FINITE ELEMENT ANALYSIS

OF

GROUNDWATER CONTAMINATION

N. Mahomed

June 1990

Submitted to the University of Cape Town in partial fulfillment of the requirements for the degree of Master of Science in Civil Engineering.

Declaration

I declare that this thesis is based upon my own work and has not been submitted in this or in a similar form for a degree at any University.

Signed by candidate

N. Mahomed

4/4/90

Date

Acknowledgments

This research was supported by a research grant from the Peninsula Technikon and computer and other equipment made available by the Department of Mechanical Engineering, Peninsula Technikon.

The Author wishes to extend sincere thanks to the following people:

- Professor ADW Sparks of the Department of Civil Engineering, University of Cape Town, for assistance towards the research.
- Mr S. Cassiem and Mr T. Mellem of the Department of Mechanical Engineering, Peninsula Technikon, for their invaluable assistance in the construction of the experimental apparatus and the monitoring of the experiments.

Also, appreciation is extended to the Vice-Rector (Academic), Peninsula Technikon, for granting study leave during the period of research.

Abstract

The purpose of this study was to develop a computational Finite Element model, validated by experimentation, to assist in the understanding of groundwater contamination problems. It was mainly aimed at studying the extent and manner of travel of contaminants in the saturated soil of unconfined aquifers which may be pumped by wells.

Previous researchers have excluded certain aspects of the problem in the modelling stage. For instance, some assumed constant groundwater velocities. Others used the Dupuit Approximation to reduce the spatial dimensions to two in order to reduce the computational effort. None, however, carried out experimental modelling alongside the mathematical modelling. Besides, all computational models thus far were developed and executed on expensive mainframe systems.

The three-dimensional coupled groundwater flow / diffusion-convection equations were solved using the Galerkin approach to the Finite Element method. Linear Lagrange isoparametric basis functions were used to interpolate the solution. The mathematical model was then computerised on the standard IBM Compatible PC using Turbo BASIC for the numerical routines and DBASEIV for the data input and output routines.

The model was tested against one-dimensional analytical solutions. The results produced were fairly accurate, even in the case of convection-dominated diffusion-convection problems.

An experimental model was developed and constructed to simulate the aquifer conditions. A specially graded sand was used as the aquifer medium. The properties of the sand, such as its permeability and porosity, were experimentally determined. An experimental method was also developed to determine the coefficient of diffusion of a solute in a porous medium.

In the experiments carried out, NaCl was used as the contaminant. The concentrations of NaCl in the groundwater were determined using the electrical conductivity method.

The results of two experiments were presented and compared with that obtained from the computational model. The first experiment was based on the determination of the free surface elevation in a high drawdown case. In the second experiment, the aquifer was contaminated with saline water recharged via a well. The contaminant transport was convection-dominated. In both experiments, the results compared reasonably well. Most of the discrepancies were ascribed to errors in the experimental technique.

The computational model required extensive computational time. The second experiment, run over a real time of 18 hours, required a computational time of approximately 30 hours. This was mainly due to iterations within a time step as well as the recalculation of the Finite Element matrices because of variations in the domain during the transient groundwater flow stages. However, the computational Finite Element model proved reliable and economical. With the advent of more powerful IBM Compatible PC systems, its speed and memory capabilities can be extensively upgraded.

Table of Contents

List of illustrations

Figures

Figures

Figures

Tables

Chapter 1

Introduction

Many developed countries utilise surface water resources, such as rivers and lakes, to meet the public demand for water. In South Africa, for instance, 87 percent of the freshwater supplies are acquired from surface water, mainly rivers flowing into dams [5]. The remainder, only 13 percent, is extracted from the ground.

The reason for this is not the abundance of surface water but rather the ease with which surface water can be trapped and controlled. Or maybe it stems from the inability of man to recognise and evaluate the potential of existing groundwater resources. This may be true if one considers the fact that groundwater accounts for approximately 98 percent of the world's freshwater.

The situation in Third World countries is somewhat different. With generally higher populations coupled with a lack of financial resources to develop surface water supplies, the trend is towards the use of groundwater. Although, in its natural form, groundwater may be of a higher quality than surface water, many cases of groundwater contamination have arisen.

Groundwater contamination can be defined as the addition of chemical, physical or biological substances which causes deterioration in the natural quality, generally through the activities of man. This impairment of the water quality adversely affects such waters for domestic, agricultural and industrial use.

The list of potential contaminants is extensive. One, for instance, is the improper control, treatment and disposal of sewage. In 1980, the World Health organisation estimated that 1320 million people (57 percent) of the developing world (excluding China) were without a clean water supply, while 1730 million (75 percent) were without adequate sanitation. The results are startling. At least 30 000 people die per day in the Third World because of inadequate water and sanitation facilities [1].

Groundwater pollution is not simply a hydrological or geological problem. Social and economic factors sometimes make the control of groundwater pollution impossible. Even political factors sometimes contribute to the problem. Paul Findley, a former U.S. Congressman, author of the book "They Dare To Speak Out" reported that Israeli soldiers in the Gaza Strip purposefully overpump the aquifers in the near-coastal region. This causes salt-water intrusion, adversely affecting the only water supply of the inhabitants. This, according to Findley, is an attempt to force the Palestinians to leave the region.

Many mining and industrial concerns, who are responsible for the disposal of improperly treated wastes, deny that their wastes are potentially hazardous to the quality of groundwater. The main argument is that the deposition of wastes above the water table as well as the slow rate of travel of the contaminants will not allow the groundwater to become polluted. However, water tables may fluctuate. Also, infiltration of rain water can cause the contaminants to reach the aquifers. Even if the contaminants do travel slowly, the case of pollution may go undetected for a number of years before it is realised, sometimes a long distance away from the source of pollution.

The purpose of this study is to develop a computational model, validated by experimentation, to assist in the understanding of groundwater contamination problems. It is mainly aimed at studying the extent and manner of travel of contaminants in the saturated soil of unconfined aquifers which may be pumped by wells.

The approach will be to use a Finite Element method to solve the mathematical model of the problem. The Finite Element model will then be computerised on the standard IBM Compatible PC. An experimental simulation of the problem will be carried out alongside the computational model. This will enable the verification of the computational model.

Chapter 2

Groundwater contamination: A literature survey

2.1 Causes and effects of groundwater contamination

Numerous cases of groundwater contamination have been reported [1]. These include groundwater contamination arising from arsenic pesticides (Hungary), insecticide disposal (Rumania), nitrates (UK and Hungary), lead (Italy), metal plating wastes (USA), hydrocarbons (Czechoslovakia), sugar waste disposal (Cuba), bacterial contamination (Mexico and Italy), road salt (Canada), oil/gas production (Hungary) and sewerage (UK). In the latter case, 1000 people were affected by gastroenteritis as a result of a borehole becoming contaminated by a combination of a leaking sewer and a polluted surface stream.

A soil-aquifer has a self-cleansing ability. This is due to the biological (such as the biodegradation of bacteria), physical (such as filtration), and chemical (such as oxidation and precipitation) processes that occur during the migration of the pollutants. It is apparent, though, that this self-cleansing ability depends upon the physical and chemical form of the pollutant, the nature of the aquifer material and the way in which the pollutant enters the ground.

The greatest danger of groundwater pollution is from surface sources such as farm animals, man, sewers, polluted streams, refuse disposal sites, industrial waste disposal sites, etc. In fact, the list is endless. Areas with thin soil cover or where the aquifer is exposed, such as the recharge area, are most susceptible to the establishment of pollution sources. Some of the more common forms of groundwater pollution will be discussed further.

Domestic waste disposal

A major threat to the quality of groundwater arises from the disposal of domestic and commercial wastes in landfill sites. A large percentage of this untreated waste, about 60%, are comprised of biodegradable solids (such as paper, metal and vegetable matter). Leachate is formed when

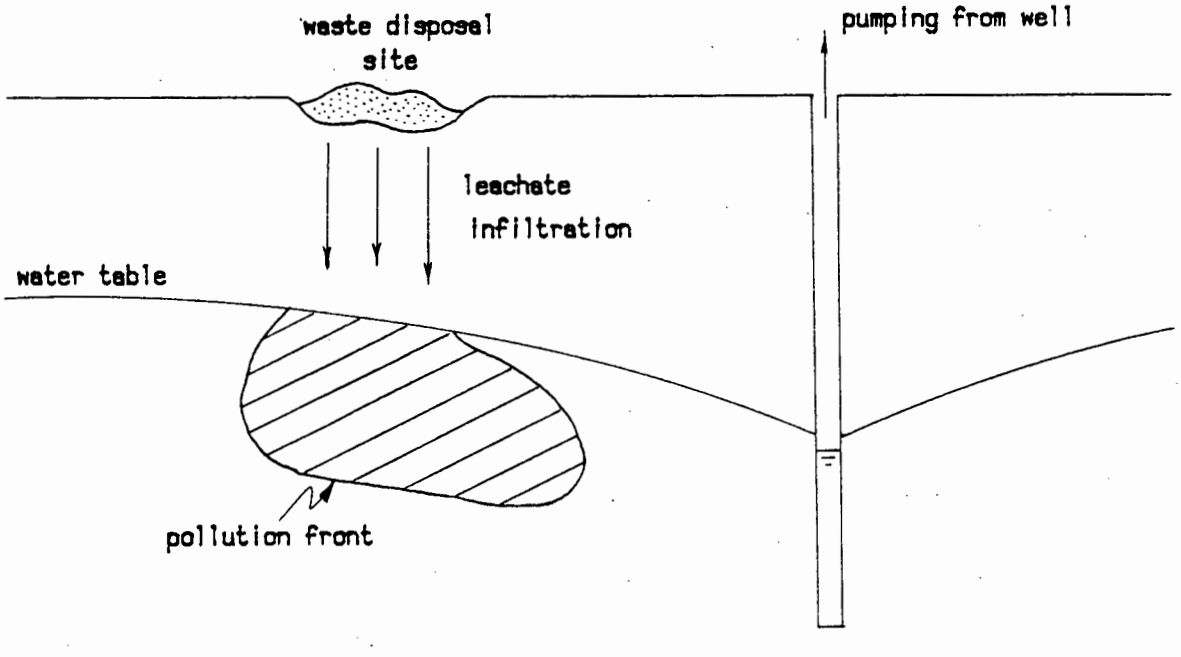


Figure 2.1 The formation and travel of leachate

liquids such as rainwater infiltrate the landfill and dissolve the soluble wastes. Leachates often have a high concentration of organic substances, bacteria and viruses.

Governmental legislation, to a certain degree, control the disposal of wastes [5]. Such legislation controls the location of landfill sites (based on the permeability of the substrata, etc.), the treatment of the waste, the monitoring of the groundwater quality of nearby aquifers, and other facets in order to ensure a minimum pollution potential. However, in lower economic areas, legislation sometimes does not exist, and the pollution potential is increased.

Sewerage effluent disposal

The disposal or leakage of sewerage sludge, whether in the ocean or on land, could lead to the faecal contamination of groundwater. If this happens, then harmful pathogenic bacteria are existent within the groundwater. Besides bacteria, the sludge may contain contaminants such as nitrates and cadmium. The latter can also originate as metal wastes from mine workings and can cause kidney damage in humans.

Saline intrusion

Near the coast, an interface exists between the groundwater and the salt groundwater. This interface, although assumed to be sharp in mathematical models [33], consists of a transition zone from freshwater to salt-water. The zone may

sometimes have a thickness of up to 100m.

Excessive lowering of the water table, due to overpumping, will cause the salt-water front to travel landwards, as shown in figure 2.2. If the wedge-shaped salt-water body reaches the area of pumping of a well, then the quality of the pumped water will be affected. The situation could be remedied by allowing the aquifer to recharge, pushing the salt-water front seawards.

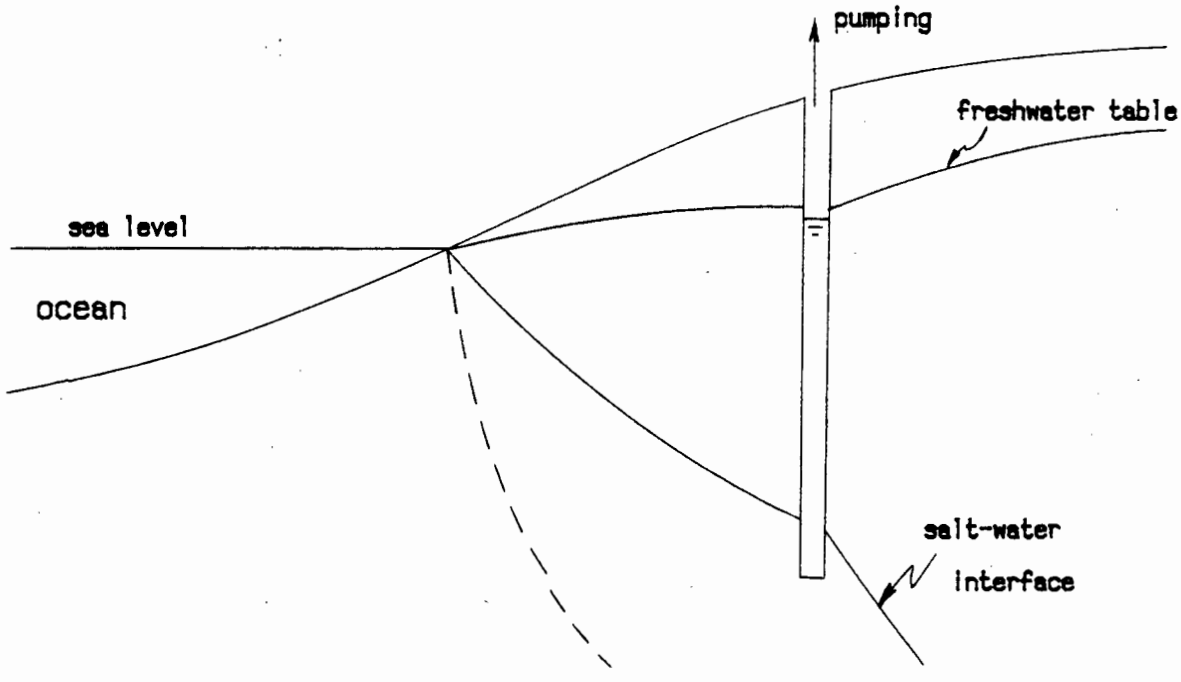


Figure 2.2 The salt-water interface in a coastal aquifer

Nitrate pollution

The use of synthetic nitrogenous or natural organic fertilizers for purposes of cultivation leads to the production of nitrates. This ion is neither adsorbed nor precipitated in the soil [1] and is therefore easily leached

by infiltrating water. Excessive amounts of nitrate, of the order of more than 50mg/l, can cause reduction of the bloodstream's oxygen-carrying capacity. Furthermore, cases of stomach cancer have been traced to high nitrate levels of drinking water.

2.2 Developments in groundwater contamination studies

France [27] studied the transient behaviour of the free surface of water in porous media. Applications were directed towards drainage problems rather than groundwater flow. The problem was solved by considering the solution to be a number of steady state problems at small intervals of time Δt apart.

A similar method was developed and applied by Desai [26]. Stability criteria were developed for the Finite Element transient free surface flow model. In addition, the characteristics and effects of non-Darcy flow and the flow of fluid in discontinuous porous media were fully discussed.

