet \\ \bullet \\ \bullet \\ \delta G_{1N} \\ \delta G_{2N} \\ \delta G_{3N} \end{bmatrix} = \begin{bmatrix} -f_1 \\ -g_1 \\ -h_1 \\ \bullet \\ \bullet \\ \bullet \\ -f_i \\ -g_i \\ -h_i \\ \bullet \\ \bullet \\ \bullet \\ -f_N \\ -g_N \\ -h_N \end{bmatrix} \tag{4.2.28}$$

where

$$F_{3i-2,3j-2} = \frac{\partial f_i}{\partial G_{1j}} \;,\; F_{3i-2,3j-1} = \frac{\partial f_i}{\partial G_{2j}} \;,\; F_{3i-2,3j} = \frac{\partial f_i}{\partial G_{3j}}$$

$$F_{3i-1,3j-2} = \frac{\partial g_i}{\partial G_{1j}} \;,\; F_{3i-1,3j-1} = \frac{\partial g_i}{\partial G_{2j}} \;,\; F_{3i-1,3j} = \frac{\partial g_i}{\partial G_{3j}} \tag{4.2.29}$$

$$F_{3i,3j-2} = \frac{\partial h_i}{\partial G_{1j}} \;,\; F_{3i,3j-1} = \frac{\partial h_i}{\partial G_{2j}} \;,\; F_{3i,3j} = \frac{\partial h_i}{\partial G_{3j}}$$

i=1,2,3,·····, N

j=1,2,3,·····, N

$3i$=3*i

$3j$=3*j

$3N$=3*N

Equation (4.2.21) is the discretized form of the organic component equation at node *i*. The other two equations also can be expressed numerically. Then, the elements of the main matrix in equation(4.2.28) can be obtained by applying equation(4.2.29) to the three discretized equations.

However, the matrix equation(4.2.28) has an infinite number of solutions. To obtain a unique solution that corresponds to particular problems, initial and boundary conditions are required. Subsection 2.5.2 introduced two types of boundary condition. Dirichlet condition provides the general primary variables with fixed values on boundaries. The following boundary condition may be adopted at node *i*:

$$\begin{aligned} G_{1_i} &= C_1 , \quad & B_{1i}(G_1, G_2, G_3) &= G_{1i} - C_1 \\ G_{2i} &= C_2 , \quad & B_{2i}(G_1, G_2, G_3) &= G_{2i} - C_2 \\ G_{3_i} &= C_3 , \quad & B_{3i}(G_1, G_2, G_3) &= G_{3i} - C_3 \end{aligned} \tag{4.2.30}$$

where C_1, C_2, C_3 : constants

B_{1_i} , B_{2_i} , B_{3_i} : boundary conditions

The boundary conditions replace the mass balance equation at node *i*. The Newton Raphson method equally applies to equation(4.2.31), resulting that

$$\begin{aligned} F_{3i-2,3i-2} &= \frac{\partial B_{1i}}{\partial G_{1i}} = 1 , \quad & -B_{1i} &= -(G_{1i} - C_1) \\ F_{3i-1,3i-1} &= \frac{\partial B_{2i}}{\partial G_{2i}} = 1 , \quad & -B_{2i} &= -(G_{2i} - C_2) \\ F_{3i,3i} &= \frac{\partial B_{3i}}{\partial G_{3i}} = 1 , \quad & -B_{3i} &= -(G_{3i} - C_3) \end{aligned} \tag{4.2.31}$$

All other elements of the i th row of the matrix $[\mathbf{F}]$ are zero except $F_{3i-2,3i-2}$, $F_{3i-1,3i-1}$, $F_{3i,3i}$ and $-f_i$, $-g_i$, $-h_i$ are replaced by $-B_{1i}$, $-B_{2i}$, $-B_{3i}$.

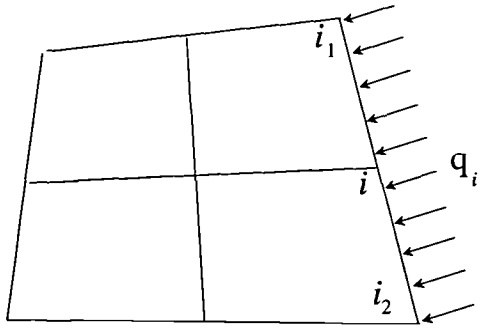


Figure 4.2.4 Specified flow boundary condition

The Neuman condition defines fluxes on boundaries as mentioned in subsection 2.5.2. Thus, if the normal flux, q_t , is specified at a boundary node i(see figure 4.2.4), the boundary terms such as $q_{out}(t)_i$, $\Psi_{out}(t)_i$ in equation(4.2.21) can be replaced as follows:

$$q_{out}(t)_t + \Psi_{out}(t)_i = -\int_{i1}^{i} q_t \phi_t(\Gamma) d\Gamma - \int_{i}^{i2} q_t \phi_i(\Gamma) d\Gamma \tag{4.2.32}$$

q_t is positive when they represent inflow through the boundary. The integrals in equation(4.2.32) can be simplified as follows:

$$-\int_{i1}^{i} q_t \phi_t(\Gamma) d\Gamma - \int_{t}^{i2} q_t \phi_i(\Gamma) d\Gamma = -q_i \frac{\overline{ii_1}}{2} - q_i \frac{\overline{ii_2}}{2} \tag{4.2.33}$$

where $\overline{ii_1}$ and $\overline{ii_2}$ are the distances between two nodes. The boundary conditions are incorporated into the column vector of the right hand side of equation(4.2.28).
Combining the governing equations with the boundary conditions, the whole system of equations is completed.

4.2.3 Partition, density and compressibility

As mentioned before, the mass of e component in the organic phase may transfer to the water phase or gas phase by dissolution or evaporation. Two constitutive relations are required to close the system of equations. Experiments have been conducted to prove that the transfer of soluble hydrocarbons to the water phase might be modelled as a single-stage extraction process. So partition coefficients are determined on the basis of that. Van der Waarden, et al(1971), Kappler and Wuhrmann(1978) and Fried, et al(1979) conclude that the use of a partition coefficient determines solute concentrations.

Henry's law is the thermodynamic expression which relates the vapour pressure of a solute to the mole fraction of this solute in the liquid phase. Henry's constant represents proportionality between these two variables. Generally Henry's constant is a function of both pressure and composition(Wark, 1971). There is a dimensionless Henry's law constant that relates the concentration of a compound in the gas phase to its concentration in the liquid phase(Lyman, et al(1982)). Therefore, this dimensionless constant can be considered as a partition coefficient.

Local equilibrium between the three fluid phases are established for the e component .

$$\omega_e^{gw} = \frac{W_e^g}{W_e^w} \tag{4.2.34}$$

$$\omega_e^{wo} = \frac{W_e^w}{W_e^o} \tag{4.2.35}$$

where ω_e^{gw} and ω_e^{wo} are partition coefficients which must be determined empirically. So W_e^w and W_e^g are presented as follows:

$$W_e^w = \omega_e^{wo} W_e^o \tag{4.2.36}$$

$$W_e^g = \omega_e^{gw} \omega_e^{wo} W_e^o \tag{4.2.37}$$

As the model, COMPO, considers that the fluid phases are compressible, their densities vary according to the fluid pressures and mass fractions. Assuming that there is only one component in the α-phase, the density of the α-phase can be represented from equation (3.2.19) as follows:

$$\rho^{\alpha} = \rho^{\alpha b} \exp\left[\beta^{\alpha}\left(P_{\alpha} - P^{\alpha b}\right)\right] \tag{4.2.38}$$

where $P^{\alpha b}$: reference pressure

$\rho^{\alpha b}$: fluid density at the reference pressure

Crookson, et al (1979) propose the following method of determining density of organic phases that are mixture of the individual components:

$$\rho^{o} = \frac{1}{\sum_{i} \frac{W_i^o}{\rho_i^o}} \tag{4.2.39}$$

where ρ_i^o : density of i component at the phase pressure P_o.

ρ_i^o can be obtained by using equation (4.2.38). Since this research assumes that the organic phase is made up of two components, organic and e components, equation (4.2.39) can be rewritten as follows:

$$\rho^{o} = \frac{1}{\frac{W_e^o}{\rho_e^o} + \frac{W_o^o}{\rho_o^o}} = \frac{\rho_e^o \rho_o^o}{W_e^o \rho_o^o + W_o^o \rho_e^o} \tag{4.2.40}$$

By using equation(3.2.19), the organic phase compressibilities can be derived by differentiation and manipulation of equation (4.2.39).

$$\beta_w^o = \frac{\rho_o^o - \rho_e^o}{W_o^o \rho_e^o + W_e^o \rho_o^o} \tag{4.2.41}$$

$$\beta_p^o = \beta_o^o + \beta_e^o - \frac{W_e^o \beta_o^o \rho_o^o + W_o^o \beta_e^o \rho_e^o}{W_o^o \rho_e^o + W_e^o \rho_o^o} \tag{4.2.42}$$

where β_w^o: compressibility of organic phase with respect to mass fraction

β_p^o : compressibility of organic phase with respect to pressure

β_o^o : compressibility of organic component in the organic phase

β_e^o : compressibility of e component in the organic phase

The density of gas phase is subject to the gas law:

$$\rho^g = \frac{P_g M_g}{Z R_u T} \tag{4.2.43}$$

where P_g : gas pressure(constant)

M_g : molecular weight of the gas mixture

R_u : universal gas constant

Z : compressibility factor

T : temperature

The gas phase is assumed to consist of air and e components. Thus, assuming that P_g is constant, the following expression of the compressibility of the gas phase can be derived from equation (3.2.19), (4.2.39) and (4.2.43):

$$\beta^g = \frac{M_e - M_a}{M_e\left(1 - W_e^g\right) + M_a W_e^g} \tag{4.2.44}$$

where M_e , M_a : Molecular weight of component e and air

4.3 Apparatus for generalizing a numerical model

The fixed primary variables may be obstacles for general applications. The general variables G_1, G_2, and G_3 are very useful for applying a numerical model to the various pollution patterns(see table 3.2.1). In this section, the pollution patterns are categorized by the property of solid matrix, fluid phases in a porous medium, and interphase mass transfer. Table 4.3.1 summarizes the variations of the general primary variables, according to the pollution patterns.

	G_1	G_2	G_3
W - G	P_{gw}	·	W_e^w
O - G	·	P_{go}	$\left(W_e^g\right)^*$
W - O (**h**)	P_{ow}	P_w	$\left(W_e^o\right)^*$
W - O (**o**)	P_{wo}	P_o	$\left(W_e^o\right)^*$
W - O - G (**h**)	P_{ow}	P_{go}	$\left(W_e^o\right)^*$
W - O - G (**o**)	P_{wo}	P_{gw}	$\left(W_e^o\right)^*$

where

W, O, G : water, organic and gas phase respectively

h, **o** : hydrophilic and oliophilic solid matrix

$\left(W_e^\alpha\right)^*$: G_3 is considered under the condition of interphase mass exchange.

Table 4.3.1 Variations of the general primary variables

Table 4.3.2 presents the temporal derivative of the α-phase saturation according to the pollution patterns.

	$\frac{\partial S_\alpha}{\partial t} = \frac{\partial S_\alpha}{\partial G_1}\frac{\partial G_1}{\partial t} + \frac{\partial S_\alpha}{\partial G_2}\frac{\partial G_2}{\partial t}$
W - G	$\frac{\partial S_\alpha}{\partial P_{gw}}\frac{\partial P_{gw}}{\partial t}$
O - G	$\frac{\partial S_\alpha}{\partial P_{go}}\frac{\partial P_{go}}{\partial t}$
W - O (**h**)	$\frac{\partial S_\alpha}{\partial P_{ow}}\frac{\partial P_{ow}}{\partial t}$
W - O (**o**)	$\frac{\partial S_\alpha}{\partial P_{wo}}\frac{\partial P_{wo}}{\partial t}$
W - O - G (**h**)	$\frac{\partial S_\alpha}{\partial P_{ow}}\frac{\partial P_{ow}}{\partial t} + \frac{\partial S_\alpha}{\partial P_{go}}\frac{\partial P_{go}}{\partial t}$
W - O - G (**o**)	$\frac{\partial S_\alpha}{\partial P_{wo}}\frac{\partial P_{wo}}{\partial t} + \frac{\partial S_\alpha}{\partial P_{gw}}\frac{\partial P_{gw}}{\partial t}$

Table 4.3.2 Presentation of time derivative of the α-phase saturation

Table 4.3.3 summarizes the water, organic and average fluid pressure, P_w , P_o ,and P_{av} which are expressed in terms of the general primary variables, and the derivatives of the three pressures with respect to the general primary variables, G_1 and G_2.

For a general application, it is also required to consider various ways of defining the relations between saturations and capillary pressures, and relative permeabilities and capillary pressures. They may be presented by mathematical expressions or experimental data.

Employing the finite element method and the generalizing procedures, the numerical model, COMPO, is developed in terms of FORTRAN language. Appendix B shows the variable list coded in the model.

	P_w	P_o	P_{av}	$\frac{\partial P_w}{\partial G_1}$	$\frac{\partial P_w}{\partial G_2}$	$\frac{\partial P_o}{\partial G_1}$	$\frac{\partial P_o}{\partial G_2}$	$\frac{\partial P_{av}}{\partial G_1}$	$\frac{\partial P_{av}}{\partial G_2}$
W-G	$-G_1 + P_g$	·	P_w	−1	0	0	0	−1	0
O-G	·	$-G_2 + P_g$	P_o	0	0	0	−1	0	−1
W-O (**h**)	G_2	$G_1 + G_2$	$kP_o + (1-k)P_w$	0	1	1	1	k	1
W-O (**o**)	$G_1 + G_2$	G_2	$kP_o + (1-k)P_w$	1	1	0	1	$1-k$	1
W-O-G (**h**)	$-G_1 - G_2 + P_g$	$-G_2 + P_g$	$kP_o + (1-k)P_w$	−1	−1	0	−1	$k-1$	−1
W-O-G (**o**)	$-G_2 + P_g$	$-G_1 - G_2 + P_g$	$kP_o + (1-k)P_w$	0	−1	−1	−1	$-k$	−1

Table 4.3.3 Presentations of pressures and derivatives in terms of the general primary variables.

CHAPTER V
VERIFICATIONS

The numerical model is applied to the four cases of subsurface contamination to verify its algorithm. Section 5.1 deals with the pollution pattern (I). Water saturation and concentration of a pollutant are anticipated over the domain in the course of time. Section 5.2 treats the pollution pattern (II). The numerical model shows how the organic phase replace the water phase in the course of time. In section 5.3, a porous medium is assumed to be occupied by the three fluid phases-water, gas and organic phase. Here, as in the pollution pattern (VII), interphase mass exchange is allowed among the three fluid phases. Section 5.4 deals with a full two-dimensional problem of the pollution pattern (VII). Finally section 5.5 shows the extended use of the numerical model by applying it to a tracer problem.

5.1 Pollution pattern (I)

This contamination scenario is given by Van Genuchten(1982). A vertical column which is 1.2 m long is assumed to be filled with clear water and gas initially. The numerical model, COMPO1D, meshes the domain with 120 elements. So each 1-D element is 1 cm, Δy = 1 cm. The vertical coordinate, y, is measured downward from the top of the column. The time step is constant at Δt = 30 sec. The simulation is carried out for 9 hours of infiltration. The parameters for this simulation are summarized in Table 5.1.1.

	parameters	value	units
water	μ_w	1.0×10^{-3}	$kg / (m \cdot sec)$
	β_w	0	m^2 / N
	ρ^w	1000	kg / m^3

solid matrix	ε	0.38	
	$\mathbf{k}$	4.4558×10^{-13}	m^2
	β_s	0	m^2 / N
gravity	$\mathbf{g}$	9.81	$m / \sec^2$
dispersion &	D^{mw}	0	$m^2 / \sec$
diffusion	$\mathbf{a}^w$	0.01	m

Table 5.1.1 Parameters for example 1

The water saturation is represented in terms of the capillary pressure, P_{gw} , and the relative permeability of water is a function of water saturation. The relationships are shown as follows:

$$S_w = 1.52208 - 0.0718947 \ln (P_{gw})$$
$$\text{for } 1421.96 \le P_{gw} < 2892.38 \; kg / (m \cdot \sec^2)$$
$$S_w = 2.94650 - 0.250632 \ln (P_{gw}) \qquad (5.1.1.)$$
$$\text{for } P_{gw} \ge 2892.38 \; kg / (m \cdot \sec^2)$$

$$k_r = 1.235376 \times 10^{-6} \exp (13.604 \; S_w) \qquad (5.1.2)$$

Initial water saturation along the vertical column is expressed as follows:

$$S_w(y, t = 0) = \begin{cases} 0.394737 + 0.219287 y & 0 < y \le 0.6 \\ 0.526316 & 0.6 < y \le 1.2 \end{cases} \qquad (5.1.3)$$

Using equation (5.1.1), the initial capillary pressure distribution can be obtained. The top boundary consists of an infiltration pond, where water saturation is S_w=1.0 .

During the simulation the capillary pressure, P_{gw} , is fixed to be 1421.96 kg / (m·sec^2) at the top boundary. The bottom boundary is held at a specified saturation of S_w=0.526316, by specification of pressure, P_{gw}= 15616.5 kg / (m·sec^2). No flow occurs across either side boundary, but flow enters the top boundary due to the pressure gradient. For the first 168 minutes, contaminated water enters from the top boundary, where the solute concentration is c=209 meq / liter. After that, clean water flows again from the pond.

As shown in table 4.3.1, the general primary variables become $G_1 = P_{gw}$, $G_3 = W_e^w = c$. The initial and boundary conditions are summarized in table 5.1.2.

Boundary condition	
upper boundary	$G_1 = P_{gw} = 1421.96\ kg/(m\cdot\sec^2)$
	$G_3 = W_e^w = c = 209$ meq / liter ($t \le 168$ min)
	$G_3 = W_e^w = c = 0$ meq / liter ($t > 168$ min)
lower boundary	$G_1 = P_{gw} = 15616.5\ kg/(m\cdot\sec^2)$
	$\frac{\partial G_3}{\partial y} = \frac{\partial W_e^w}{\partial y} = 0$
Initial condition	The initial distribution of $G_1 = P_{gw}$ is calculated from equation (5.1.1) and (5.1.3)
	$G_3 = W_e^w = c = 0$ meq / liter

Table 5.1.2 Boundary and initial conditions

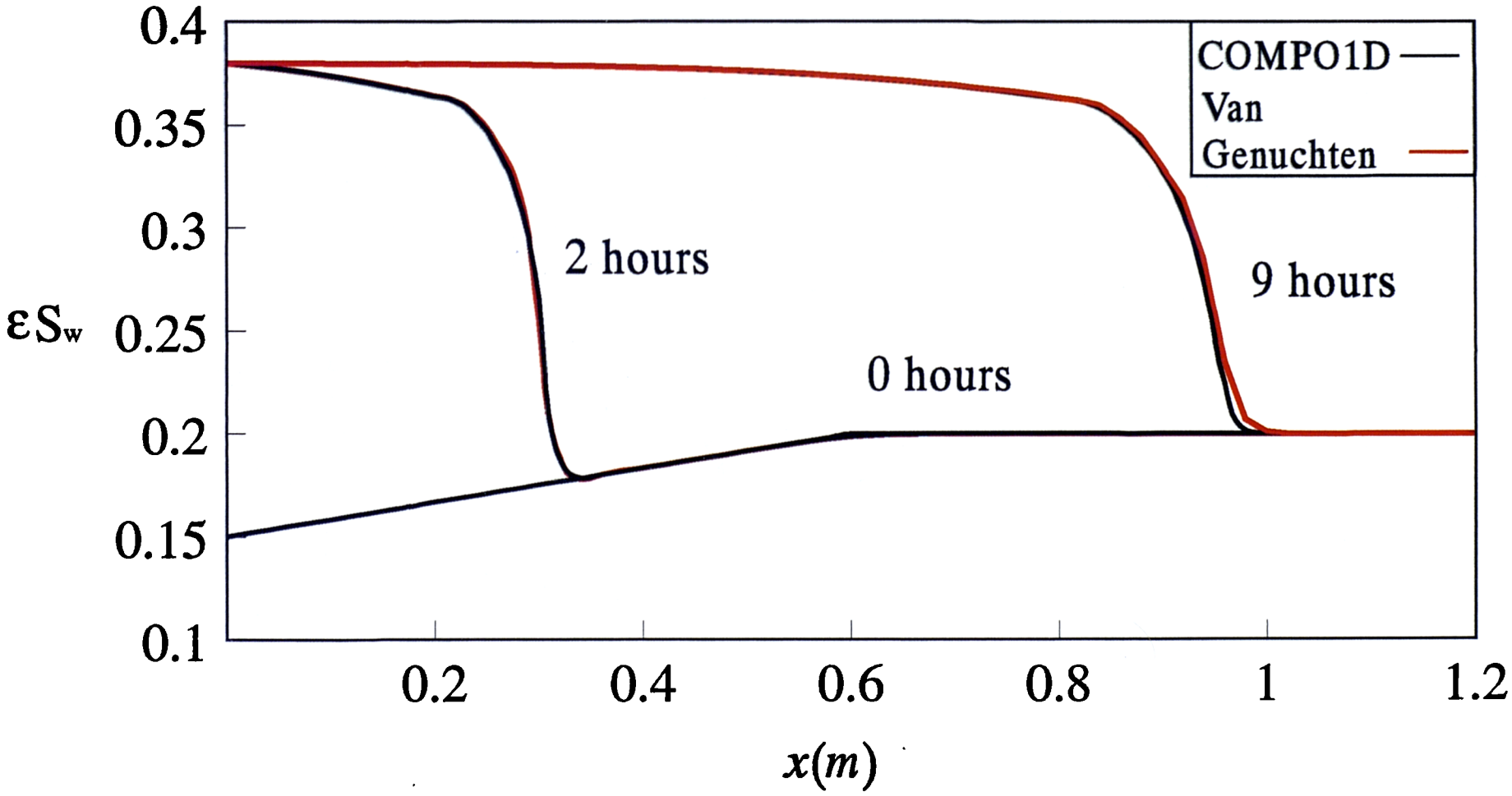


Figure 5.1.1 Propagation of moisture content

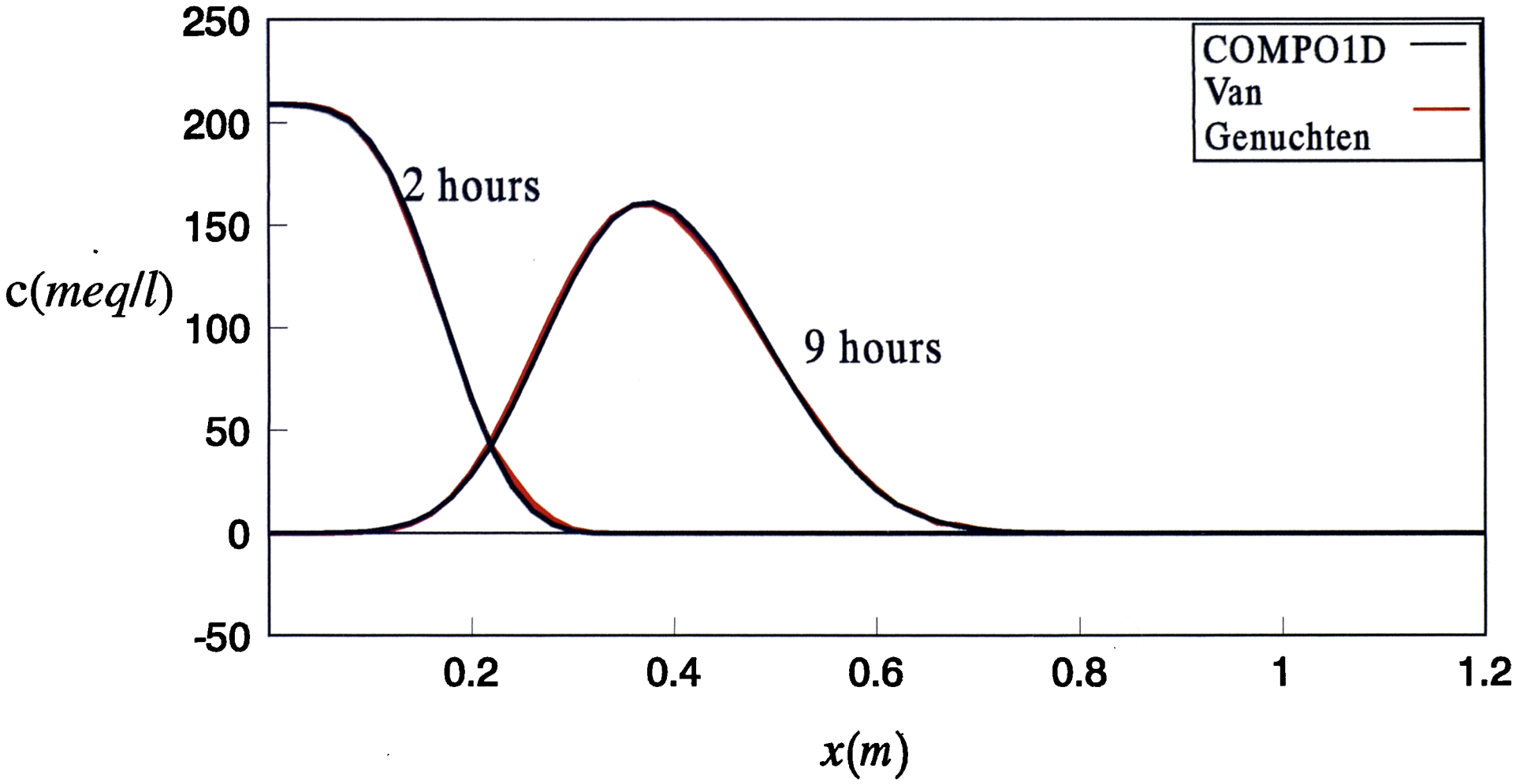


Figure 5.1.2 Propagation of solute concentration

Note that the concentration units are arbitrary (need not to be mass fraction) because this is a constant density simulation. Simulation results are shown in figure 5.1.1 and 5.1.2 . The simulation results of COMPO1D and Van Genuchten (1982) coincide almost exactly for water content and solute concentration. Thus it can be said that COMPO1D is applicable to solute transport problems in water-gas system.

5.2 Pollution pattern (II)

This example is originated from Gamiel(1989). Table 5.2.1 shows the properties of solid matrix, water, and TCE. TCE(Trichloroethylene) is a degreasing agent commonly used by industry and private households. To know the physical properties of TCE in more detail, refer to Gallant(1966) or Lyman, et al (1982). TCE is a liquid at room temperature and is both volatile and slightly water soluble. Thus, it may transfer to water and gas phases while TCE flows through porous media. However, the interphase mass exchange is not considered in this example. The compressibility of TCE is assumed to be negligible.

	parameters	value	units
water	μ_w	1.0019×10^{-2}	poise
	β_w	4.532×10^{-11}	$cm^2 / dyne$
	ρ^w	0.9997964	g / cm^3
TCE	μ_o	5.8×10^{-1}	poise
	β_o	0	$cm^2 / dyne$
	ρ^{ob}	1.4657	g / cm^3
solid matrix	ε	0.36	
	k	5.8231×10^{-7}	cm^2
	β_s	2.0×10^{-10}	$cm^2 / dyne$
gravity	g	9.80665×10^2	$cm / \sec^2$
residual	S_{wr}	0.306	
saturation	S_{om}	0.17	

	S_{wr}	0.306	
parameters for	S_{ws}	0.9998	
van Genuchten	a_{sw}	0.11	cm^{-1}
equation (3.1.8)	n_{sw}	6.5	
and (5.2.1)	a_{kw}	0.108	cm^{-1}
	n_{kw}	6.60	

Table 5.2.1 parameters for example 2

It adopts Van Genuchten's equation for establishing the relation between water saturation and capillary pressure, P_{ow} :

$$S_\alpha = S_{\alpha r} + \frac{S_{\alpha s} - S_{\alpha r}}{\left[1+(ah)^n\right]^m} \tag{5.2.1}$$

Where $S_{\alpha r}$: residual saturation of α-phase

$S_{\alpha s}$: maximum saturation of α-phase

To decide the parameters for equation (3.1.8) and (5.2.1), the experiments are conducted by Lin, et al (1982). Figure 5.2.1 shows the relationship between saturations and capillary pressure calculated from equation (5.2.1). TCE saturation S_o is given by a simple constraint, $S_o = 1 - S_w$.

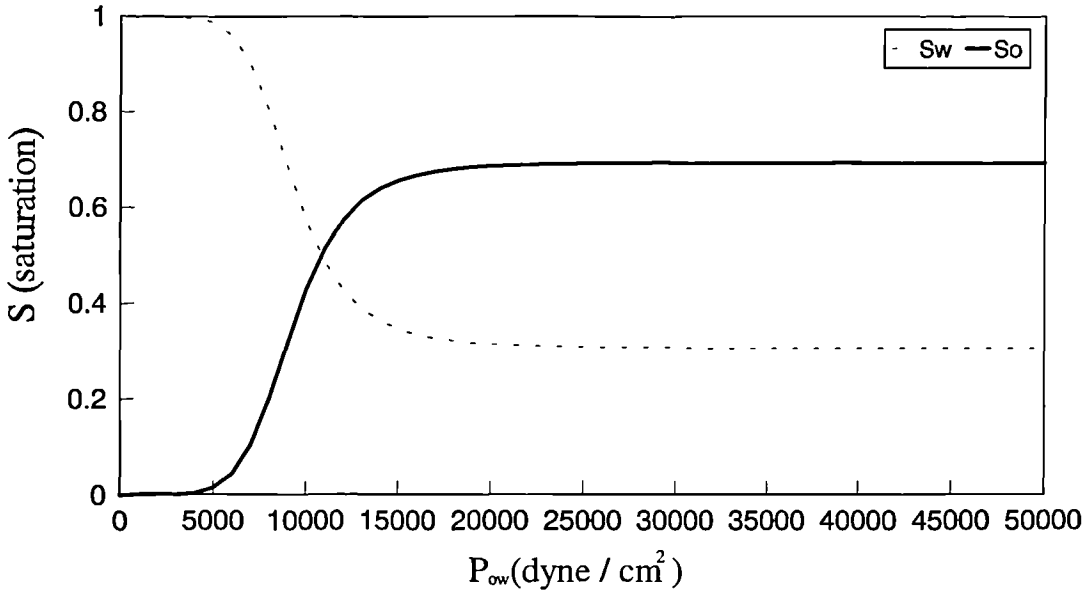


Figure 5.2.1 Capillary pressure curve in water-TCE system

The relative permeabilities of water and TCE are presented in figure 5.2.2, which is based on the data of Lin, et al (1982).