In applications on unconfined groundwater aquifers, the free surface can be considered steady state if the recharge boundary is at a constant potential and the pumping rates at wells are kept constant. Connor and Brebbia [22] were amongst those who analysed such steady state drawdown. Since the initial position of the free surface is not known beforehand an initial guess of the free surface is made. After each iteration, the values of the potential heads at

the free surface was compared to the elevation. If they were different, the Finite Element mesh was moved to satisfy the condition that the potential head at the free surface is equal to the elevation. This adjustment of the free surface renders the problem non-linear. The difficulty with variable mesh problems is that, after each adjustment, the stiffness matrices have to be recalculated. Nevertheless, Connor and Brebbia reported a convergence to within a tolerance of 0.1% difference between the potential head and the elevation after 5 iterations.

Taylor [14] modeled the problem in the exact same way, and reported that two iterations were required to obtain "satisfactory convergence". The aim here was two-fold. Firstly to solve the elemental velocities of the water in the aquifer and, secondly, to substitute these into the convection-diffusion equation to study the transport of a contaminant through the aquifer.

Before Taylor, many other investigators considered similar problems, that of groundwater degradation by a contaminating substance. Van Genuchten [9] used the Galerkin Finite Element approach to study the diffusion-dispersion equation. He used a time-centered Crank-Nicholson scheme, second order correct in time, to estimate the time derivative. Although the method, referred to as the dispersion corrected scheme, can easily be extended to two- and three-dimensional domains, he limited the applications to one dimensional problems. Linear, quadratic and cubic

basis functions were used in applications. The results, compared to analytical and finite difference methods, proved to be fairly accurate although a degree of oscillation was experienced. The higher order basis functions did not produce markedly better results than that of the linear basis functions. The increase in accuracy as opposed to the increase in computational time is not justified. Minimal oscillations were found immediately upstream and downstream of the contaminant front when the contaminant front became steep. Steep contaminant fronts arise when the problem becomes convection dominated. Although it may seem that this is due to the increasing non-symmetrical nature of the stiffness matrix, he ascribed this to the approximation of the time derivative. A major shortcoming of the study was that the velocity within the domain was assumed to be constant. This would render the problem impractical in more complex three-dimensional aquifers where the components of velocity may vary with position.

Ehlig [7] concentrated on groundwater contamination modeling in two-dimensions. The groundwater flow equation was first solved to find the fluid velocities. These were then substituted into the diffusion-convection equation and the nodal values of concentrations were solved. The object was to compare the degree of oscillations produced by the Finite Element method to that of the central and non-central finite difference methods used by past investigators. Oscillation criteria, based on the eigenvalues of the stiffness matrix

and the nodal spacing, were derived by numerical experimentation. These criteria showed that the Finite Element procedure required fewer nodes and fewer time-steps in order to contain the oscillations within a specified tolerance.

Segol [8] developed a three-dimensional groundwater contamination model for saturated-unsaturated porous media. The Galerkin procedure was used with isoparametric elements to obtain the coupled Finite Element approximation equations. The model was used to study one-dimensional infiltration problems. The results compared favourably with existing numerical and experimental data. In tests on two- and three-dimensional problems, a 40% increase in storage and 800% increase in computational time was reported. These constraints made the model too expensive to be applied to fully three-dimensional field situations.

The discharge of industrial and municipal waste effluents into the sea could lead to the contamination of the nearshore zone. This may in turn affect the aquatic environment even though the effluents, such as sewage, are biochemically and chemically treated. Such situations give rise to the propagation of contaminants with steep concentration gradients. Lam [31] developed a computational Finite Element procedure to simulate nearshore contamination. He was mainly concerned with the negative and oscillatory nature of solutions generated by finite difference methods. The one-dimensional diffusion

convection equation was solved with the fluid medium's velocities assumed to be uniform. Results compared favourably with analytical solutions, exhibiting a much lesser degree of oscillation compared with that of finite difference methods.

In all the work discussed so far, it was shown that the Finite Element method gives acceptable results for diffusion dominated flow problems, but when the problem becomes convection dominated, the results exhibit oscillations and negative oscillations, even in one dimensional problems. A reduction of the time step or a reduction of the nodal spacing is not sufficient to eliminate such effects [9], [30]. Neither higher order finite elements [9] nor higher order integration schemes [31] remedy the oscillations in convection dominated flow problems. Varoglu and Finn [11] and Jensen and Finlayson [12] introduced a novel method. The diffusion-convection equation was solved using a moving coordinate system (MCS). By allowing the domain to travel at the same speed as the velocity of the medium, the diffusion-convection equation was reduced to the simple diffusion form. The effect of the convection term is included by the altered geometry of the region. The results obtained from the Galerkin Finite Element method [11] and a finite difference method [12] showed that the MCS method eliminates oscillations even for relatively large nodal spacing. In both cases, however, the problem was not coupled to the groundwater flow problem and only the one-

dimensional case was examined. It seems that the method will break down if different elements within the domain have different values of velocities, especially in multi-dimensional problems where the velocity vectors are displayed as components. This would mean that the local coordinate axes of each element could alter to a different degree, affecting the global continuity requirements.

The most useful model, by far, was that developed by Taylor [14]. The Galerkin Finite Element method was used to solve the coupled groundwater flow - diffusion-convection equations in three dimensions. A computer program was coded and applied to the case of three-dimensional unconfined aquifer. The domain, discretised into 368 brick elements and 500 nodes, included a constant contaminant source and a well pumped at a constant rate. Two iterations were required to obtain a "satisfactory convergence" of the initial free surface. The results conformed intuitively, but no analytical or experimental data was available to prove its validity.

More recently, Nwaogazie [32] presented a two-dimensional computational model called SOTRAN, developed for the IBM System 360. The option of either linear or quadratic isoparametric quadrilateral elements for use with the Galerkin Finite Element model was provided. The model included the effects due to adsorption, biodegradation (applicable to the transport of effluent), volatilization, precipitation, and radio-active decay.

In the coupled problem, the groundwater-flow equation was simplified using the Dupuit Approximation. This approximation neglects the seepage surface in drawdown problems.

Chapter 3

The development of the Finite Element model

3.1 The mathematical model

The problem of mass transport, or contaminant transport, in a porous medium is governed by the diffusion-convection equation. The three dimensional form of this equation may be expressed as follows

$$\frac{1}{n}\frac{\partial c}{\partial t} - \left[D_X\frac{\partial^2 c}{\partial x^2} + D_Y\frac{\partial^2 c}{\partial y^2} + D_Z\frac{\partial^2 c}{\partial z^2}\right] + \left[v_X\frac{\partial c}{\partial x} + v_Y\frac{\partial c}{\partial y} + v_Z\frac{\partial c}{\partial z}\right] = 0 \qquad (3.1)$$

where c is the concentration of the solute and n is the effective porosity of the porous medium. D_X, D_Y and D_Z are the diffusion coefficients in the different directions, being equal in the case of an isotropic medium. The velocity components, v_X, v_Y and v_Z, give rise to the convective term in the equation.

In order to solve equation (3.1), the boundary conditions as well as the initial conditions (time-dependent problem) have to be specified. Two types of boundary conditions may prevail here. Firstly, the value of the concentration at

the boundary Γ_1 may be specified as $\varsigma(x, y, z, t)$, giving

$$c - \varsigma = 0 \qquad \text{on } \Gamma_1 \tag{3.2}$$

A boundary condition of this form is commonly referred to as a Dirichlet, or essential, boundary condition.

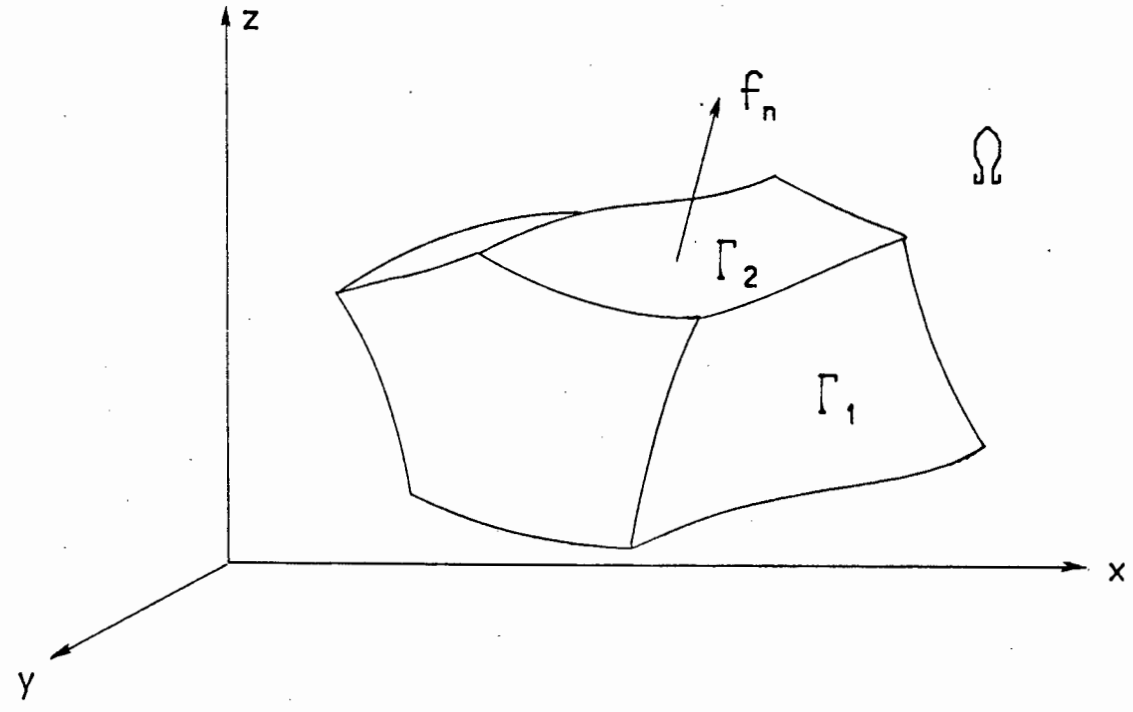


Figure 3.1 An arbitrary domain of the problem.

Secondly, the values of the concentration gradient, or outward flux, normal to the boundary Γ_2 may be specified as f_n, giving

$$\left[D_x \frac{\partial c}{\partial x} \nu_x + D_y \frac{\partial c}{\partial y} \nu_y + D_z \frac{\partial c}{\partial z} \nu_z \right] + f_n = 0 \qquad \text{on } \Gamma_2 \tag{3.3}$$

where ν_x, ν_y and ν_z are the direction cosines in the x, y

and z directions respectively. A boundary condition of this form is commonly referred to as a Neumann, or natural, boundary condition.

If a zero Dirichlet boundary condition prevails, i.e. $\varphi(x, y, t) = 0$, then the flux normal to that boundary is also zero, i.e. $f(x, y, t) = 0$. Such a boundary is called an impermeable boundary. If, on the other hand, a non-zero Dirichlet boundary condition is specified, then the flux normal to that boundary is also non-zero, except at time t=0, and varies until steady-state conditions are reached. In this case, the Neumann boundary condition becomes a natural boundary condition, and the unknown solved by the system is the value of $f(x,y,t)$ since the concentration c is known. As was mentioned previously, the initial values of the concentration, i.e. $c(x,y,0)$, must also be specified.

The velocity components v_x, v_y and v_z depend on the potential gradients that exist within the domain, and may vary from element to element. These components can be found by solving the groundwater flow equation which, for a three dimensional domain, may be expressed as follows:

$$S_s \frac{\partial h}{\partial t} = \left[k_x \frac{\partial^2 h}{\partial x^2} + k_y \frac{\partial^2 h}{\partial y^2} + k_z \frac{\partial^2 h}{\partial z^2} \right] - Q \qquad (3.4)$$

where h is the fluid potential head in metres. S_s is the specific storage, defined as the volume discharge per unit

volume of aquifer per unit decrease in potential head, and k_x, k_y and k_z are the coefficients of permeability or hydraulic conductivity. Q is the discharge, or sometimes referred to as the internal fluid flux, defined as the volume rate of discharge per unit volume of aquifer.

Figure 3.2 represents the type of domain for which equation (3.4) will be solved. It shows a cross-section of an unconfined aquifer which may be pumped by wells resulting in a drawdown of the free surface or water table.

A number of different boundary conditions may prevail. The boundary Γ_1 represents an impermeable or confined boundary, i.e. a boundary where no fluid passes through. The fluid flux normal to the boundary, q_n, is zero. The boundary Γ_2 is referred to as a constant head boundary, the potential

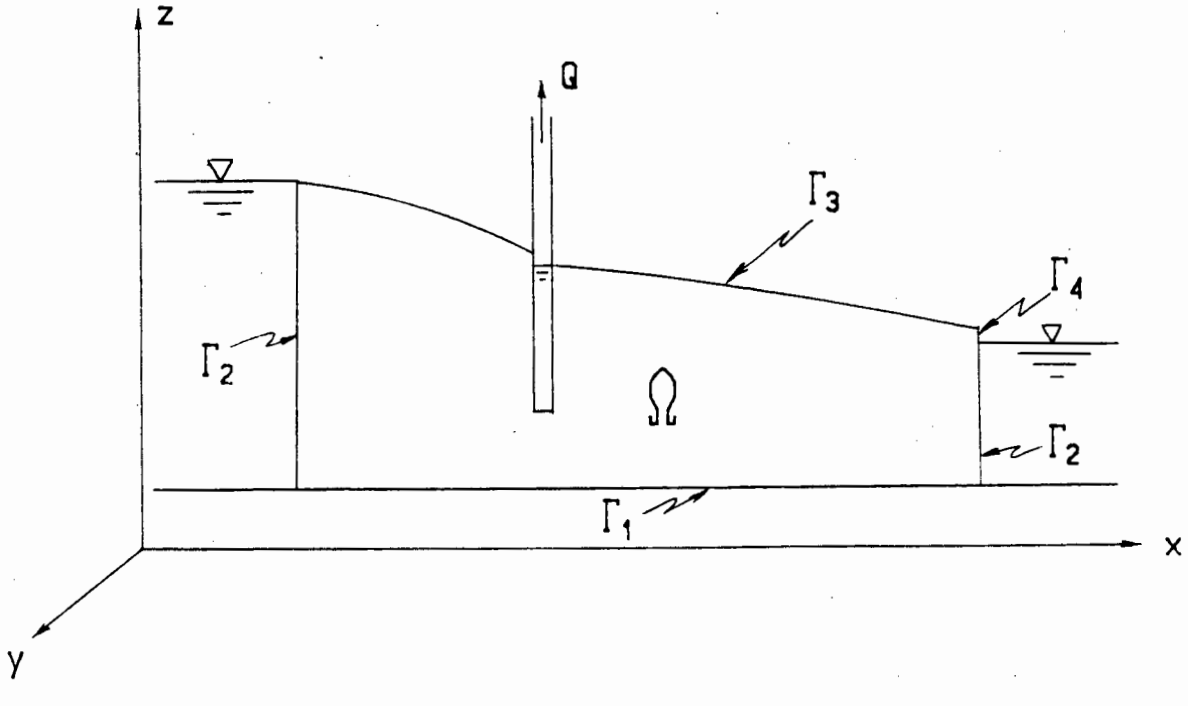


Figure 3.2 A flow domain of an unconfined aquifer.

head being specified as follows

$$h = \bar{h} \qquad \text{on } \Gamma_2 \tag{3.5}$$

This is an equipotential line. On this boundary, the fluid flux q_n, given by

$$\left[K_x\frac{\partial h}{\partial x}\nu_x + K_y\frac{\partial h}{\partial y}\nu_y + K_z\frac{\partial h}{\partial z}\nu_z\right] + q_n = 0 \tag{3.6}$$

is variable under unsteady conditions, otherwise constant.