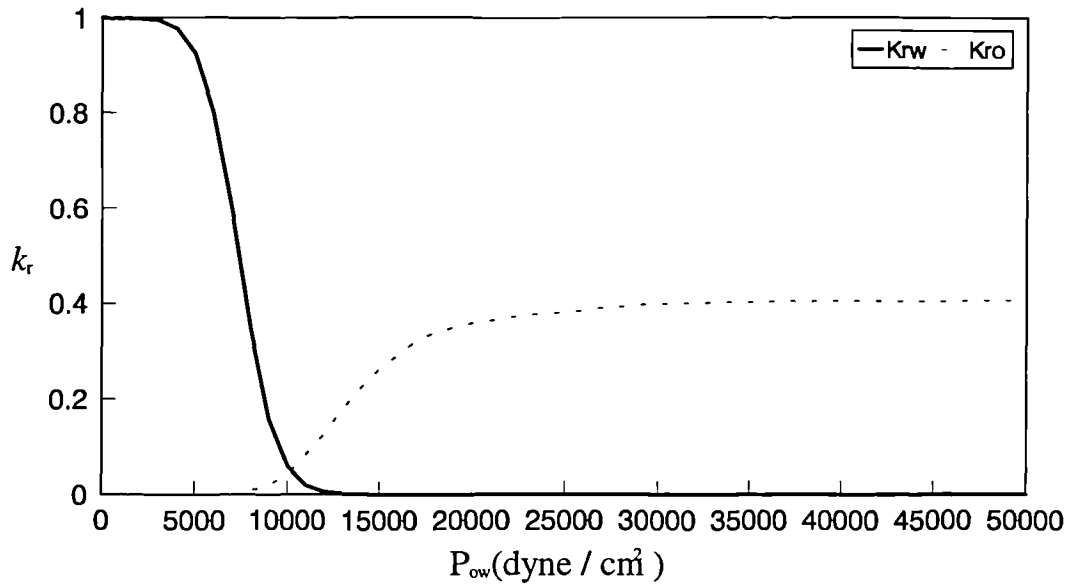


Figure 5.2.2 Relative permeabilities of water and TCE in water-TCE system

Consider a horizontal soil column initially saturated with TCE and water. The length of the column is 19 *cm*. Both fluids are above their residual level. The initial water pressure is 1000 dyne/cm^2 throughout the horizontal column and the organic pressure is 11000 dyne/cm^2 . Thus the initial capillary pressure is P_{ow}= 10000 $dyne/cm^2$. At one end, organic pressure abruptly increases to 15610 dyne/cm^2. No organic flow condition is set at the other end. Water pressure is fixed at its initial conditions at both ends. Table 5.2.2 shows the boundary and initial conditions.

Boundary condition

upper boundary	$G_1 = P_{ow} = 14610\ dyne/cm^2$
	$G_2 = P_w = 1000\ dyne/cm^2$
lower boundary	$\frac{\partial G_1}{\partial x} = \frac{\partial P_{ow}}{\partial x} = 0$
	$G_2 = P_w = 1000\ dyne/cm^2$
Initial condition	$G_1 = P_{ow} = 10000\ dyne/cm^2$
	$G_2 = P_w = 1000\ dyne/cm^2$

Table 5.2.2 Boundary and initial conditions

As shown in table 4.3.1, the general primary variables become $G_1 = P_{ow}$ and $G_2 = P_w$. The water and organic phase equations are used to analyze the pollution pattern (II). The relevant derivatives are expressed in terms of the general primary variables in the code(see table 4.3.2). The column is divided by 18 1-D elements. The length of each element is Δx = 1.0 *cm* equally and the time step, Δt , is 1 *sec*. Simulation results are shown in figure 5.2.3 and 5.2.4 . The comparison of the pressure and saturation

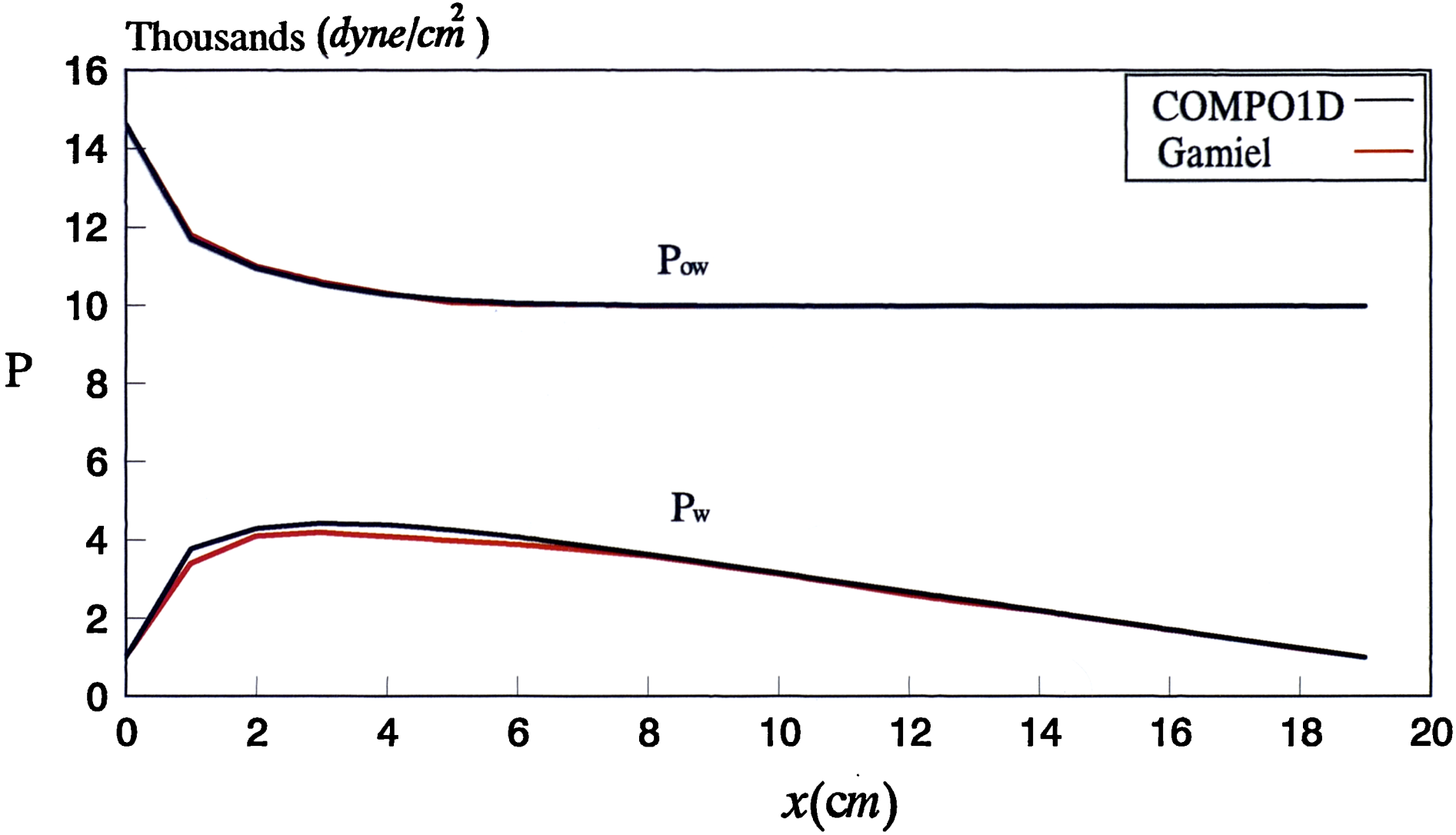


Figure 5.2.3 Capillary pressure and water pressure at time=50 sec

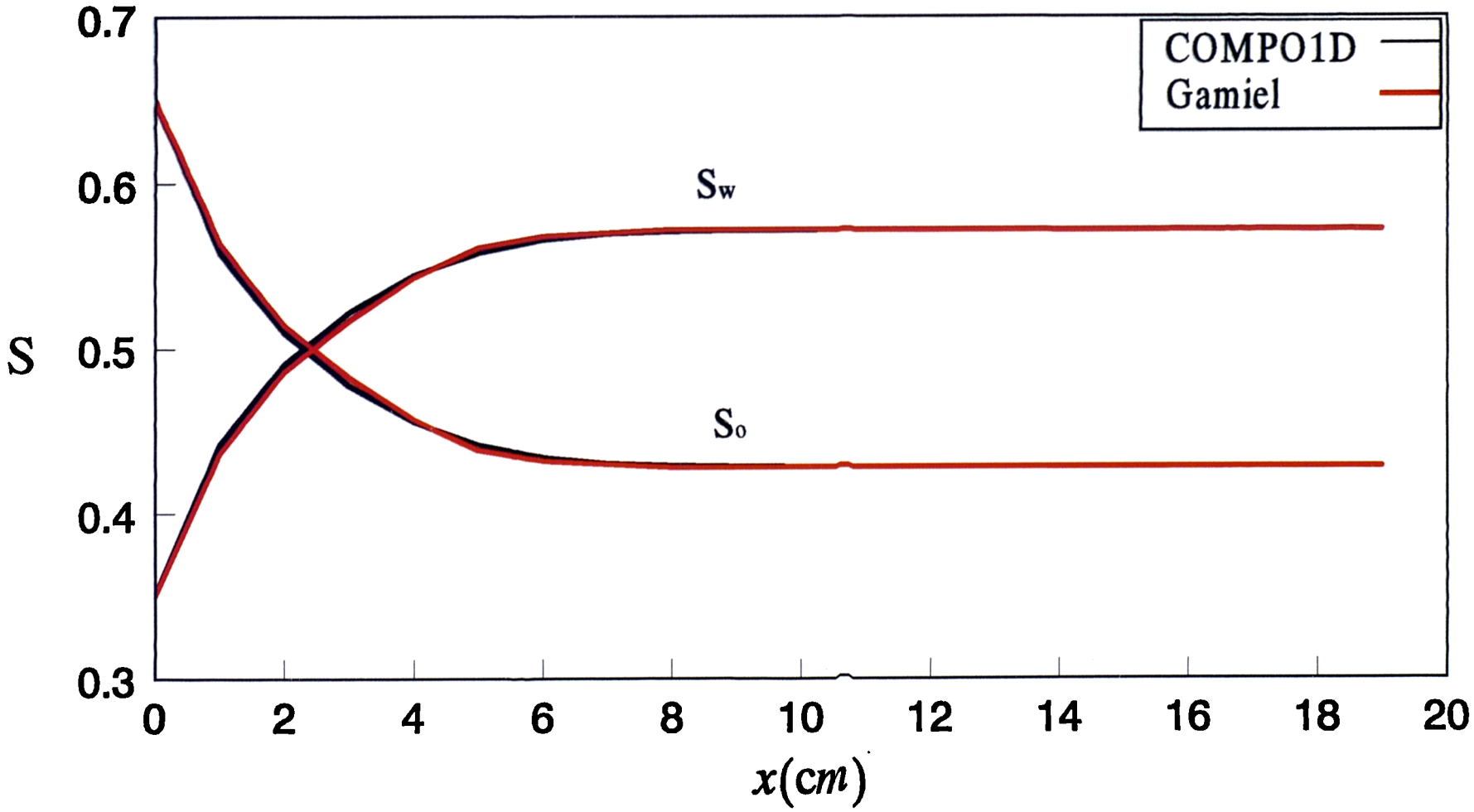


Figure 5.2.4 Saturations at time=50 sec

distributions obtained by COMPO1D and Gamiel's model verifies the applicability of COMPO1D to the pollution pattern (II). The Gamiel's model adopts the finite element scheme to anticipate flows in the subsurface. Thus the comparison of the results of this research and the Gamiel's model also conveys the accuracy of COMPO1D for organic-water system.

5.3 Pollution pattern (VII)

Water, gas and organic phases are present in pore space. Assuming that the porous matrix consists of hydrophilic material, water is the wetting phase. Gas is the nonwetting phase with respect to both of the other liquids. The wettability of TCE phase is between water and gas. No direct contact between the water and gas phase is assumed. The parameters for this example are given by Lin, et al (1982) and Abriola(1983). Table 5.3.1 summarizes the parameters of the fluid phases and solid matrix.

	parameters	value	units
water	M_w	18.02	
	μ_w	1.0019×10^{-2}	poise
	β_w	4.532×10^{-11}	$cm^2 / dyne$
	ρ^{wb}	0.9997964	g / cm^3
	P^{wb}	1.0133×10^6	$dyne / cm^2$
TCE	M_o	131.4	
	μ_o	5.8×10^{-1}	poise
	β_o	0	
	ρ^{ob}	1.4657	g / cm^3
	P^{ob}	1.0133×10^6	$dyne / cm^2$
air	M_A	28.97	
	Z	1.0	
	P^g	1.0133×10^6	$dyne / cm^2$

	g	9.80665×10^{2}	$cm / \sec^2$
	T	293.15	°K
solid matrix	ε	0.36	
	k	5.8231×10^{-7}	cm^2
	β_s	2.0×10^{-10}	cm^2 / *dyne*
residual	S_{wr}	0.306	
saturation	S_{om}	0.17	
partition	ω_e^{wo}	3.018×10^{-4}	
coefficients	ω_e^{gw}	5.549×10^{2}	
dispersion &	D^{mg}	0.039	cm^2 / sec
diffusion	D^{mw}	8.434×10^{-6}	cm^2 / sec
	a^w	0.1	*cm*
	D^{mo}	0	
	a^o	0	

Table 5.3.1 Parameters for example 3

This example assumes that the water relative permeability and saturation are functions of the capillary pressure, P_{ow}, alone even for a three-phase system. It means that the experimental data of the two-phase(water and organic) system can be applied to the three-phase system, as far as water relative permeability and saturation are concerned. Similarly the gas relative permeability and saturation are functions of P_{go} alone in a three-phase system. So they can be determined from the experiments of the two-phase system, although this example deals with the three-phase flow. Figure 5.3.1, 5.3.2, 5.3.3 and 5.3.4 show the relevant relations obtained from the experiments of the two-phase systems, TCE-gas and TCE-water.

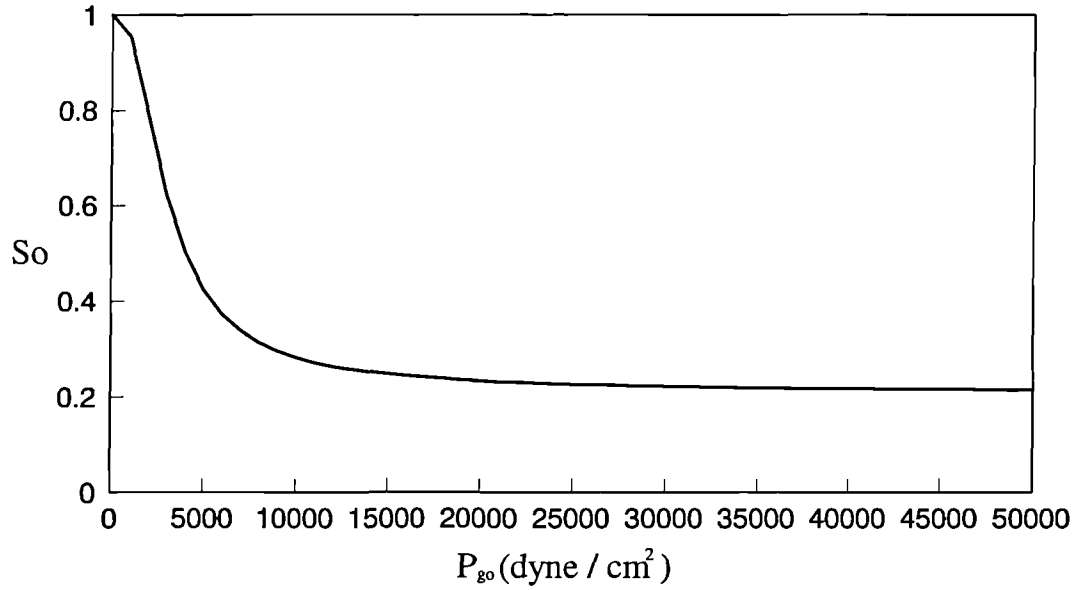


Figure 5.3.1 Capillary pressure curve in TCE-gas system

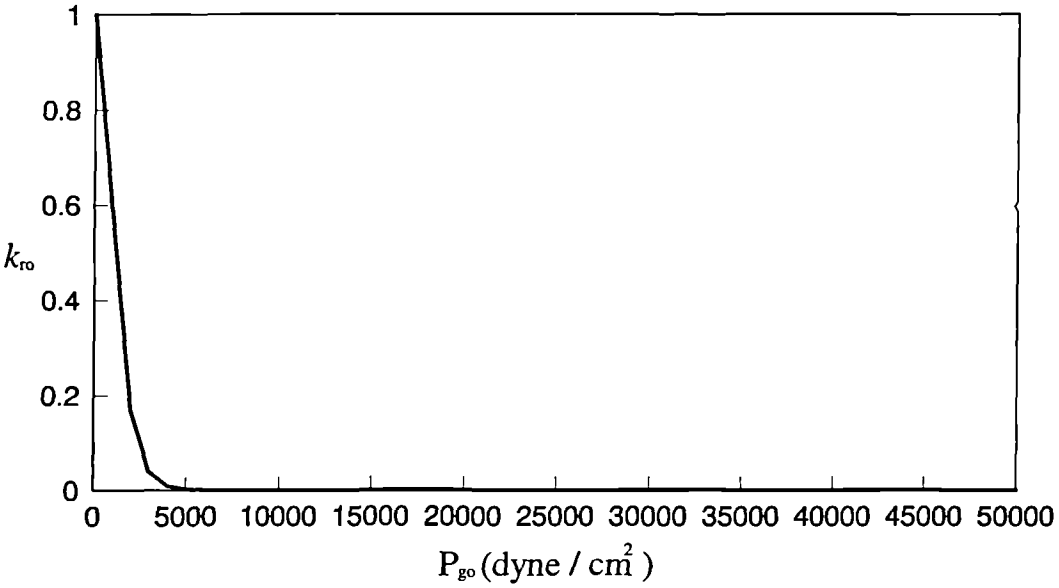


Figure 5.3.2 Relative permeability of TCE in TCE-gas system

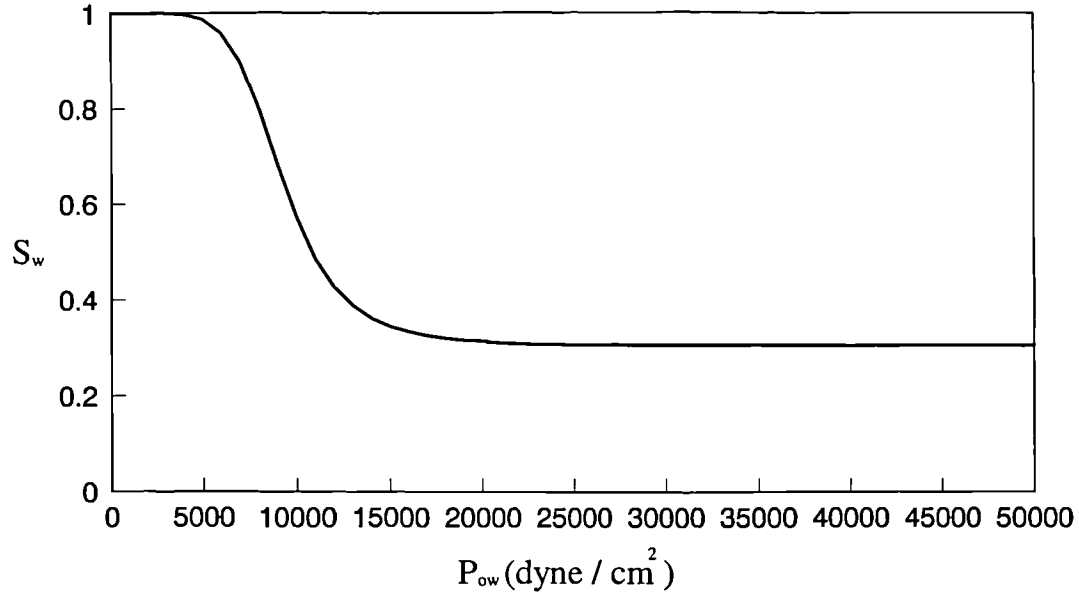


Figure 5.3.3 Capillary pressure curve in TCE-water system

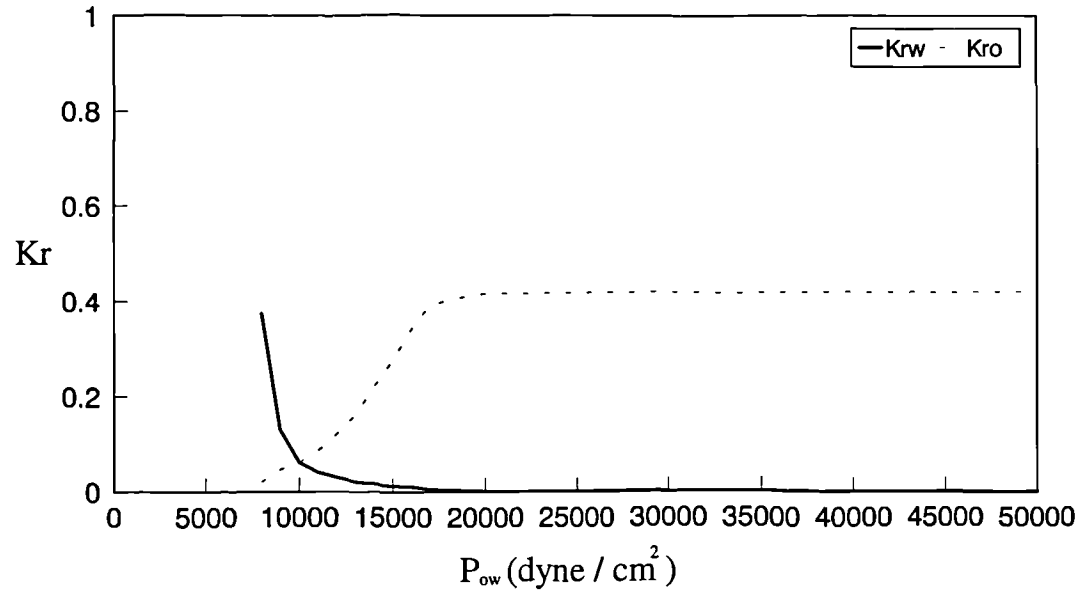


Figure 5.3.4 Relative permeability of TCE and water in TCE-water system

Gas saturation, S_g , and water saturation, S_w , are obtained from figure 5.3.1 and 5.3.3 respectively. Then the saturation of TCE is determined from the constraint ($S_o = 1 - S_w - S_g$). The relative permeability of water is able to be obtained from figure

5.3.4 . The Stone's method, equation(3.1.9), is adopted to determine the organic relative permeability.

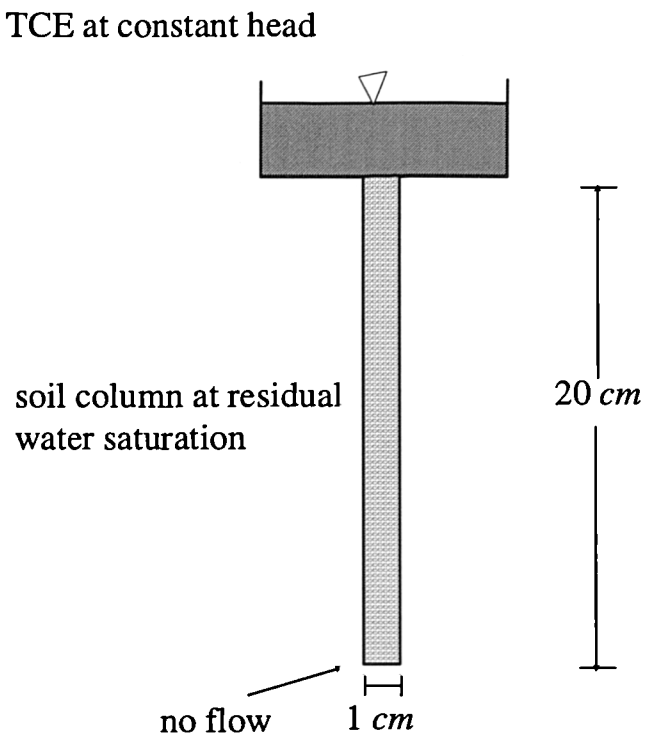


Figure 5.3.5 Sketch of a soil column

A sketch of a TCE soil column which is 20 *cm* long and 1 *cm* wide is shown in figure 5.3.5. TCE infiltrates into the soil column at residual water saturation from the top boundary. Table 4.3.1 and 4.3.2 summarize the general primary variables and relevant derivatives that COMP1D adopts for the pollution pattern (VII). The boundary and initial conditions for the 1-D domain are presented in table 5.3.2 .

Boundary condition

upper boundary $G_1 = P_{ow} = 50000\ dyne / cm^2$

$G_2 = P_{go} = 1500\ dyne / cm^2$

$$G_3 = W_e^o = 0.5$$

lower boundary

$$G_1 = P_{ow} = 50000\ dyne / cm^2$$

$$\frac{\partial G_2}{\partial y} = -\frac{\partial P_o}{\partial y} = -\rho^o g \qquad \leftarrow \text{no TCE flow}$$

$$\frac{\partial G_3}{\partial y} = \frac{\partial W_e^o}{\partial y} = 0$$

Initial condition

$$G_1 = P_{ow} = 50000\ dyne / cm^2$$

$$G_2 = P_{go} = 8300\ dyne / cm^2$$

$$G_3 = W_e^o = 1$$

$$S_w = 0.306$$

$$S_o \cong 0$$

Table 5.3.2 Boundary and initial conditions

To simulate the hypothetical transient flow, a negligible saturation of TCE is assumed at each node at the start of the simulation. However, in the case that TCE saturation is below its residual level, the derivatives of TCE saturation with respect to capillary pressures become zero theoretically. It may cause numerical difficulties, because the mobility, τ_o, becomes zero too. So a minimum value for the saturation derivatives has been incorporated into the numerical model. Numerical experiments have proved that a minimum value of 10^{-6} appears optimum for this simulation. Interphase mass exchange and dispersion/diffusion are considered in this example.

Firstly the domain is considered as a 1-dimensional problem. It is divided into 1 *cm* increments for 21 nodes. Figure 5.3.6, 5.3.7, and 5.3.8 display the simulation results obtained by the 1-D numerical model COMPO1D.