The free surface boundary Γ_3 is representative of a streamline, the potential head being equal to the elevation head at that point, that is

$$h = \bar{z} \qquad \text{on } \Gamma_2 \tag{3.7}$$

Similar to an impermeable boundary, the fluid flux is zero.

The seepage surface, Γ_4, is neither a streamline nor an equipotential line. At this boundary, fluid seeps out into the atmosphere. It will, however, be assumed to be an impermeable boundary.

In addition to the above boundary conditions, the initial condition h(x,y,z,0) need also be specified. In the case of the free surface, the initial condition will have to be guessed as will be seen later.

Equations (3.1) and (3.4) have to be solved simultaneously. Equation (3.4) must be solved first in order to establish the potential heads, and hence the velocity components using Darcy's Law

$$v_x = - k_x \frac{\partial h}{\partial x} \qquad (3.8)$$

These components can then be substituted into equation (3.1) to solve for the concentrations. The problem is therefore said to be non-linear.

The mathematical model described above may also be used to solve problems such as:

(i) seepage through dam walls

(ii) seepage through confined porous media such as confined aquifers

(iii) contaminant transport in non-porous media, such as canals, where the porosity is unity.

3.2 The derivation of the Finite Element equations

A detailed account of the theory of functional analysis and finite element analysis will not be given here. The reader is referred to Reddy and Rasmussen [20], Zienkiewicz and Morgan [15], and Carey and Oden [21].

The diffusion-convection differential equation is non-self-adjoint [16] [13] because of the presence of the first order spatial derivatives. There is no energy formulation equivalent for non-self-adjoint differential equations. Since the Galerkin approach to the Finite Element method can be used with all types of equations, this approach will be used to develop the Finite Element forms of the diffusion-convection and the groundwater flow equations. Furthermore, it should be noted that non-self-adjoint differential equations will lead to unsymmetrical stiffness matrices.

Groundwater Flow Equation

Multiplying equation (3.4) by a function ψ, called a test function, and integrating over the domain,

$$\int_{\Omega} \left[K_x \frac{\partial^2 h}{\partial x^2} + K_y \frac{\partial^2 h}{\partial y^2} + K_z \frac{\partial^2 h}{\partial z^2} \right] \psi \, d\Omega = \int_{\Omega} \left[S_s \frac{\partial h}{\partial t} + Q \right] \psi \, d\Omega \qquad (3.9)$$

where Ω is the domain of integration. Integrating the first term by parts gives

$$K_x \int_{\Omega} \frac{\partial^2 h}{\partial x^2} \psi \, d\Omega = K_x \int_{\Gamma} \frac{\partial h}{\partial x} \nu_x \psi \, d\Gamma - K_x \int_{\Omega} \frac{\partial h}{\partial x} \frac{\partial \psi}{\partial x} \, d\Omega$$

$$K_y \int_\Omega \frac{\partial^2 h}{\partial y^2} \psi \, d\Omega = K_y \int_\Gamma \frac{\partial h}{\partial y} \nu_y \psi \, d\Gamma - K_y \int_\Omega \frac{\partial h}{\partial y} \frac{\partial \psi}{\partial y} \, d\Omega$$

$$K_z \int_\Omega \frac{\partial^2 h}{\partial z^2} \psi \, d\Omega = K_z \int_\Gamma \frac{\partial h}{\partial z} \nu_z \psi \, d\Gamma - K_z \int_\Omega \frac{\partial h}{\partial z} \frac{\partial \psi}{\partial z} \, d\Omega$$

where ν_x, ν_y and ν_z are the direction cosines in the x, y and z directions respectively. The reason for integrating by parts is to reduce the order of the differential equation so that the test/trial functions, to be introduced later, have relaxed continuity and integrability requirements. Equation (3.9) thus becomes

$$\int_\Omega \left[K_x \frac{\partial h}{\partial x} \frac{\partial \psi}{\partial x} + K_y \frac{\partial h}{\partial y} \frac{\partial \psi}{\partial y} + K_z \frac{\partial h}{\partial z} \frac{\partial \psi}{\partial z} \right] d\Omega - \int \left[K_x \frac{\partial h}{\partial x} \nu_x + K_y \frac{\partial h}{\partial y} \nu_y + K_z \frac{\partial h}{\partial z} \nu_z \right] \psi \, d\Gamma$$

$$+ \int_\Omega \left[S_s \frac{\partial h}{\partial t} + Q \right] \psi \, d\Omega = 0 \qquad (3.10)$$

which will be used to approximate the solution. At all points in Ω, the solution $h(x,y,t)$ can be approximated by

$$h(x,y,z,t) = \sum_{j=1}^{n} h_j(t) \, \phi_j(x,y,z) \qquad (3.11)$$

where ϕ_j, $j=1,2,\ldots,n$, is a linearly independent set of trial, or basis, functions defined for an individual element such that the function is unity at node j and zero at all the other nodes. An arbitrary three-dimensional element is shown in figure 3.3. The coefficients h_j, $j=1,2,\ldots,n$, are the approximations of the values of potential heads at the various nodes of the element. These coefficients will be solved in the Finite Element procedure.

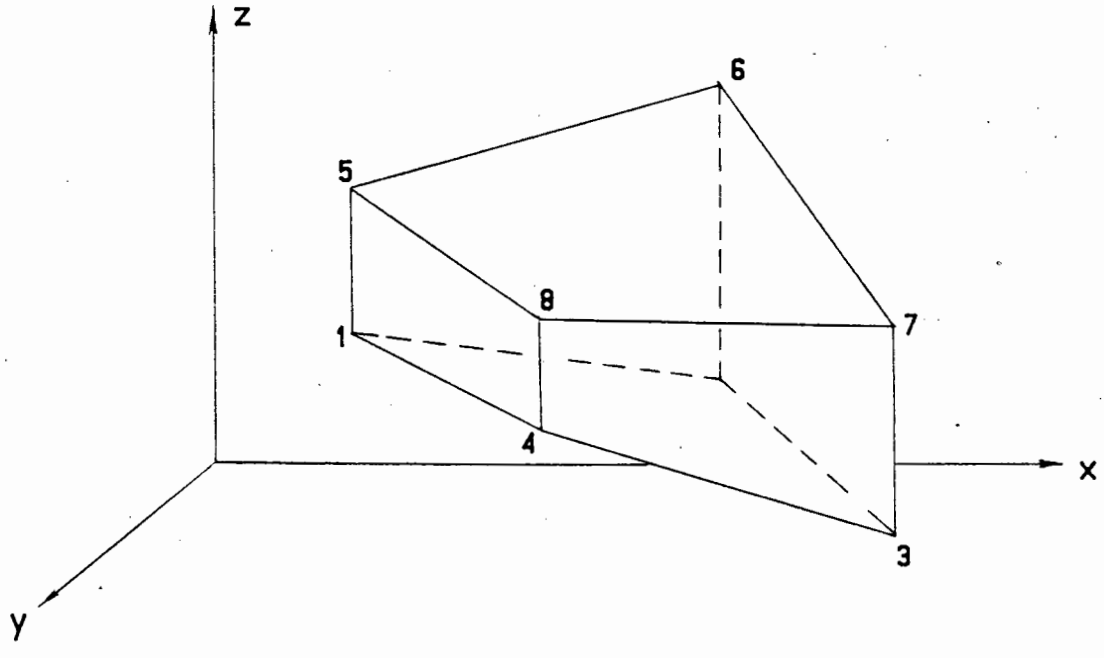


Figure 3.3 An arbitrary three-dimensional finite element.

A feature of the Galerkin method is that the test functions, or weighting functions, are chosen to be the same as the trial functions. That is

$$\psi_i(x,y) = \phi_i(v,y) \qquad i=1,2,\ldots,n \qquad (3.12)$$

It can be noted at this stage that the Neumann boundary condition, as stated in equation (3.6) can be substituted, together with equations (3.11) and (3.12), into equation (3.10) to give

$$h_i \int_\Omega \left[k_x \frac{\partial \phi_i}{\partial x} \frac{\partial \phi_j}{\partial x} + k_y \frac{\partial \phi_i}{\partial y} \frac{\partial \phi_j}{\partial y} + k_z \frac{\partial \phi_i}{\partial z} \frac{\partial \phi_j}{\partial z} \right] d\Omega + \int_\Gamma q_n \phi_j \, d\Gamma$$

$$+ \int_\Omega \left[S_s \frac{\partial h_i}{\partial t} \phi_i \phi_j + Q \phi_j \right] d\Omega = 0 \qquad (3.13)$$

Various methods can be used to approximate the time derivative. Finite difference methods have been used successfully by other investigators to solve transient problems. The forward difference (Euler) scheme, the central difference (Crank-Nicholson) scheme and the backward difference scheme are the most commonly used finite difference methods. With these types of approximations of the time derivative, the Finite Element method process results in a system of linear equations which can be easily solved.

On the other hand, if a finite difference method is not employed, then the Finite Element process generates a system of first order differential equations, which can be solved by the eigenvalue method.

Here, the backward difference scheme will be used. This results in the following approximations:

$$h_i = h_i{}^{t+\Delta t} \tag{3.14a}$$

$$\frac{\partial h_i}{\partial t} = \frac{h_i{}^{t+\Delta t} - h_i{}^{t}}{\Delta t} \tag{3.14b}$$

Equation (3.13) now becomes

$$h_i{}^{t+\Delta t}\int_\Omega \left[k_x \frac{\partial \phi_i}{\partial x}\frac{\partial \phi_j}{\partial x} + k_y \frac{\partial \phi_i}{\partial y}\frac{\partial \phi_j}{\partial y} + k_z \frac{\partial \phi_i}{\partial z}\frac{\partial \phi_j}{\partial z}\right] d\Omega + \int_\Gamma q_n \phi_j \, d\Gamma$$

$$+ \frac{h_i{}^{t+\Delta t} - h_i{}^{t}}{\Delta t}\int_\Omega S_s \frac{\partial h_i}{\partial t}\phi_i \phi_j \, d\Omega + \int_\Omega Q \phi_j \, d\Omega = 0$$

which may be rearranged to give

$$h_i{}^{t+\Delta t}\int_\Omega \left[\left[k_x \frac{\partial \phi_i}{\partial x}\frac{\partial \phi_j}{\partial x} + k_y \frac{\partial \phi_i}{\partial y}\frac{\partial \phi_j}{\partial y} + k_z \frac{\partial \phi_i}{\partial z}\frac{\partial \phi_j}{\partial z}\right] + \left[S_s \phi_i \phi_j\right]\frac{1}{\Delta t}\right] d\Omega$$

$$+ \int_\Omega Q \phi_j \, d\Omega + \int_\Gamma q_n \phi_j \, d\Gamma = h_i{}^{t}\int_\Omega \left[S_s \phi_i \phi_j\right] d\Omega \tag{3.15}$$

As was mentioned previously, ϕ_j, $j=1,2,\ldots,n$, represents a linearly independent set of basis functions. This set of basis functions can be written, for a three-dimensional cubic or brick element, in matrix form as

$$[\phi]^T = \left[\phi_1 \ \phi_2 \ \phi_3 \ \phi_4 \ \phi_5 \ \phi_6 \ \phi_7 \ \phi_8 \right] \qquad (3.16)$$

where $[\phi]^T$ is the transpose of $[\phi]$. Note that for a cubic element there are eight nodes. Equation (3.16) therefore applies to the eight-noded cubic element, each basis function having a value of unity at one particular node and a value of zero elsewhere.

The integrations in equation (3.15) is performed nxn times per element (n=number of nodes). This is more clearly seen if equation (3.16) is substituted, as shown below, into equation (3.15) resulting in a global Finite Element formulation.

$$\underset{\sim}{h}^{t+\Delta t}\int_{\Omega}\left[\left[k_x[\phi_x][\phi_x]^T + k_y[\phi_y][\phi_y]^T + k_z[\phi_z][\phi_z]^T\right] + \frac{1}{\Delta t}\left[S_s[\phi][\phi]^T\right]\right]$$

$$+ \int_{\Omega} Q[\phi]\ d\Omega + \int_{\Gamma} q_n[\phi]\ d\Gamma = \underset{\sim}{h}^{t}\int_{\Omega}\frac{1}{\Delta t}\left[S_s[\phi][\phi]^T\right]\ d\Omega \qquad (3.17)$$

where $[\phi_x]$, $[\phi_y]$ and $[\phi_z]$ are the derivative matrices of $[\phi]$

with respect to x, y and z respectively, as shown below. Also, $\underset{\sim}{h}^{t+\Delta t}$ and $\underset{\sim}{h}^{t}$ are the concentrations vectors at two adjacent time steps.

$$[\phi_x]^T = \left[\frac{\partial \phi_1}{\partial x} \; \frac{\partial \phi_2}{\partial x} \; \frac{\partial \phi_3}{\partial x} \; \frac{\partial \phi_4}{\partial x} \; \frac{\partial \phi_5}{\partial x} \; \frac{\partial \phi_6}{\partial x} \; \frac{\partial \phi_7}{\partial x} \; \frac{\partial \phi_8}{\partial x} \right]$$

$$[\phi_y]^T = \left[\frac{\partial \phi_1}{\partial y} \; \frac{\partial \phi_2}{\partial y} \; \frac{\partial \phi_3}{\partial y} \; \frac{\partial \phi_4}{\partial y} \; \frac{\partial \phi_5}{\partial y} \; \frac{\partial \phi_6}{\partial y} \; \frac{\partial \phi_7}{\partial y} \; \frac{\partial \phi_8}{\partial y} \right]$$

$$[\phi_z]^T = \left[\frac{\partial \phi_1}{\partial z} \; \frac{\partial \phi_2}{\partial z} \; \frac{\partial \phi_3}{\partial z} \; \frac{\partial \phi_4}{\partial z} \; \frac{\partial \phi_5}{\partial z} \; \frac{\partial \phi_6}{\partial z} \; \frac{\partial \phi_7}{\partial z} \; \frac{\partial \phi_8}{\partial z} \right] \qquad (3.18)$$

Equation (3.17) is the equivalent Finite Element form of the diffusion-convection equation which, as will be seen later, generates a system of linear equations of the form

$$\{[A] + [B]/\Delta t\}\underset{\sim}{h}^{t+\Delta t} = \{[B]/\Delta t\}\underset{\sim}{h}^{t} - \underset{\sim}{Q} - \underset{\sim}{g} \qquad (3.19)$$

where $$[A] = \int_{\Omega} \left[k_x [\phi_x][\phi_x]^T + k_y [\phi_y][\phi_y]^T + k_z [\phi_z][\phi_z]^T \right] d\Omega$$

$$[B] = \int_{\Omega} S_s [\phi][\phi]^T \, d\Omega$$

$$\underset{\sim}{Q} = \int_{\Omega} Q[\phi] \, d\Omega$$

$$\underset{\sim}{q} = \int_{\Gamma} q_n[\phi] \, d\Gamma \qquad (3.20)$$

The matrices [A] and [B] are symmetrical. The specified Dirichlet, or essential, boundary conditions are introduced directly into equation (3.19) in the potential heads vectors $\underset{\sim}{h}^{t+\Delta t}$ and $\underset{\sim}{h}^{t}$, while the specified Neumann, or natural, boundary conditions are introduced in equation (3.13) as was shown earlier. For a specified Dirichlet boundary condition, the unknown in equation (3.17) becomes the flux qn, or potential gradient, at the boundary. In other words, the unknowns at the boundary can occur in the potential heads vectors or the flux vector $\underset{\sim}{q}$ depending on the boundary conditions specified.