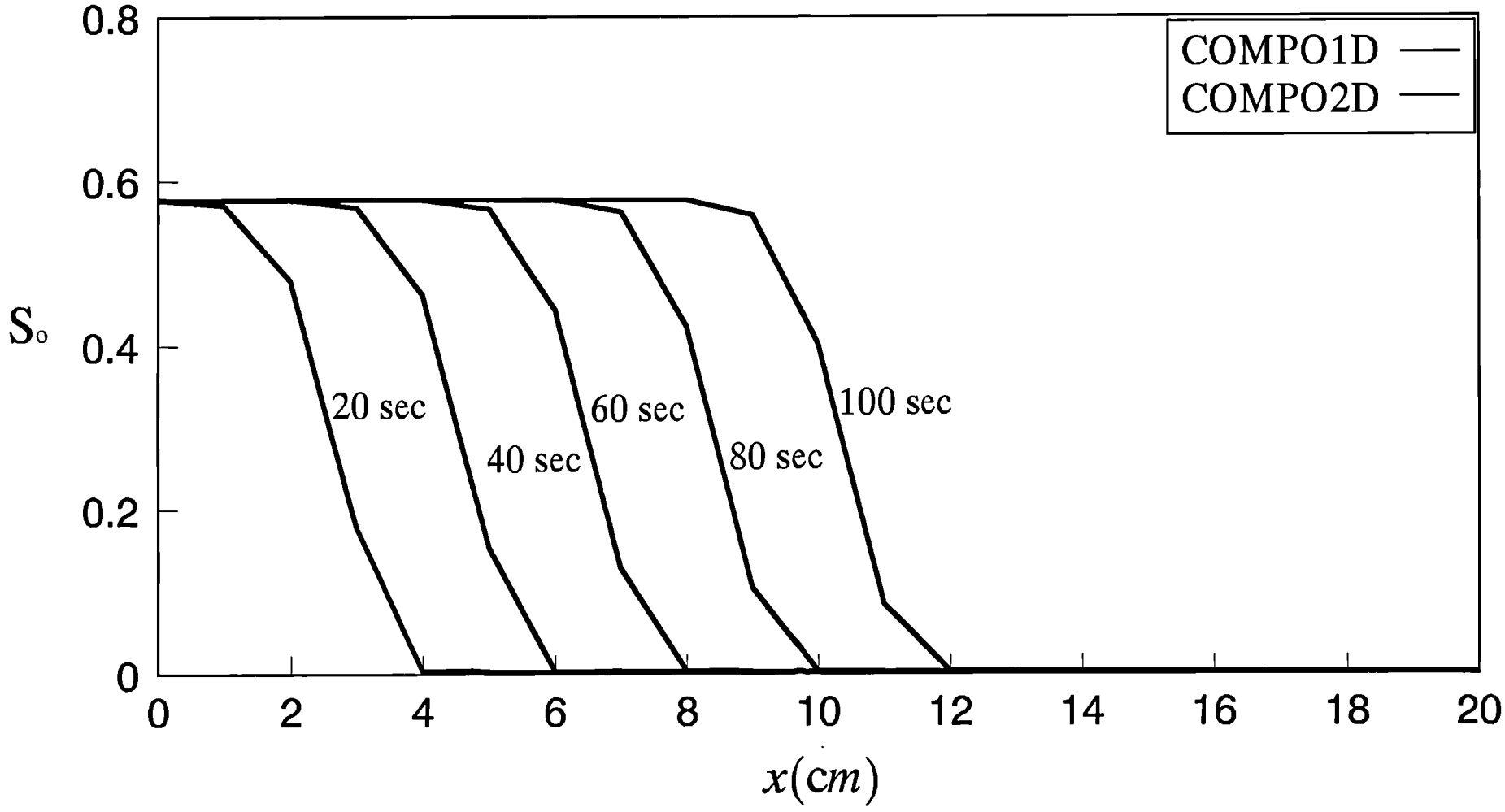


Figure 5.3.6 TCE saturation in the course of time

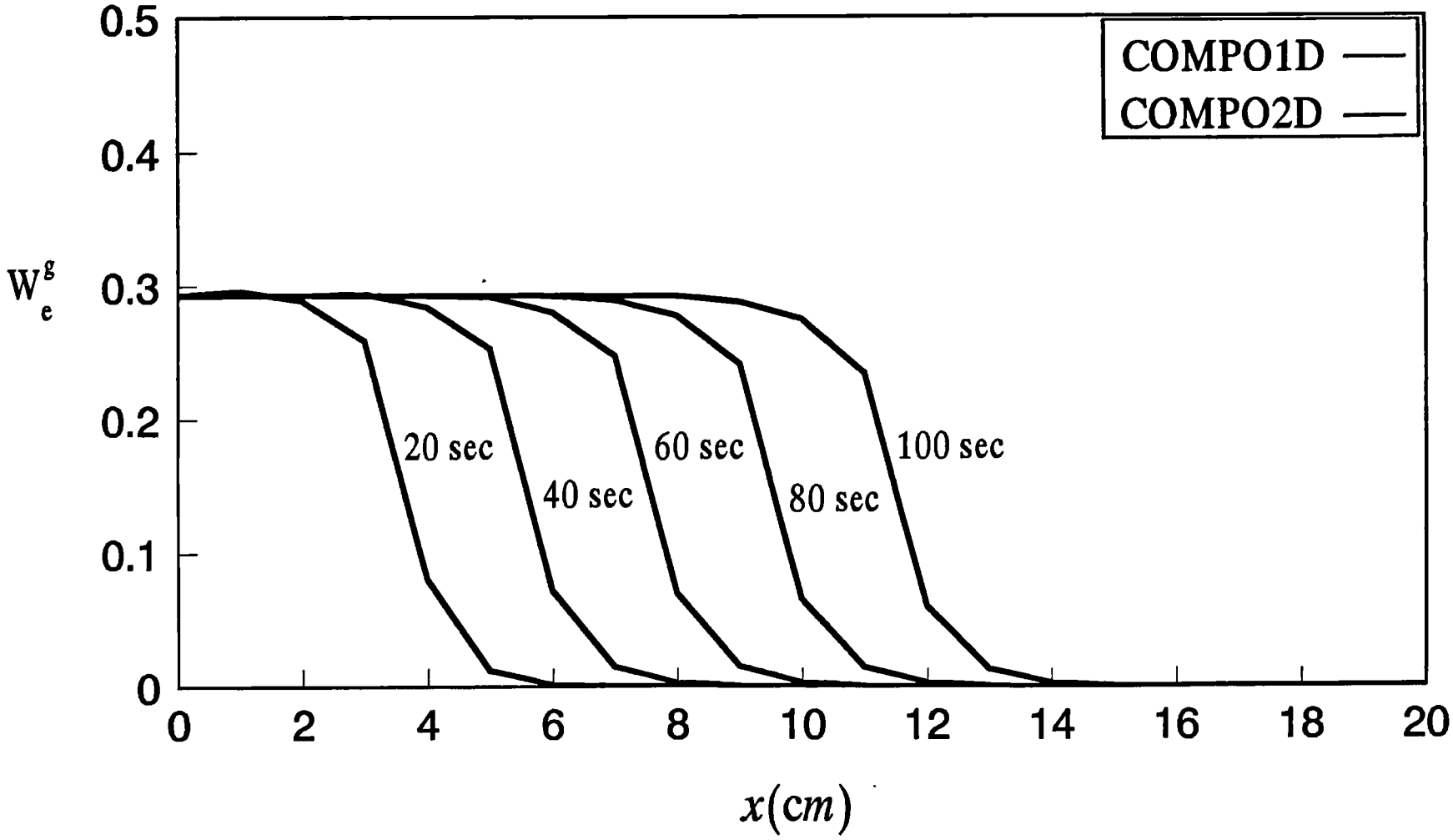


Figure 5.3.7 TCE concentration in the gas phase

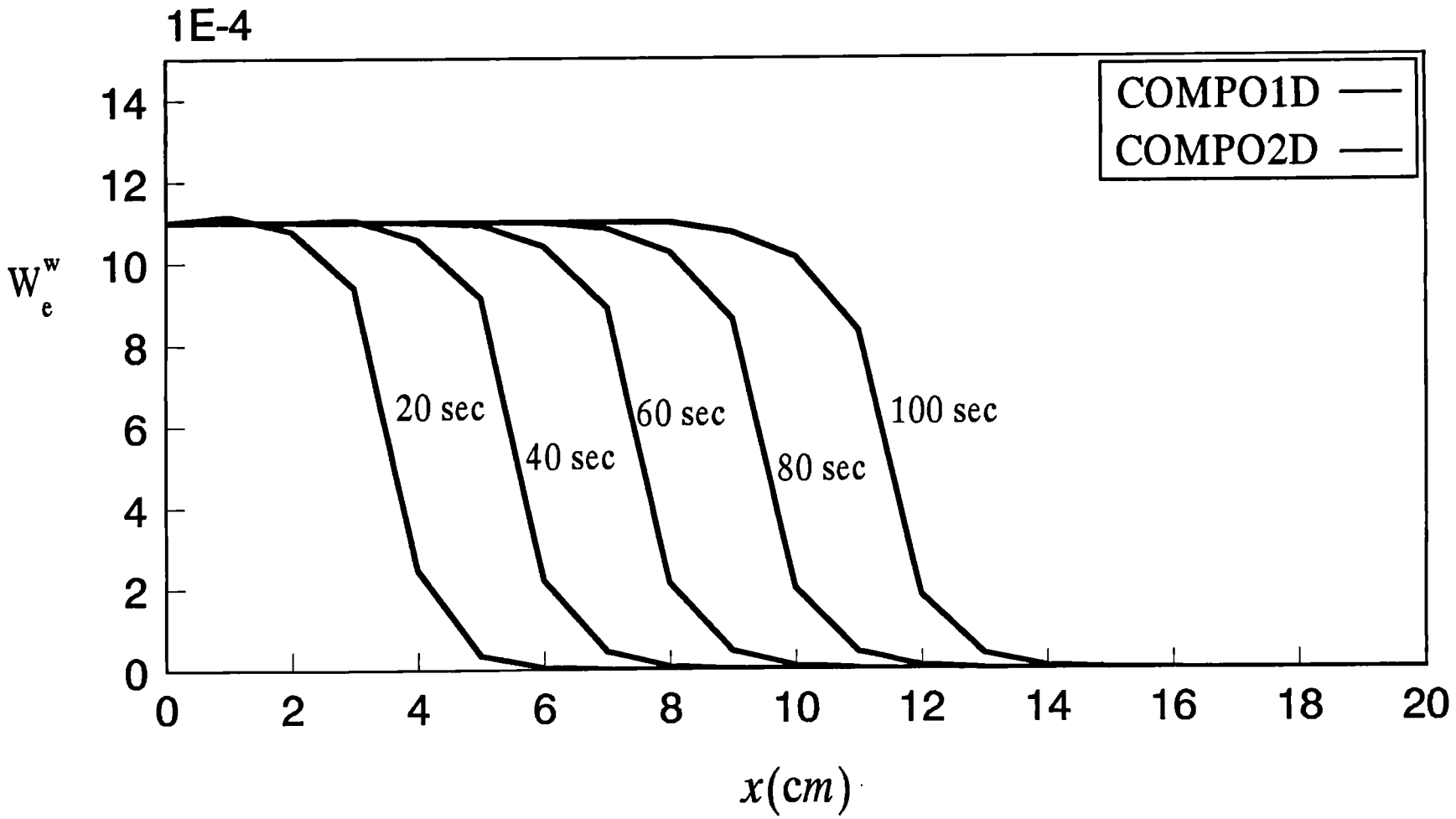


Figure 5.3.8 TCE concentration in the water phase

Figure 5.3.9(a) examines the effect of mesh refinement on the organic saturation profile. Solutions for three nodal spacing are compared at a fixed time, t=20 sec, and time step, Δt =0.5 sec. It shows that the saturation front steepens, as the spacing becomes smaller. For relatively small spacings, Δx=0.5 cm and Δx=1 cm, the simulation results are very similar. It can be said that the simulation results become closer to the true solution as the spacing is smaller. Figure 5.3.9(b) is a semi-log plot of the absolute value of the maximum error, $|\delta G_{2i}|_{max}$, versus the number of iterations. It shows that more iterations are required as the spacing becomes smaller.

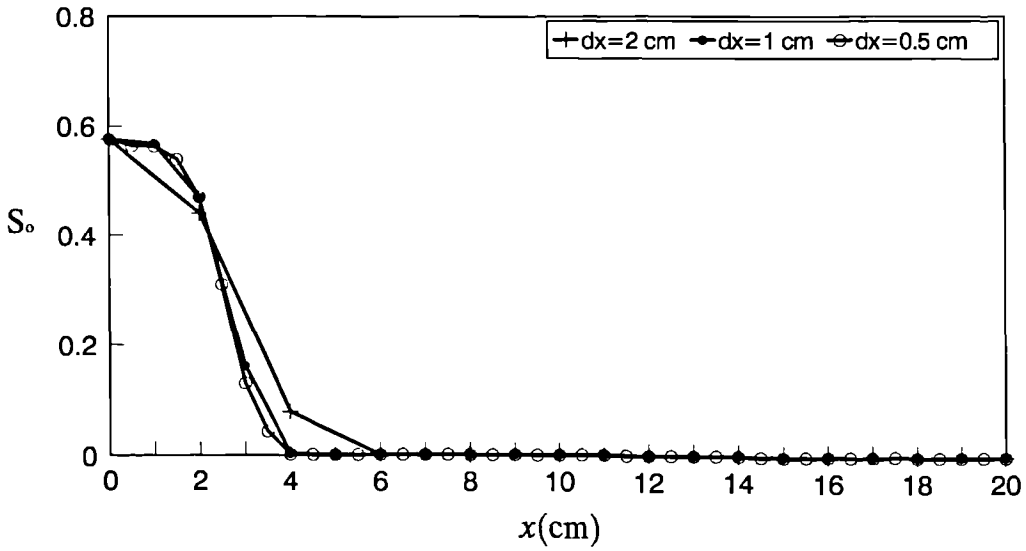


Figure 5.3.9(a) Nodal spacing comparison for Δt=0.5 sec

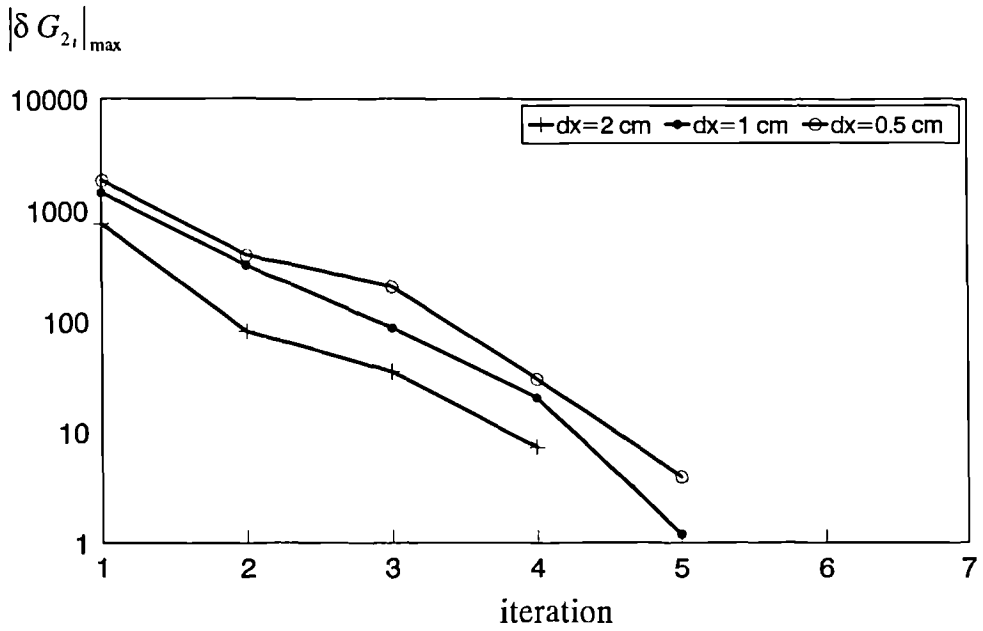


Figure 5.3.9(b) Spatial convergence for Δt=0.5 sec

Figure 5.3.10(a) illustrates the effect of time step size for a fixed nodal spacing Δx=1 cm. Solutions for three time step sizes are compared at a specific time, t=20 sec. The organic saturation curve becomes a little bit steeper as Δt decreases. However, as shown in figure 5.3.10(a), the three time step sizes do not influence the simulation results considerably. Figure 5.3.10(b) shows that solutions are converging for the three different time steps. However, there is a small jump at iteration 4 for Δt =0.1 sec. It also can be observed that higher number of iterations is required for bigger time step size.

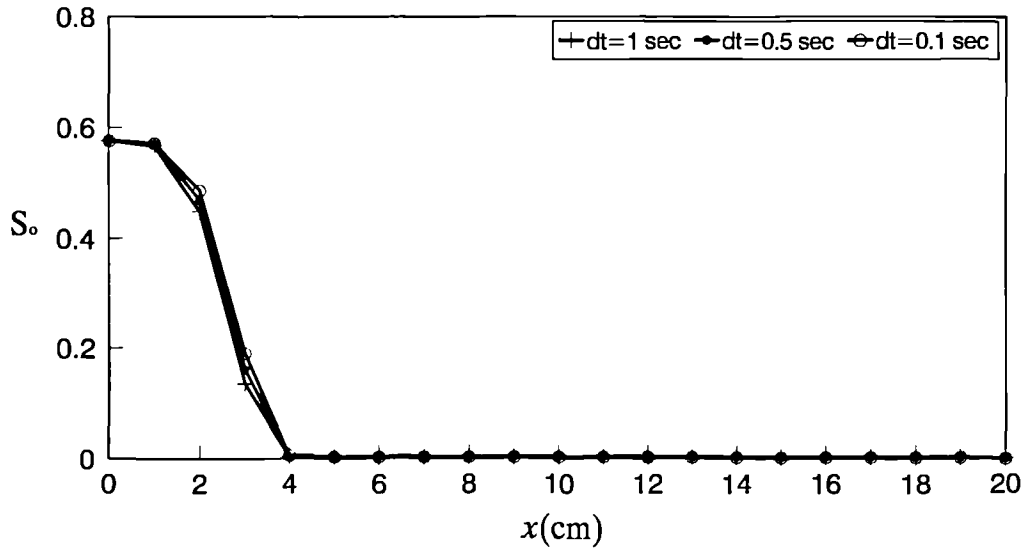


Figure 5.3.10(a) Time step comparison for Δx=1 cm

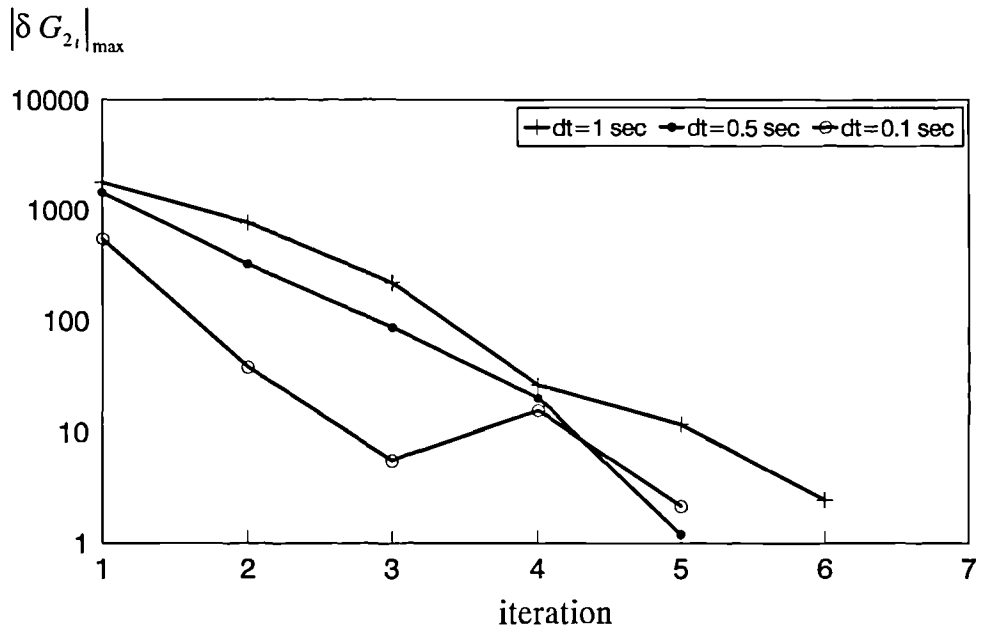


Figure 5.3.10(b) Convergence in time for Δx=1 cm

Secondly the domain is treated as a 2-dimensional problem to verify the 2-D model, COMPO2D. The numbers of nodes and elements is 42 and 20 respectively. Each element is 1 *cm* wide and 1 *cm* high. The simulation results of the 2-D FEM code almost overlap those of the 1-D FEM code. Thus the numerical models, COMPO1D and COMPO2D have been proved to be applicable to the pollution pattern (VII).

5.4 Pollution pattern (VII) - two dimensional simulation

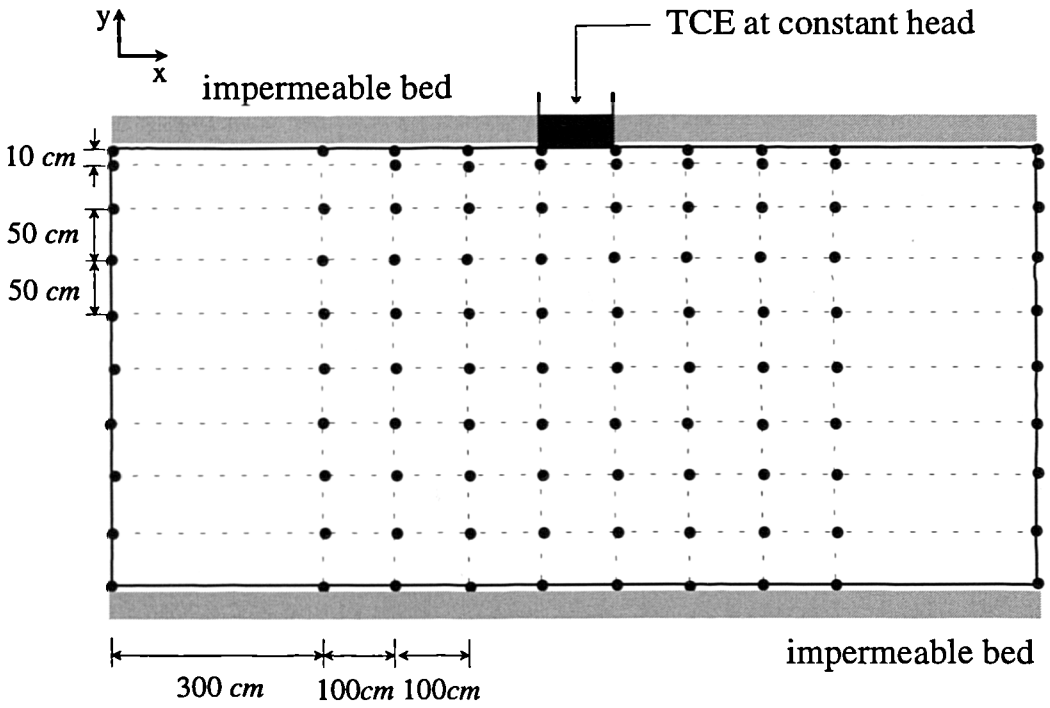


Figure 5.4.1 Sketch for 2-D simulation

This section deals with the full two-dimensional problem depicted in figure 5.4.1. The domain is 4 *m* thick and it extends 13 *m* laterally. As shown in the figure, the domain is composed of 100 nodes (10 rows x 10 columns). TCE infiltrates into the region through a rift in the center of the upper boundary. Initially the region is assumed to contain water at its residual saturation level.

	parameters	value	units
water	M_w	18.02	
	μ_w	1.0019×10^{-2}	poise
	β_w	4.532×10^{-11}	cm^2 / $dyne$
	ρ^{wb}	0.9997964	g / cm^3
	P^{wb}	1.0133×10^6	$dyne$ / cm^2
TCE	M_o	131.4	
	μ_o	5.8×10^{-1}	poise
	β_o	0	
	ρ^{ob}	1.4657	g / cm^3
	P^{ob}	1.0133×10^6	$dyne$ / cm^2
air	M_A	28.97	
	Z	1.0	
	P^g	1.0133×10^6	$dyne$ / cm^2
	g	9.80665×10^2	cm / $\sec^2$
	T	293.15	°K
solid matrix	ε	0.36	
	k	5.8231×10^{-7}	cm^2
	β_s	2.0×10^{-10}	cm^2 / $dyne$
residual	S_{wr}	0.306	
saturation	S_{om}	0.17	
partition	ω_e^{wo}	3.018×10^{-4}	
coefficients	ω_e^{gw}	5.549×10^2	
dispersion &	D^{mg}	0.039	cm^2 / sec
diffusion	D^{mw}	8.434×10^{-6}	cm^2 / sec
	a^w	0.1	cm
	D^{mo}	0	
	a^o	0	

Table 5.4.1 Parameters for example 4

The simulation parameters are presented in table 5.4.1. They are almost same with those in the previous example. Thus the fluid saturations and relative permeabilities are determined by the same procedures used in section 5.3 . The boundary and initial conditions are listed in table 5.4.2 .

Boundary condition

upper boundary

$$G_1 = P_{ow} = 50000\ dyne / cm^2$$

$$\frac{\partial G_2}{\partial y} = -\frac{\partial P_o}{\partial y} = -\rho^o g$$

$$\frac{\partial G_3}{\partial y} = \frac{\partial W_e^o}{\partial y} = 0$$

at nodes 41 & 51 (rift)

$$G_1 = P_{ow} = 50000\ dyne / cm^2$$

$$G_2 = P_{go} = 1500\ dyne / cm^2$$

$$G_3 = W_e^o = 0.5$$

lower boundary

$$G_1 = P_{ow} = 50000\ dyne / cm^2$$

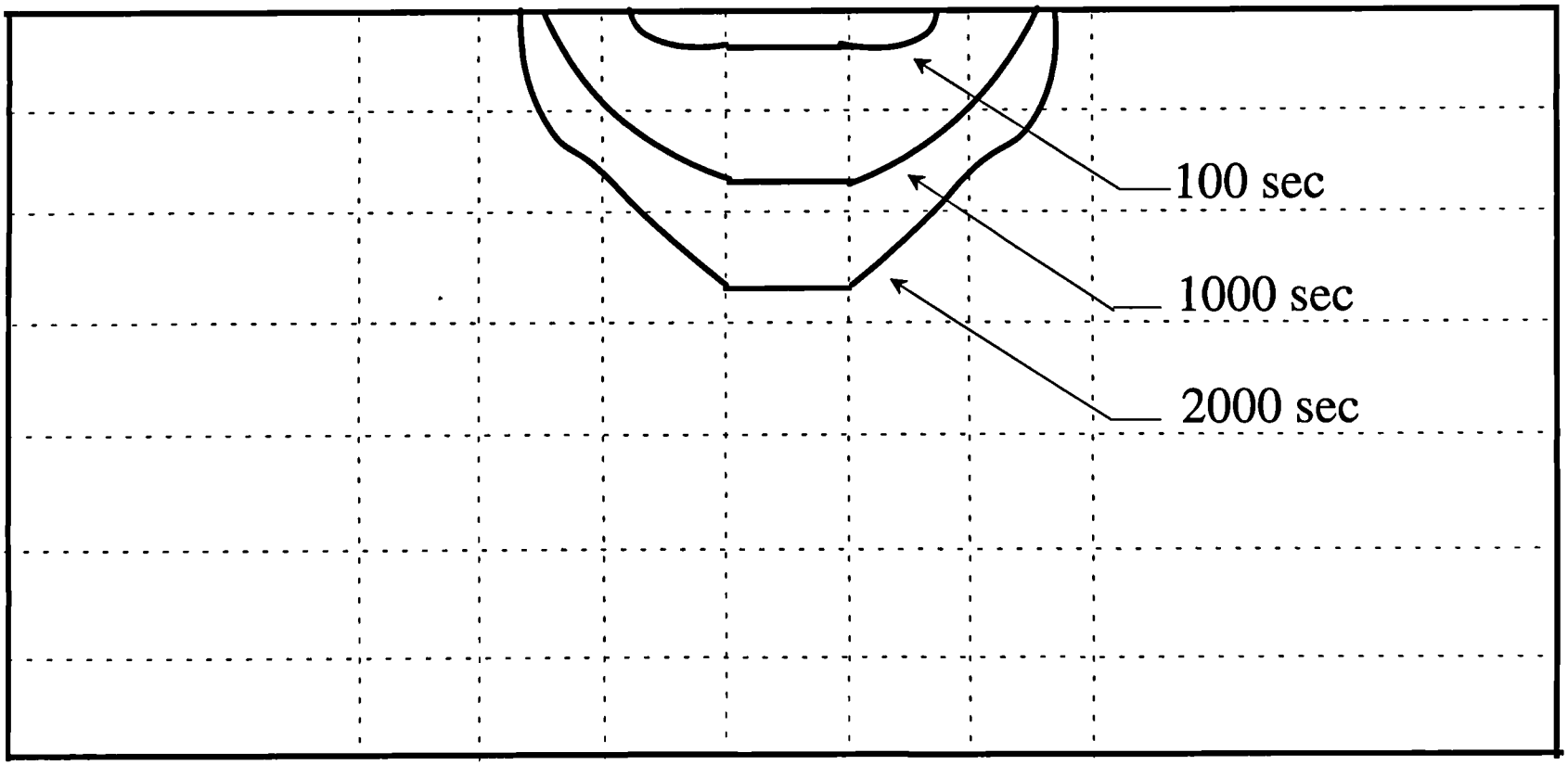
$$\frac{\partial G_2}{\partial y} = -\frac{\partial P_o}{\partial y} = -\rho^o g$$

$$\frac{\partial G_3}{\partial y} = \frac{\partial W_e^o}{\partial y} = 0$$

left boundary	$G_1 = P_{ow} = 50000\ dyne/cm^2$
	$G_2 = P_{go} = 8300\ dyne/cm^2$
	$G_3 = W_e^o = 1$
right boundary	$G_1 = P_{ow} = 50000\ dyne/cm^2$
	$G_2 = P_{go} = 8300\ dyne/cm^2$
	$G_3 = W_e^o = 1$
Initial condition	$G_1 = P_{ow} = 50000\ dyne/cm^2$
	$G_2 = P_{go} = 8300\ dyne/cm^2$
	$G_3 = W_e^o = 1$
	$S_w = 0.306$
	$S_o \equiv 0$

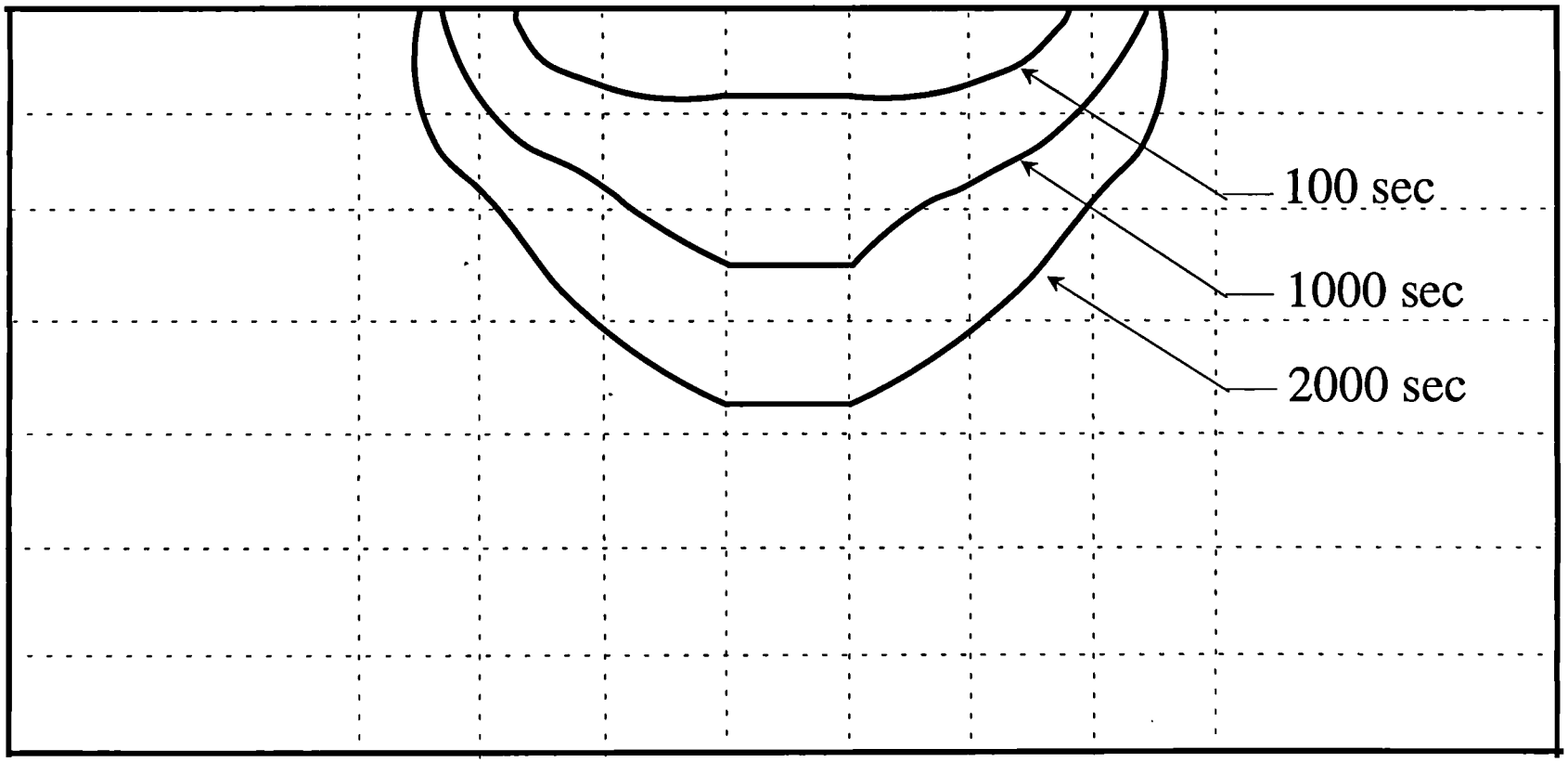
Table 5.4.2 Boundary and initial conditions

COMPO2D is used to solve this scenario. Simulation results are shown in figure 5.4.2, 5.4.3, and 5.4.4. TCE propagation is presented in figure 5.4.2 for various time steps. Three different simulation times are presented. Each contour delineates the region within which the organic phase saturation is greater than 0.1. Figure 5.4.3 and 5.4.4 show the movements of e component in the gas and water phases respectively. Each contour in figure 5.4.3 and 5.4.4 delineates the region respectively within which W_e^w or W_e^g is greater than $1x10^{-4}$ or 0.01. Here, linear interpolation between nodal values are used to develop the contours shown in the figures. A time step of 10 seconds are employed for the grid shown in figure 5.4.1.



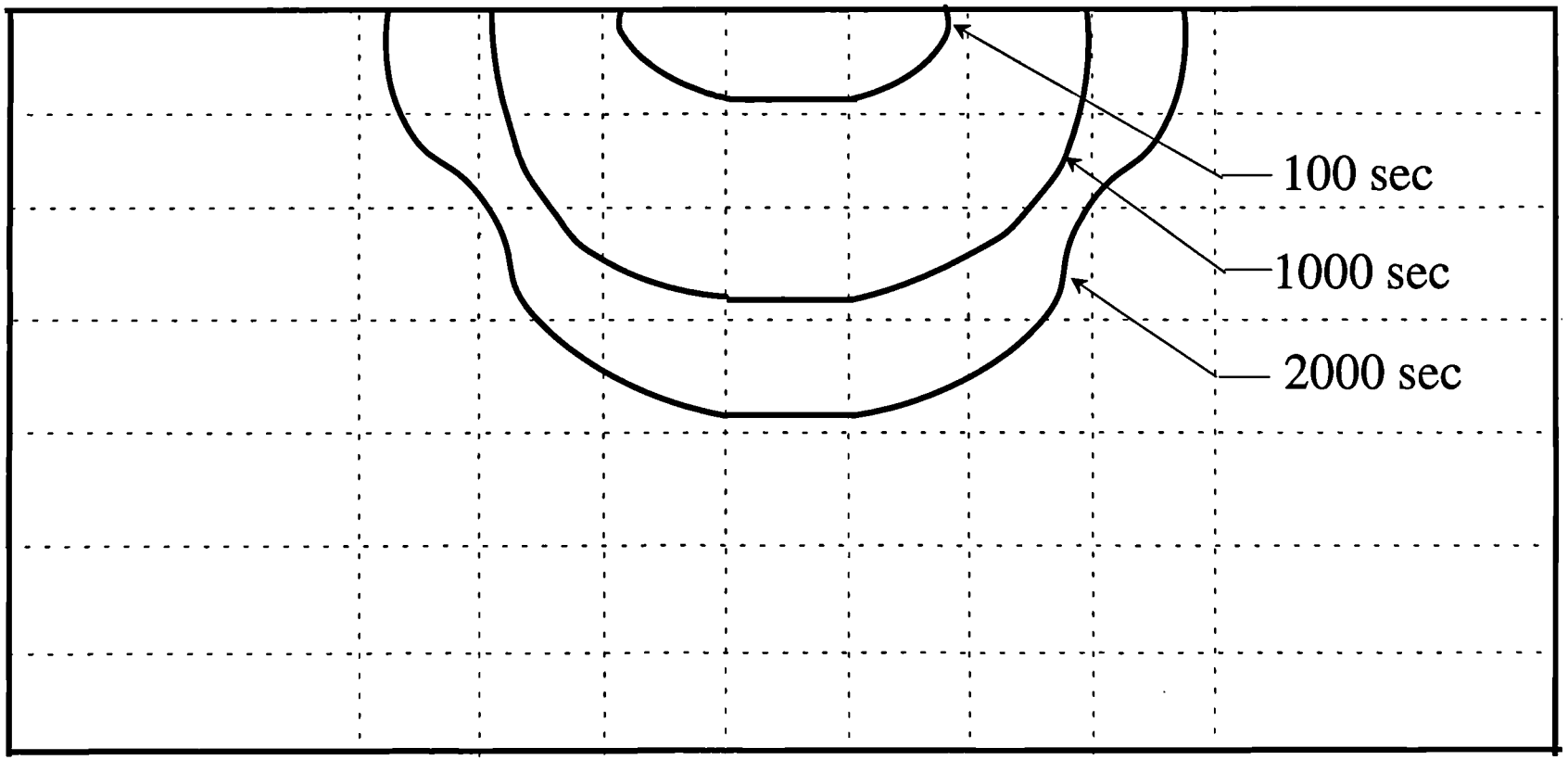
Contour represents S_o=0.1

Figure 5.4.2 TCE migration as a phase



Contour represents 100 ppm in the water phase

Figure 5.4.3 TCE migration in water phase



Contour represents 10,000 ppm in the water phase

Figure 5.4.4 TCE migration in gas phase

Two more grid types are considered in figure 5.4.5 and 5.4.6 to show the grid effect. Grid type 2 has 100 nodes as grid type 1 (see figure 5.4.1). However it has smaller spacing near the rift. On the other hand, grid type 3 divides the domain into 6 rows x 10 columns(60 nodes).

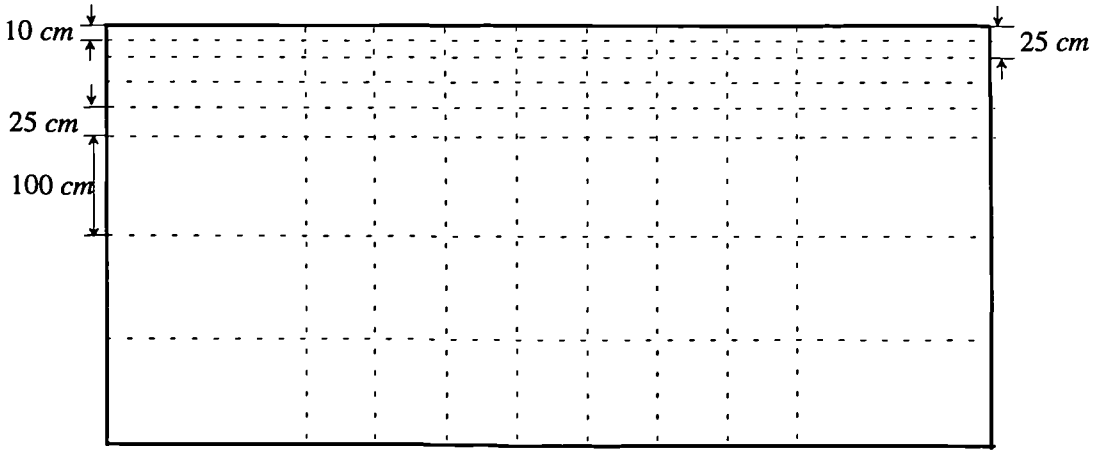


Figure 5.4.5 Grid type 2

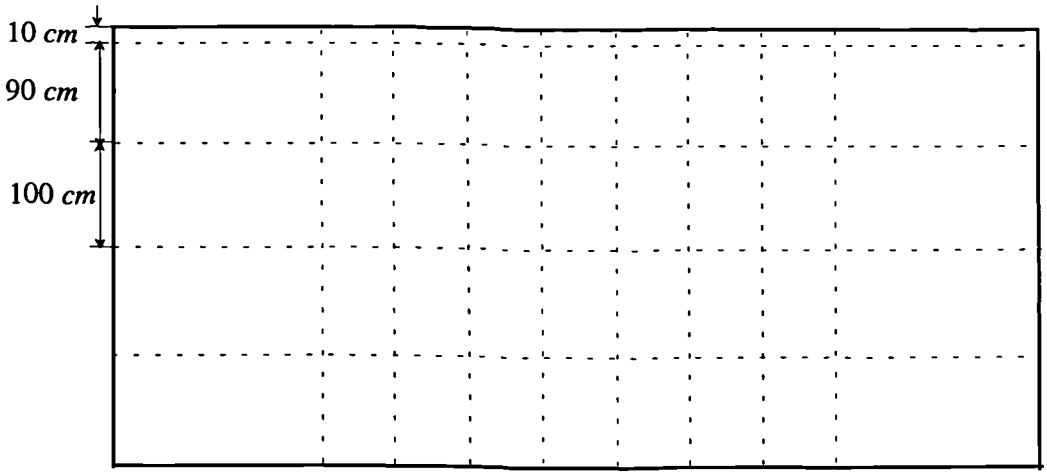


Figure 5.4.6 Grid type 3

Figure 5.4.8(a) examines the effect of mesh refinement on the organic saturation profile. Solutions for the three grid types are compared at a fixed time, t=1000 sec, and time step, Δt =10 sec. It shows that the organic phase spreads farther as the grid is more refined. Grid type 2 which has dense grid in front shows high values of the saturation of the organic phase. Thus, if more accurate information on the saturation

distribution is desired, the grid should be more refined. Figure 5.3.9(b) is a semi-log plot of the absolute value of the maximum error, $|\delta G_{2i}|_{max}$, versus the number of iterations. Figure 5.4.8(b) shows that solutions are converging for the three different time steps. However, the curve for grid 1 exhibits different behavior.

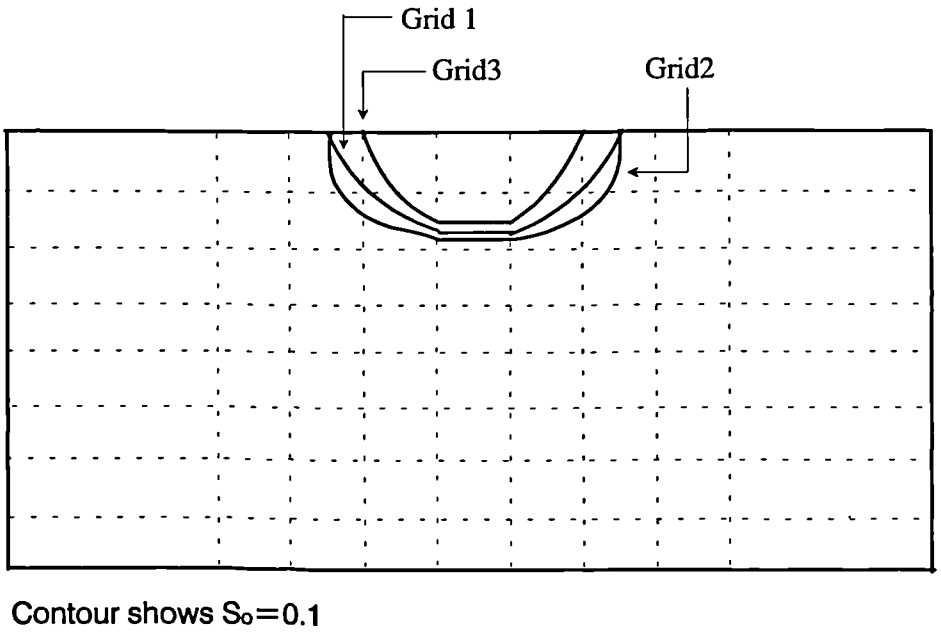


Figure 5.4.8(a) Grid type comparison for Δt=10 sec

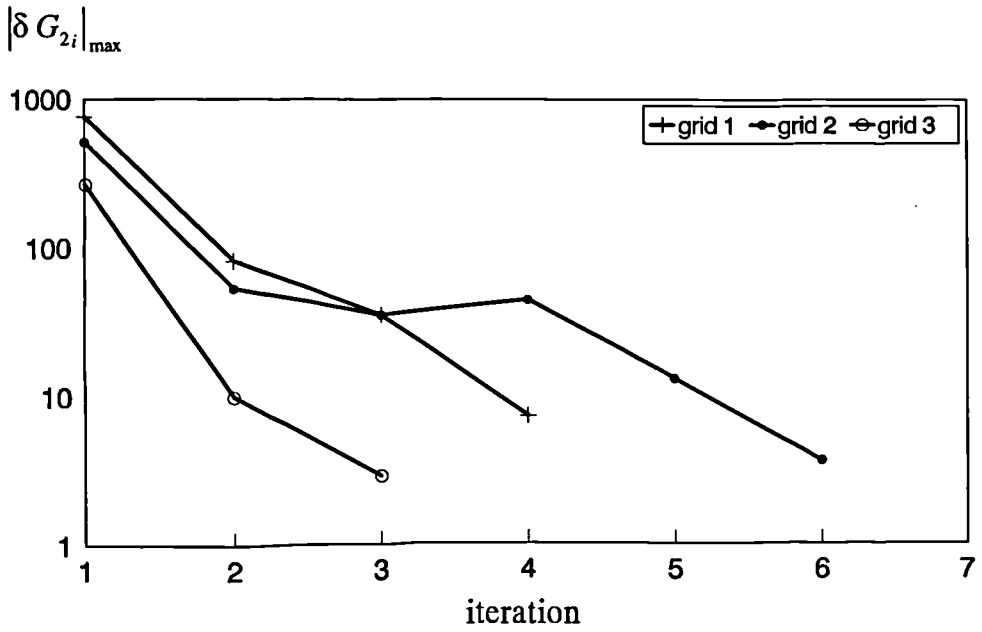


Figure 5.3.9(b) Spatial convergence for Δt=10 sec

Figure 5.4.9(a) illustrates the effect of time step size for grid type 1. Solutions for three time step sizes are compared at a specific time, t=1000 sec. As shown in figure 5.4.9(a), The simulation results are almost same for the time step sizes, Δt =5 sec and Δt =10 sec. On the other hand, TCE infiltrates a little bit farther for the time step size, Δt =20 sec. Thus, it can be said that Δt =10 sec is small enough to describe accurately the migration of the organic phase. Figure 5.4.9(b) shows that solutions are converging for the three different time steps. However, as in figure 5.4.8(b), the curve for Δt =10 sec exhibits different behavior. Because the curves in figure 5.4.8(b) and 5.4.9(b) are obtained for the worst cases, the unstable converging behavior does not continue. On average, 3-4 iterations are required per time step for the convergence criteria after a few time steps.

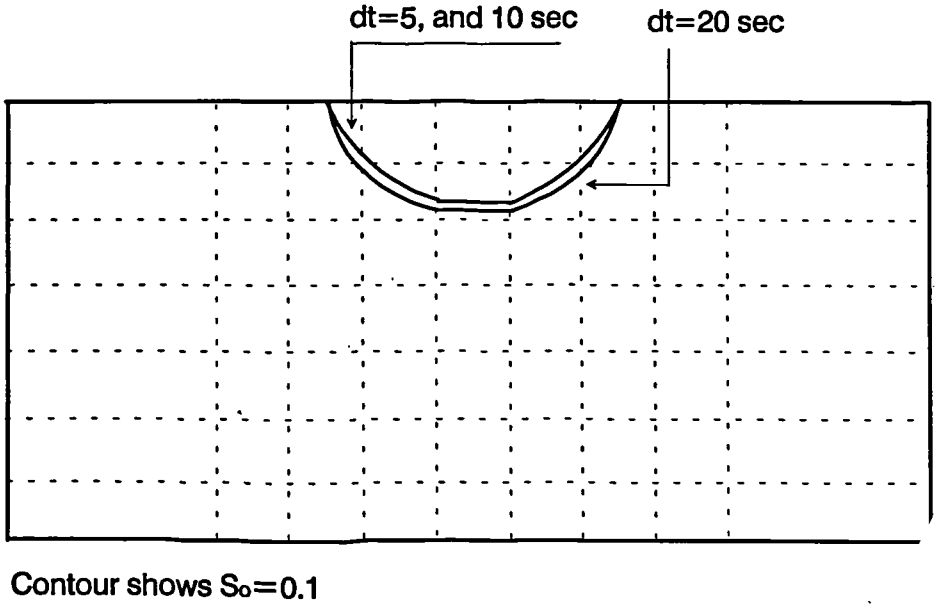


Figure 5.4.9(a) Time step comparison for grid 1

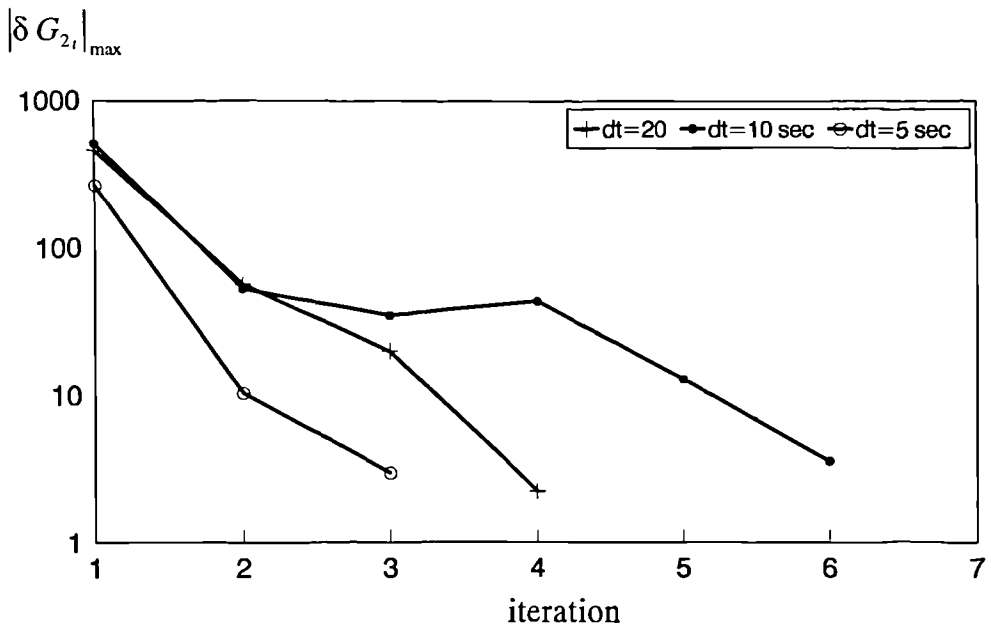


Figure 5.4.9(b) Convergence in time for grid 1

5.5 Extended use of the code (Tracer experiment)

This section is contributed to show the extended capacity of the code by applying it to a simple tracer scenario. Let us consider a horizontal soil column initially saturated with TCE and water. The length of the column is 10 cm. Both fluids are above their residual level. Table 5.2.1 are used again to show the properties of solid matrix, water, and TCE. Figure 5.2.1 and 5.2.2 are also employed to establish the relations between capillary pressure and saturations or relative permeabilities.

As in section 5.2, the initial water and organic pressures are assumed to be 1000 dyne/cm^2 and 11000 dyne/cm^2 respectively throughout the horizontal column. Thus the initial capillary pressure is P_{ow} = 10000 $dyne / cm^2$. However, at one end, an organic phase which contains a tracer of 1000 ppm is introduced, increasing organic pressure

to 15610 $dyne / cm^2$. Water pressure is fixed at its initial conditions at both ends. Table shows the boundary and initial conditions.

Boundary condition

upper boundary	$G_1 = P_{ow} = 14610\ dyne / cm^2$
	$G_2 = P_w = 1000\ dyne / cm^2$
	$G_3 = W_e^o = 0.999$
lower boundary	$G_2 = P_w = 1000\ dyne / cm^2$
Initial condition	$G_1 = P_{ow} = 10000\ dyne / cm^2$
	$G_2 = P_w = 1000\ dyne / cm^2$
	$G_3 = W_e^o = 1$

Table 5.5.1 Boundary and initial conditions

As it can be observed, this tracer example is able to be categorised as the pollution pattern (III) which allows the interphase mass exchange. To display the effect of partitioning of the tracer between water and organic phase, numerical simulations are conducted for the three partition coefficient, ω_e^{wo} = 0, 5 x 10^{-3} , and 7.5 x 10^{-3}. The length of each element is Δx = 1.0 *cm* equally and the time step size, Δt =5 *sec*.
The breakthrough curves at the other end are compared in figure 5.5.1 for the three partition coefficients. Figure 5.5.1 shows that the tracer concentration becomes smaller as the partition coefficients becomes larger. It is because the tracer transfers to the water phase in the course of migration.

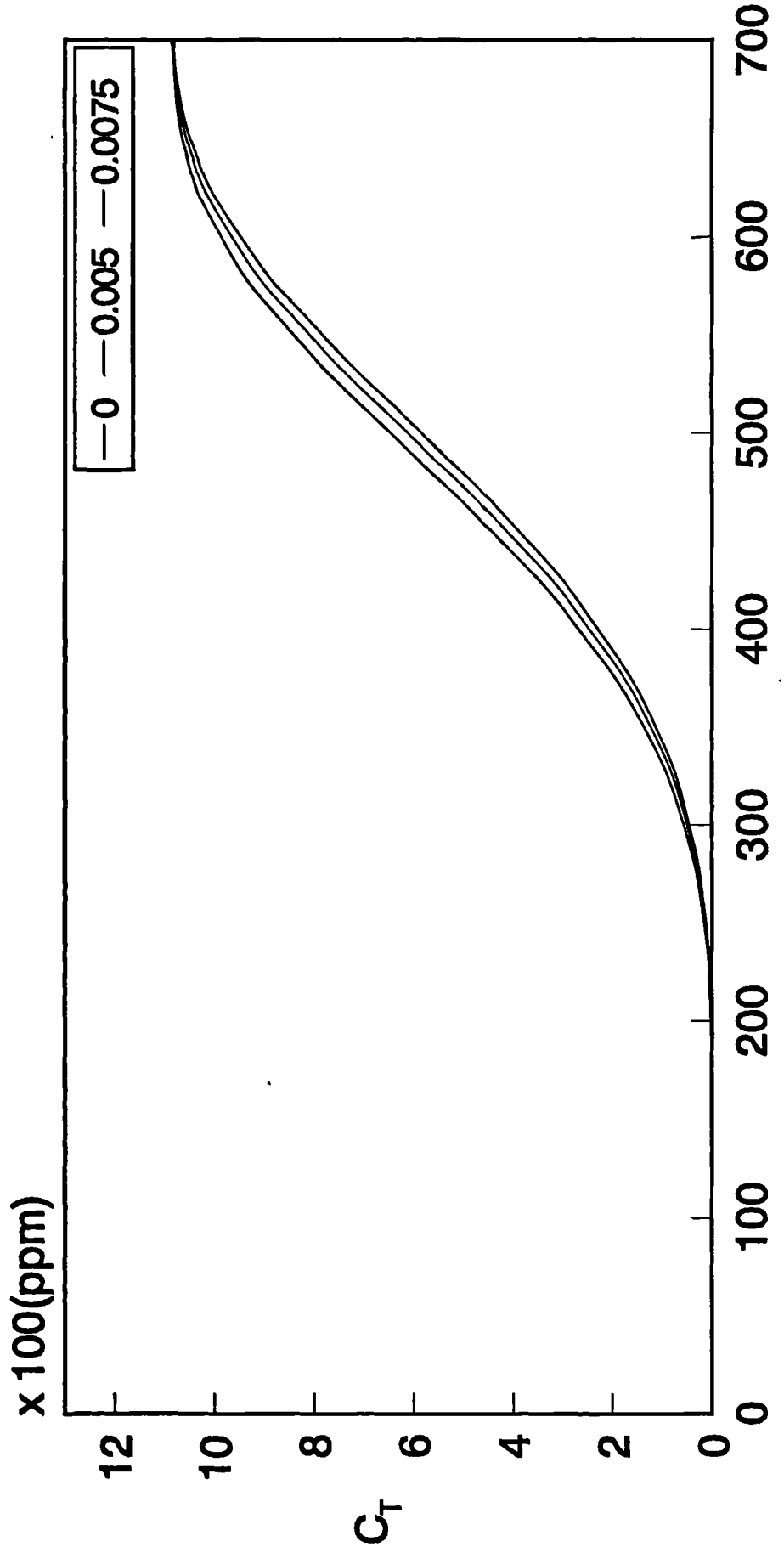


Figure 5.5.1 Tracer concentration at the end node

Figure 5.5.2 displays the saturations of organic phase at a certain time step for the varying spacings. For Δx=0.5 cm and Δx=1 cm, the simulation results are very similar. On the other hand, the curve for Δx=2 cm is slightly off from the others.

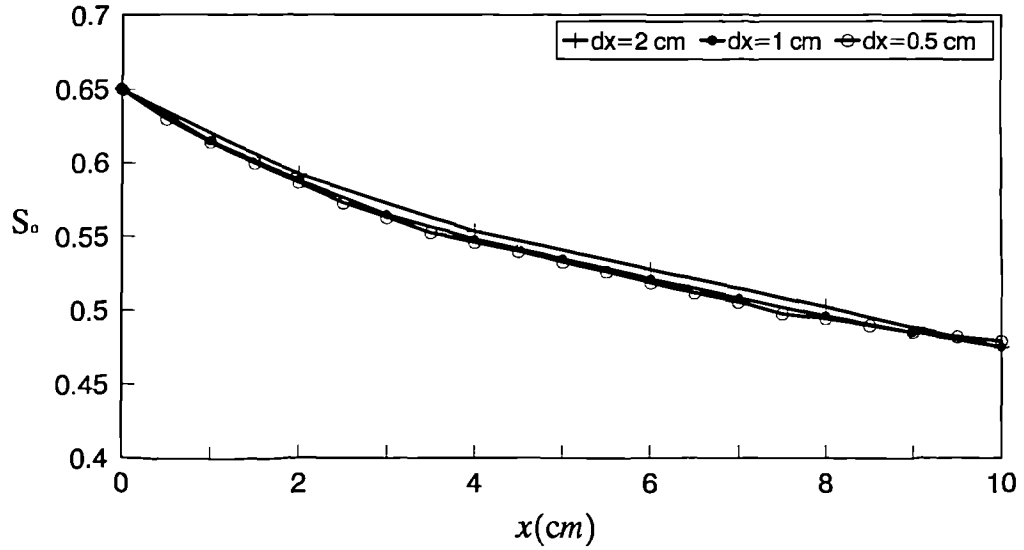


Figure 5.5.2 Nodal spacing comparison for Δt=5 sec and time=200 sec

From the simulation results in figure 5.5.3, it can be said that the effect of the time step size is minor for the three time step sizes, Δt=1 sec, Δt=5 sec, and Δt=10 sec.

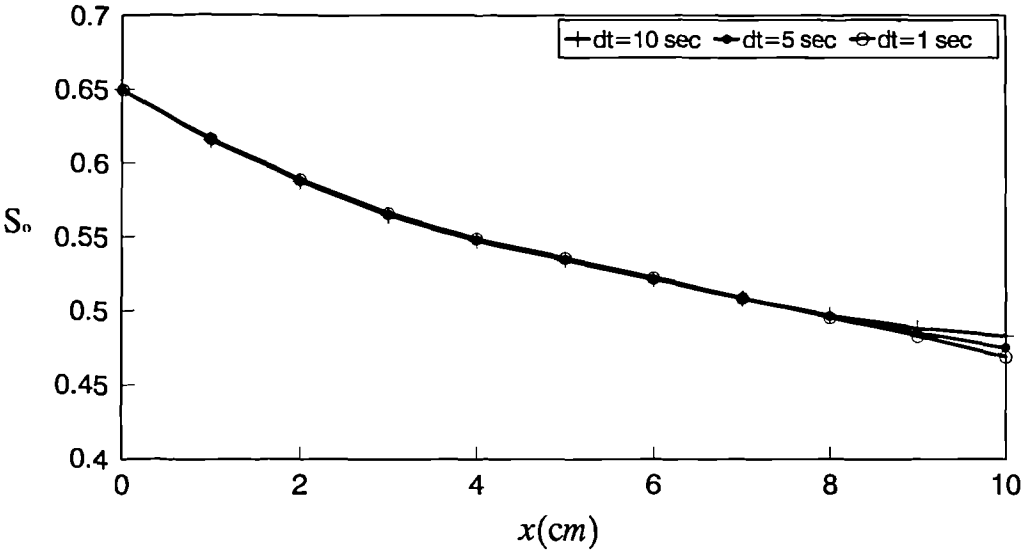


Figure 5.5.3 Time step comparison for Δx=1 cm and time=200 sec

5.6 Summary

The example of the pollution pattern(I) conveys the accuracy of the algorithm of COMPO which is concerned with the water phase equation and the e component equation. The application of COMPO to the pollution pattern(II) proves that the organic component equation is coded correctly into the model. Finally, applying COMPO to the pollution type(VII), it is proved that the concepts of diffusion/dispersion and interphase mass transfer are correctly considered in the model. Therefore, the other pollution patterns categorized in this study also can be treated by using COMPO. On top of that, the applicability of the code can be extended even to tracer experiments.

CHAPTER VI
CONCLUSIONS AND RECOMMENDATIONS

6.1 Conclusions

This thesis has been dedicated to tracing the movements of pollutants in porous media. Table 3.2.1 shows pollution patterns that can take place in the subsurface. The pollution patterns can be categorised by the property of solid matrix, fluid phases in a porous medium, and interphase mass transfer among fluid phases. The wettability order is determined by whether the solid matrix is hydrophilic or oliophilic. Four combinations of fluid phases are considered in categorizing the pollution patterns: water-gas, water-organic, organic-gas, and water-organic-gas. Interphase mass transfer is allowed in this study by dissolution and evaporation and it also contributes to determining the pollution patterns.

The general primary variables are employed to derive the mathematical model. Matrix and fluid compressibility, capillarity, diffusion and dispersion, and interphase mass transfer are all incorporated in deriving the three governing equations for the analysis of multiphase flows in heterogeneous porous media. However, the modelers do not have to use all the three governing equations for some of the pollution patterns. Table 3.2.1 indicates which governing equations are used for the pollution patterns. The variations of the general primary variables and the relevant derivatives are categorized in table 4.3.1, 4.3.2, and 4.3.3, according to the pollution patterns.

The numerical simulators for 1- and 2-dimension are developed to solve the nonlinear partial differential equations employing a finite element discretization method. A Newton-Raphson method is chosen to solve the resultant system of nonlinear algebraic equations.

To demonstrate its applicability, firstly the numerical model, COMPO1D, is applied to a two-phase flow (water and gas) system where contaminated water infilterates from the top boundary(pollution type I). The accuracy of the 1-D model, COMPO1D is verified by comparing the simulation results with those of Van Genuchten's model. Secondly a TCE and water phase flow is considered to verify the applicability of

COMPO1D(pollution type II). The comparison of the simulation results among the relevant numerical models shows that COMPO1D is able to predict the movements of TCE and water. Thirdly the three fluid phases(water, gas, and TCE) are assumed to exist in a domain of interest, taking into account interphase mass exchange(pollution type VII). For this example, the convergence properties of the code are shown by grid and time step refinement. Additionally the 2-D model, COMPO2D is applied to the same 1-D example to verify its algorithm. The results of the 1-D and 2-D models are nearly overlapped. The applicability of 2-D model, COMPO2D, is demonstrated by applying it to a full 2-dimensional TCE migration scenario. The potential applicability of the code to tracer problems is illustrated by applying it to a simple tracer problem.
In conclusion, this research has developed a generalized compositional numerical model that is able to deal with various pollution problems. The generalizing procedures have been fully presented throughout this thesis.