In the problem described earlier, the elemental velocities rather than the potential heads are of interest. Thus, by using Darcy's Law as stated in equation (3.8), the following relationships can be written

$$v_x = - k_x[\phi_x] \underset{\sim}{h}$$

$$v_y = - k_y[\phi_y] \underset{\sim}{h}$$

$$v_z = - k_z[\phi_z] \underset{\sim}{h} \qquad (3.21)$$

per element.

Darcy's law is generally considered valid for flows with values of Reynolds number, Re, less than 1. That is

$$R_e = \frac{vdR}{\mu} < 1$$

where v is the fluid velocity, d the average pore diameter, R the fluid density and μ the fluid viscosity. This implies that the the flow is laminar and that inertial forces are negligible compared to the viscous forces. This can be assumed to occur in media with particle size less than 1mm [25]. Examples of such media are clayey and silty soils, and fine to medium sands. For flows with higher values of Reynolds number, it may be necessary to use non-linear, or non-Darcy, laws. Two such non-Darcy laws commonly used are Forchheimer's law and Misbach's law, given by

$$h = av + bv^2$$

and

$$h = cv^n$$

respectively. The constants a,b,c and n are properties of the medium and can be determined from permeameter tests.

Diffusion-Convection Equation

In the Galerkin Finite Element formulation of the diffusion-convection equation, the dispersion corrected scheme of Van Genuchten [9] will be used. In this scheme, the time derivative of equation (3.1) is approximated by the equation

$$\frac{c^{t+\Delta t}-c^{t}}{\Delta t} = \frac{1}{2}\left[\frac{\partial c}{\partial t} - \frac{\Delta t}{6}\frac{\partial^2 c}{\partial t^2}\right]^{t+\Delta t} + \frac{1}{2}\left[\frac{\partial c}{\partial t} + \frac{\Delta t}{6}\frac{\partial^2 c}{\partial t^2}\right]^{t} \qquad (3.22)$$

which can be considered as a time centered Crank-Nicholson finite difference scheme with a correction factor applied to the dispersion coefficient, the value of the dispersion correction being dependent upon the time interval.

After the substitution of the time derivatives and the application of the Galerkin Finite Element method the formulation shown as equation (3.23) is derived.

$$\underset{\sim}{c}^{t+\Delta t}\int_{\Omega} n/2\Big[D_{xx1}[\phi_x][\phi_x]^T + D_{yy1}[\phi_y][\phi_y]^T + D_{zz1}[\phi_z][\phi_z]^T$$

$$+ 2D_{xy1}[\phi_x][\phi_y]^T + 2D_{xz1}[\phi_x][\phi_z]^T + 2D_{yz1}[\phi_y][\phi_z]^T$$

$$+ v_x[\phi_x][\phi]^T + v_y[\phi_y][\phi]^T + v_z[\phi_z][\phi]^T + 2/n[\phi][\phi]^T/\Delta t\Big]\, d\Omega$$

$$= \underset{\sim}{c}^{t}\int_{\Omega} n/2\Big[- D_{xx2}[\phi_x][\phi_x]^T - D_{yy2}[\phi_y][\phi_y]^T - D_{zz2}[\phi_z][\phi_z]^T$$

$$- 2D_{xy2}[\phi_x][\phi_y]^T - 2D_{xz2}[\phi_x][\phi_z]^T - 2D_{yz2}[\phi_y][\phi_z]^T$$

$$- v_x[\phi_x][\phi]^T - v_y[\phi_y][\phi]^T - v_z[\phi_z][\phi]^T + 2/n[\phi][\phi]^T/\Delta t\Big]\, d\Omega$$

$$- \int_{\Gamma} n f_n[\phi]\, d\Gamma \qquad (3.23)$$

For a detailed discussion on the above derivation, the reader may refer to APPENDIX A. The system of linear equations generated can be expressed as follows

$$\tfrac{1}{2}\{[L1]+[M]+2[N]/\Delta t\}\underset{\sim}{c}^{t+\Delta t} = \tfrac{1}{2}\{-[L1]-[M]+2[N]/\Delta t\}\underset{\sim}{c}^{t} - \underset{\sim}{g}$$

..... (3.24)

$$\text{where} \quad [L1] = \int_{\Omega} n\Big[D_{xx1}[\phi_x][\phi_x]^T + D_{yy1}[\phi_y][\phi_y]^T + D_{zz1}[\phi_z][\phi_z]^T$$

$$+ 2D_{xy1}[\phi_x][\phi_y]^T + 2D_{xz1}[\phi_x][\phi_z]^T + 2D_{yz1}[\phi_y][\phi_z]^T\Big]\, d\Omega$$

$$[L2] = \int_{\Omega} n\Big[D_{xx2}[\phi_x][\phi_x]^T + D_{yy2}[\phi_y][\phi_y]^T + D_{zz2}[\phi_z][\phi_z]^T$$

$$+ 2D_{xy2}[\phi_x][\phi_y]^T + 2D_{xz2}[\phi_x][\phi_z]^T + 2D_{yz2}[\phi_y][\phi_z]^T\Big]\, d\Omega$$

$$[M] = \int_{\Omega} n\Big[v_x[\phi_x][\phi]^T + v_y[\phi_y][\phi]^T + v_z[\phi_z][\phi]^T\Big]\, d\Omega$$

$$[N] = \int_{\Omega} [\phi][\phi]^T\, d\Omega$$

$$\underset{\sim}{g} = \int_{\Gamma} n f_n[\phi]\, d\Gamma \qquad (3.25)$$

Equations (3.25) contains the corrected dispersion coefficients given in APPENDIX A. Any specified Dirichlet, or essential, boundary conditions are introduced directly into equation (3.24) in the concentrations vectors $\underset{\sim}{c}^{t+\Delta t}$ and $\underset{\sim}{c}^{t}$. For a specified Dirichlet boundary condition, the unknown in equation (3.24) becomes the flux, or concentration gradient, at the boundary.

The stiffness matrices are unsymmetrical because of the presence of the convection term. In previous studies, this has lead to oscillations in the solution when the convection terms became dominant.

Equations (3.19) and (3.24) constitute two separate systems of algebraic equations which can be solved sequentially for each time step. The system is non-linear due to the unknown free surface in the case of an unconfined aquifer. This requires iteration within a time step until the nodal values of equation (3.19) converge to a prescribed tolerance.

During each time step, the nodal values of potential head at the previous time step are used to compute the nodal values of potential head at the present time step using equation (3.19). These nodal potential heads are then used to calculate the elemental velocities using equations (3.21). The velocity components and the nodal values of concentration are substituted into equation (3.24) and the nodal values of concentration at the present time step are computed.

In the case of unconfined aquifers, the position of the free surface is initially guessed. However, the condition that the potential head must equal the elevation of the free surface must be satisfied. After the time step, this condition is checked. If the difference exceeds a prescribed tolerance, then a further iteration is performed, until the solution converges to a desired accuracy.

3.3 Finite Element interpolation and discretization

Consider an arbitrary, irregular domain, as shown in figure 3.4. The continuum is discretized into a number of linear brick elements. The discretized body may only approximate the irregular shaped domain, since the latter may constitute curved boundaries. However, it should be noted that the inclusion of curved boundaries in a finite element formulation poses no great difficulty, except from a computational point of view. Also, for large domains, such as aquifers, such accuracy in the discretization is not warranted, as may be the case with, say, an aerofoil boundary.

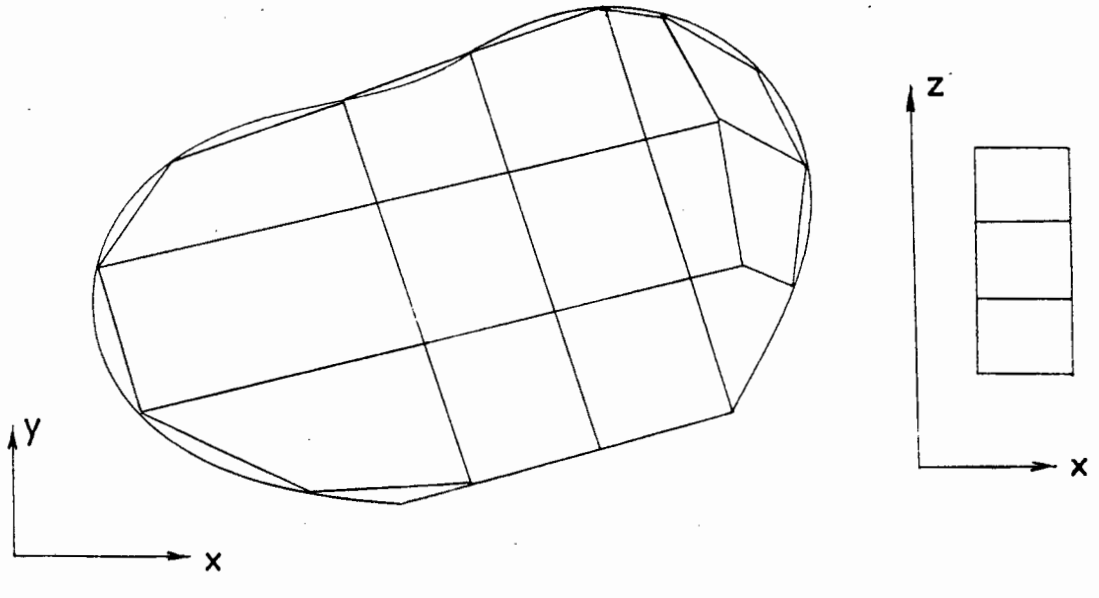


Figure 3.4 Discretization of an irregular shaped domain.

Lets now apply the finite element formulations on a typical brick element which is also chosen to be a boundary element.

It is geometrically convenient to perform the integration on the element along its own local axes and then to make use of transformation matrices to transform the elemental axes to the global axes. Such a transformation is called an isoparametric transformation and is depicted in figure 3.5.

The integration is performed on the isoparametric element along the local ξ-η-ω axes and then transformed back to the global x-y-z axes by means of a determinant called the Jacobian. Isoparametric elements are widely used and a feature of these elements is that their geometry is interpolated in exactly the same way as the unknown function is interpolated. The reader may refer to Connor and Brebbia [22] for further information regarding isoparametric elements.

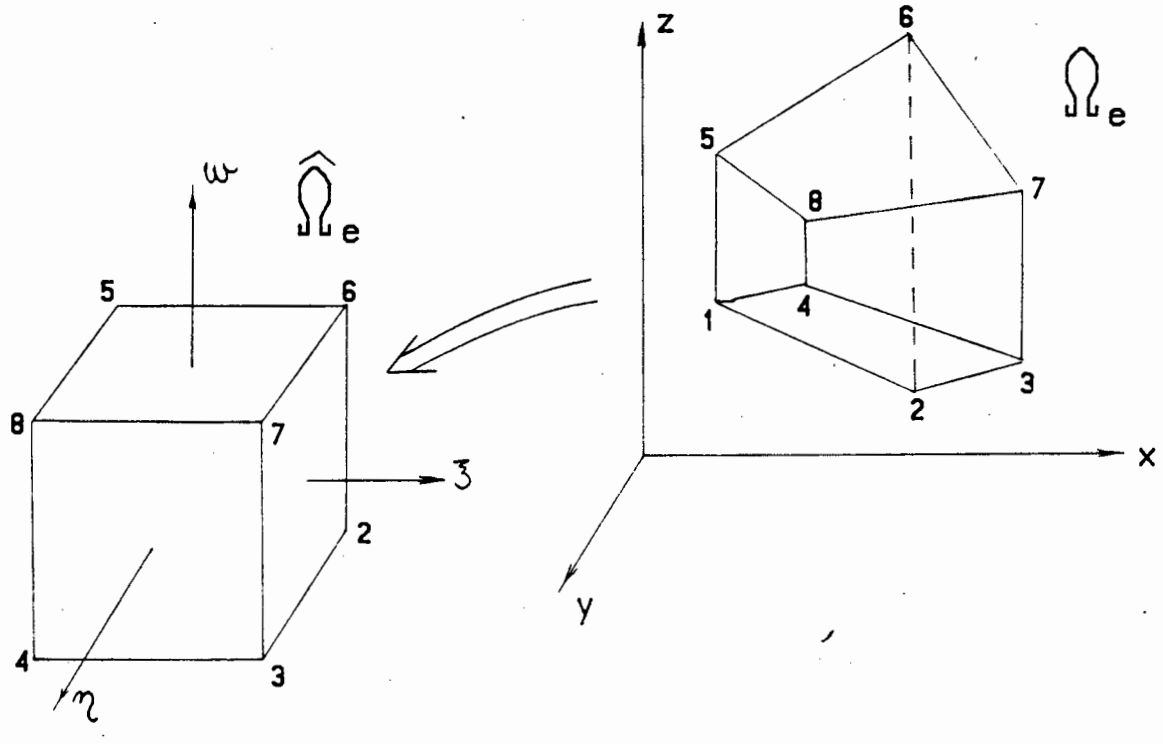


Figure 3.5 The isoparametric transformation.

For the isoparametric element, the following approximations therefore hold:

$$c = \sum_{j=1}^{n} \phi_j \, c_j = [\phi]^T \, \underset{\sim}{c} \qquad (3.26a)$$

$$x = \sum_{j=1}^{n} \phi_j \, x_j = [\phi]^T \, \underset{\sim}{x} \qquad (3.26b)$$

$$y = \sum_{j=1}^{n} \phi_j \, y_j = [\phi]^T \, \underset{\sim}{y} \qquad (3.26c)$$

$$z = \sum_{j=1}^{n} \phi_j \, z_j = [\phi]^T \, \underset{\sim}{z} \qquad (3.26d)$$

where the interpolation functions, ϕ_j, which are usually Lagrange polynomials, are now written for the transformed element. Due to this, the elements of the derivative matrices $[\phi_x]$, $[\phi_y]$ and $[\phi_z]$ cannot be found directly since they are functions of ξ, η and ω. The infinitesimals $d\xi$, $d\eta$ and $d\omega$ transform into dx, dy and dz according to the chain rule as follows

$$dx = \frac{\partial x}{\partial \xi} d\xi + \frac{\partial x}{\partial \eta} d\eta + \frac{\partial x}{\partial \omega} d\omega \qquad (3.27a)$$

$$dy = \frac{\partial y}{\partial \xi} d\xi + \frac{\partial y}{\partial \eta} d\eta + \frac{\partial y}{\partial \omega} d\omega \qquad (3.27b)$$

$$dz = \frac{\partial z}{\partial \xi} d\xi + \frac{\partial z}{\partial \eta} d\eta + \frac{\partial z}{\partial \omega} d\omega \qquad (3.27c)$$

or, in matrix form

$$\begin{bmatrix} dx \\ dy \\ dz \end{bmatrix} = \begin{bmatrix} \partial x/\partial \xi & \partial x/\partial \eta & \partial x/\partial \omega \\ \partial y/\partial \xi & \partial y/\partial \eta & \partial y/\partial \omega \\ \partial z/\partial \xi & \partial z/\partial \eta & \partial z/\partial \omega \end{bmatrix} \begin{bmatrix} d\xi \\ d\eta \\ d\omega \end{bmatrix}$$

$$= [J] \begin{bmatrix} d\xi \\ d\eta \\ d\omega \end{bmatrix} \qquad (3.28)$$

where [J] is the Jacobian matrix. For the transformation to be invertible, the Jacobian matrix must be non-singular. In other words, the determinant of the Jacobian matrix, simply referred to as the Jacobian, must be non-zero, that is

$$|J| = \frac{\partial x}{\partial \xi}\left[\frac{\partial y}{\partial \eta}\frac{\partial z}{\partial \omega} - \frac{\partial z}{\partial \eta}\frac{\partial y}{\partial \omega}\right] - \frac{\partial x}{\partial \eta}\left[\frac{\partial y}{\partial \xi}\frac{\partial u}{\partial \omega} - \frac{\partial u}{\partial \xi}\frac{\partial y}{\partial \omega}\right] + \frac{\partial x}{\partial \omega}\left[\frac{\partial y}{\partial \xi}\frac{\partial u}{\partial \eta} - \frac{\partial u}{\partial \xi}\frac{\partial y}{\partial \eta}\right] \neq 0$$

..... (3.29)

and it follows that

$$\begin{bmatrix} d\xi \\ d\eta \\ d\omega \end{bmatrix} = [J]^{-1} \begin{bmatrix} dx \\ dy \\ dz \end{bmatrix} \qquad (3.30)$$

where the inverse of the Jacobian matrix is given by

$$[J]^{-1} = \frac{1}{|J|}\begin{bmatrix} \left[\frac{\partial y}{\partial \eta}\frac{\partial u}{\partial \omega} - \frac{\partial u}{\partial \eta}\frac{\partial y}{\partial \omega}\right] & -\left[\frac{\partial x}{\partial \eta}\frac{\partial u}{\partial \omega} - \frac{\partial u}{\partial \eta}\frac{\partial x}{\partial \omega}\right] & \left[\frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \omega} - \frac{\partial y}{\partial \eta}\frac{\partial x}{\partial \omega}\right] \\ -\left[\frac{\partial y}{\partial \xi}\frac{\partial u}{\partial \omega} - \frac{\partial u}{\partial \xi}\frac{\partial y}{\partial \omega}\right] & \left[\frac{\partial x}{\partial \xi}\frac{\partial u}{\partial \omega} - \frac{\partial u}{\partial \xi}\frac{\partial x}{\partial \omega}\right] & -\left[\frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \omega} - \frac{\partial y}{\partial \xi}\frac{\partial x}{\partial \omega}\right] \\ \left[\frac{\partial y}{\partial \xi}\frac{\partial u}{\partial \eta} - \frac{\partial u}{\partial \xi}\frac{\partial y}{\partial \eta}\right] & -\left[\frac{\partial x}{\partial \xi}\frac{\partial u}{\partial \eta} - \frac{\partial u}{\partial \xi}\frac{\partial x}{\partial \eta}\right] & \left[\frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial y}{\partial \xi}\frac{\partial x}{\partial \eta}\right] \end{bmatrix}$$

..... (3.31)

Also, the infinitesimals dx, dy and dz transform into $d\xi$, $d\eta$ and $d\omega$ according to

$$\begin{bmatrix} d\xi \\ d\eta \\ d\omega \end{bmatrix} = \begin{bmatrix} \partial\xi/\partial x & \partial\xi/\partial y & \partial\xi/\partial z \\ \partial\eta/\partial x & \partial\eta/\partial y & \partial\eta/\partial z \\ \partial\omega/\partial x & \partial\omega/\partial y & \partial\omega/\partial z \end{bmatrix} \begin{bmatrix} dx \\ dy \\ dz \end{bmatrix} \qquad (3.32)$$

Equating equations (3.30) and (3.32) and making use of the inverse Jacobian matrix in equation (3.31), it follows that

$$\frac{\partial \xi}{\partial x} = \frac{1}{|J|}\left[\frac{\partial y}{\partial \eta}\frac{\partial u}{\partial \omega} - \frac{\partial u}{\partial \eta}\frac{\partial y}{\partial \omega}\right]$$

$$\frac{\partial \xi}{\partial y} = -\frac{1}{|J|}\left[\frac{\partial x}{\partial \eta}\frac{\partial u}{\partial \omega} - \frac{\partial u}{\partial \eta}\frac{\partial x}{\partial \omega}\right]$$

.
.
.

$$\frac{\partial \omega}{\partial z} = \frac{1}{|J|}\left[\frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \eta} - \frac{\partial y}{\partial \xi}\frac{\partial x}{\partial \eta}\right] \qquad (3.33)$$

Using the chain rule the derivatives of the basis functions ϕ_j can now be computed as follows

$$\frac{\partial \phi_j}{\partial x} = \frac{\partial \phi_j}{\partial \xi}\frac{\partial \xi}{\partial x} + \frac{\partial \phi_j}{\partial \eta}\frac{\partial \eta}{\partial x} + \frac{\partial \phi_j}{\partial \omega}\frac{\partial \omega}{\partial x}$$

$$\frac{\partial \phi_j}{\partial y} = \frac{\partial \phi_j}{\partial \xi}\frac{\partial \xi}{\partial y} + \frac{\partial \phi_j}{\partial \eta}\frac{\partial \eta}{\partial y} + \frac{\partial \phi_j}{\partial \omega}\frac{\partial \omega}{\partial y}$$

$$\frac{\partial \phi_j}{\partial z} = \frac{\partial \phi_j}{\partial \xi}\frac{\partial \xi}{\partial z} + \frac{\partial \phi_j}{\partial \eta}\frac{\partial \eta}{\partial z} + \frac{\partial \phi_j}{\partial \omega}\frac{\partial \omega}{\partial z} \qquad (3.34)$$

Since $d\Omega = dxdy$, by making use of equations (3.33), it can be deduced that

$$d\Omega = |J|\,d\xi d\eta \qquad (3.35)$$

A relationship for the infinitesimal $d\Gamma$ is still required. For an arbitrary piece of the boundary on a global element (figure 3.6)

$$dA = S_1 \times S_2$$

where S_1 and S_2 are the sides of the rectangular boundary element. In terms of the global coordinates

$$dA = \sqrt{(dx^2 + dy^2 + dz^2)}\,.\,\sqrt{(dx^2 + dy^2 + dz^2)} \qquad (3.36)$$

where the first term S_1 is transformed to the side ω=constant and the second term S_2 is transformed to the side ζ=constant, as can be seen from figure 3.6. Note that η=constant everywhere on the boundary. This leaves only $d\zeta$ non-zero on S_1 and only $d\omega$ non-zero on S_2. Equation (3.36) therefore becomes, in terms of the local coordinates

$$d\Gamma = \sqrt{\left(\left[\frac{\partial x}{\partial \zeta}\right]^2 d\zeta^2 + \left[\frac{\partial y}{\partial \zeta}\right]^2 d\zeta^2 + \left[\frac{\partial u}{\partial \zeta}\right]^2 d\zeta^2\right).\sqrt{\left(\left[\frac{\partial x}{\partial \omega}\right]^2 d\omega^2 + \left[\frac{\partial y}{\partial \omega}\right]^2 d\omega^2 + \left[\frac{\partial u}{\partial \omega}\right]^2 d\omega^2\right)}}$$

and, for purposes of integration

$$d\Gamma = \sqrt{\left(\left[\frac{\partial x}{\partial \zeta}\right]^2 + \left[\frac{\partial y}{\partial \zeta}\right]^2 + \left[\frac{\partial u}{\partial \zeta}\right]^2\right).\sqrt{\left(\left[\frac{\partial x}{\partial \omega}\right]^2 + \left[\frac{\partial y}{\partial \omega}\right]^2 + \left[\frac{\partial u}{\partial \omega}\right]^2\right)}\, d\zeta d\omega$$

$$\eta = \text{constant} \qquad \ldots\ldots\ldots (3.37)$$

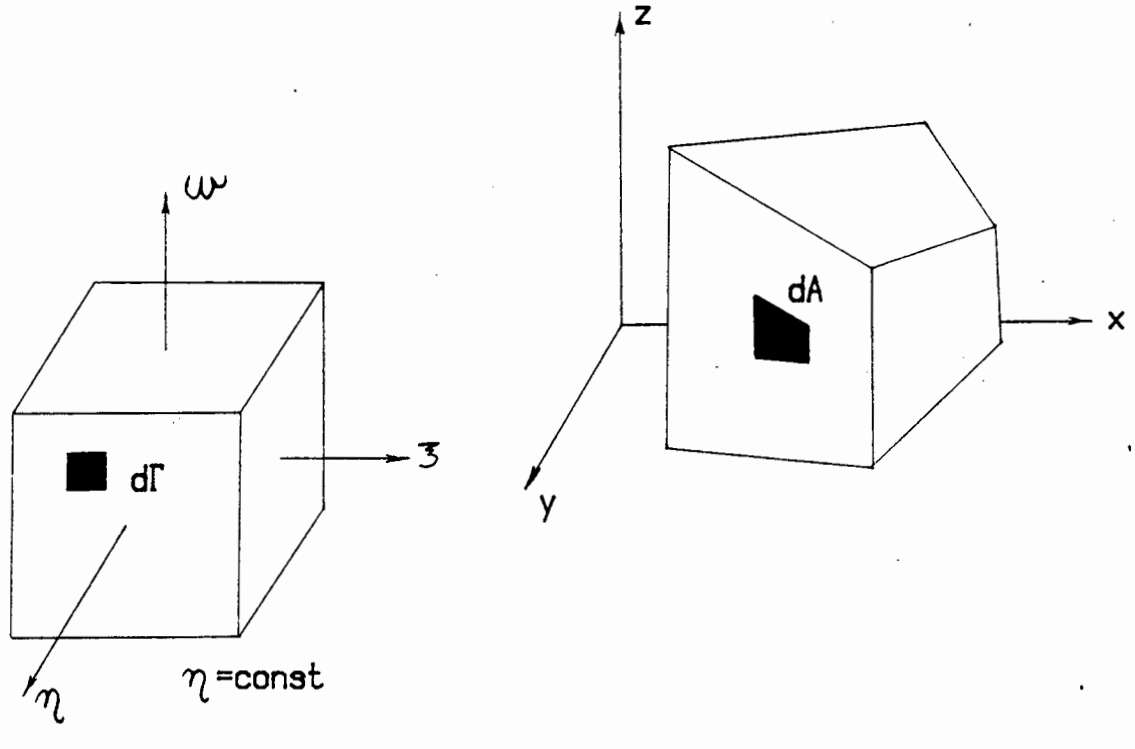


Figure 3.6 Isoparametric transformation of a boundary

Similar expressions can be developed for the transformed boundary belonging to a side where ξ=constant or ω=constant.

Equations (3.29), (3.31), (3.33), (3.34), (3.35) and (3.37) define the relationships required in order to effect the integration in the Finite Element formulations of equations (3.19) and (3.24) on the transformed isoparametric element. These Finite Element formulations can now finally be written, for the transformed coordinate system, as

Groundwater Flow:

$$\underset{\sim}{h}^{t+\Delta t}\int_{\Omega}\left[\left[k_x[\phi_x][\phi_x]^T + k_y[\phi_y][\phi_y]^T + k_z[\phi_z][\phi_z]^T\right] + \frac{1}{\Delta t}\left[S_s[\phi][\phi]^T\right]\right]$$

$$+ \int_{\Omega} Q[\phi]\ |J|\,d\xi d\eta d\omega = \underset{\sim}{h}^{t}\int_{\Omega}\frac{1}{\Delta t}\left[S_s[\phi][\phi]^T\right]\ |J|\,d\xi d\eta d\omega$$

$$- \int_{\Gamma} q_n[\phi]\ \sqrt{\left(\left[\frac{\partial x}{\partial \xi}\right]^2 + \left[\frac{\partial y}{\partial \xi}\right]^2 + \left[\frac{\partial u}{\partial \xi}\right]^2\right).\sqrt{\left(\left[\frac{\partial x}{\partial \omega}\right]^2 + \left[\frac{\partial y}{\partial \omega}\right]^2 + \left[\frac{\partial u}{\partial \omega}\right]^2\right)}}\ d\xi d\omega \qquad (3.38)$$

for η=constant

Diffusion-Convection:

$$\underset{\sim}{\varsigma}^{t+\Delta t}\int_{\Omega} n/2\Big[D_{xx1}[\phi_x][\phi_x]^T + D_{yy1}[\phi_y][\phi_y]^T + D_{zz1}[\phi_z][\phi_z]^T$$

$$+ 2D_{xy1}[\phi_x][\phi_y]^T + 2D_{xz1}[\phi_x][\phi_z]^T + 2D_{yz1}[\phi_y][\phi_z]^T$$

$$+ v_x[\phi_x][\phi]^T + v_y[\phi_y][\phi]^T + v_z[\phi_z][\phi]^T + 2/n[\phi][\phi]^T/\Delta t\Big]\,|J|\,d\xi d\eta d\omega$$

$$= \underset{\sim}{\varsigma}^{t}\int_{\Omega} n/2\Big[- D_{xx2}[\phi_x][\phi_x]^T - D_{yy2}[\phi_y][\phi_y]^T - D_{zz2}[\phi_z][\phi_z]^T$$

$$- 2D_{xy2}[\phi_x][\phi_y]^T - 2D_{xz2}[\phi_x][\phi_z]^T - 2D_{yz2}[\phi_y][\phi_z]^T$$

$$- v_x[\phi_x][\phi]^T - v_y[\phi_y][\phi]^T - v_z[\phi_z][\phi]^T + 2/n[\phi][\phi]^T/\Delta t\Big]\,|J|\,d\xi d\eta d\omega$$

$$- \int_{\Gamma} n f_n[\phi]\,\sqrt{\left(\left[\frac{\partial x}{\partial \xi}\right]^2+\left[\frac{\partial y}{\partial \xi}\right]^2+\left[\frac{\partial u}{\partial \xi}\right]^2\right)}\cdot\sqrt{\left(\left[\frac{\partial x}{\partial \omega}\right]^2+\left[\frac{\partial y}{\partial \omega}\right]^2+\left[\frac{\partial u}{\partial \omega}\right]^2\right)}\,d\xi d\omega \qquad (3.39)$$

for η=constant

The interpolation functions to be used in the above equations will be the linear Lagrange polynomials given in equations (3.40).

$$\phi_1 = 1/8(1-\xi)(1-\eta)(1-\omega)$$

$$\phi_2 = 1/8(1+\xi)(1-\eta)(1-\omega)$$

$$\phi_3 = 1/8(1+\xi)(1+\eta)(1-\omega)$$

$$\phi_4 = 1/8(1-\xi)(1+\eta)(1-\omega)$$

$$\phi_5 = 1/8(1-\xi)(1-\eta)(1+\omega)$$

$$\phi_6 = 1/8(1+\xi)(1-\eta)(1+\omega)$$

$$\phi_7 = 1/8(1+\xi)(1+\eta)(1+\omega)$$

$$\phi_8 = 1/8(1-\xi)(1+\eta)(1+\omega) \qquad (3.40)$$

The numbering of the nodes of the transformed isoparametric element is shown in figure 3.7, and is in accordance with the basis functions of equations (3.40) where the function is unity at the subscripted node and zero elsewhere.