6.2 Recommendations

Basically it is very complex to accurately describe subsurface flows. The complexity increases much more, if multiphase flows are considered. On top of this, there could be various pollution types in the subsurface as this study presents. Thus, more than one numerical model are required to solve these different types of pollution problems. The completely different pollution mechanisms may take place within a domain of concern, because of the state of the existing fluids in a porous medium or the change of the property of solid matrix. However, most of the established models have very restricted range of application. The newly developed numerical model, COMPO, is able to overcome these difficulties. If the primary variables are fixed as in most of models, their experimental data also should be obtained on the basis of the fixed assumptions. Thus, many experimental data for the subsurface contaminations may be useless to some modelers who make use of only one or two models. However, the numerical model, COMPO, can handle with many kinds of experimental data and experimental methods.

Although COMPO has big advantages, the idea of combining procedures is not difficult. Thus, if a numerical code is formulated for anticipating the movements of three-phase flows in the subsurface, it is recommendable to adopt the concept of the general primary variables and the generalizing procedures. Changing part of the code, its application range can be wider. Furthermore, if the code is developed under the condition of interphase mass exchange, it can deal with all the pollution patterns categorized in this study.

This model could be very useful for tracer problems. However, according to the properties of tracers, it may be required to change part of the code which is concerned with partitions between tracers and fluid phases.

REFERENCES

Abifadal, N. "On the Modeling of Two-Phase Media by the Finite Element Method." PhD, Colorado University at Boulder, 1991.

Abriola, Linda. M. "Mathematical Modeling of the Multiphase Migration of Organic Compounds in a Porous Medium." Ph.D. Thesis, Univ. Princeton, 1983.

Abriola, Linda. M. and George F. Pinda. "A Mulitple Approach to the Modeling Porous Media Contaminant at Organic Compounds: 2. Numerical Simulation." Water Res. Res. 21(1) (1985a): 19-26.

Abriola, Linda. M. and George F. Pinda. "A Mulitple Approach to the Modeling Porous Media Contaminant at Organic Compounds: 1. Equation Development." Water Res. Res. 21(1) (1985b): 11-18.

Atkin, R.J. and R.E. Crain. "Continuum Theories of Mixtures: Applications." J. Inst. Math. Appl. 17 (1976):

Atkin, R.J. and R.E. Crain. "Continuum Theories of Mixtures: Basic Theory and Historical Developments." Q. J. Mech. Appl. Math. 29 (1976):

Aziz, K. and A. Settari. Petroleum Reservoir Simulation. London: Applied Science Publ., 1979.

Bachmat, Y. "Spatial Macroscopization of processes in Hetrogeneous Systems." Israel J. Technol. 10 (1972): 391-403.

Baehr, Arthur L. "Selective Transport of Hydrocarbons in the Unsaturated Zone Due to Aqueous and Vapor Phase Partitioning." Water Res. Res. 23 (10 1987): 1926-1938.

Bear, Jacob. Dynamics of Fluids in Porous Media. American Elsevier, 1972.

Bear, Jacob. Hydraulics of Groundwater. London: McGraw-Hill Book Co., 1979.

Bear, Jacob and Y. Bachmat. "A Generalized Theory on Hydrodynamic Dispersion in Porous Media." In IASH Symp. on Artificial Recharge and Management of Aquifers in Haifa, Israel, IASH Publ., 7-16, Year.

Bear, Jacob and Y. Bachmat. Introduction to Modeling of Transport Phenomena in Porous Media. Netherlands: Kluwer Academic Publishers, 1990.

Bear, Jacob and J. M. Buchlin. Modeling and Applications of Transport Phenomena in Porous Media. Netherlands: Kluwer Academic Publishers, 1991.

Bowen, R. M. "Theory of Mixtures." Continuum Physics 3 (1 1976):

Brooks, R. H. and A. T. Corey. "Hydraulic Properties of Porous Media." Hydrol. Pap. 3 (1964):

Brooks, R. H. and A. T. Corey. "Prosperties of Porous Media Affecting Fluid Flow." J. Irr. and Drain. Div. ASCE 92 (IR2 1966): 61-88.

Butterworth, D. and G. F. Hewitt. Two-Phase Flow and Heat Transfer. London: Oxford Univ. Press, 1977.

Coats, K. H. "In-Situ Combustion Model." Soc. Pet. Eng. J. (1980): 533-553.

Coats, K. H. "Reservoir Simulation: State of Art." J. Pet. Tech. 34 (8 1982): 533-553.

Corapcioglu, M. Yavuz and Arthur L. Baehr. "A Compositional Multiphase Model for Groundwater Contamination by Petroleum Products: 1. Theoretical Considerations." Water Res. Res. 23 (1 1987a): 191-200.

Corapcioglu, M. Yavuz and Arthur L. Baehr. "A Compositional Multiphase Model for Groundwater Contamination by Petroleum Products: 2. Numerical Soultion." Water Res. Res. 23 (1 1987b): 201-213.

Corey, A. T. Mechnics of Immiscible Fluids in Porous Media. Littleton, Colo.: Water Resources Publishers, 1986.

Corey, A. T., C. H. Rathjens, J. H. Henderson, and M. R. J. Wyllie. "Three-Phase Relitive Permeability." Trans. AIME 207 (1956): 349-351.

Crookston, R. B., W. E. Culham, and W. H. Chen. "A Numerical Simulation Model for Thermal Recovery Processes." Soc. Pet. Eng. J. (1979): 37-58.

Dilling, W. L. "Interphase Transfer Process: II. Evaporation Rates of Chloro Methanes, Ethanes, Ethylens, Propanes, and Propylenes from Dilute Aqueous Soultions:Comparison with Theoretical Predictions." Environ. Sci. Tech. 11 (4 1977): 405-409.

Eringen, A. C. and J. D. Ingram. "A Continuum Theory of Chemically Reacting Media." I. Int J. Eng. Sci. 3 (1965):

Faust, Charles R. "Transport of Immiscible Fluids within and below the Unsaturated Zone: A Numerical Model." Water Res. Res. 21 (4 1985): 587-596.

Fried, J. J., P. Muntzer, and L. Zilliox. "Groundwater Pollution by Transfer of Oil Hydrocarbons." Ground Water 17 (6 1979): 586-594.

Gallant, R. H. "Physical Properties of Hydrocarbons, Pt. 6: Chlorinated Ethylenes." Hydrocarbon Processing 45 (6 1966):

Gamiel, A. "Simulation of Immiscible Multiphase Flow in Porous Media Using a Moving Finite Element Algorithm." PhD, Michigan University, 1989.

Gerald, Curtis F. and Patric O. Wheatley. Applied Numerical Analysis. 4 ed., Addison-Wesley Publishing Company, 1989.

Gouse, S. W. An Index to the Two-Phase Gas-Liquid Flow Literature. Cambridge, Mass.: M.I.T. press, 1966.

Gray, W. G. "Derivation of Vertically Averaged Equations Describing Multiphase Flow in Porous Media." Water Res. Res. 18 (6 1982): 1705-1712.

Gray, W. G. and P. C. Y. Lee. "On the Theorems for Local Volume Averaging of Multiphase Systems." Int. J. Multiphase Flow. 3 (1977): 333-340.

Gray, W. G. and K. O'Neill. "On the General Equations for Flow in Porous Media and their Reduction to Darcy's Law." Water Res. Res. 12 (2 1976): 148-154.

Hassanizadeh, M. and W. G. Gray. "General Conservation Equations for Multi-Phase systems: 1. Averaging Procedure." Adv. in Water Res. 2 (1979): 131-144.

Hassanizadeh, M. and W. G. Gray. "General Conservation Equations for Multi-Phase systems: 2. Mass, Momenta, Energy, and Entrophy Equations." Adv. in Water Res. 2 (1979): 191-203.

Hochmuth, D. P. "Two-Phase Flow of Immiscible Fluids in Groundwater Systems." MSc, Colorado State University, 1981.

Kappeler, T. and K. Wuhrmann. "Microbial Degradation of the Water-Soluble Fraction of Gas Oil-I." Water Res. 12 (5 1978): 327-333.

Kia, Sheila F. "Subsurface multiphase flow of organic contaminants: Model development and validation." Wat. Res. 25 (1991): 1225-1236.

Kilzer, L., I. Scheunert, H. Geyer, W. Klein, and F. Korte. "Laboratory Screening of the Volatilization Rates of Organic Chemicals from Water and Soil." Chemosphere 8 (10 1979): 751-761.

Kuppusamy, T., J. Sheng, J. C. Parker, and R. J. Lenhard. "Finite-Element Analysis of Multiphase Immiscible Flow through Soils." Water Res. Res. 23 (4 1987): 625-631.

Langsrud, O. "Simulation of Two-Phase Flow by Finite Element Method." In The 4'th Synposium of Numerical Simulation of Reservoir Performance of the Soc. of Petroleum Engineers of AIME in Los Angeles, Year.

Lenhard, R.J. and J.C.Parker. "Experimental Validation of the Theory of Extending Two-Phase Saturation-Pressure Relations to Three-Fluid Phase Systems for Monotonic Drainage Paths." Water Res. Res. 24 (3 1988): 373-380.

Leverett, M. C. "Capillary Behavior in Porous Solids." Trans. AIME 142 (1941): 152-160.

Leverett, M. C. and W. B. Lewis. "Steady Flow of Gas-Oil-Water Mixtures through Unconsolidated Sands." Trans. AIME 142 (1941): 107-113.

Lin, C. "Modeling the Flow of Immiscible Fluids in Soils." Soil Science 143 (4 1987): 293-300.

Lin, C., G. F. Pinder, and E. F. Wood. Water and Trichloroethylene as Immiscible Fluids in Porous Media. Princeton University, 1982. Water Resources Prog. Rept. 83-WR-2.

Lujan S., Carlos A. "Three-Phase Flow Analysis of Oil Spills in Partially Water-Saturated Soils." PhD, Colorado State University, USA, 1985.

Lyman, W. J., W. F. Reehl, and D. H. Rosenblatt. Handbook of Chemical Property Estimation Methods, Environmental Behavior of Organic Compounds. McGraw-Hill, N. Y., 1982.

Lymann, W. J., W. F. Reehl, and D. H. Rosenblatt. Handbook of Chemical Property Estimation Methods, Environmental Behavior of Organic Compounds. McGrqw-Hill, 1982.

Marly, C. M. Multiphase Flow in Porous Media. Paris: Gulf Publishing Company, 1981.

Mercer, J. W. and Charles R. Faust. "The Application of Finite-Element Techniques to Immiscible Flow in Porous Media." Finite Element in Water Resources (1977): 1.21-1.57.

Mualem, Y. "A New Model for Predicting the Hydraulic Conductivity of Unsaturated Porous Media." Water Res. Res. 12 (3 1976): 513-522.

Mull, R. "Migration of Oil Products in the Subsoil with regard to Groundwater Pollution by Oil." In Adv. Water Poll. Res.Proc. 5th Int. Conf. in Pergamon, Oxford, England, , 1-8, Year.

Mull, R. "Calculations and Experimental Investigations of the Migration of Hydrocarbons in Natural Soils." In IAH Int. Synp. on Groundwater Poll. by Oil Hydrocarbons in Natural Soils in Prague, , 167-181, Year.

Muller, I. "A Thermodynamic Theory of Mixtures of Fluids." Arch. Rat. Mech. Anal. 28 (1968):

Naar, J., R. J. Wygal, and J. H. Henderson. "Imbibition Relative Permeability in Unconsolidated Porous Media." Soc. Pet. Eng. J. 2 (1962): 13-17.

Nikolaevskii, V. N. "Convective Diffusion in Porous Media." PMM J. Appl. Math. Mech. 23 (6 1959): 1042-1050.

Nolen, James S. "Tests of the Stability and Time-Step Sensitivity of Semi-Implicit Reservoir Simulation Techniques." Soc. Pet. Eng. J. (1972):

Osborne, M. and J. Sykes. "Numerical Modeling of Immiscible Organic Transport at the Hyde Park Landfill." Water Res. Res. 22 (1 1986): 25-33.

Pelikan, V., M. Kucera, and M. polenka. "The Application of Soil Air Analysis in order to determine the Extent of Groundwater Contamination due to Petroleum Products." In IAH Int. Symp. on Groundwater Poll. by Oil Hydrocarbons, Proc. in Prague, , 73-81, Year.

Saraf, D. N. "Measurement of Fluid Saturations by Nuclear Magnetic Respondance and its Application to Three-Phase Relative Permeability Studies." Califonia University, Berkeley, 1966.

Schiegg, H. O. "Experimental Contribution to the Dynamic Capillary Fringe." In Symp. on Hydrodynamic Diffusion and Dispersion in Porous Media in Pavia, Italia, Year.

Sheidegger, A. E. Physics of Flow through Porous Media. Ontario: University of Toronto Press, 1974.

Shodja, H. M. "Analysis of Two-Phase Flow of Immiscible Fluids through Non-deformable Porous Media using Moving Finite Elements." Northwestern University, USA, 1990.

Slattery, J. C. "Flow of Viscoelastic Fluids through Porous Media." Am. Ins. Chem. Eng. J. 13 (1967): 1066-1071.

Sleep, Brent. E and Jonathan F. Sykes. "Compositional Simulation of Groundwater Contamination by Organic Compounds: 1. Model Development and Verification." Water Res. Res. 29 (1993a): 1697-1708.

Sleep, Brent. E and Jonathan F. Sykes. "Compositional Simulation of Groundwater Contamination by Organic Compounds: 1. Model Applications." Water Res. Res. 29 (1993b): 1709-1718.

Soo, S. L. Fluis Dynamics of Multiphase System. Waltham, Mass: Blaisdell, 1967.

Stone, H. L. "Probability Model for Estimating Three-Phase Relative Permeability." J. Pet. Tech. 22 (1 1970): 214-218.

Stone, H. L. "Estimation of Three-Phase Relative Permeability and Residual Oil Data." J. Can. Pet. Tech. 12 (4 1973): 53-61.

Su, C. and R. H. Brooks. "Water Retension Measurement for Soils." J. Irrig. Drain. Div. Am. Soc. Civ. Eng. 106 (IR2 1980): 105-112.

Van Dam, J. "The Migration of Hydrocarbons in a Water-bearing Stratum." In The Joint Problems of the Oil and Water industries in edited by ed. P. Hepple, Elsevier Publ., Co., N. Y., 55-96, Year.

Van der Waarden, M., L. A. Bridie, and W. M. Groenewoud. "Transport of MineralOil Components to Groundwater-I." Water Res. 5 (1971): 213-226.

Van Genuchten, M. Th. Calculation the Unsaturated Hydraulic Conductivity withh a New Closed-Form Analytical Model. Princeton University, 1978. Water Resources Prog. Rept. 78-WR-08.

Van Genuchten, M. Th. "A Closed Form Equation for Predicting the Hydraulic Conductivity of Unsaturated Soils." Soil Sci. Am. J. 44 (1980): 892-898.

Van Genuchten, M. Th. "A Comparison of Numerical Solutions of the One Dimensional Unsaturated-Saturated Flow and Mass Transport Equations." Adv. in Water Res. 5 (1982): 47-54.

Voss, Clifford I. A Finite Element Simulation Model for Saturated-Unsaturated Fluid-Density-Dependent Groundwater Flow with Energy Transport or Chemically Reactive Single-Species Solute Transport. U. S. Geological Survey, 1984. Water-Resources Report 84-4369.

Wang, H. F. and Mary P. Anderson. Intoduction to Groundwater Modeling: Finite Difference and Finite Element Method. USA: W. H. Freeman and Company, 1982.

Wark, K. Thermodynamics. N. Y.: McGraw-Hill Book Co, 1971.

Whitaker, S. "The Equation of Motion in Porous Media." Chem. Eng. Sci. 21 (1966): 291-300.

Whitaker, S. "Diffusion and Dispersion in Porous Media." Am. Ins. Chem. Eng. J. 13 (1967): 420-427.

Whitaker, S. "Advances in Theory of Fluid Motion in Porous Media." Ind. Eng. Chem. 61 (1969): 14-28.

Zienkiewicz, O. C. and K. Morgan. Finite Element and Approximation. USA: John Wiley & Sons, Inc., 1983.

APPENDIX A

Cellwise discretization

This numerical model employs both concepts of the finite difference method and finite element method. Time derivative terms are discretized by borrowing the concept of the finite difference technique, while convective terms adopt the finite element technique. Thus, coefficients also should be adjusted to be suitable for the analysis techniques.

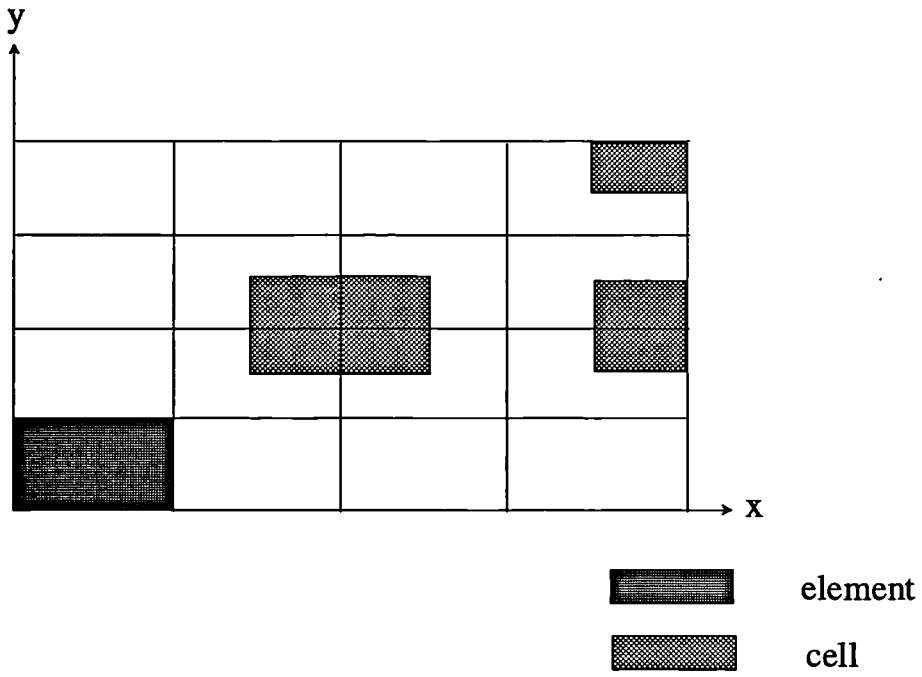


Figure A.1 cell and element

Cellwise discretization is used to present coefficients for time derivative terms. Figure A.1 shows the three types of cells, according to the locations of node: inside, side and corner. The area of cell changes according to the locations of node. The following equation describes water saturation over the entire domain, which is based on cellwise discretization.

$$S_w(x,y) = \sum_{i=1}^{N} S_{wi}(x,y) \qquad \text{(A1)}$$

where $S_{wi}(x, y)$ is constant for (x, y) coordinates within the cell and zero outside the cell. Thus, $S_{w_i}(x, y)$ is a flat topped box standing on a cell i.

APPENDIX B

The variable list for COMPO2D

nn	=	number of nodes	n3	=	nn x 3
ne	=	number of elements	nr	=	number of rows
dt	=	time step	nc	=	number of columns
nt	=	number of time steps	time	=	current time
itype	=	pollution type	kount	=	iteration number
x	=	x coordinate	y	=	y coordinate
e	=	ξ coordinate	n	=	η coordinate
ge1	=	G_1 at previous time	gen1	=	G_1 at current time
ge2	=	G_2 at previous time	gen2	=	G_2 at current time
ge3	=	G_3 at previous time	gen3	=	G_3 at current time
pwg	=	P_w at previous time	pwgn	=	P_w at current time
pog	=	P_o at previous time	pogn	=	P_o at current time
ww2	=	W_e^w at previous time	wwn2	=	W_e^w at current time
wg2	=	W_e^g at previous time	wgn2	=	W_e^g at current time
row	=	ρ^w at previous time	rowm	=	average of ρ^w over an element
roo	=	ρ^o at previous time	room	=	average of ρ^o over an element
rogn	=	ρ^g at current time	pg	=	P_g constant
alpa	=	β_s constant	bew	=	β_p^w constant
bop	=	β_p^o at previous time	bo1	=	β_w^o at previous time
beg	=	β^g at current time	g	=	**g** gravity
rx, ry	=	λ_x and λ_y	rkp	=	k weighting factor at previous time