The integrations in equations (3.38) and (3.39) can be effected using a three-dimensional Gauss-Legendre integration. A two point integration scheme will suffice since the highest order of the polynomials is 2.

3.4 The computational model

For each eight-noded element in the three-dimensional domain, the finite element equations will generate seven 8X8 matrices and one 8X1 vector as follows

$$[\phi][\phi]^T \qquad [\phi_x][\phi_x]^T \qquad [\phi_x][\phi]^T \qquad [\phi]$$

$$[\phi_y][\phi_y]^T \qquad [\phi_y][\phi]^T$$

$$[\phi_z][\phi_z]^T \qquad [\phi_z][\phi]^T$$

the elements of which have to be integrated. These elements

then have to be positioned into global matrices encompassing the whole domain. If this process is performed for each element, a system of linear equations will result. This system will have to be solved for each time-step. A computational model is therefore essential for this, in fact any, finite element procedure because of the extensive numerical processing required.

Finite Element programs require large computer storage for larger domains. To date most applications were therefore carried out on mainframe or minicomputers. The microcomputer such as the IBM Compatible PC with its limited working memory capacity therefore seems, at the outset, incapable of handling these procedures. To a large extent this problem could be overcome with efficient programming. The PC has an extensive disk storage capacity. Working memory could be made available for use by writing unused information, that is otherwise memory resident, to disk. This refinement of the program is an ongoing process as the need for solving larger domains arise.

Turbo BASIC, a product of Borland International Inc., was used in the development of the computational model. It is a compiler version of BASIC. The advantage of the compiler is that the program can be compiled into machine code as an operating system executable (.EXE) file. This means that the entire RAM (except that occupied by the system files) is available for use by the program since the compiler itself is not required during the execution. The data capturing

routines and selection menus were written in dBaseIV, a product of Ashton-Tate. An apparent advantage of this computational model is that it's use is easily accessible since PC's can easily be made available compared to larger mainframe or minicomputer systems.

Essentially, the program, called AQUIFEM, uses three dimensional brick elements with linear Lagrange interpolation functions to solve the coupled groundwater flow - diffusion-convection problem as defined by the Finite Element formulations of equations (3.38) and (3.39). Figure 3.7 shows the menu structure of AQUIFEM.

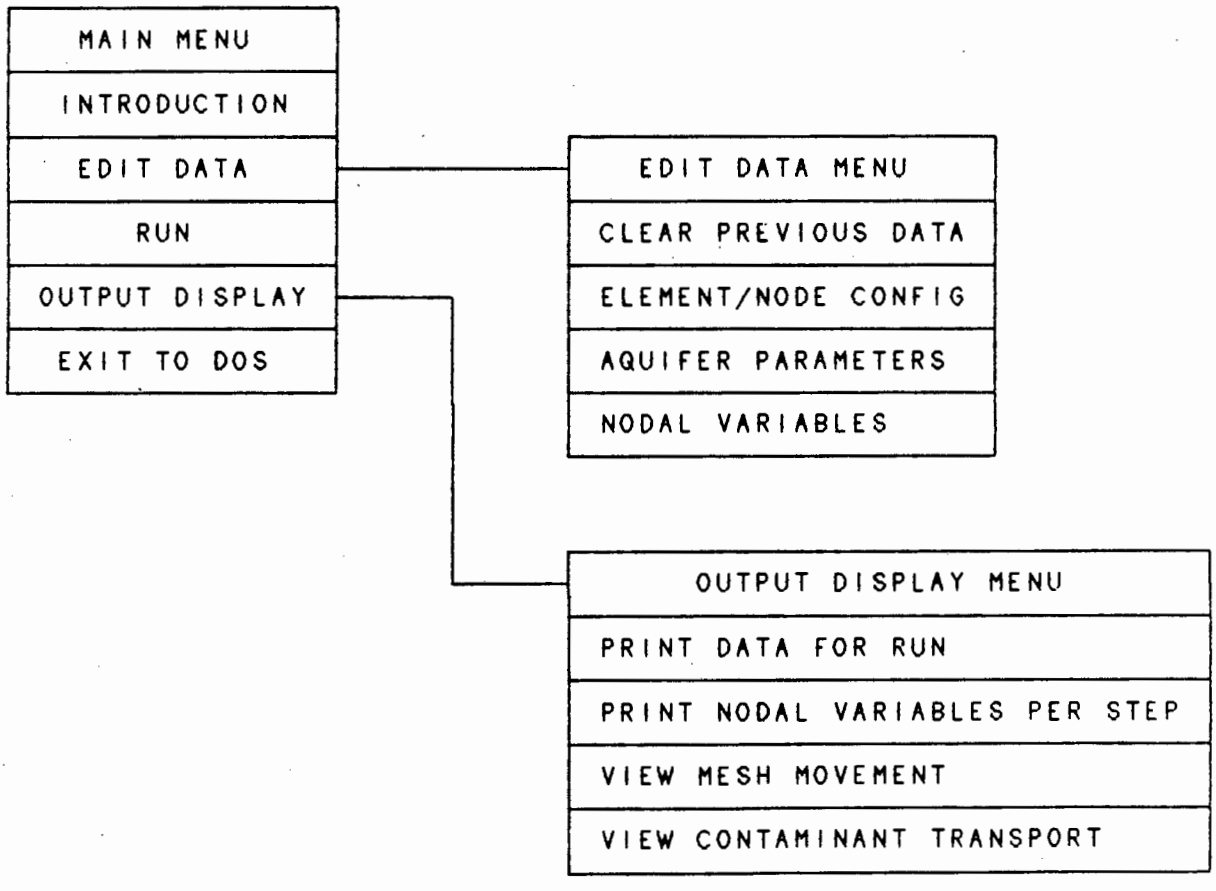


Figure 3.7 AQUIFEM main menu and submenus

An outline of the function of each selection follows. Note that any names within square brackets refer to the names of subprograms effecting the procedures under discussion.

INTRODUCTION — This gives an overview of the program, the methods employed and the types of problems that could be solved.

EDIT DATA — This selection allows user entry and editing of the input data. Extensive error trapping routines are used to avoid the entry of incorrect and insufficient data as well as data exceeding the limitations of the program. Data is entered via the following submenus as follows:

DOMAIN PARAMETERS	The number of elements and the number of nodes.
ELEMENT/NODE CONFIG	The assignment of nodes within the domain to the elements.
AQUIFER PARAMETERS	The porosity and the coefficients of specific storage, permeability and dispersion per element.
NODAL VARIABLES	The spatial coordinates, the initial values of potential head and concentrations, and rate of discharge or recharge. In addition, the nodes have to be marked as internal, boundary or free surface nodes.

The data, which is captured in dBaseIV, is converted into ASCII format [CONVERT.FEM] to enable the program execution files, written in Turbo BASIC, to read it.

RUN — After the entry of the data, the user may then execute the Finite Element procedure. The time interval and the number of steps for execution has to be specified. When the run has completed, the user may execute a further number of steps not necessarily with

the same time interval. The program called MAIN.FEM controls the 13 subprograms involved in the execution process. Figure 3.8 depicts a brief flowchart of the subprograms and table 3.1 gives the functions performed by each subprogram.

OUTPUT DISPLAY This allows the user to print, via the screen or printer, the input and output data. The following selections are available:

PRINT DATA FOR RUN	Allows the printing of the input data. [OUTPUT1.FEM]
PRINT NODAL VARIABLES PER TIME STEP	Allows the printing, either to screen or printer, of the results. A range of time steps could be specified. [OUTPUT2.FEM]
VIEW MESH MOVEMENT	The three dimensional mesh is graphically produced per time step. This is useful in the case of free surface problems where the mesh may vary per time step until a steady state is reached. Parameters governing the size and the view angle have to be specified. [DRW.FEM]
VIEW CONTAMINANT TRANSPORT	A contour map of the contaminant concentrations are plotted per time step or range of time steps. Since the domain is three dimensional, the user has to specify either an x-, y-, or z-plane and its coordinate on which the contours have to be generated. Furthermore, the desired contour level has to be specified. [CONTOUR.FEM]

Listings of all the programs are given in APPENDIX C.

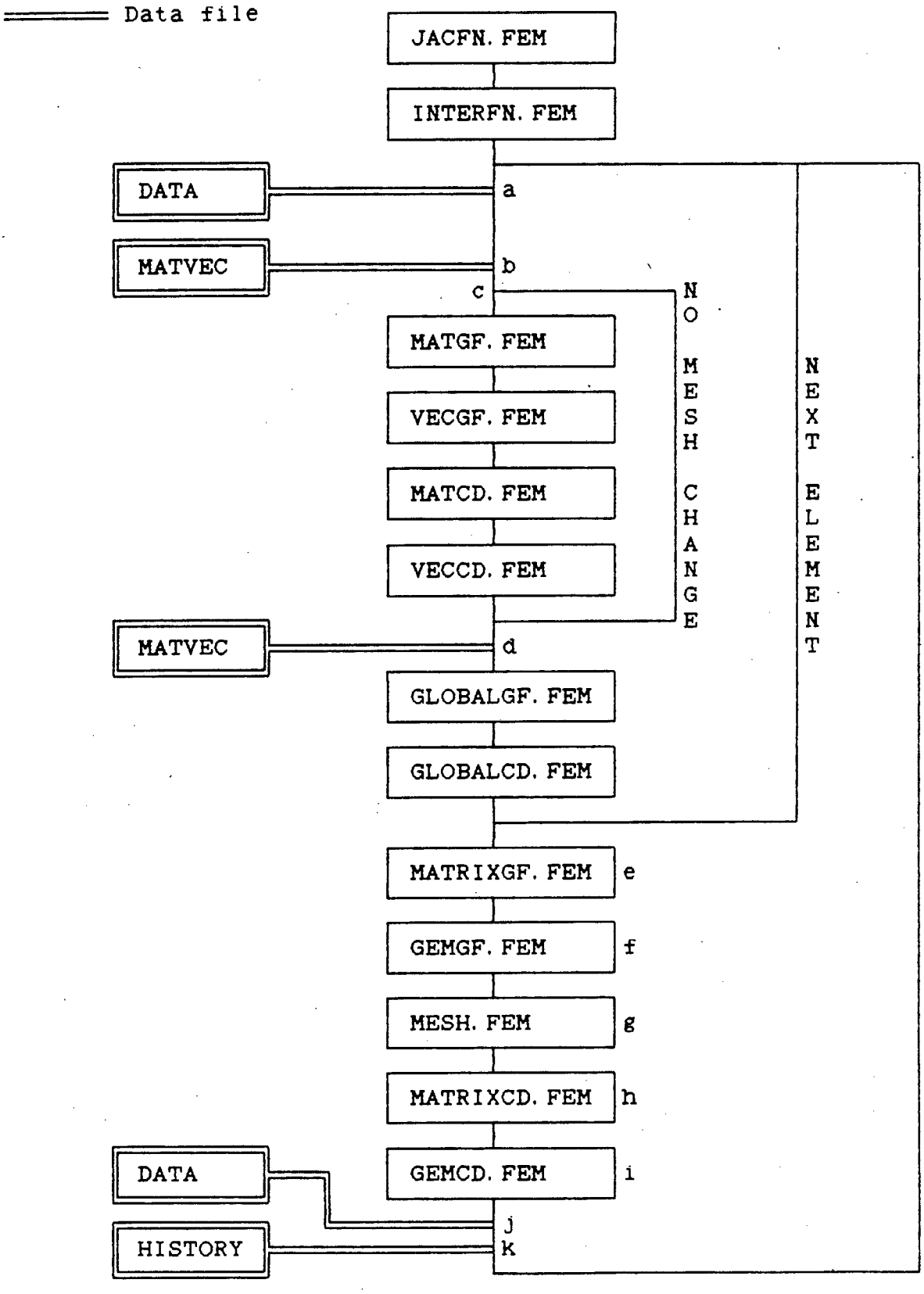


Figure 3.8 Flowchart of MAIN.FEM - the Finite Element execution program

Table 3.1 The subprograms of MAIN.FEM and their functions.

Program name	Function
JACFN.FEM	Defines the functions leading up to, and including, the Jacobian function.
INTERFN.FEM	Defines the FE interpolation functions and the elements of the matrices used in the FE equns.
MATGF.FEM	Calculates the matrices $\int [\phi][\phi]^T d\Omega$, $\int [\phi_u][\phi_u]^T d\Omega$ u=x, y, z
VECGF.FEM	Calculates the integrals of the vectors $\int [\phi] d\Omega$, $\int [\phi] d\Gamma$ $d\Gamma$=boundaries of constant potential
MATCD.FEM	Calculates the matrices $\int [\phi u][\phi]^T d\Omega$ u=x, y, z
VECCD.FEM	Calculates the integrals of the vector $\int [\phi] d\Gamma$ $d\Gamma$=boundaries of constant concentration
GLOBALGF.FEM	Arranges element matrices globally into the groundwater flow FE formulation.
GLOBALCD.FEM	Arranges element matrices globally into the diffusion-convection FE formulation.
MATRIXGF.FEM	Rearranges the groundwater flow FE formulation into a form $[U]\underset{\sim}{u}=\underset{\sim}{s}$, u=unknowns vector.
GEMGF.FEM	Solves the system of linear equations generated by MATRIXGF using Gauss Elimination.
MESH.FEM	Adjusts the mesh due to changes in the free surface boundaries.
MATRIXCD.FEM	Rearranges the diffusion-convection FE formulation into a form $[U]\underset{\sim}{u}=\underset{\sim}{s}$, u=unknowns vector.
GEMCD.FEM	Solves the system of linear equations generated by MATRIXCD using Gauss Elimination.

Certain aspects of MAIN.FEM will now be discussed The reader should refer to the labels a, b, c, ... , K in the flowchart of figure 3.8.

a. The data relating to the physical domain (element/node configurations, aquifer properties, and nodal characteristics and values) are read from a data file on disk.

b. After the first time step, the elements of the Finite Element matrices are written (at d) into a data file and stored on disk. During the second and subsequent time steps, these matrices are not necessarily regenerated. These matrices depend on the shape and size of the elements. They will hold as long as the elements within the mesh have not changed in shape and size. The data are therefore at first read from disk.

c. In the case of free surface boundaries, the mesh is readjusted after each time step (see h). If the change in the mesh exceeds a prescribed tolerance, then the Finite Element matrices are regenerated per element (i.e. certain elements in the mesh might not have changed as much as others). Otherwise the same matrix data used in the previous time step is used.