sw = S_w at previous time

swn = S_w at current time

so = S_o at previous time

son = S_o at current time

sg = S_g at previous time

sgn = S_g at current time

twx = $\tau_{wx} = \frac{k_x k_{rw}}{\mu_w}$ at current time

twy = $\tau_{wy} = \frac{k_y k_{rw}}{\mu_w}$ at current time

tox = $\tau_{ox} = \frac{k_x k_{ro}}{\mu_o}$ at current time

toy = $\tau_{oy} = \frac{k_y k_{ro}}{\mu_o}$ at current time

dswge1= $\frac{\partial S_w}{\partial G_1}$ at current time

dswg11= $\frac{\partial^2 S_w}{\partial G_1^2}$ at current time

dswge2= $\frac{\partial S_w}{\partial G_2}$ at current time

dswg22= $\frac{\partial^2 S_w}{\partial G_2^2}$ at current time

dswg21= $\frac{\partial^2 S_w}{\partial G_2 \partial G_1}$ at current time

dswg12= $\frac{\partial^2 S_w}{\partial G_1 \partial G_2}$ at current time

dsoge1 = $\frac{\partial S_o}{\partial G_1}$ at current time

dsog11 = $\frac{\partial^2 S_o}{\partial G_1^2}$ at current time

dsoge2 = $\frac{\partial S_o}{\partial G_2}$ at current time

dsog22 = $\frac{\partial^2 S_o}{\partial G_2^2}$ at current time

dsog21= $\frac{\partial^2 S_o}{\partial G_2 \partial G_1}$ at current time

dsog12 = $\frac{\partial^2 S_o}{\partial G_1 \partial G_2}$ at current time

dsgge1 = $\frac{\partial S_g}{\partial G_1}$ at current time

dsgg11 = $\frac{\partial^2 S_g}{\partial G_1^2}$ at current time

dsgge2 = $\frac{\partial S_g}{\partial G_2}$ at current time

dsgg22 = $\frac{\partial^2 S_g}{\partial G_2^2}$ at current time

dsgg21= $\frac{\partial^2 S_g}{\partial G_2 \partial G_1}$ at current time

dsgg12 = $\frac{\partial^2 S_g}{\partial G_1 \partial G_2}$ at current time

twxge1 = $\frac{\partial \tau_{wx}}{\partial G_1}$ at current time

twyge1 = $\frac{\partial \tau_{wy}}{\partial G_1}$ at current time

twxge2 = $\frac{\partial \tau_{wx}}{\partial G_2}$ at current time

twyge2 = $\frac{\partial \tau_{wy}}{\partial G_2}$ at current time

toxge1 = $\frac{\partial \tau_{ox}}{\partial G_1}$ at current time

toyge1 = $\frac{\partial \tau_{oy}}{\partial G_1}$ at current time

toxge2 = $\frac{\partial \tau_{ox}}{\partial G_2}$ at current time

toyge2 = $\frac{\partial \tau_{oy}}{\partial G_2}$ at current time

dwwge1= $\frac{\partial W_e^w}{\partial G_1}$ at current time

dwwge2= $\frac{\partial W_e^w}{\partial G_2}$ at current time

dwwge3= $\frac{\partial W_e^w}{\partial G_3}$ at current time

dwgge1= $\frac{\partial W_e^g}{\partial G_1}$ at current time

dwgge2= $\frac{\partial W_e^g}{\partial G_2}$ at current time

dwgge3= $\frac{\partial W_e^g}{\partial G_3}$ at current time

dbgdwo= $\frac{\partial \beta^g}{\partial G_3}$ at current time

drgge3 = $\frac{\partial \rho^g}{\partial G_3}$ at current time

dwxx = element of D_{xy}^w at previous time

dwxy = element of D_{xy}^w at previous time

dwyx = element of D_{xy}^w at previous time

dwyy = element of D_{xy}^w at previous time

doxx = element of D_{xy}^o at previous time

doxy = element of D_{xy}^o at previous time

doyx = element of D_{xy}^o at previous time

doyy = element of D_{xy}^o at previous time

dg = diffusion coefficient of gas, constant

APPENDIX C

Source Code of COMPO2D

```
      program compo2d
      common /moon/ge1(200),ge2(200),ge3(200),pwg(200),pog(200),
     +       wo2(200),ww2(200),wg2(200),bop(200),bo1(200),sw(200),
     +       rowm(200),room(200),rkp(200),so(200),sg(200)
      common /sun/ gen1(200),gen2(200),gen3(200),pwgn(200),pogn(200),
     +       won2(200),wwn2(200),wgn2(200),swn(200),son(200),sgn(200),
     +       twx(200),twy(200),tox(200),toy(200),beg(200),dbgdwo(200),
     +       twxge1(200),twyge1(200),twxge2(200),twyge2(200),
     +       toxge1(200),toyge1(200),toxge2(200),toyge2(200),
     +       dwwge1(200),dwwge2(200),dwwge3(200),
     +       dwgge1(200),dwgge2(200),dwgge3(200),
     +       rogn( 200),drgge3(200)
      common /dove/dwxx(200),dwxy(200),dwyx(200),dwyy(200),
     +       doxx(200),doxy(200),doyx(200),doyy(200),
     +       dmw,atw,alw,dmo,ato,alo,dmg,dg(200)
      common /lake/gintx1(4,4,2),ginty1(4,4,3),gintx2(4,4,4,3),
     +       ginty2(4,4,4,3),gintx3(4,4,4,4,3),ginty3(4,4,4,4,3),
     +       gintx4(4,4,4,4,4,2),ginty4(4,4,4,4,4,2)
      common /may/ dswge1(200),dswg11(200),dswg21(200),
     +          dswge2(200),dswg22(200),dswg12(200),
     +          dsoge1(200),dsog11(200),dsog21(200),
     +          dsoge2(200),dsog22(200),dsog12(200),
     +          dsgge1(200),dsgg11(200),dsgg21(200),
     +          dsgge2(200),dsgg22(200),dsgg12(200)
      common /door/asw,rnsw,swr,sws,akw,rnkw,aso,rnso,sor,sos,ako,rnko,
     +          ako1,rnko1,ako2,rnko2,swir,som
      common /hill/ru,t,z
      common /june/rkx,rky,rx,ry,g
      common /july/a(600,600),b(600),u(600),rmax
      common /wind/nn,pg,rm1,rm2,rkpg,rkpw,rmw,rma,uo(200),row(200),
     +          roo(200)
      common /tree/pwb,pob1,pob2,bew,be1,be2,rwb,rob1,rob2,
     +          rmuw,rmu1,rmu2,theta,eps,itype
      common /bird/x(200),y(200),nne(200,4),e(4),n(4)
      common /phic/ dpc(20),drw(20),dro(20),dro1(20),dro2(20),
     +          dsw(20),dso(20)
      dimension tge1(200),tge2(200),bc(200),qwbo(200),qobo(200)
c
c  DATA INPUT ACCORDING TO FLOW PATTERNS
c
      open(50, file='da21')
      open(51, file='da22')
      read(50,*) itype
      if(itype.eq.4.or.itype.eq.5) then
       go to 901
      else
       read(50,*) rmw,rmuw,bew,rwb,pwb
      endif
```

```
 901 if(itype.eq.1) go to 902
     read(50,*) rm1,rmu1,be1,rob1,pob1
     if(itype.eq.2.or.itype.eq.4) go to 902
     if(itype.eq.6) go to 902
     read(50,*) rm2,rmu2,be2,rob2,pob2
 902 read(50,*) eps,alpa,rkx,rky,rx,ry,g,pg,swir,som
     if(itype.eq.5.or.itype.eq.7) then
      read(50,*) rma,z,t,ru
      read(50,*) dmw,dmo,dmg,atw,alw,ato,alo,rkpw,rkpg
     endif
     if(itype.eq.3) then
      read(50,*) dmw,dmo,dmg,atw,alw,ato,alo,rkpw,rkpg
     endif
     if(itype.eq.1) read(50,*) dmw,atw,alw
c
c  isw,iso,irw,iro,iro1, AND iro2 DEFINE DATA TYPES WHICH RELATE
c  SATURATIONS AND RELATIVE PERMEABILITIES TO CAPILLARY PRESSURES
c  THREE TYPES ARE DEFINED IN THIS CODE.
c
c  1: Van Genuchten
c  2: Experimental Data
c  3: Others
c
     if(itype.eq.1) then
     read(50,*) isw,irw
     if(isw.eq.1) read(50,*) asw,rnsw,swr,sws
     if(isw.eq.2) then
      read(50,*) inum
      read(50,*) (dpc(i),i=1,inum)
      read(50,*) (dsw(i),i=1,inum)
     endif
     if(irw.eq.1) read(50,*) akw,rnkw
     if(irw.eq.2) then
      read(50,*) inum
      read(50,*) (dpc(i),i=1,inum)
      read(50,*) (drw(i),i=1,inum)
     endif
     endif
c
     if(itype.eq.2.or.itype.eq.3) then
     read(50,*) isw,irw,iro
     if(isw.eq.1) read(50,*) asw,rnsw,swr,sws
     if(isw.eq.2) then
      read(50,*) inum
      read(50,*) (dpc(i),i=1,inum)
      read(50,*) (dsw(i),i=1,inum)
     endif
     if(irw.eq.1) read(50,*) akw,rnkw
     if(irw.eq.2) then
      read(50,*) inum
      read(50,*) (dpc(i),i=1,inum)
      read(50,*) (drw(i),i=1,inum)
     endif
     if(iro.eq.1) read(50,*) ako,rnko
     if(iro.eq.2) then
      read(50,*) inum
      read(50,*) (dpc(i),i=1,inum)
      read(50,*) (dro(i),i=1,inum)
```

```
      endif
      endif
c
      if(itype.eq.4.or.itype.eq.5) then
      read(50,*) iso,iro
      if(iso.eq.1) read(50,*) aso,rnso,sor,sos
      if(iso.eq.2) then
       read(50,*) inum
       read(50,*) (dpc(i),i=1,inum)
       read(50,*) (dso(i),i=1,inum)
      endif
      if(iro.eq.1) read(50,*) ako,rnko
      if(iro.eq.2) then
       read(50,*) inum
       read(50,*) (dpc(i),i=1,inum)
       read(50,*) (dro(i),i=1,inum)
      endif
      endif
c
      if(itype.eq.6.or.itype.eq.7) then
      read(50,*) isw,iso,irw,iro1,iro2
      if(isw.eq.1) read(50,*) asw,rnsw,swr,sws
      if(isw.eq.2) then
       read(50,*) inum
       read(50,*) (dpc(i),i=1,inum)
       read(50,*) (dsw(i),i=1,inum)
      endif
      if(iso.eq.1) read(50,*) aso,rnso,sor,sos
      if(iso.eq.2) then
       read(50,*) inum
       read(50,*) (dpc(i),i=1,inum)
       read(50,*) (dso(i),i=1,inum)
      endif
      if(irw.eq.1) read(50,*) akw,rnkw
      if(irw.eq.2) then
       read(50,*) inum
       read(50,*) (dpc(i),i=1,inum)
       read(50,*) (drw(i),i=1,inum)
      endif
      if(iro1.eq.1) read(50,*) ako1,rnko1
      if(iro1.eq.2) then
       read(50,*) inum
       read(50,*) (dpc(i),i=1,inum)
       read(50,*) (dro(i),i=1,inum)
      endif
      if(iro2.eq.1) read(50,*) ako2,rnko2
      if(iro2.eq.2) then
       read(50,*) inum
       read(50,*) (dpc(i),i=1,inum)
       read(50,*) (dro1(i),i=1,inum)
      endif
      endif
      read(50,*) ne,nn,nr,nc,nt,dt
      data e(1),e(2),e(3),e(4)/-1,1,1,-1/
      data n(1),n(2),n(3),n(4)/-1,-1,1,1/
      rx=3.14159/180.*rx
      ry=3.14159/180.*ry
c
```

```
c  NODE AND ELEMENT NUMBERS ARE DECIDED AND EACH NODE IS DECIDED IN
c  THE LOCAL COODINATE SYSTEM FROM THE SUBROUTINE coor.
c
      call coor(ne,nn)
c
      do 13 i=1,nn
      ge1(i)=0
      ge2(i)=0
      ge3(i)=0
      pwg(i)=0
      pog(i)=0
      wo2(i)=0
      wg2(i)=0
      ww2(i)=0
      sw(i)=0
      so(i)=0
      swn(i)=0
      son(i)=0
      sgn(i)=0
      gen1(i)=0
      gen2(i)=0
      gen3(i)=0
      pwgn(i)=0
      pogn(i)=0
      won2(i)=0
      wgn2(i)=0
      wwn2(i)=0
      uo(i)=0
      bop(i)=0
      bo1(i)=0
      row(i)=0
      rowm(i)=0
      roo(i)=0
      room(i)=0
      twx(i)=0
      twy(i)=0
      tox(i)=0
      toy(i)=0
      beg(i)=0
      rogn(i)=0
      rkp(i)=0
      dswge1(i)=0
      dswg11(i)=0
      dswg21(i)=0
      dswge2(i)=0
      dswg22(i)=0
      dswg12(i)=0
      dsoge1(i)=0
      dsog11(i)=0
      dsog21(i)=0
      dsoge2(i)=0
      dsog22(i)=0
      dsog12(i)=0
      dsgge1(i)=0
      dsgg11(i)=0
      dsgg21(i)=0
      dsgge2(i)=0
      dsgg22(i)=0
```

```
      dsgg12(i)=0
      twxge1(i)=0
      twyge1(i)=0
      twxge2(i)=0
      twyge2(i)=0
      toxge1(i)=0
      toyge1(i)=0
      toxge2(i)=0
      toyge2(i)=0
      dwwge1(i)=0
      dwwge2(i)=0
      dwwge3(i)=0
      dwgge1(i)=0
      dwgge2(i)=0
      dwgge3(i)=0
      drgge3(i)=0
      dwxx(i)=0
      dwxy(i)=0
      dwyx(i)=0
      dwyy(i)=0
      doxx(i)=0
      doxy(i)=0
      doyx(i)=0
      doyy(i)=0
      dg(i)=0
      bc(i)=0
      qwbo(i)=0
      qobo(i)=0
   13 continue
c
c  SUBROUTINE inicon AND bouncon ARE CONCERNED WITH INITIAL CONDITION
c  AND BOUNDARY CONDITION
c
      call inicon(ge1,ge2,ge3,gen1,gen2,gen3,nn)
      call bouncon(ge1,ge2,ge3,gen1,gen2,gen3,nr,nc,nn)
c
      n3=nn*3
      time=0
      nk=1
c
c  DO LOOP 1000 IS FOR CALCULATING THE ELEMENTS OF THE RESULTANT MATRIX
c  EQUATION,
c
      do 1000 idt=1,nt
      if(idt.eq.11) dt=dt*10
      ikount=0
      kount=0
      time=time+dt
 1001 do 17 i=1,n3
      b(i)=0
      u(i)=0
      do 17 j=1,n3
      a(i,j)=0
   17 continue
      if(ikount.eq.1) go to 1056
      tune=0.5
      kount=kount+1
 1056 if(kount.gt.30) then
```

```
      stop
      endif
      do 133 i=1,nn
c
c  WATER PRESSURE, ORGANIC PRESSURE, DERIVATIVES OF WATER AND ORGANIC
c  PRESSURES WITH RESPECT TO THE PRIMARY VARIABLES ARE REPRESENTED
c  ACCORDIN TO FLOW PATTERNS.
c
      if(itype.eq.1) then
       pwg(i)=-ge1(i)+pg
       pwgn(i)=-gen1(i)+pg
       pog(i)=0
       pogn(i)=0
       pwgge1=-1
       pwgge2=0
       pogge1=0
       pogge2=0
       go to 133
      endif
      if(itype.eq.2.or.itype.eq.3) then
       pwg(i)=ge2(i)
       pwgn(i)=gen2(i)
       pog(i)=ge1(i)+ge2(i)
       pogn(i)=gen1(i)+gen2(i)
       pwgge1=0
       pwgge2=1
       pogge1=1
       pogge2=1
       go to 133
      endif
      if(itype.eq.4.or.itype.eq.5) then
       pwg(i)=0
       pwgn(i)=0
       pog(i)=-ge2(i)+pg
       pogn(i)=-gen2(i)+pg
       pwgge1=0
       pwgge2=0
       pogge1=0
       pogge2=-1
       go to 133
      endif
      if(itype.eq.6.or.itype.eq.7) then
       pwg(i)=-ge1(i)-ge2(i)+pg
       pwgn(i)=-gen1(i)-gen2(i)+pg
       pog(i)=-ge2(i)+pg
       pogn(i)=-gen2(i)+pg
       wo2(i)=1-ge3(i)
       won2(i)=1-gen3(i)
       pwgge1=-1
       pwgge2=-1
       pogge1=0
       pogge2=-1
      endif
  133 continue
      if(kount.gt.1) go to 221
c
c  SUBROUTINE ocof IS TO CALCULATE COEFFICIENTS THAT IS RELATED TO
c  THE PREVIOUS TIME.
```

```
c
      call ocof(nr,nc)
c
c  SUBROUTINE coef IS TO CALCULATE COEFFICIENTS THAT IS RELATED TO
c  THE CURRENT TIME.
c
  221 call coef
c
      kk1=0
      do 108 ie=1,ne
      n1=nne(ie,1)
      n2=nne(ie,2)
      nn3=nne(ie,3)
      n4=nne(ie,4)
      do 110 i4=1,4
      if(i4.eq.1) n0=n1
      if(i4.eq.2) n0=n2
      if(i4.eq.3) n0=nn3
      if(i4.eq.4) n0=n4
      if(n0.eq.1.or.n0.eq.21) then
       a(3*n0-2,3*n0-2)=1
       a(3*n0-1,3*n0-1)=1
       a(3*n0,3*n0)=1
       go to 110
      endif
      do 19 i19=1,nk
       if(n0.eq.bc(i19)) then
       ih=bc(i19)
       a(ih*3-2,ih*3-2)=1
       a(ih*3-1,ih*3-1)=1
       a(ih*3,ih*3)=1
       go to 110
       endif
   19 continue
      if(n0.eq.20.or.n0.eq.40) then
       a(3*n0-2,3*n0-2)=1
       a(3*n0-1,3*n0-1)=1
       a(3*n0,3*n0)=1
       go to 110
      endif
       do 92 ih=2,nr-1
       if(n0.eq.ih) then
       rbo=abs(y(n1)-y(n4))/2
       go to 793
       endif
       jh=nr*(nc-1)+ih
       if(n0.eq.jh) then
       rbo=abs(y(n1)-y(n4))/2
       go to 793
       endif
   92  continue
       do 93 ih=1,nc
       jh=nr*(ih-1)+1
       if(n0.eq.jh) then
       rbo=abs(x(n1)-x(n2))/2
       go to 793
       endif
       lh=nr*ih
```

```
      if(n0.eq.lh) then
      rbo=abs(x(n1)-x(n2))/2
      go to 793
      endif
   93 continue
  793 pav=rkp(n0)*pog(n0)+(1-rkp(n0))*pwg(n0)
      pavn=rkp(n0)*pogn(n0)+(1-rkp(n0))*pwgn(n0)
c
c  DERIVATIVES OF THE AVERAGE FLUID PRESSURE ARE PRESENTED ACCORDING TO
c  FLOW PATTERNS.
c
      if(itype.eq.1) then
       pavge1=-1
       pavge2=0
       go to 1333
      endif
      if(itype.eq.2.or.itype.eq.3) then
       pavge1=rkp(n0)
       pavge2=1
       go to 1333
      endif
      if(itype.eq.4.or.itype.eq.5) then
       pavge1=0
       pavge2=-1
       go to 1333
      endif
      if(itype.eq.6.or.itype.eq.7) then
       pavge1=rkp(n0)-1
       pavge2=-1
      endif
 1333 continue
      rowm(n0)=(row(n1)+row(n2)+row(nn3)+row(n4))/4
      room(n0)=(roo(n1)+roo(n2)+roo(nn3)+roo(n4))/4
      qvol=vol(n1,n2,nn3,n4,i4)
      kk1=kk1+1
c
c  SUBROUTINE gint IS FOR NUMERICAL INTEGRATIONS BY GAUSSIAN QUAGRATURE.
c
      call gint(n1,n2,nn3,n4,i4)
c
      if(swn(n0).lt.swir) go to 453
      a(3*n0-2,3*n0-2)=a(3*n0-2,3*n0-2)
     +       +(eps*(dswg11(n0)*(gen1(n0)-ge1(n0))/dt
     +       +dswge1(n0)/dt+dswg12(n0)*(gen2(n0)-ge2(n0))/dt)
     +       +sw(n0)*bew*eps/dt*pwgge1
     +       +sw(n0)*alpa*(1-eps)*pavge1/dt)*qvol
      a(3*n0-2,3*n0-1)=a(3*n0-2,3*n0-1)
     +       +(eps*(dswg21(n0)*(gen1(n0)-ge1(n0))/dt
     +       +dswg22(n0)*(gen2(n0)-ge2(n0))/dt+dswge2(n0)/dt)
     +       +sw(n0)*alpa*(1-eps)*pavge2/dt+sw(n0)*bew*eps*pwgge2/dt)
     +       *qvol
      b(3*n0-2)=b(3*n0-2)
     +       -(eps*(dswge1(n0)*(gen1(n0)-ge1(n0))/dt+dswge2(n0)
     +       *(gen2(n0)-ge2(n0))/dt)+sw(n0)*bew*eps
     +       *(pwgn(n0)-pwg(n0))/dt
     +       +sw(n0)*alpa*(1-eps)*(pavn-pav)/dt)*qvol
     +       -qwbo(i)*rbo
  453 if(itype.eq.1) go to 1311
```

```
     a(3*n0-1,3*n0-2)=a(3*n0-1,3*n0-2)
    +        +(ge3(n0)*so(n0)*eps*bop(n0)/dt*pogge1
    +        +eps*gen3(n0)*(dsog11(n0)*(gen1(n0)-ge1(n0))/dt
    +        +dsoge1(n0)/dt+dsog12(n0)*(gen2(n0)-ge2(n0))/dt)
    +        +so(n0)*ge3(n0)*alpa*(1-eps)*pavge1/dt)*qvol
     a(3*n0-1,3*n0-1)=a(3*n0-1,3*n0-1)
    +        +(ge3(n0)*so(n0)*eps*bop(n0)/dt*pogge2
    +        +eps*gen3(n0)*(dsog21(n0)*(gen1(n0)-ge1(n0))/dt
    +        +dsog22(n0)*(gen2(n0)-ge2(n0))/dt+dsoge2(n0)/dt)
    +        +so(n0)*ge3(n0)*alpa*(1-eps)/dt*pavge2)*qvol
     a(3*n0-1,3*n0)=a(3*n0-1,3*n0)
    +        +(ge3(n0)*so(n0)*eps*bo1(n0)/dt+eps*so(n0)/dt
    +        +eps*(dsoge1(n0)*(gen1(n0)-ge1(n0))/dt+dsoge2(n0)
    +        *(gen2(n0)-ge2(n0))/dt))*qvol
     b(3*n0-1)=b(3*n0-1)
    +        -(ge3(n0)*so(n0)*eps*(bop(n0)*(pogn(n0)-pog(n0))/dt
    +        +bo1(n0)*(gen3(n0)-ge3(n0))/dt)+eps*so(n0)
    +        *(gen3(n0)-ge3(n0))/dt+eps*gen3(n0)
    +        *(dsoge1(n0)*(gen1(n0)-ge1(n0))/dt+dsoge2(n0)
    +        *(gen2(n0)-ge2(n0))/dt)+so(n0)*ge3(n0)*alpa*(1-eps)
    +        *(pavn-pav)/dt)*qvol
    +        -qobo(i)*rbo
     if(swn(n0).lt.swir) go to 454
c
 1311 a(3*n0,3*n0-2)=a(3*n0,3*n0-2)
    +    +(row(n0)*eps*sw(n0)/dt*dwwge1(n0)
    +    +row(n0)*eps*dwwge1(n0)*(dswge1(n0)*(gen1(n0)-ge1(n0))/dt
    +    +dswge2(n0)*(gen2(n0)-ge2(n0))/dt)+row(n0)*eps*wwn2(n0)
    +    *(dswg11(n0)*(gen1(n0)-ge1(n0))/dt+dswge1(n0)/dt+dswg12(n0)
    +    *(gen2(n0)-ge2(n0))/dt)+row(n0)*wwn2(n0)*sw(n0)
    +    *bew*eps/dt*pwgge1)*qvol
    +    +(alpa*(1-eps)*(row(n0)*sw(n0)*dwwge1(n0)+row(n0)*dswge1(n0)
    +    *wwn2(n0))*(pavn-pav)/dt
    +    +alpa*(1-eps)*(row(n0)*sw(n0)*wwn2(n0))*pavge1/dt)*qvol
     a(3*n0,3*n0-1)=a(3*n0,3*n0-1)
    +    +(row(n0)*eps*sw(n0)/dt*dwwge2(n0)
    +    +row(n0)*eps*dwwge2(n0)*(dswge1(n0)*(gen1(n0)-ge1(n0))/dt
    +    +dswge2(n0)*(gen2(n0)-ge2(n0))/dt)+row(n0)*eps*wwn2(n0)
    +    *(dswg21(n0)*(gen1(n0)-ge1(n0))/dt+dswg22(n0)
    +    *(gen2(n0)-ge2(n0))/dt+dswge2(n0)/dt)+row(n0)*wwn2(n0)
    +    *sw(n0)*bew*eps/dt*pwgge2)*qvol
    +    +(alpa*(1-eps)*(row(n0)*sw(n0)*dwwge2(n0)+row(n0)*dswge2(n0)
    +    *wwn2(n0))*(pavn-pav)/dt
    +    +alpa*(1-eps)*(row(n0)*sw(n0)*wwn2(n0))*pavge2/dt)*qvol
     a(3*n0,3*n0)=a(3*n0,3*n0)
    +    +(row(n0)*eps*sw(n0)/dt*dwwge3(n0)+row(n0)*eps
    +    *dwwge3(n0)*(dswge1(n0)*(gen1(n0)-ge1(n0))/dt+dswge2(n0)
    +    *(gen2(n0)-ge2(n0))/dt)
    +    +alpa*(1-eps)*row(n0)*sw(n0)*dwwge3(n0)*(pavn-pav)/dt)*qvol
     b(3*n0)=b(3*n0)
    +    -(row(n0)*eps*sw(n0)*(wwn2(n0)-ww2(n0))/dt+row(n0)*eps
    +    *wwn2(n0)*(dswge1(n0)*(gen1(n0)-ge1(n0))/dt+dswge2(n0)
    +    *(gen2(n0)-ge2(n0))/dt)+row(n0)*ww2(n0)*sw(n0)*bew*eps
    +    *(pwgn(n0)-pwg(n0))/dt
    +    +alpa*(1-eps)*row(n0)*sw(n0)*wwn2(n0)*(pavn-pav)/dt)*qvol
c
  454 if(itype.eq.1) go to 1211
c
```

```
      a(3*n0,3*n0-2)=a(3*n0,3*n0-2)
     +    +(roo(n0)*eps*won2(n0)*(dsog11(n0)
     +    *(gen1(n0)-ge1(n0))/dt+dsoge1(n0)/dt+dsog12(n0)
     +    *(gen2(n0)-ge2(n0))/dt)
     +    +roo(n0)*wo2(n0)*so(n0)*eps*bop(n0)/dt*pogge1+rogn(n0)*eps
     +    *sg(n0)*(1+wg2(n0)*beg(n0))/dt*dwgge1(n0)
     +    +rogn(n0)*eps*dwgge1(n0)*(dsgge1(n0)*(gen1(n0)-ge1(n0))/dt
     +    +dsgge2(n0)*(gen2(n0)-ge2(n0))/dt)
     +    +rogn(n0)*eps*wgn2(n0)*(dsgg11(n0)*(gen1(n0)-ge1(n0))/dt
     +    +dsgge1(n0)/dt+dsgg12(n0)*(gen2(n0)-ge2(n0))/dt))*qvol
      a(3*n0,3*n0-2)=a(3*n0,3*n0-2)
     +    +(alpa*(1-eps)*(sg(n0)*rogn(n0)*dwgge1(n0)+roo(n0)*dsoge1(n0)
     +    *won2(n0)+rogn(n0)*dsgge1(n0)*wgn2(n0))*(pavn-pav)/dt
     +    +alpa*(1-eps)*(roo(n0)*so(n0)*won2(n0)
     +    +sg(n0)*rogn(n0)*wgn2(n0))*pavge1/dt)*qvol
      a(3*n0,3*n0-1)=a(3*n0,3*n0-1)
     +    +(roo(n0)*eps*won2(n0)*(dsog21(n0)*(gen1(n0)-ge1(n0))/dt
     +    +dsog22(n0)*(gen2(n0)-ge2(n0))/dt+dsoge2(n0)/dt)
     +    +roo(n0)*wo2(n0)*so(n0)*eps*bop(n0)/dt*pogge2+rogn(n0)*eps
     +    *sg(n0)*(1+wg2(n0)*beg(n0))/dt*dwgge2(n0)
     +    +rogn(n0)*eps*dwgge2(n0)*(dsgge1(n0)*(gen1(n0)-ge1(n0))/dt
     +    +dsgge2(n0)*(gen2(n0)-ge2(n0))/dt)
     +    +rogn(n0)*eps*wgn2(n0)*(dsgg21(n0)*(gen1(n0)-ge1(n0))/dt
     +    +dsgg22(n0)*(gen2(n0)-ge2(n0))/dt+dsgge2(n0)/dt))*qvol
      a(3*n0,3*n0-1)=a(3*n0,3*n0-1)
     +    +(alpa*(1-eps)*(sg(n0)*rogn(n0)*dwgge2(n0)+roo(n0)*dsoge2(n0)
     +    *won2(n0)+rogn(n0)*dsgge2(n0)*wgn2(n0))*(pavn-pav)/dt
     +    +alpa*(1-eps)*(roo(n0)*so(n0)*won2(n0)
     +    +sg(n0)*rogn(n0)*wgn2(n0))*pavge2/dt)*qvol
      a(3*n0,3*n0)=a(3*n0,3*n0)
     +    +(roo(n0)*eps*so(n0)*(-1)/dt+roo(n0)*eps*(-1)*(dsoge1(n0)
     +    *(gen1(n0)-ge1(n0))/dt+dsoge2(n0)*(gen2(n0)-ge2(n0))/dt)
     +    +roo(n0)*wo2(n0)*so(n0)*eps*bo1(n0)/dt+drgge3(n0)*eps
     +    *sg(n0)*(1+wg2(n0)*beg(n0))*(wgn2(n0)-wg2(n0))/dt
     +    +rogn(n0)*eps*sg(n0)*(1+wg2(n0)*beg(n0))/dt*dwgge3(n0)
     +    +eps*(drgge3(n0)*wgn2(n0)+rogn(n0)*dwgge3(n0))*(dsgge1(n0)
     +    *(gen1(n0)-ge1(n0))/dt+dsgge2(n0)*(gen2(n0)-ge2(n0))/dt)
     +    +alpa*(1-eps)*(roo(n0)*so(n0)*(-1)
     +    +sg(n0)*(drgge3(n0)*wgn2(n0)
     +    +rogn(n0)*dwgge3(n0)))*(pavn-pav)/dt)*qvol
      b(3*n0)=b(3*n0)
     +    -(roo(n0)*eps*so(n0)*(won2(n0)-wo2(n0))/dt+roo(n0)*eps
     +    *won2(n0)*(dsoge1(n0)*(gen1(n0)-ge1(n0))/dt+dsoge2(n0)
     +    *(gen2(n0)-ge2(n0))/dt)
     +    +roo(n0)*won2(n0)*so(n0)*eps*(bop(n0)*(pogn(n0)-pog(n0))
     +    /dt+bo1(n0)*(gen3(n0)-ge3(n0))/dt)+rogn(n0)*eps*sg(n0)
     +    *(1+wg2(n0)*beg(n0))*(wgn2(n0)-wg2(n0))/dt
     +    +rogn(n0)*eps*wgn2(n0)*(dsgge1(n0)*(gen1(n0)-ge1(n0))/dt
     +    +dsgge2(n0)*(gen2(n0)-ge2(n0))/dt)
     +    +alpa*(1-eps)*(roo(n0)*so(n0)*won2(n0)
     +    +sg(n0)*rogn(n0)*wgn2(n0))
     +    *(pavn-pav)/dt)*qvol
c
 1211 do 111 ii=1,4
      if(ii.eq.1) i=n1
      if(ii.eq.2) i=n2
      if(ii.eq.3) i=nn3
      if(ii.eq.4) i=n4
```

```
      do 112 jj=1,4
      if(jj.eq.1) j=n1
      if(jj.eq.2) j=n2
      if(jj.eq.3) j=nn3
      if(jj.eq.4) j=n4
      if(swn(n0).lt.swir) go to 455
c
      a(3*n0-2,3*i-2)=a(3*n0-2,3*i-2)
     +      +pwgn(j)*twxge1(i)*gintx1(ii,jj,1)
     +      +pwgge1*twx(j)*gintx1(jj,ii,1)
     +      -twxge1(i)*row(j)*g*cos(rx)*gintx1(ii,jj,2)
     +      +pwgn(j)*twyge1(i)*ginty1(ii,jj,1)
     +      +pwgge1*twy(j)*ginty1(jj,ii,1)
     +      -twyge1(i)*row(j)*g*cos(ry)*ginty1(ii,jj,2)
      a(3*n0-2,3*i-1)=a(3*n0-2,3*i-1)
     +      +twx(j)*gintx1(jj,ii,1)*pwgge2
     +      +pwgn(j)*twxge2(i)*gintx1(ii,jj,1)
     +      -twxge2(i)*row(j)*g*cos(rx)*gintx1(ii,jj,2)
     +      +twy(j)*ginty1(jj,ii,1)*pwgge2
     +      +pwgn(j)*twyge2(i)*ginty1(ii,jj,1)
     +      -twyge2(i)*row(j)*g*cos(ry)*ginty1(ii,jj,2)
      b(3*n0-2)=b(3*n0-2)
     +      -(pwgn(j)*twx(i)*gintx1(ii,jj,1)
     +      -twx(i)*row(j)*g*cos(rx)*gintx1(ii,jj,2))
     +      -(pwgn(j)*twy(i)*ginty1(ii,jj,1)
     +      -twy(i)*row(j)*g*cos(ry)*ginty1(ii,jj,2))
  455 do 113 kk=1,4
      if(kk.eq.1) k=n1
      if(kk.eq.2) k=n2
      if(kk.eq.3) k=nn3
      if(kk.eq.4) k=n4
      if(itype.eq.1) go to 1312
      a(3*n0-1,3*i-2)=a(3*n0-1,3*i-2)
     +    +gen3(k)*toxge1(i)*(pogn(j)
     +    *gintx2(ii,jj,kk,1)
     +    -roo(j)*g*cos(rx)*gintx2(ii,jj,kk,2))
     +    +gen3(k)*tox(j)*gintx2(jj,ii,kk,1)*pogge1
     +    +gen3(k)*toyge1(i)*(pogn(j)
     +    *ginty2(ii,jj,kk,1)
     +    -roo(j)*g*cos(ry)*ginty2(ii,jj,kk,2))
     +    +gen3(k)*toy(j)*ginty2(jj,ii,kk,1)*pogge1
     +    +eps*dsoge1(i)*(doxx(k)*gen3(j)*gintx2(ii,jj,kk,1)
     +    +doxy(k)*gen3(j)*gintx2(ii,jj,kk,3))
     +    +eps*dsoge1(i)*(doyx(k)*gen3(j)*ginty2(ii,jj,kk,1)
     +    +doyy(k)*gen3(j)*ginty2(ii,jj,kk,3))
      a(3*n0-1,3*i-1)=a(3*n0-1,3*i-1)
     +    +gen3(k)*toxge2(i)*(pogn(j)
     +    *gintx2(ii,jj,kk,1)
     +    -roo(j)*g*cos(rx)*gintx2(ii,jj,kk,2))
     +    +gen3(k)*tox(j)*gintx2(jj,ii,kk,1)*pogge2
     +    +gen3(k)*toyge2(i)*(pogn(j)
     +    *ginty2(ii,jj,kk,1)
     +    -roo(j)*g*cos(ry)*ginty2(ii,jj,kk,2))
     +    +gen3(k)*toy(j)*ginty2(jj,ii,kk,1)*pogge2
     +    +eps*dsoge2(i)*(doxx(k)*gen3(j)*gintx2(ii,jj,kk,1)
     +    +doxy(k)*gen3(j)*gintx2(ii,jj,kk,3))
     +    +eps*dsoge2(i)*(doyx(k)*gen2(j)*ginty2(ii,jj,kk,1)
     +    +doyy(k)*gen3(j)*ginty2(ii,jj,kk,3))
```

```
      a(3*n0-1,3*i)=a(3*n0-1,3*i)
     +     +tox(k)*(pogn(j)*gintx2(ii,jj,kk,1)
     +     -roo(j)*g*cos(rx)*gintx2(ii,jj,kk,2))
     +     +toy(k)*(pogn(j)*ginty2(ii,jj,kk,1)
     +     -roo(j)*g*cos(ry)*ginty2(ii,jj,kk,2))
     +     +eps*son(j)*(doxx(k)*gintx2(jj,ii,kk,1)
     +     +doxy(k)*gintx2(jj,ii,kk,3))
     +     +eps*son(j)*(doyx(k)*ginty2(jj,ii,kk,1)
     +     +doyy(k)*ginty2(jj,ii,kk,3))
      b(3*n0-1)=b(3*n0-1)
     +     -gen3(k)*tox(i)*(pogn(j)*gintx2(ii,jj,kk,1)
     +     -roo(j)*g*cos(rx)*gintx2(ii,jj,kk,2))
     +     -gen3(k)*toy(i)*(pogn(j)*ginty2(ii,jj,kk,1)
     +     -roo(j)*g*cos(ry)*ginty2(ii,jj,kk,2))
     +     -eps*son(i)*(doxx(k)*gen3(j)*gintx2(ii,jj,kk,1)
     +     +doxy(k)*gen3(j)*gintx2(ii,jj,kk,3))
     +     -eps*son(i)*(doyx(k)*gen3(j)*ginty2(ii,jj,kk,1)
     +     +doyy(k)*gen3(j)*ginty2(ii,jj,kk,3))
c
      if(swn(n0).lt.swir) go to 456
c
 1312 a(3*n0,3*i-2)=a(3*n0,3*i-2)
     +      +rowm(n0)*dwwge1(i)*twx(k)*(pwgn(j)
     +      *gintx2(ii,jj,kk,1)
     +      -row(j)*g*cos(rx)*gintx2(ii,jj,kk,2))
     +      +rowm(n0)*wwn2(k)*twxge1(i)*(pwgn(j)
     +      *gintx2(kk,jj,ii,1)
     +      -row(j)*g*cos(rx)*gintx2(kk,jj,ii,2))
     +      +rowm(n0)*wwn2(j)*twx(k)*gintx2(jj,ii,kk,1)*pwgge1
     +      +rowm(n0)*dwwge1(i)*twy(k)*(pwgn(j)
     +      *ginty2(ii,jj,kk,1)
     +      -row(j)*g*cos(ry)*ginty2(ii,jj,kk,2))
     +      +rowm(n0)*wwn2(k)*twyge1(i)*(pwgn(j)
     +      *ginty2(kk,jj,ii,1)
     +      -row(j)*g*cos(ry)*ginty2(kk,jj,ii,2))
     +      +rowm(n0)*wwn2(j)*twy(k)*ginty2(jj,ii,kk,1)*pwgge1
      a(3*n0,3*i-2)=a(3*n0,3*i-2)
     +      +rowm(n0)*eps*dswge1(i)*(dwxx(k)*wwn2(j)
     +      *gintx2(ii,jj,kk,1)
     +      +dwxy(k)*wwn2(j)*gintx2(ii,jj,kk,3))
     +      +rowm(n0)*eps*swn(j)*(dwxx(k)*dwwge1(i)
     +      *gintx2(jj,ii,kk,1)
     +      +dwxy(k)*dwwge1(i)*gintx2(jj,ii,kk,3))
     +      +rowm(n0)*eps*dswge1(i)*(dwyx(k)*wwn2(j)
     +      *ginty2(ii,jj,kk,3)
     +      +dwyy(k)*wwn2(j)*ginty2(ii,jj,kk,1))
     +      +rowm(n0)*eps*swn(j)*(dwyx(k)*dwwge1(i)
     +      *ginty2(jj,ii,kk,3)
     +      +dwyy(k)*dwwge1(i)*ginty2(jj,ii,kk,1))
      a(3*n0,3*i-1)=a(3*n0,3*i-1)
     +      +rowm(n0)*dwwge2(i)*twx(k)*(pwgn(j)
     +      *gintx2(ii,jj,kk,1)
     +      -row(j)*g*cos(rx)*gintx2(ii,jj,kk,2))
     +      +rowm(n0)*wwn2(k)*twxge2(i)*(pwgn(j)
     +      *gintx2(kk,jj,ii,1)
     +      -row(j)*g*cos(rx)*gintx2(kk,jj,ii,2))
     +      +rowm(n0)*wwn2(j)*twx(k)*gintx2(jj,ii,kk,1)*pwgge2
     +      +rowm(n0)*dwwge2(i)*twy(k)*(pwgn(j)
```

```
     +     *ginty2(ii,jj,kk,1)
     +     -row(j)*g*cos(ry)*ginty2(ii,jj,kk,2))
     +     +rowm(n0)*wwn2(k)*twyge2(i)*(pwgn(j)
     +     *ginty2(kk,jj,ii,1)
     +     -row(j)*g*cos(ry)*ginty2(kk,jj,ii,2))
     +     +rowm(n0)*wwn2(j)*twy(k)*ginty2(jj,ii,kk,1)*pwgge2
      a(3*n0,3*i-1)=a(3*n0,3*i-1)
     +     +rowm(n0)*eps*dswge2(i)*(dwxx(k)*wwn2(j)
     +     *gintx2(ii,jj,kk,1)
     +     +dwxy(k)*wwn2(j)*gintx2(ii,jj,kk,3))
     +     +rowm(n0)*eps*swn(j)*(dwxx(k)*dwwge2(i)
     +     *gintx2(jj,ii,kk,1)
     +     +dwxy(k)*dwwge2(i)*gintx2(jj,ii,kk,3))
     +     +rowm(n0)*eps*dswge2(i)*(dwyx(k)*wwn2(j)
     +     *ginty2(ii,jj,kk,3)
     +     +dwyy(k)*wwn2(j)*ginty2(ii,jj,kk,1))
     +     +rowm(n0)*eps*swn(j)*(dwyx(k)*dwwge2(i)
     +     *ginty2(jj,ii,kk,3)
     +     +dwyy(k)*dwwge2(i)*ginty2(jj,ii,kk,1))
      a(3*n0,3*i)=a(3*n0,3*i)
     +     +rowm(n0)*dwwge3(i)*twx(k)*(pwgn(j)
     +     *gintx2(ii,jj,kk,1)
     +     -row(j)*g*cos(rx)*gintx2(ii,jj,kk,2))
     +     +rowm(n0)*dwwge3(i)*twy(k)*(pwgn(j)
     +     *ginty2(ii,jj,kk,1)
     +     -row(j)*g*cos(ry)*ginty2(ii,jj,kk,2))
     +     +rowm(n0)*eps*swn(j)*(dwxx(k)*dwwge3(i)
     +     *gintx2(jj,ii,kk,1)
     +     +dwxy(k)*dwwge3(i)*gintx2(jj,ii,kk,3))
     +     +rowm(n0)*eps*swn(j)*(dwyx(k)*dwwge3(i)
     +     *ginty2(jj,ii,kk,3)
     +     +dwyy(k)*dwwge3(i)*ginty2(jj,ii,kk,1))
      b(3*n0)=b(3*n0)
     +     -rowm(n0)*wwn2(i)*twx(k)*(pwgn(j)
     +     *gintx2(ii,jj,kk,1)
     +     -row(j)*g*cos(rx)*gintx2(ii,jj,kk,2))
     +     -rowm(n0)*wwn2(i)*twy(k)*(pwgn(j)
     +     *ginty2(ii,jj,kk,1)
     +     -row(j)*g*cos(ry)*ginty2(ii,jj,kk,2))
     +     -rowm(n0)*eps*swn(i)*(dwxx(k)*wwn2(j)
     +     *gintx2(ii,jj,kk,1)
     +     +dwxy(k)*wwn2(j)*gintx2(ii,jj,kk,3))
     +     -rowm(n0)*eps*swn(i)*(dwyx(k)*wwn2(j)
     +     *ginty2(ii,jj,kk,3)
     +     +dwyy(k)*wwn2(j)*ginty2(ii,jj,kk,1))
c
  456 if(itype.eq.1) go to 113
c
      a(3*n0,3*i-2)=a(3*n0,3*i-2)
     +     +room(n0)*won2(k)*toxge1(i)*(pogn(j)
     +     *gintx2(kk,jj,ii,1)
     +     -roo(j)*g*cos(rx)*gintx2(kk,jj,ii,2))
     +     +room(n0)*won2(j)*tox(k)*gintx2(jj,ii,kk,1)*pogge1
     +     +room(n0)*won2(k)*toyge1(i)*(pogn(j)
     +     *ginty2(kk,jj,ii,1)
     +     -roo(j)*g*cos(ry)*ginty2(kk,jj,ii,2))
     +     +room(n0)*won2(j)*toy(k)*ginty2(jj,ii,kk,1)*pogge1
      a(3*n0,3*i-2)=a(3*n0,3*i-2)
```

```
     +     +room(n0)*eps*dsoge1(i)*(doxx(k)*won2(j)
     +     *gintx2(ii,jj,kk,1)
     +     +doxy(k)*won2(j)*gintx2(ii,jj,kk,3))
     +     +room(n0)*eps*dsoge1(i)*(doyx(k)*won2(j)
     +     *ginty2(ii,jj,kk,3)
     +     +doyy(k)*won2(j)*ginty2(ii,jj,kk,1))
      a(3*n0,3*i-1)=a(3*n0,3*i-1)
     +     +room(n0)*won2(k)*toxge2(i)*(pogn(j)
     +     *gintx2(kk,jj,ii,1)
     +     -roo(j)*g*cos(rx)*gintx2(kk,jj,ii,2))
     +     +room(n0)*won2(j)*tox(k)*gintx2(jj,ii,kk,1)*pogge2
     +     +room(n0)*won2(k)*toyge2(i)*(pogn(j)
     +     *ginty2(kk,jj,ii,1)
     +     -roo(j)*g*cos(ry)*ginty2(kk,jj,ii,2))
     +     +room(n0)*won2(j)*toy(k)*ginty2(jj,ii,kk,1)*pogge2
     +     +room(n0)*eps*dsoge2(i)*(doxx(k)*won2(j)
     +     *gintx2(ii,jj,kk,1)
     +     +doxy(k)*won2(j)*gintx2(ii,jj,kk,3))
     +     +room(n0)*eps*dsoge2(i)*(doyx(k)*won2(j)
     +     *ginty2(ii,jj,kk,3)
     +     +doyy(k)*won2(j)*ginty2(ii,jj,kk,1))
      a(3*n0,3*i)=a(3*n0,3*i)
     +     +room(n0)*(-1)*tox(k)*(pogn(j)
     +     *gintx2(ii,jj,kk,1)
     +     -roo(j)*g*cos(rx)*gintx2(ii,jj,kk,2))
     +     +room(n0)*(-1)*toy(k)*(pogn(j)
     +     *ginty2(ii,jj,kk,1)
     +     -roo(j)*g*cos(ry)*ginty2(ii,jj,kk,2))
     +     +room(n0)*eps*son(j)*(doxx(k)*(-1)
     +     *gintx2(jj,ii,kk,1)
     +     +doxy(k)*(-1)*gintx2(jj,ii,kk,3))
     +     +room(n0)*eps*son(j)*(doyx(k)*(-1)
     +     *ginty2(jj,ii,kk,3)
     +     +doyy(k)*(-1)*ginty2(jj,ii,kk,1))
      b(3*n0)=b(3*n0)
     +     -room(n0)*won2(i)*tox(k)*(pogn(j)
     +     *gintx2(ii,jj,kk,1)
     +     -roo(j)*g*cos(rx)*gintx2(ii,jj,kk,2))
     +     -room(n0)*won2(i)*toy(k)*(pogn(j)
     +     *ginty2(ii,jj,kk,1)
     +     -roo(j)*g*cos(ry)*ginty2(ii,jj,kk,2))
     +     -room(n0)*eps*son(i)*(doxx(k)*won2(j)
     +     *gintx2(ii,jj,kk,1)
     +     +doxy(k)*won2(j)*gintx2(ii,jj,kk,3))
     +     -room(n0)*eps*son(i)*(doyx(k)*won2(j)
     +     *ginty2(ii,jj,kk,3)
     +     +doyy(k)*won2(j)*ginty2(ii,jj,kk,1))
c
      do 115 ll=1,4
      if(ll.eq.1) l=n1
      if(ll.eq.2) l=n2
      if(ll.eq.3) l=nn3
      if(ll.eq.4) l=n4
      a(3*n0,3*i-2)=a(3*n0,3*i-2)
     +     +eps*rogn(l)*dsgge1(i)*dg(k)*wgn2(j)
     +     *gintx3(ll,jj,kk,ii,1)
     +     +eps*rogn(j)*sgn(k)*dg(l)*dwgge1(i)
     +     *gintx3(jj,ii,kk,ll,1)
```

```
+     +eps*rogn(l)*dsgge1(i)*dg(k)*wgn2(j)
+     *ginty3(ll,jj,kk,ii,1)
+     +eps*rogn(j)*sgn(k)*dg(l)*dwgge1(i)
+     *ginty3(jj,ii,kk,ll,1)
 a(3*n0,3*i-1)=a(3*n0,3*i-1)
+     +eps*rogn(l)*dsgge2(i)*dg(k)*wgn2(j)
+     *gintx3(ll,jj,kk,ii,1)
+     +eps*rogn(j)*sgn(k)*dg(l)*dwgge2(i)
+     *gintx3(jj,ii,kk,ll,1)
+     +eps*rogn(l)*dsgge2(i)*dg(k)*wgn2(j)
+     *ginty3(ll,jj,kk,ii,1)
+     +eps*rogn(j)*sgn(k)*dg(l)*dwgge2(i)
+     *ginty3(jj,ii,kk,ll,1)
 a(3*n0,3*i)=a(3*n0,3*i)
+     +eps*drgge3(i)*sgn(k)*dg(l)*wgn2(j)
+     *gintx3(ii,jj,kk,ll,1)
+     +eps*rogn(j)*sgn(k)*dg(l)*dwgge3(i)
+     *gintx3(jj,ii,kk,ll,1)
+     +eps*drgge3(i)*sgn(k)*dg(l)*wgn2(j)
+     *ginty3(ii,jj,kk,ll,1)
+     +eps*rogn(j)*sgn(k)*dg(l)*dwgge3(i)
+     *ginty3(jj,ii,kk,ll,1)
 b(3*n0)=b(3*n0)
+     -eps*rogn(i)*sgn(k)*dg(l)*wgn2(j)
+     *gintx3(ii,jj,kk,ll,1)
+     -eps*rogn(i)*sgn(k)*dg(l)*wgn2(j)
+     *ginty3(ii,jj,kk,ll,1)
 do 116 mm=1,4
 if(mm.eq.1) m=n1
 if(mm.eq.2) m=n2
 if(mm.eq.3) m=nn3
 if(mm.eq.4) m=n4
 a(3*n0-1,3*i-2)=a(3*n0-1,3*i-2)
+     -gen3(k)*toxge1(i)*(pogn(j)*bop(l)*pogn(m)
+     *gintx4(ii,jj,kk,ll,mm,1)
+     +pogn(j)*bo1(l)*gen3(m)*gintx4(ii,jj,kk,ll,mm,1)
+     -roo(j)*g*cos(rx)*bop(l)*pogn(m)
+     *gintx4(ii,jj,kk,ll,mm,2)
+     -roo(j)*g*cos(rx)*bo1(l)*gen3(m)
+     *gintx4(ii,jj,kk,ll,mm,2))
+     -gen3(m)*tox(k)*pogge1*(2*bop(l)*pogn(j)
+     *gintx4(mm,jj,kk,ll,ii,1)
+     +bo1(l)*gen3(j)*gintx4(mm,jj,kk,ll,ii,1)
+     -roo(j)*g*cos(rx)*bop(l)
+     *gintx4(mm,jj,kk,ll,ii,2))
 a(3*n0-1,3*i-2)=a(3*n0-1,3*i-2)
+     -gen3(k)*toyge1(i)*(pogn(j)*bop(l)*pogn(m)
+     *ginty4(ii,jj,kk,ll,mm,1)
+     +pogn(j)*bo1(l)*gen3(m)*ginty4(ii,jj,kk,ll,mm,1)
+     -roo(j)*g*cos(ry)*bop(l)*pogn(m)
+     *ginty4(ii,jj,kk,ll,mm,2)
+     -roo(j)*g*cos(ry)*bo1(l)*gen3(m)
+     *ginty4(ii,jj,kk,ll,mm,2))
+     -gen3(m)*toy(k)*pogge1*(2*bop(l)*pogn(j)
+     *ginty4(mm,jj,kk,ll,ii,1)
+     +bo1(l)*gen3(j)*ginty4(mm,jj,kk,ll,ii,1)
+     -roo(j)*g*cos(ry)*bop(l)
+     *ginty4(mm,jj,kk,ll,ii,2))
```

```
 a(3*n0-1,3*i-1)=a(3*n0-1,3*i-1)
+     -gen3(k)*toxge2(i)*(pogn(j)*bop(l)*pogn(m)
+     *gintx4(ii,jj,kk,ll,mm,1)
+     +pogn(j)*bo1(l)*gen3(m)*gintx4(ii,jj,kk,ll,mm,1)
+     -roo(j)*g*cos(rx)*bop(l)*pogn(m)
+     *gintx4(ii,jj,kk,ll,mm,2)
+     -roo(j)*g*cos(rx)*bo1(l)*gen3(m)
+     *gintx4(ii,jj,kk,ll,mm,2))
+     -gen3(m)*tox(k)*pogge2*(2*bop(l)*pogn(j)
+     *gintx4(mm,jj,kk,ll,ii,1)
+     +bo1(l)*gen3(j)*gintx4(mm,jj,kk,ll,ii,1)
+     -roo(j)*g*cos(rx)*bop(l)
+     *gintx4(mm,jj,kk,ll,ii,2))
 a(3*n0-1,3*i-1)=a(3*n0-1,3*i-1)
+     -gen3(k)*toyge2(i)*(pogn(j)*bop(l)*pogn(m)
+     *ginty4(ii,jj,kk,ll,mm,1)
+     +pogn(j)*bo1(l)*gen3(m)*ginty4(ii,jj,kk,ll,mm,1)
+     -roo(j)*g*cos(ry)*bop(l)*pogn(m)
+     *ginty4(ii,jj,kk,ll,mm,2)
+     -roo(j)*g*cos(ry)*bo1(l)*gen3(m)
+     *ginty4(ii,jj,kk,ll,mm,2))
+     -gen3(m)*toy(k)*pogge2*(2*bop(l)*pogn(j)
+     *ginty4(mm,jj,kk,ll,ii,1)
+     +bo1(l)*gen3(j)*ginty4(mm,jj,kk,ll,ii,1)
+     -roo(j)*g*cos(ry)*bop(l)
+     *ginty4(mm,jj,kk,ll,ii,2))
 a(3*n0-1,3*i)=a(3*n0-1,3*i)
+     -tox(k)*(pogn(j)*bop(l)*pogn(m)
+     *gintx4(kk,jj,ii,ll,mm,1)
+     +pogn(j)*bo1(l)*gen3(m)*gintx4(kk,jj,ii,ll,mm,1)
+     -roo(j)*g*cos(rx)*bop(l)*pogn(m)
+     *gintx4(kk,jj,ii,ll,mm,2)
+     -roo(j)*g*cos(rx)*bo1(l)*gen3(m)
+     *gintx4(kk,jj,ii,ll,mm,2))
+     -gen3(m)*tox(k)*(pogn(j)*bo1(l)
+     *gintx4(mm,jj,kk,ll,ii,1)
+     -roo(j)*g*cos(rx)*bo1(l)
+     *gintx4(mm,jj,kk,ll,ii,2))
 a(3*n0-1,3*i)=a(3*n0-1,3*i)
+     -toy(k)*(pogn(j)*bop(l)*pogn(m)
+     *ginty4(kk,jj,ii,ll,mm,1)
+     +pogn(j)*bo1(l)*gen3(m)*ginty4(kk,jj,ii,ll,mm,1)
+     -roo(j)*g*cos(ry)*bop(l)*pogn(m)
+     *ginty4(kk,jj,ii,ll,mm,2)
+     -roo(j)*g*cos(ry)*bo1(l)*gen3(m)
+     *ginty4(kk,jj,ii,ll,mm,2))
+     -gen3(m)*toy(k)*(pogn(j)*bo1(l)
+     *ginty4(mm,jj,kk,ll,ii,1)
+     -roo(j)*g*cos(ry)*bo1(l)
+     *ginty4(mm,jj,kk,ll,ii,2))
 b(3*n0-1)=b(3*n0-1)
+     +gen3(k)*tox(i)*(pogn(j)*bop(l)*pogn(m)
+     *gintx4(ii,jj,kk,ll,mm,1)
+     +pogn(j)*bo1(l)*gen3(m)*gintx4(ii,jj,kk,ll,mm,1)
+     -roo(j)*g*cos(rx)*bop(l)*pogn(m)
+     *gintx4(ii,jj,kk,ll,mm,2)
+     -roo(j)*g*cos(rx)*bo1(l)*gen3(m)
+     *gintx4(ii,jj,kk,ll,mm,2))
```

```
+     +gen3(k)*toy(i)*(pogn(j)*bop(l)*pogn(m)
+     *ginty4(ii,jj,kk,ll,mm,1)
+     +pogn(j)*bo1(l)*gen3(m)*ginty4(ii,jj,kk,ll,mm,1)
+     -roo(j)*g*cos(ry)*bop(l)*pogn(m)
+     *ginty4(ii,jj,kk,ll,mm,2)
+     -roo(j)*g*cos(ry)*bo1(l)*gen3(m)
+     *ginty4(ii,jj,kk,ll,mm,2))
 a(3*n0,3*i-2)=a(3*n0,3*i-2)
+     -room(n0)*won2(k)*toxge1(i)*(pogn(j)*bop(l)*pogn(m)
+     *gintx4(ii,jj,kk,ll,mm,1)
+     +pogn(j)*bo1(l)*gen3(m)*gintx4(ii,jj,kk,ll,mm,1)
+     -roo(j)*g*cos(rx)*bop(l)*pogn(m)
+     *gintx4(ii,jj,kk,ll,mm,2)
+     -roo(j)*g*cos(rx)*bo1(l)*gen3(m)
+     *gintx4(ii,jj,kk,ll,mm,2))
+     -room(n0)*won2(m)*tox(k)*pogge1*(2*bop(l)*pogn(j)
+     *gintx4(mm,jj,kk,ll,ii,1)
+     +bo1(l)*gen3(j)*gintx4(mm,jj,kk,ll,ii,1)
+     -roo(j)*g*cos(rx)*bop(l)
+     *gintx4(mm,jj,kk,ll,ii,2))
 a(3*n0,3*i-2)=a(3*n0,3*i-2)
+     -room(n0)*won2(k)*toyge1(i)*(pogn(j)*bop(l)*pogn(m)
+     *ginty4(ii,jj,kk,ll,mm,1)
+     +pogn(j)*bo1(l)*gen3(m)*ginty4(ii,jj,kk,ll,mm,1)
+     -roo(j)*g*cos(ry)*bop(l)*pogn(m)
+     *ginty4(ii,jj,kk,ll,mm,2)
+     -roo(j)*g*cos(ry)*bo1(l)*gen3(m)
+     *ginty4(ii,jj,kk,ll,mm,2))
+     -room(n0)*won2(m)*toy(k)*pogge1*(2*bop(l)*pogn(j)
+     *ginty4(mm,jj,kk,ll,ii,1)
+     +bo1(l)*gen3(j)*ginty4(mm,jj,kk,ll,ii,1)
+     -roo(j)*g*cos(ry)*bop(l)
+     *ginty4(mm,jj,kk,ll,ii,2))
 a(3*n0,3*i-1)=a(3*n0,3*i-1)
+     -room(n0)*won2(k)*toxge2(i)*(pogn(j)*bop(l)*pogn(m)
+     *gintx4(ii,jj,kk,ll,mm,1)
+     +pogn(j)*bo1(l)*gen3(m)*gintx4(ii,jj,kk,ll,mm,1)
+     -roo(j)*g*cos(rx)*bop(l)*pogn(m)
+     *gintx4(ii,jj,kk,ll,mm,2)
+     -roo(j)*g*cos(rx)*bo1(l)*gen3(m)
+     *gintx4(ii,jj,kk,ll,mm,2))
+     -room(n0)*won2(m)*tox(k)*pogge2*(2*bop(l)*pogn(j)
+     *gintx4(mm,jj,kk,ll,ii,1)
+     +bo1(l)*gen3(j)*gintx4(mm,jj,kk,ll,ii,1)
+     -roo(j)*g*cos(rx)*bop(l)
+     *gintx4(mm,jj,kk,ll,ii,2))
 a(3*n0,3*i-1)=a(3*n0,3*i-1)
+     -room(n0)*won2(k)*toyge2(i)*(pogn(j)*bop(l)*pogn(m)
+     *ginty4(ii,jj,kk,ll,mm,1)
+     +pogn(j)*bo1(l)*gen3(m)*ginty4(ii,jj,kk,ll,mm,1)
+     -roo(j)*g*cos(ry)*bop(l)*pogn(m)
+     *ginty4(ii,jj,kk,ll,mm,2)
+     -roo(j)*g*cos(ry)*bo1(l)*gen3(m)
+     *ginty4(ii,jj,kk,ll,mm,2))
+     -room(n0)*won2(m)*toy(k)*pogge2*(2*bop(l)*pogn(j)
+     *ginty4(mm,jj,kk,ll,ii,1)
+     +bo1(l)*gen3(j)*ginty4(mm,jj,kk,ll,ii,1)
+     -roo(j)*g*cos(ry)*bop(l)
```

```
     +      *ginty4(mm,jj,kk,ll,ii,2))
      a(3*n0,3*i)=a(3*n0,3*i)
     +      -room(n0)*(-1)*tox(k)*(pogn(j)*bop(l)*pogn(m)
     +      *gintx4(ii,jj,kk,ll,mm,1)
     +      +pogn(j)*bo1(l)*gen3(m)*gintx4(ii,jj,kk,ll,mm,1)
     +      -roo(j)*g*cos(rx)*bop(l)*pogn(m)
     +      *gintx4(ii,jj,kk,ll,mm,2)
     +      -roo(j)*g*cos(rx)*bo1(l)*gen3(m)
     +      *gintx4(ii,jj,kk,ll,mm,2))
     +      -room(n0)*won2(m)*tox(k)*(pogn(j)*bo1(l)
     +      *gintx4(mm,jj,kk,ll,ii,1)
     +      -roo(j)*g*cos(rx)*bo1(l)
     +      *gintx4(mm,jj,kk,ll,ii,2))
      a(3*n0,3*i)=a(3*n0,3*i)
     +      -room(n0)*(-1)*toy(k)*(pogn(j)*bop(l)*pogn(m)
     +      *ginty4(ii,jj,kk,ll,mm,1)
     +      +pogn(j)*bo1(l)*gen3(m)*ginty4(ii,jj,kk,ll,mm,1)
     +      -roo(j)*g*cos(ry)*bop(l)*pogn(m)
     +      *ginty4(ii,jj,kk,ll,mm,2)
     +      -roo(j)*g*cos(ry)*bo1(l)*gen3(m)
     +      *ginty4(ii,jj,kk,ll,mm,2))
     +      -room(n0)*won2(m)*toy(k)*(pogn(j)*bo1(l)
     +      *ginty4(mm,jj,kk,ll,ii,1)
     +      -roo(j)*g*cos(ry)*bo1(l)
     +      *ginty4(mm,jj,kk,ll,ii,2))
      b(3*n0)=b(3*n0)
     +      +room(n0)*won2(k)*tox(i)*(pogn(j)*bop(l)*pogn(m)
     +      *gintx4(ii,jj,kk,ll,mm,1)
     +      +pogn(j)*bo1(l)*gen3(m)*gintx4(ii,jj,kk,ll,mm,1)
     +      -roo(j)*g*cos(rx)*bop(l)*pogn(m)
     +      *gintx4(ii,jj,kk,ll,mm,2)
     +      -roo(j)*g*cos(rx)*bo1(l)*gen3(m)
     +      *gintx4(ii,jj,kk,ll,mm,2))
     +      +room(n0)*won2(k)*toy(i)*(pogn(j)*bop(l)*pogn(m)
     +      *ginty4(ii,jj,kk,ll,mm,1)
     +      +pogn(j)*bo1(l)*gen3(m)*ginty4(ii,jj,kk,ll,mm,1)
     +      -roo(j)*g*cos(ry)*bop(l)*pogn(m)
     +      *ginty4(ii,jj,kk,ll,mm,2)
     +      -roo(j)*g*cos(ry)*bo1(l)*gen3(m)
     +      *ginty4(ii,jj,kk,ll,mm,2))
    116 continue
    115 continue
    113 continue
    112 continue
    111 continue
    110 continue
    108 continue
        do 338 i=1,nn
        do 339 j=1,3*nn
        if(3*i-2.eq.j) then
         a(3*i-2,j)=1
        else
         a(3*i-2,j)=0
        endif
    339 continue
        b(3*i-2)=0
    338 continue
c
```

```
c  SUBROUTINE solv IS TO SOLVE THE MATRIX EQUATION.
c
      call solv(n3)
c
      irmm=abs(rmm)*100
      irmax=abs(rmax)*100
  278 format(1x,' rmax =',f10.3)
      if(kount.eq.1) go to 1008
      if(irmm.le.irmax) then
      do 1020 i=2,nn
      if(i.eq.21) go to 1020
      gen3(i)=tge1(i)+u(i*3)*tune
      gen2(i)=tge2(i)+u(i*3-1)*tune
 1020 continue
      ikount=1
      num=num+1
      conu=0.9
      if(num.ge.40) then
      conu=1.1
      isa=-1
      tune=tune*conu*isa
      if(tune.gt.0) tune=isa*tune
      else
      isa=1
      tune=tune*conu*isa
      endif
      go to 1001
      endif
  172 format(1x,/,'rmax stopping',/,1x ' previous maximum residual = ',
     +      f10.2,/,' current maximum residual = ',f10.2)
 1008 rmm=rmax
      tune=0.9
      isa=1
      num=0
      ikount=0
c
c  DO LOOP 117 IS TO GET NEW TRIAL VALUE
c
      do 117 i=2,nn
      do 219 ip=1,nr*(nc-1)+1,nr
      if(i.eq.ip) go to 117
  219 continue
      do 217 ip=1,nk
       if(i.eq.bc(ip)) go to 117
  217 continue
      gen1(i)=gen1(i)+u(i*3-2)
      gen2(i)=gen2(i)+u(i*3-1)
      gen3(i)=gen3(i)+u(i*3)
      nge2=gen2(i)
      if(nge2.le.1510) then
      gen2(i)=1500
      gen3(i)=0.5
      bc(nk)=i
      nk=nk+1
      endif
      tge1(i)=gen3(i)
      tge2(i)=gen2(i)
  117 continue
```

```
c
c  DO LOOP 123 IS TO CHECK THE CONVERGENCY CRITERIA.
c
      do 123 l=1,n3-2,3
      l1=l+1
      l2=l+2
      if(abs(u(l)).gt.10) go to 1001
      if(abs(u(l1)).gt.10) go to 1001
      if(abs(u(l2)).gt.1e-3) go to 1001
  123 continue
      do 65 i=1,nn
      ge1(i)=gen1(i)
      ge2(i)=gen2(i)
      ge3(i)=gen3(i)
      ww2(i)=wwn2(i)
      wg2(i)=wgn2(i)
   65 continue
c
c  OUTPUT OF THE RESULTS
c
      if(time.eq.20.or.time.eq.40) then
      print 999,time
  999 format(1x,//,'TIME = ',f10.4)
      do 211 i=1,nn
      print 899,i,son(i),i,wwn2(i),i,wgn2(i)
  211 continue
  899 format(1x,'so(',i2,') =',f6.4,3x,'wwn2(',i2,') =',e9.3,
     +      3x,'wgn2(',i2,') =',e9.3)
      endif
      if(time.eq.60.or.time.eq.80) then
      print 999,time
      do 311 i=1,nn
      print 899,i,son(i),i,wwn2(i),i,wgn2(i)
  311 continue