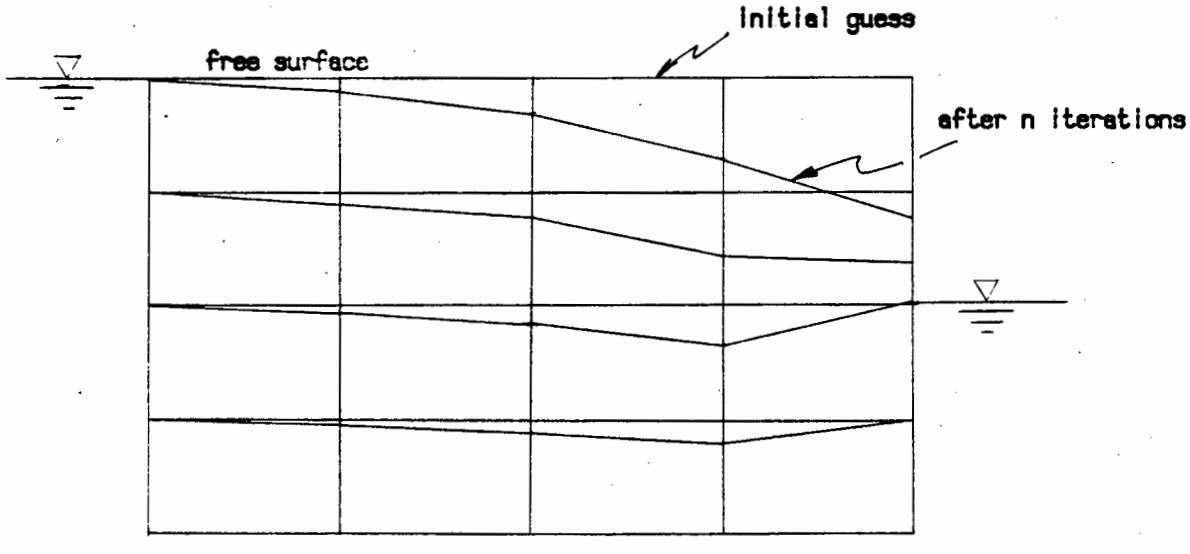


Figure 3.9 Adjusting of FE mesh due to changes in the free surface boundary.

e. In this subprogram, the FE formulation of the groundwater flow equation given in equation (3.19)

$$\{[A] + [B]/\Delta t\}\underset{\sim}{h}^{t+\Delta t} = \{[B]/\Delta t\}\underset{\sim}{h}^{t} - \underset{\sim}{Q} - \underset{\sim}{q}$$

is reduced to the form

$$[U]\underset{\sim}{u} = \underset{\sim}{s}$$

where u is the unknowns vector and s is the constants vector. For a detailed discussion of this procedure, the reader may refer to APPENDIX B.

f. The system of linear algebraic equations

$$[U]\underset{\sim}{u} = \underset{\sim}{s}$$

is solved using the Gauss elimination method. This results in the solution of the unknowns vector which represents the nodal values of potential head and the fluid flux at the boundaries where constant potential heads are specified.

g. Adjustments in the mesh are not only effected at the boundary nodes, but spread throughout the domain. Nodes further away from the free surface boundaries are, however, affected to a lesser extent than those nearby, as shown in figure 3.9.

Besides adjusting the mesh, this subprogram also calculates the element velocity components v_x, v_y and v_z by substituting the nodal values of potential head into Darcy's equation.

h. A similar procedure to that performed in e is carried out here. This time, the FE formulation of the diffusion-convection equation given in equation (3.24)

$$\tfrac{1}{2}\{[L1]+[M]+2[N]/\Delta t\}\underset{\sim}{c}^{t+\Delta t} = \tfrac{1}{2}\{-[L1]-[M]+2[N]/\Delta t\}\underset{\sim}{c}^{t} - \underset{\sim}{g}$$

is reduced to the form

$$[U]\underset{\sim}{u} = \underset{\sim}{s}$$

i. The Gauss elimination method used at f is again used here to calculate the unknowns vector representing the nodal values of concentration and the flux at the boundaries where constant potential heads were specified.

j. During each time step, the mesh coordinates and the nodal values of potential head and concentration may change. The new set of data is stored to disk for use in the next step. Storage to disk instead of memory is necessary since the program may be run for a certain number of steps and then rerun later.

k. The FE data on disk only constitutes that of the most recent time step. It is therefore necessary to write the data, per time step, into history files on disk. In addition to the mesh coordinates and the nodal variables, the element velocities and the number of iterations performed is also stored.

Chapter 4

Verification of the computational model.

Results obtained from the model will be compared to analytical results in order to determine the model's accuracy. Analytical solutions can be obtained for simplified forms of the problem.

4.1 Groundwater Flow

Although the model is a coupled groundwater flow and diffusion-convection model, the two phases can be isolated for the purpose of comparison with theoretical results.

4.1.1 Test Problem 1 - Confined aquifer

Consider the simple case of an confined aquifer as shown in figure 4.1. The two permeable boundaries experience constant potential heads. The equation governing the flow is the one-dimensional steady state equation

$$\frac{\partial}{\partial x} k \frac{\partial h}{\partial x} = Q \qquad (4.1)$$

which, for zero discharge/recharge Q, simplifies to the Laplace equation

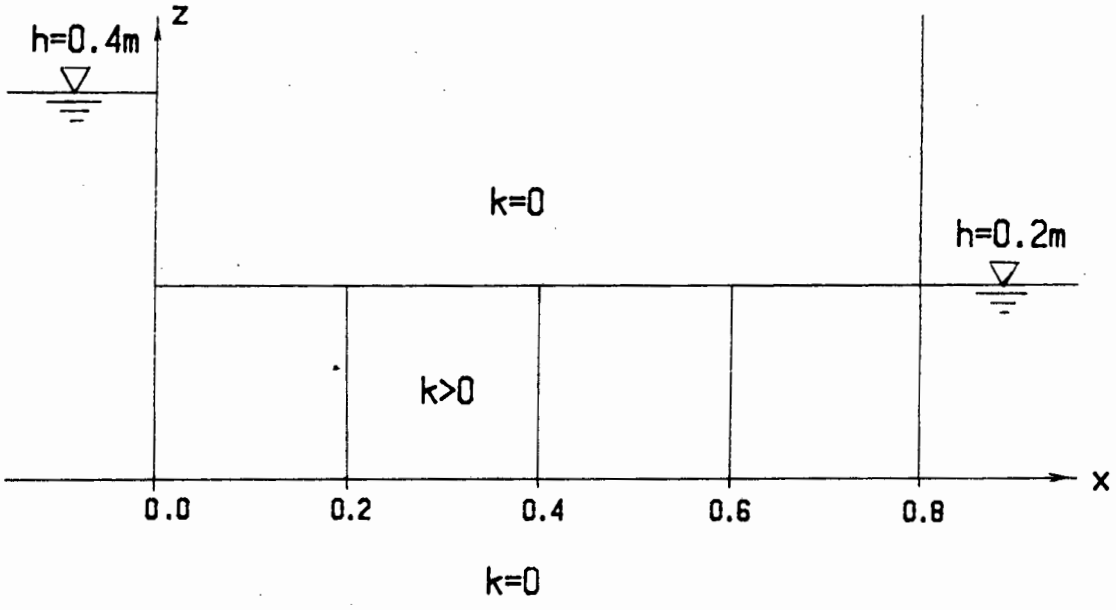


Figure 4.1 Test Problem 1: A confined aquifer

$$\frac{\partial^2 h}{\partial x^2} = 0 \qquad (4.2)$$

Integrating the above equation gives

$$\frac{\partial h}{\partial x} = c_1$$

$$h = c_1 x + c_2$$

Using the boundary conditions to solve for the constants c_1 and c_2, the exact solution to the problem can be written as

$$h = -.25x + .4$$

The solution is tabulated, together with that of the model, in table 4.1. A zero percent error is encountered.

Table 4.1 Test Problem 1: Comparison of results

x (m)	h_{theory} (m)	h_{model} (m)
0	0.400	0.400
0.2	0.350	0.350
0.4	0.300	0.300
0.6	0.250	0.250
0.8	0.200	0.200

4.1.2 Test Problem 2 - Unconfined aquifer

The two-dimensional unconfined aquifer problem depicted in figure 4.2 is no longer governed by the Laplace equation because of the existence of a free surface. An equation, called the Boussinesq equation [26], can be used to analytically solve free surface flow. The one-dimensional, steady-state form of this equation is given by

$$\frac{\partial}{\partial x} k_x h \frac{\partial h}{\partial x} = 0 \qquad (4.3)$$

where h is the height of the free surface. In the derivation of equation (4.3), a simplification known as the Dupuit approximation was used. This assumes (a) that velocities are horizontal and therefore (b) the potential heads do not change along any vertical line within the aquifer. This is often used when the horizontal dimensions of aquifers are orders of magnitude larger than the vertical dimensions, i.e. aquifers with high slenderness ratios.

The main disadvantage of the Dupuit approximation is that it

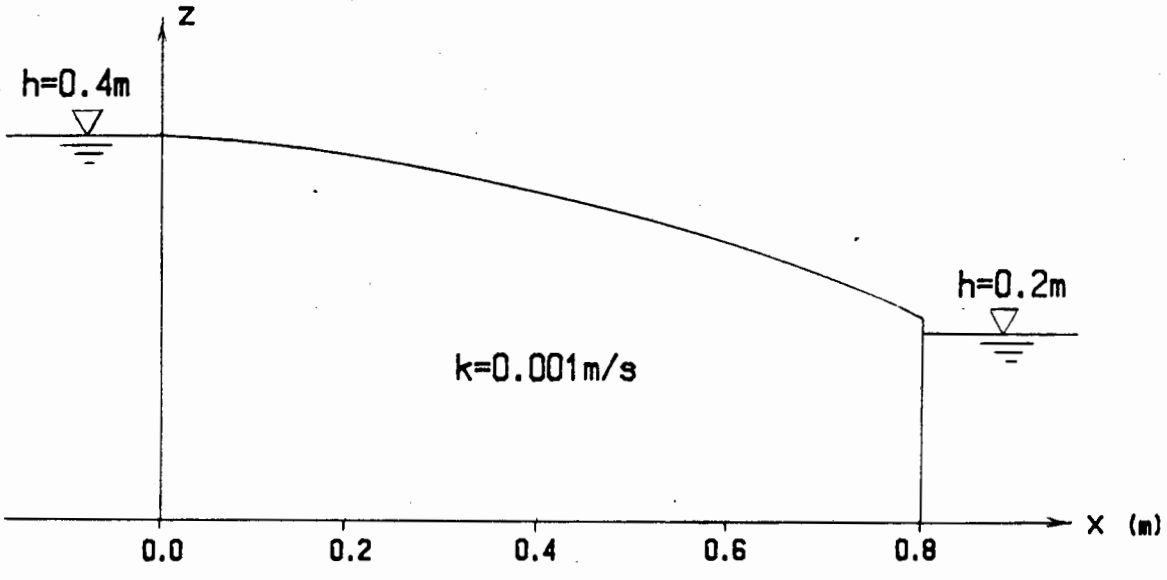


Figure 4.2 Test Problem 2: An unconfined aquifer

does not take into account the seepage surface shown in figure 4.2. However, the error involved is generally small and confined to a short distance from the well [22]. A major advantage, however, is that the problem is mathematically two-dimensional which is computationally more desirable.

It is thus anticipated that the free surface height obtained from equation (4.3) will be lower than that from the computational model since the Dupuit approximation was not used in the latter case. However, it would be interesting to note the general shapes of the free surfaces.

Integrating equation (4.3) gives

$$h\frac{\partial h}{\partial x} = c_1$$

$$\frac{h^2}{2} = c_1 x + c_2$$

Using the boundary conditions, that is

at x = 0 , h = 0.4
at x = 0.8 , h = 0.2

the following solution is obtained

$$h = \sqrt{(-1.50x + 0.16)} \qquad (4.4)$$

In the computational model, the domain was discretized into eight elements and 30 nodes, as shown in figure 4.3. (The free surface was assumed to be a horizontal line initially.)

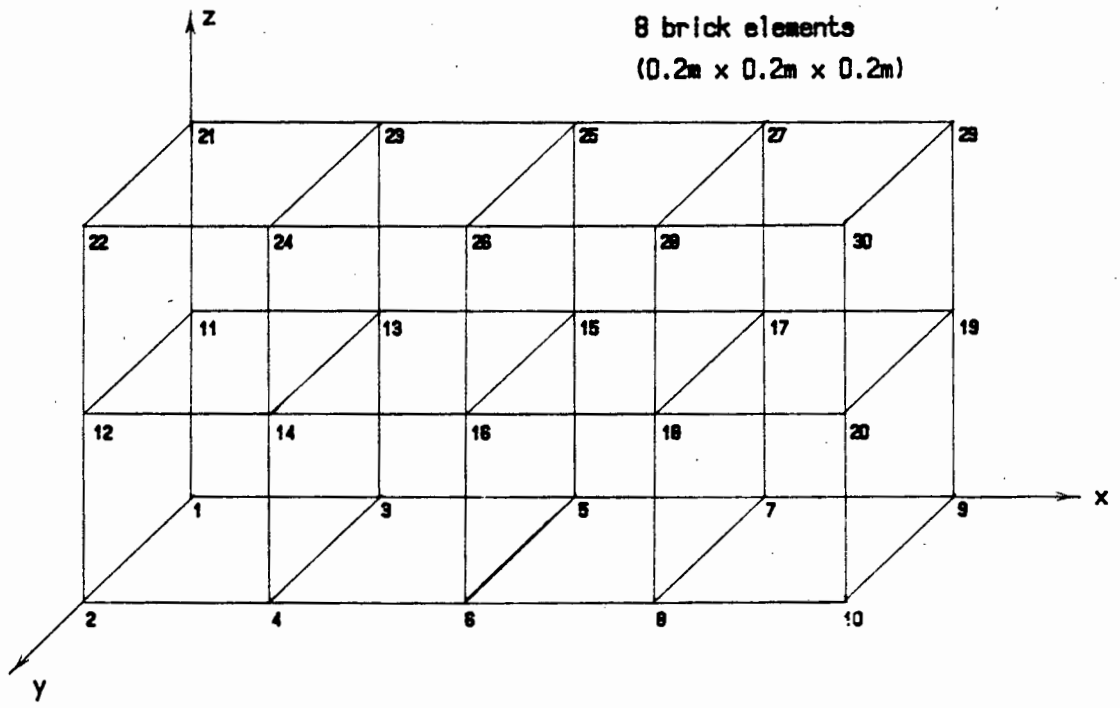


Figure 4.3 Test Problem 2: Finite Element discretization.

A 2% iteration tolerance was used in the program. Five iterations were required for convergence within the specified tolerance.

Figure 4.4 and table 4.2 show that the results obtained are in good agreement with each other. As mentioned earlier, the model's results are higher due to the effect of the seepage surface. More detailed results obtained from computer printouts are given in Appendix D.