      endif
      if(time.eq.100) then
      print 999,time
      do 411 i=1,nn
      print 899,i,son(i),i,wwn2(i),i,wgn2(i)
  411 continue
      endif
 1000 continue
      stop
      end
c
      subroutine ocof(nr,nc)
      common /moon/ge1(200),ge2(200),ge3(200),pwg(200),pog(200),
     +      wo2(200),ww2(200),wg2(200),bop(200),bo1(200),sw(200),
     +      rowm(200),room(200),rkp(200),so(200),sg(200)
      common /dove/dwxx(200),dwxy(200),dwyx(200),dwyy(200),
     +      doxx(200),doxy(200),doyx(200),doyy(200),
     +      dmw,atw,alw,dmo,ato,alo,dmg,dg(200)
      common /door/asw,rnsw,swr,sws,akw,rnkw,aso,rnso,sor,sos,ako,rnko,
     +          ako1,rnko1,ako2,rnko2,swir,som
      common /june/rkx,rky,rx,ry,g
      common /wind/nn,pg,rm1,rm2,rkpg,rkpw,rmw,rma,uo(200),row(200),
     +          roo(200)
      common /tree/pwb,pob1,pob2,bew,be1,be2,rwb,rob1,rob2,
```

```
     +          rmuw,rmu1,rmu2,theta,eps,itype
      common /bird/x(200),y(200),nne(200,4),e(4),n(4)
      nm=nn-1
c
c  IT CALCULATES DENSITIES, COMPRESSIBILITIES, VISCOSITIES, AND
c  DIFFUSION COEFFICIENTIES
c
      do 63 i=1,nn
      row(i)=rwb*exp(bew*(pwg(i)-pwb))
      row(i)=rwb
      roo1=rob1*exp(be1*(pog(i)-pob1))
      roo2=rob2*exp(be2*(pog(i)-pob2))
      roo(i)=1./(ge3(i)/roo1+wo2(i)/roo2)
      bo1(i)=(roo1-roo2)/(ge3(i)*roo2+wo2(i)*roo1)
      bop(i)=be1+be2-(wo2(i)*be1*roo1+ge3(i)*be2*roo2)/(ge3(i)*roo2+
     +     wo2(i)*roo1)
      x2=rm1*(1-ge3(i))/(rm1-rm1*ge3(i)+rm2*ge3(i))
      x1=1-x2
      uo(i)=rmu1**x1*rmu2**x2
      xo2=rm1*(1-ge3(i))/(rm1-rm1*ge3(i)+rm2*ge3(i))
      xw2=rkpw*xo2
      xg2=rkpg*rkpw*xo2
      wg2(i)=xg2*rm2/(rma-xg2*rma+xg2*rm2)
      ww2(i)=xw2*rm2/(rmw-xw2*rmw+xw2*rm2)
      sw(i)=sat(swr,sws,asw,ge1(i),rnsw,row(i))
      sl=sat(sor,sos,aso,ge2(i),rnso,row(i))
      so(i)=sl-sw(i)
      sg(i)=1-sl
      rkrw=graw(ge1(i))
      if(sw(i).lt.swir) rkrw=0
      if(so(i).lt.som) then
       rkro=0
      else
       swi=sw(i)
       soi=so(i)
       rkrow=grao(ge1(i))
       rkrog=per(ako2,ge2(i),rnko2)
       soak=(soi-som)/(1-swir-som)
       if(sw(i).gt.swir) then
        swak=(swi-swir)/(1-swir-som)
       else
        swak=0
       endif
       sgak=(1-swi-soi)/(1-swir-som)
       omw=rkrow/(1-swak)
       omg=rkrog/(1-sgak)
       rkro=soak*omw*omg
      endif
c
c  CALCULATING VELOCOTIES FOR DIFFUSION AND DISPERSION COEFFICIENTS
c
      do 53 it=1,nr
      if(i.eq.it) then
      dpwdx=(pwg(i+nr)-pwg(i))/(x(i+nr)-x(i))
      if(it.eq.1) then
      dpwdy=(pwg(i+1)-pwg(i))/(y(i+1)-y(i))
      go to 43
      endif
```

```
  if(it.eq.nr) then
  dpwdy=(pwg(i)-pwg(i-1))/(y(i)-y(i-1))
  go to 43
  endif
  dpwdy=(pwg(i+1)-pwg(i-1))/(y(i+1)-y(i-1))
  go to 43
  endif
53 continue
  do 54 it=nr*(nc-1)+1,nr*nc
  if(i.eq.it) then
  dpwdx=(pwg(i)-pwg(i-nr))/(x(i)-x(i-nr))
  if(it.eq.nr*(nc-1)+1) then
  dpwdy=(pwg(i+1)-pwg(i))/(y(i+1)-y(i))
  go to 43
  endif
  if(it.eq.nr*nc) then
  dpwdy=(pwg(i)-pwg(i-1))/(y(i)-y(i-1))
  go to 43
  endif
  dpwdy=(pwg(i+1)-pwg(i-1))/(y(i+1)-y(i-1))
  go to 43
  endif
54 continue
  do 55 it=nr+1,nr*(nc-2)+1,nr
  if(i.eq.it) then
  dpwdx=(pwg(i+nr)-pwg(i-nr))/(x(i+nr)-x(i-nr))
  dpwdy=(pwg(i+1)-pwg(i))/(y(i+1)-y(i))
  go to 43
  endif
55 continue
  do 56 it=2*nr,nr*(nc-1),nr
  if(i.eq.it) then
  dpwdx=(pwg(i+nr)-pwg(i-nr))/(x(i+nr)-x(i-nr))
  dpwdy=(pwg(i)-pwg(i-1))/(y(i)-y(i-1))
  go to 43
  endif
56 continue
  dpwdx=(pwg(i+nr)-pwg(i-nr))/(x(i+nr)-x(i-nr))
  dpwdy=(pwg(i+1)-pwg(i-1))/(y(i+1)-y(i-1))
43 continue
  temw=rkrw/(rmuw*eps*sw(i))
  vwx=+temw*rkx*(dpwdx-row(i)*g*cos(rx))
  vwy=+temw*rky*(dpwdy-row(i)*g*cos(ry))
  vvw=(vwx**2+vwy**2)**(1./2.)
  if(vvw.le.0) then
  dwxx(i)=dmw
  dwxy(i)=0
  dwyx(i)=0
  dwyy(i)=dmw
  else
  dwxx(i)=dmw+(atw*vwy**2+alw*vwx**2)/vvw
  dwxy(i)=(alw-atw)*vwx*vwy/vvw
  dwyx(i)=dwxy(i)
  dwyy(i)=dmw+(atw*vwx**2+alw*vwy**2)/vvw
  endif
  temo=rkro/(uo(i)*eps*so(i))
  do 73 it=1,nr
  if(i.eq.it) then
```

```
      dpodx=(pog(i+nr)-pog(i))/(x(i+nr)-x(i))
      if(it.eq.1) then
      dpody=(pog(i+1)-pog(i))/(y(i+1)-y(i))
      go to 83
      endif
      if(it.eq.nr) then
      dpody=(pog(i)-pog(i-1))/(y(i)-y(i-1))
      go to 83
      endif
      dpody=(pog(i+1)-pog(i-1))/(y(i+1)-y(i-1))
      go to 83
      endif
   73 continue
      do 74 it=nr*(nc-1)+1,nr*nc
      if(i.eq.it) then
      dpodx=(pog(i)-pog(i-nr))/(x(i)-x(i-nr))
      if(it.eq.nr*(nc-1)+1) then
      dpody=(pog(i+1)-pog(i))/(y(i+1)-y(i))
      go to 83
      endif
      if(it.eq.nr*nc) then
      dpody=(pog(i)-pog(i-1))/(y(i)-y(i-1))
      go to 83
      endif
      dpody=(pog(i+1)-pog(i-1))/(y(i+1)-y(i-1))
      go to 83
      endif
   74 continue
      do 75 it=nr+1,nr*(nc-2)+1,nr
      if(i.eq.it) then
      dpodx=(pog(i+nr)-pog(i-nr))/(x(i+nr)-x(i-nr))
      dpody=(pog(i+1)-pog(i))/(y(i+1)-y(i))
      go to 83
      endif
   75 continue
      do 76 it=2*nr,nr*(nc-1),nr
      if(i.eq.it) then
      dpodx=(pog(i+nr)-pog(i-nr))/(x(i+nr)-x(i-nr))
      dpody=(pog(i)-pog(i-1))/(y(i+1)-y(i-1))
      go to 83
      endif
   76 continue
      dpodx=(pog(i+nr)-pog(i-nr))/(x(i+nr)-x(i-nr))
      dpody=(pog(i+1)-pog(i-1))/(y(i+1)-y(i-1))
   83 continue
      temo=rkro/(uo(i)*eps*so(i))
      vox=+temo*rkx*(dpodx-roo(i)*g*cos(rx))
      voy=+temo*rky*(dpody-roo(i)*g*cos(ry))
      vvo=(vox**2+voy**2)**(1./2.)
      if(vvo.le.0) then
      doxx(i)=dmo
      doxy(i)=0
      doyx(i)=0
      doyy(i)=dmo
      else
      doxx(i)=dmo+(ato*voy**2+alo*vox**2)/vvo
      doxy(i)=(alo-ato)*vox*voy/vvo
      doyx(i)=doxy(i)
```

```
      doyy(i)=dmo+(ato*vox**2+alo*voy**2)/vvo
      endif
      dg(i)=dmg
      rkp(i)=so(i)/(so(i)+sw(i))
   63 continue
      return
      end
c
      subroutine coef
      common /sun/ gen1(200),gen2(200),gen3(200),pwgn(200),pogn(200),
     +       won2(200),wwn2(200),wgn2(200),swn(200),son(200),sgn(200),
     +       twx(200),twy(200),tox(200),toy(200),beg(200),dbgdwo(200),
     +       twxge1(200),twyge1(200),twxge2(200),twyge2(200),
     +       toxge1(200),toyge1(200),toxge2(200),toyge2(200),
     +       dwwge1(200),dwwge2(200),dwwge3(200),
     +       dwgge1(200),dwgge2(200),dwgge3(200),
     +       rogn(200),drgge3(200)
      common /may/ dswge1(200),dswg11(200),dswg21(200),
     +          dswge2(200),dswg22(200),dswg12(200),
     +          dsoge1(200),dsog11(200),dsog21(200),
     +          dsoge2(200),dsog22(200),dsog12(200),
     +          dsgge1(200),dsgg11(200),dsgg21(200),
     +          dsgge2(200),dsgg22(200),dsgg12(200)
      common /door/asw,rnsw,swr,sws,akw,rnkw,aso,rnso,sor,sos,ako,rnko,
     +          ako1,rnko1,ako2,rnko2,swir,som
      common /hill/ru,t,z
      common /june/rkx,rky,rx,ry,g
      common /wind/nn,pg,rm1,rm2,rkpg,rkpw,rmw,rma,uo(200),row(200),
     +          roo(200)
      common /tree/pwb,pob1,pob2,bew,be1,be2,rwb,rob1,rob2,
     +          rmuw,rmu1,rmu2,theta,eps,itype
c
c THIS SUBROUTINE CALCULATES SATURATIONS, RELATIVE PERMEABILITIES,
c MOBILITIES, DERIVATIVES OF SATURATION AND MOBILITIES WITH RESPECT TO
c THE PRIMARY VARIABLES.
c
      wkx=rmuw/rkx
      wky=rmuw/rky
      do 10 i=1,nn
      okx=uo(i)/rkx
      oky=uo(i)/rky
      dp=-1
      pc1=gen1(i)
      pc3=gen2(i)
      pc11=gen1(i)+dp
      pc10=gen1(i)-dp
      pc31=pc3+dp
      pc30=pc3-dp
      swn(i)=sat(swr,sws,asw,pc1,rnsw,row(i))
      if(swn(i).lt.swir) then
       sw1=swn(i)
       sw0=swn(i)
      endif
      sw1=sat(swr,sws,asw,pc11,rnsw,row(i))
      sw0=sat(swr,sws,asw,pc10,rnsw,row(i))
      dswge1(i)=0
      dswge2(i)=0
      dswg11(i)=0
```

```
dswg12(i)=0
dswg22(i)=0
dswg21(i)=0
sl=sat(sor,sos,aso,pc3,rnso,row(i))
sl1=sat(sor,sos,aso,pc31,rnso,row(i))
sl0=sat(sor,sos,aso,pc30,rnso,row(i))
son(i)=sl-swn(i)
so11=sl-sw1
so10=sl-sw0
so21=sl1-swn(i)
so20=sl0-swn(i)
dsoge1(i)=(so11-son(i))/dp
dsoge1(i)=0
dsog11(i)=(so11-2*son(i)+so10)/(2*dp)
dsog11(i)=0
dsoge2(i)=(so21-son(i))/dp
dsog22(i)=(so21-2*son(i)+so20)/(2*dp)
 dsog12(i)=0
 dsog21(i)=0
sgn(i)=1-sl
sg1=1-sl1
sg0=1-sl0
dsgge2(i)=(sg1-sgn(i))/dp
dsgg22(i)=(sg1-2*sgn(i)+sg0)/(2*dp)
twx(i)=graw(pc1)/wkx
twy(i)=graw(pc1)/wky
tw1=graw(pc11)
tw0=graw(pc10)
twxge1(i)=(tw1-tw0)/(2*dp)
twyge1(i)=(tw1-tw0)/(2*dp)
twxge2(i)=0
twyge2(i)=0
if(swn(i).lt.swir) then
 twx(i)=0
 twy(i)=0
 twxge1(i)=0
 twyge1(i)=0
endif
if(son(i).lt.som) then
 rkro=0
 tox(i)=0
 toy(i)=0
 toxge1(i)=0
 toyge1(i)=0
 toxge2(i)=0
 toyge2(i)=0
else
 swi=swn(i)
 soi=son(i)
 do 95 ii=1,3
  rkrow=grao(pc1)
  rkrog=per(ako2,pc3,rnko2)
  soak=(soi-som)/(1-swir-som)
  if(swn(i).gt.swir) then
   swak=(swi-swir)/(1-swir-som)
  else
   swak=0
  endif
```

```
      sgak=(1-swi-soi)/(1-swir-som)
      omw=rkrow/(1-swak)
      omg=rkrog/(1-sgak)
      if(ii.eq.1) then
      rkro=soak*omw*omg
      endif
      pc1=pc1+dp
      swi=sw1
      soi=so11
      if(ii.eq.3) go to 97
      if(ii.eq.2) then
       rkro1=soak*omw*omg
       go to 96
      endif
      go to 95
 96  pc1=pc1-2*dp
      pc3=pc3+dp
      swi=swn(i)
      soi=so21
 97   rkro2=soak*omw*omg
 95 continue
    toxge1(i)=((rkro1-rkro)/okx)/dp
    toyge1(i)=((rkro1-rkro)/oky)/dp
    toxge1(i)=0
    toyge1(i)=0
    toxge2(i)=((rkro2-rkro)/okx)/dp
    toyge2(i)=((rkro2-rkro)/oky)/dp
    tox(i)=rkro/okx
    toy(i)=rkro/oky
    endif
    if(itype.eq.7) go to 891
    if(itype.eq.1) then
    dwwge1(i)=0
    dwwge2(i)=0
    dwwge3(i)=1
    go to 10
    endif
    son(i)=1-swn(i)
    so1=1-sw1
    so0=1-sw0
    if(son(i).lt.som+0.001) then
    dsoge1(i)=1e-6
    dsog11(i)=0
    else
    dsoge1(i)=-dswge1(i)
    dsog11(i)=-dswg11(i)
    dsoge2(i)=0
    endif
    dsog22(i)=0
    dsog12(i)=0
    dsog21(i)=0
    twx(i)=graw(pc1)
    twy(i)=graw(pc1)
    tw1=graw(pc11)
    tw0=graw(pc10)
    twxge1(i)=(tw1-tw0)/(2*dp)
    twyge1(i)=(tw1-tw0)/(2*dp)
    twxge2(i)=0
```

```
      twyge2(i)=0
      if(son(i).lt.som) then
      rkro=0
      tox(i)=0
      toy(i)=0
      toxge1(i)=0
      toyge1(i)=0
      toxge2(i)=0
      toyge2(i)=0
      else
      tox(i)=grao(pc1)
      toy(i)=grao(pc1)
      to1=grao(pc11)
      to0=grao(pc10)
      toxge1(i)=(to1-to0)/(2*dp)
      toyge1(i)=(to1-to0)/(2*dp)
      toxge2(i)=0
      toyge2(i)=0
      endif
      if(itype.eq.1) then
      dwwge1(i)=0
      dwwge2(i)=0
      dwwge3(i)=1
      go to 10
      endif
  891 dww=-0.0001
      won11=gen3(i)+dww
      xo2=rm1*(1-gen3(i))/(rm1-rm1*gen3(i)+rm2*gen3(i))
      xo21=rm1*(1-won11)/(rm1-rm1*won11+rm2*won11)
      xw2=rkpw*son(i)/(son(i)+1e-4)*xo2
      xw21=rkpw*so11/(so11+1e-4)*xo2
      xw22=rkpw*so21/(so21+1e-4)*xo2
      xw23=rkpw*son(i)/(son(i)+1e-4)*xo21
      wwn2(i)=xw2*rm2/(rmw-xw2*rmw+xw2*rm2)
      wwn21=xw21*rm2/(rmw-xw21*rmw+xw21*rm2)
      wwn22=xw22*rm2/(rmw-xw22*rmw+xw22*rm2)
      wwn23=xw23*rm2/(rmw-xw23*rmw+xw23*rm2)
      dwwge1(i)=(wwn21-wwn2(i))/dp
      dwwge2(i)=(wwn22-wwn2(i))/dp
      dwwge3(i)=(wwn23-wwn2(i))/dww
      xg2=rkpg*rkpw*son(i)/(son(i)+1e-4)*xo2
      xg21=rkpg*rkpw*so11/(so11+1e-4)*xo2
      xg22=rkpg*rkpw*so21/(so21+1e-4)*xo2
      xg23=rkpg*rkpw*son(i)/(son(i)+1e-4)*xo21
      wgn2(i)=xg2*rm2/(rma-xg2*rma+xg2*rm2)
      wgn21=xg21*rm2/(rma-xg21*rma+xg21*rm2)
      wgn22=xg22*rm2/(rma-xg22*rma+xg22*rm2)
      wgn23=xg23*rm2/(rma-xg23*rma+xg23*rm2)
      dwgge1(i)=(wgn21-wgn2(i))/dp
      dwgge2(i)=(wgn22-wgn2(i))/dp
      dwgge3(i)=(wgn23-wgn2(i))/dww
      rmg=rm2*rma/(wgn2(i)*rma+(1-wgn2(i))*rm2)
      rmg1=rm2*rma/(wgn23*rma+(1-wgn23)*rm2)
      rogn(i)=pg/(z*ru*t)*rmg
      rog1=pg/(z*ru*t)*rmg1
      drgge3(i)=(rog1-rogn(i))/dww
      beg(i)=(rm2-rma)/(wgn2(i)*rma+(1-wgn2(i))*rm2)
      beg1=(rm2-rma)/(wgn23*rma+(1-wgn23)*rm2)
```

```
      dbgdwo(i)=(beg1-beg(i))/dww
    10 continue
       return
       end
c
c  THIS FUNCTION IS VAN GENUCHTEN'S EQUATION FOR CALCULATING
c  RELATIVE PERMEABILITIES
c
       function per(a,pc,rn)
       g=980.7
       h=pc/980.456
       if(h.le.0) then
       per=1
       else
       rm=1-1/rn
       per=(1-(a*h)**(rn-1)*(1+(a*h)**rn)**(-rm))**2/(1+(a*h)**rn)
      +    **(rm/2.)
       endif
       return
       end
c
c  IT IS ANOTHER TYPE OF FUNCTION STATEMENT FOR CALCULATING RELATIVE
c  PERMEABILITIES WHIC IS USED BY ABRIOLA.
c
       function pero(b0,b1,b2,b3,sat)
       pero=b0+b1*sat+b2*sat**2+b3*sat**3
       if(pero.lt.0) then
       print 294,pero,sat
   294 format(' STOP BECAUSE RELATIVE PERMEABILITY IS NEGATIVE ',/,
      +       ' pero = ',e10.4,'  Sw = ',e10.4)
       stop
       endif
       return
       end
c
c  IT IS A MATHEMATICAL EXPRESSION FOR CALCULATING SATURATION
c
       function sat1(pc)
       if(pc.gt.-2892.38.and.pc.le.-1421.96) then
        sat1=1.52208-0.0718947*log(-pc)
       else
        sat1=2.94650-0.250632*log(-pc)
       endif
       return
       end
c
c  IT IS A MATHEMATICAL EXPRESSION FOR CALCULATING PERMEABILITY
c
       function per1(sat)
        per1=(1.235376e-6)*exp(13.604*sat)
       return
       end
c
c  IT IS VAN GENUCHTEN'S EQUATIONS FOR CALCULATING SATURATIONS
c
       function sat(sar,sas,a,pc,rn,r)
       g=980.7
       h=pc/980.456
```

```
      if(h.le.0) then
      sat=sas
      else
      rm=1-1/rn
      sat=sar+(sas-sar)/(1+(a*h)**rn)**rm
      endif
      return
      end
c
c  FUNCTION graw AND grao ARE FOR TREATING EXPERIMENTAL DATA OF
c  RELATIVE PERMEABILITIES
c
      function graw(pc)
      common /phic/ dpc(20),drw(20),dro(20),dro1(20),dro2(20),
     +          dsw(20),dso(20)
      do 10 i=1,15
      if(pc.ge.dpc(i).and.pc.lt.dpc(i+1)) then
      graw=(drw(i+1)-drw(i))/(dpc(i+1)-dpc(i))*(pc-dpc(i))+drw(i)
      go to 20
      endif
   10 continue
   20 return
      end
c
      function grao(pc)
      common /phic/ dpc(20),drw(20),dro(20),dro1(20),dro2(20),
     +          dsw(20),dso(20)
      do 10 i=1,15
      if(pc.ge.dpc(i).and.pc.lt.dpc(i+1)) then
      grao=(dro(i+1)-dro(i))/(dpc(i+1)-dpc(i))*(pc-dpc(i))+dro(i)
      go to 20
      endif
   10 continue
   20 return
      end
c
      subroutine coor(ne,nn)
      common /bird/x(200),y(200),nne(200,4),e(4),n(4)
      do 10 i=1,ne
      read(51,*) (nne(i,j),j=1,4)
   10 continue
      do 20 i=1,nn
      read(51,*) x(i),y(i)
   20 continue
      return
      end
c
      subroutine inicon(ge1,ge2,ge3,gen1,gen2,gen3,nn)
      dimension ge1(200),ge2(200),ge3(200),gen1(200),gen2(200),gen3(200)
      read(51,*) gei1,gei2,gei3
      do 10 i=1,nn
      ge1(i)=gei1
      ge2(i)=gei2
      ge3(i)=gei3
      gen1(i)=ge1(i)
      gen2(i)=ge2(i)
      gen3(i)=ge3(i)
   10 continue
```

```
      return
      end
c
      subroutine bouncon(ge1,ge2,ge3,gen1,gen2,gen3,nr,nc,nn)
      dimension ge1(200),ge2(200),ge3(200),gen1(200),gen2(200),
     +        gen3(200),dc(60)
      read(51,*) (dc(i),i=1,2*nr+2*nc-4)
      k=0
      do 10 i=1,nr
      k=k+1
      if(dc(k).eq.0) go to 10
      read(51,*) ge1(i),ge2(i),ge3(i)
      gen1(i)=ge1(i)
      gen2(i)=ge2(i)
      gen3(i)=ge3(i)
   10 continue
      do 12 i=1,nr
      k=k+1
      if(dc(k).eq.0) go to 12
      j=nr*(nc-1)+i
      read(51,*) ge1(j),ge2(j),ge3(j)
      gen1(j)=ge1(j)
      gen2(j)=ge2(j)
      gen3(j)=ge3(j)
   12 continue
      do 20 i=2,nc-1
      k=k+1
      if(dc(k).eq.0) go to 20
      j=nr*(i-1)+1
      read(51,*) ge1(j),ge2(j),ge3(j)
      gen1(j)=ge1(j)
      gen2(j)=ge2(j)
      gen3(j)=ge3(j)
   20 continue
      do 22 i=2,nc-1
      k=k+1
      if(dc(k).eq.0) go to 22
      l=nr*i
      read(51,*) ge1(l),ge2(l),ge3(l)
      gen1(l)=ge1(l)
      gen2(l)=ge2(l)
      gen3(l)=ge3(l)
   22 continue
      return
      end
c
      subroutine gint(n1,n2,n3,n4,i4)
      common /bird/x(200),y(200),nne(200,4),e(4),n(4)
      common /lake/gintx1(4,4,2),ginty1(4,4,3),gintx2(4,4,4,3),
     +        ginty2(4,4,4,3),gintx3(4,4,4,4,3),ginty3(4,4,4,4,3),
     +        gintx4(4,4,4,4,4,2),ginty4(4,4,4,4,4,2)
      a1=0
      a2=0
      b=0
      c1=0
      c2=0
      d=0
      r=0
```

```
  do 5 ia=1,3
  do 5 i=1,4
  do 5 j=1,4
  if(ia.eq.3) go to 6
  gintx1(i,j,ia)=0
6 ginty1(i,j,ia)=0
  do 5 k=1,4
  gintx2(i,j,k,ia)=0
  ginty2(i,j,k,ia)=0
  do 5 l=1,4
  gintx3(i,j,k,l,ia)=0
  ginty3(i,j,k,l,ia)=0
  do 5 m=1,4
  if(ia.eq.3) go to 5
  gintx4(i,j,k,l,m,ia)=0
  ginty4(i,j,k,l,m,ia)=0
5 continue
  do 10 ii=1,4
  if(ii.eq.1) i=n1
  if(ii.eq.2) i=n2
  if(ii.eq.3) i=n3
  if(ii.eq.4) i=n4
  a1=a1+y(i)*n(ii)
  a2=a2+y(i)*e(ii)
  b=b+y(i)*n(ii)*e(ii)
  c1=c1+x(i)*e(ii)
  c2=c2+x(i)*n(ii)
  d=d+x(i)*n(ii)*e(ii)
10 continue
  te=(3./5.)**0.5
  tn=(3./5.)**0.5
  do 20 ig=-1,1
  if(ig) 23,24,25
23 ge=-te
  wx=5./9.
  go to 32
24 ge=0
  wx=8./9.
  go to 32
25 ge=te
  wx=5./9.
32 do 20 jg=-1,1
  if(jg) 27,28,29
27 gn=-tn
  wy=5./9.
  go to 31
28 gn=0
  wy=8./9.
  go to 31
29 gn=tn
  wy=5./9.
31 tem=(c1+d*gn)*(a1+b*ge)-(c2+d*ge)*(a2+b*gn)
  ajac=tem/16
  gx2=((a1+b*ge)*(e(i4)+e(i4)*n(i4)*gn)-(a2+b*gn)
 +      *(n(i4)+e(i4)*n(i4)*ge))/tem
  gy2=((c1+d*gn)*(n(i4)+e(i4)*n(i4)*ge)-(c2+d*ge)
 +      *(e(i4)+e(i4)*n(i4)*gn))/tem
  ginn=0.25*(1+e(i4)*ge)*(1+n(i4)*gn)
```

```
      do 20 i=1,4
      gin1=0.25*(1+e(i)*ge)*(1+n(i)*gn)
      do 20 j=1,4
      giin=0.25*(1+e(j)*ge)*(1+n(j)*gn)
      gx1=((a1+b*ge)*(e(j)+e(j)*n(j)*gn)-(a2+b*gn)
     +     *(n(j)+e(j)*n(j)*ge))/tem
      gy1=((c1+d*gn)*(n(j)+e(j)*n(j)*ge)-(c2+d*ge)
     +     *(e(j)+e(j)*n(j)*gn))/tem
      gintx1(i,j,1)=gintx1(i,j,1)+wx*wy*gin1*gx1*gx2*ajac
      gintx1(i,j,2)=gintx1(i,j,2)+wx*wy*gin1*abs(gx1)*gx2*ajac
      ginty1(i,j,1)=ginty1(i,j,1)+wx*wy*gin1*gy1*gy2*ajac
      ginty1(i,j,2)=ginty1(i,j,2)+wx*wy*gin1*giin*gy2*ajac
      ginty1(i,j,3)=ginty1(i,j,3)+wx*wy*gin1*giin*gy2*ajac
      do 20 k=1,4
      gin2=0.25*(1+e(k)*ge)*(1+n(k)*gn)
      gintx2(i,j,k,1)=gintx2(i,j,k,1)+wx*wy*gin1*gin2*gx1*gx2*ajac
      gintx2(i,j,k,2)=gintx2(i,j,k,2)+wx*wy*gin1*gin2*abs(gx1)*gx2*ajac
      gintx2(i,j,k,3)=gintx2(i,j,k,3)+wx*wy*gin1*gin2*gy1*gx2*ajac
      ginty2(i,j,k,1)=ginty2(i,j,k,1)+wx*wy*gin1*gin2*gy1*gy2*ajac
      ginty2(i,j,k,2)=ginty2(i,j,k,2)+wx*wy*gin1*gin2*giin*gy2*ajac
      ginty2(i,j,k,3)=ginty2(i,j,k,3)+wx*wy*gin1*gin2*gx1*gy2*ajac
      do 20 l=1,4
      gin3=0.25*(1+e(l)*ge)*(1+n(l)*gn)
      gintx3(i,j,k,l,1)=gintx3(i,j,k,l,1)+wx*wy*gin1*gin2*gin3*gx1*gx2
     +              *ajac
      gintx3(i,j,k,l,2)=gintx3(i,j,k,l,2)+wx*wy*gin1*gin2*gin3
     +             *abs(gx1)*gx2*ajac
      gintx3(i,j,k,l,3)=gintx3(i,j,k,l,3)+wx*wy*gin1*gin2*gin3*gy1*gx2
     +              *ajac
      ginty3(i,j,k,l,1)=ginty3(i,j,k,l,1)+wx*wy*gin1*gin2*gin3*gy1*gy2
     +              *ajac
      ginty3(i,j,k,l,2)=ginty3(i,j,k,l,2)+wx*wy*gin1*gin2*gin3
     +              *giin*gy2*ajac
      ginty3(i,j,k,l,3)=ginty3(i,j,k,l,3)+wx*wy*gin1*gin2*gin3*gx1*gy2
     +              *ajac
      do 20 m=1,4
      gx3=((a1+b*ge)*(e(m)+e(m)*n(m)*gn)-(a2+b*gn)
     +     *(n(m)+e(m)*n(m)*ge))/tem
      gy3=((c1+d*gn)*(n(m)+e(m)*n(m)*ge)-(c2+d*ge)
     +     *(e(m)+e(m)*n(m)*gn))/tem
      gintx4(i,j,k,l,m,1)=gintx4(i,j,k,l,m,1)+wx*wy*ginn*gin1*gin2*gin3
     +               *gx1*gx3*ajac
      gintx4(i,j,k,l,m,2)=gintx4(i,j,k,l,m,2)+wx*wy*ginn*gin1*gin2*gin3
     +               *abs(gx1)*gx3*ajac
      ginty4(i,j,k,l,m,1)=ginty4(i,j,k,l,m,1)+wx*wy*ginn*gin1*gin2*gin3
     +               *gy1*gy3*ajac
      ginty4(i,j,k,l,m,2)=ginty4(i,j,k,l,m,2)+wx*wy*ginn*gin1*gin2*gin3
     +               *abs(gy1)*gy3*ajac
   20 continue
      return
      end
c
c  IT IS FOR CALCULATING THE VOLUME OF A CELL
c
      function vol(n1,n2,n3,n4,i4)
      common /bird/x(200),y(200),nne(200,4),e(4),n(4)
      a1=0
      a2=0
```

```
      b=0
      c1=0
      c2=0
      d=0
      r=0
      do 10 ii=1,4
      if(ii.eq.1) i=n1
      if(ii.eq.2) i=n2
      if(ii.eq.3) i=n3
      if(ii.eq.4) i=n4
      a1=a1+y(i)*n(ii)
      a2=a2+y(i)*e(ii)
      b=b+y(i)*n(ii)*e(ii)
      c1=c1+x(i)*e(ii)
      c2=c2+x(i)*n(ii)
      d=d+x(i)*n(ii)*e(ii)
   10 continue
      ge=1./3**0.5
      gn=1./3**0.5
      vol=0
      do 20 i=1,2
      ge=ge*(-1)
      do 20 j=1,2
      gn=gn*(-1)
      vol=vol+1./64*(1+e(i4)*ge)*(1+n(i4)*gn)*((c1+d*gn)*(a1+b*ge)
     +   -(c2+d*ge)*(a2+b*gn))
   20 continue
      return
      end
      subroutine solv(n3)
      common /july/a(600,600),b(600),u(600),rmax
      do 40 i=1,n3
      pivot=a(i,i)
      do 30 j=i+1,n3
      dult=a(j,i)/pivot
      a(j,i)=0
      do 25 k=i+1,n3
      a(j,k)=a(j,k)-dult*a(i,k)
   25 continue
      b(j)=b(j)-dult*b(i)
   30 continue
   40 continue
      u(n3)=b(n3)/a(n3,n3)
      do 20 i=n3-1,1,-1
      quot=b(i)
      do 10 j=i+1,n3
      quot=quot-a(i,j)*u(j)
   10 continue
      u(i)=quot/a(i,i)
   20 continue
      rmax=0
      do 29 i=1,n3
      if(abs(u(i)).gt.abs(rmax)) then
      rmax=u(i)
      endif
   29 continue
      return
      end
```