Table 4.2 Test Problem 2: Comparison of results

x (m)	h_{theory} (m)	h_{model} (m)
0	0.400	0.400
0.2	0.361	0.365
0.4	0.316	0.322
0.6	0.265	0.271
0.8	0.200	0.203

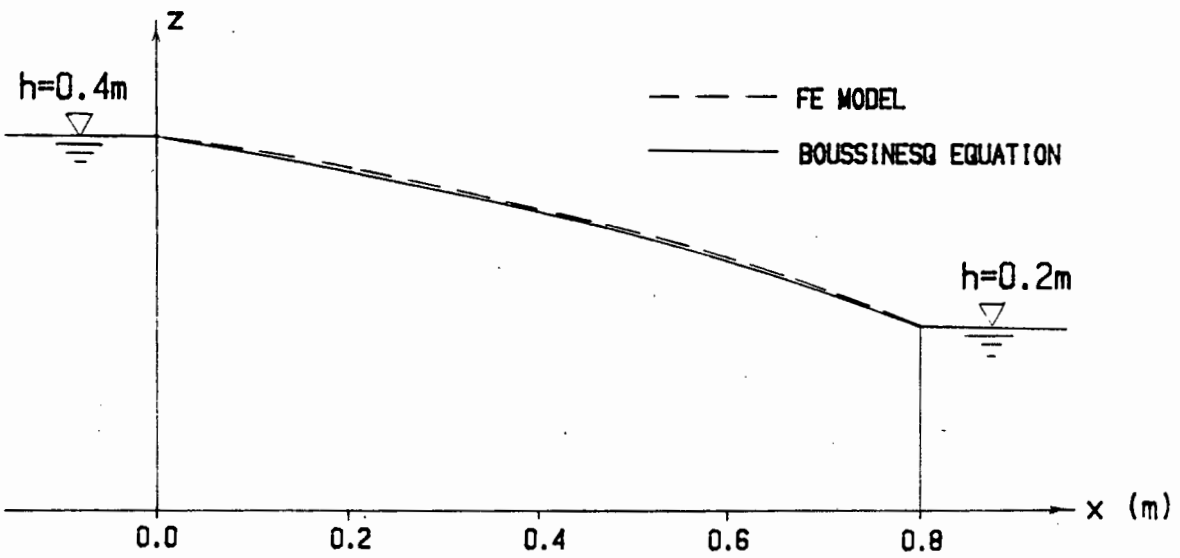


Figure 4.4 Test Problem 2: Comparison of results.

The seepage surface is not very pronounced in this problem mainly because the seepage in the vertical direction is comparable to that in the horizontal direction. Lets consider the same aquifer, but with an increased permeability in the x-direction, i.e. $k_x=5k_y$. A three dimensional view of the free surface, as generated by the computational model, is shown in figure 4.5. The seepage surface is more clearly defined.

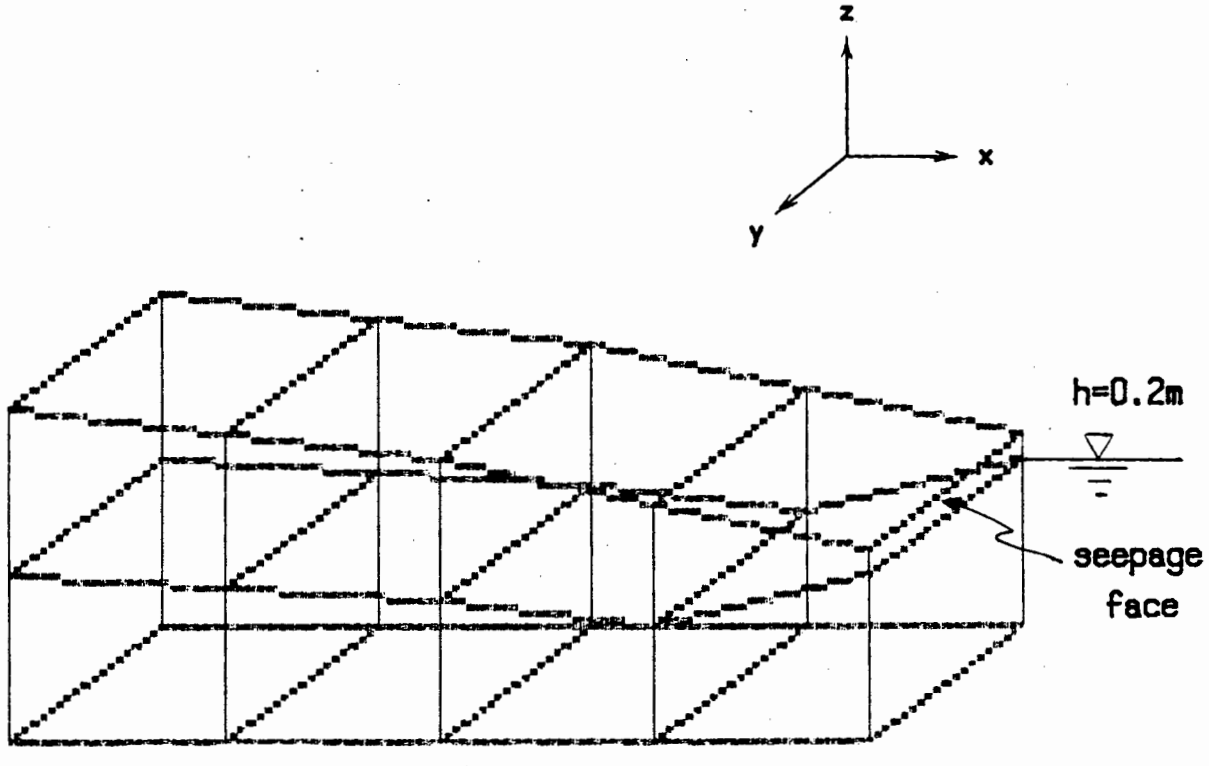


Figure 4.5 Unconfined aquifer with $k_x=5k_y$

4.2 Diffusion-Convection

4.2.1 Test Problem 3 - One-dimensional diffusion-convection

Van Genuchten [9] presented the analytical solution, first developed by Lapidus and Amundsen, to the one-dimensional diffusion-convection problem as

$$c = \tfrac{1}{2}\mathrm{erfc}\left[\frac{x - vt}{2(Dt)^{\frac{1}{2}}}\right] + \tfrac{1}{2}\exp(vx/D)\ \mathrm{erfc}\left[\frac{x + vt}{2(Dt)^{\frac{1}{2}}}\right]$$

where x is the spatial dimension, v the fluid velocity, D the diffusion coefficient and t the time. Erfc is the complimentary error function to be described later.

A modified form applicable to porous media was used by Ehlig [7] and is given as follows

$$c = \tfrac{1}{2}\mathrm{erfc}\left[\frac{x - vnt}{2(Dnt)^{\frac{1}{2}}}\right] + \tfrac{1}{2}\exp(vx/D)\ \mathrm{erfc}\left[\frac{x + vnt}{2(Dnt)^{\frac{1}{2}}}\right] \qquad (4.5)$$

where n is the porosity of the medium. Equation (4.5) is valid for a constant value of concentration, c=1, at x=0. For a source concentration other than unity, values of concentration throughout the domain can be calculated by proportion. Furthermore, equation (4.5) assumes a semi-infinite medium. This implies that the downstream boundary is at an infinite value of x.

According to Kreyszig [28], the error function erf x can be approximated by the following two infinite series

$$\operatorname{erf} x = \frac{2}{\sqrt{\pi}} \left[x - \frac{x^3}{1!3} + \frac{x^5}{2!5} - \frac{x^7}{3!7} + \cdots\cdots\cdots \right] \qquad (4.6a)$$

valid for $0 \leq x \leq 1.5$

$$\operatorname{erf} x = 1 - \frac{1}{\sqrt{\pi}} e^{-x^2} \left[\frac{1}{x} - \frac{1}{2x^3} + \frac{1.3}{2.2x^5} - \frac{1.3.5}{2.3x^7} + \cdots \right] \qquad (4.6b)$$

valid for $x \geq 1.5$

Equations (4.5) and (4.6) can easily be computerised to obtain the exact solution to one-dimensional diffusion-convection problems.

The confined flow region shown in figure 4.6 was used. The upstream and downstream potential heads were chosen such that a fluid velocity of 0.0002 m/s could be obtained.

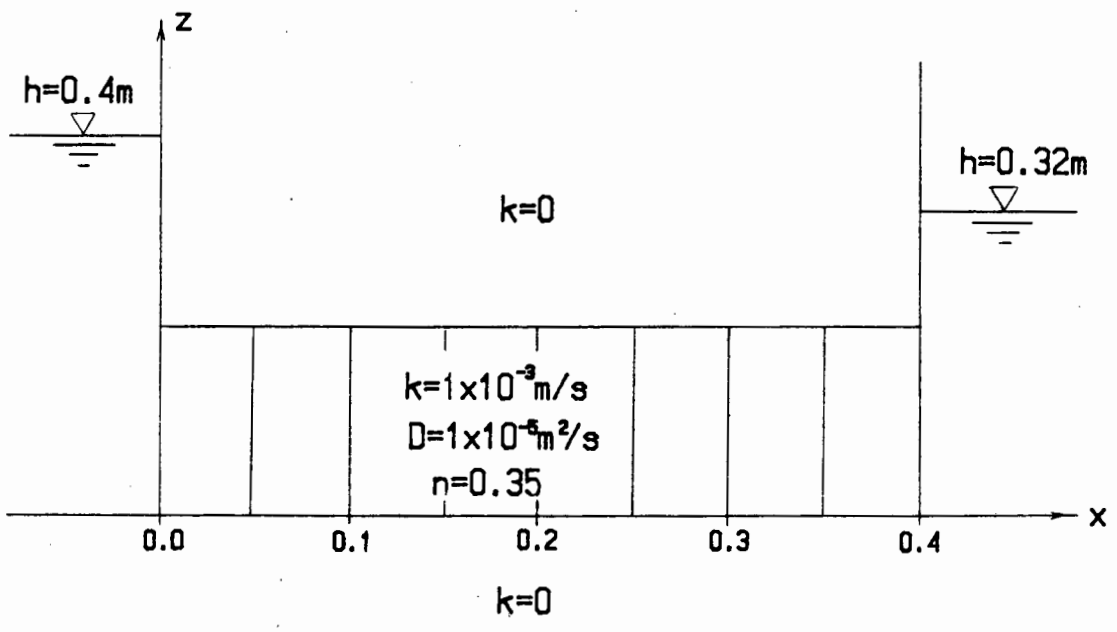


Figure 4.6 Test Problem 3: Contaminant transport domain

To prevent instability due to oscillations of the solution, the time step should be carefully chosen. The following relationship,

$$\frac{v\,\Delta t}{\Delta x} < K \qquad (4.7)$$

originally derived for the stability analysis of finite difference meshes [22], could be used to estimate the time step. The coefficient K is characteristic of the scheme used, and can be found by tests carried out with the model. A value of K less than 0.5 was consequently found to generate stable results. That is

$$\Delta t < 0.5\Delta x/v$$

This implies that a time step of less than 125 seconds should be used for this test problem. A value of 60 seconds was used which is well below this limit.

The results presented in figure 4.7 and table 4.3 demonstrate the accuracy of the Finite Element model. Eight brick elements were used in the mesh generation. The solutions after 1200 seconds and 1800 seconds are compared to the analytical solution from equation (4.5). More detailed results are presented in Appendix D.

Table 4.3 Test Problem 3: Comparison of results

Δt = 60 secs porosity = 0.35

D = $1X10^{-5}$ m^2/s v = 0.0002 m/s

x (m)	t = 1200 seconds		t = 1800 seconds	
	C_{theory}	C_{model}	C_{theory}	C_{model}
0.00	1.000	1.000	1.000	1.000
0.05	0.840	0.852	0.910	0.915
0.10	0.588	0.626	0.746	0.769
0.15	0.347	0.382	0.562	0.580
0.20	0.156	0.186	0.357	0.385
0.25	0.055	0.068	0.195	0.220
0.30	0.016	0.017	0.090	0.105
0.35	0.003	0.002	0.034	0.041
0.40	0.000	0.000	0.012	0.020

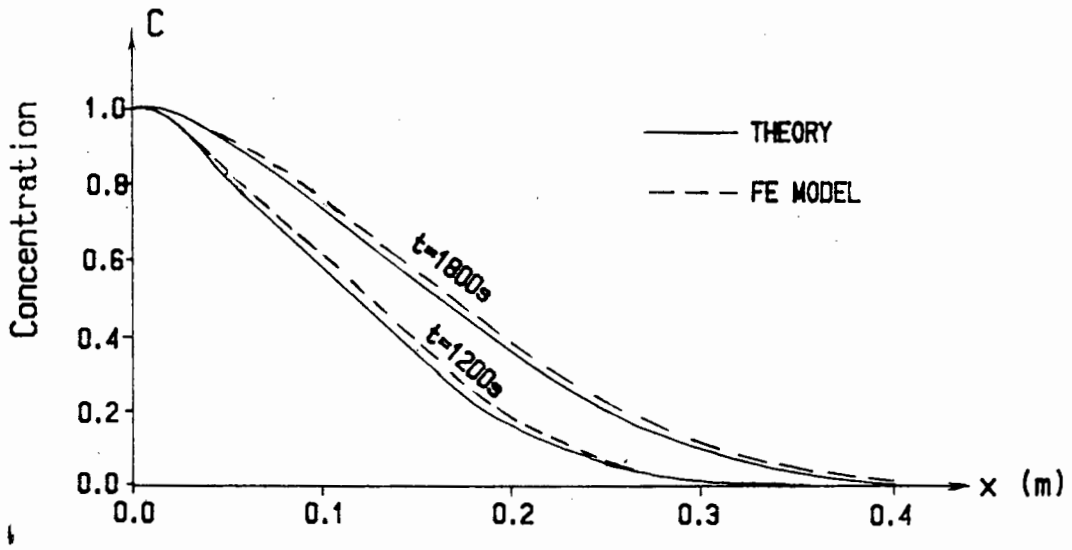


Figure 4.7 Test Problem 3: Concentration curves

The computational model has a facility to generate the concentration level contours, commonly referred to as the contaminant plume, at any time step. The concentration contour map at t=1200 seconds is given in figure 4.8. A contour interval of 0.1 was used.

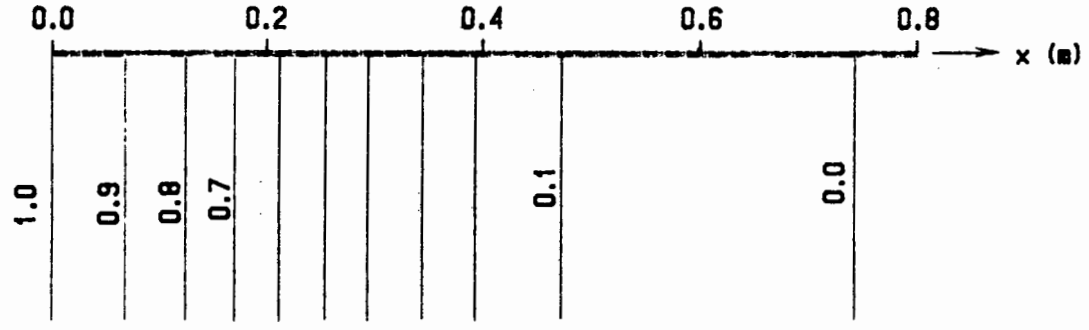


Figure 4.8 Test Problem 3: Contaminant contour map at t=1200 secs

4.2.2 Test Problem 4 - Convection dominated diffusion-convection

The problem of figure 4.6 is neither diffusion dominated nor convection dominated since both terms are relatively small. Previous researchers have reported cases of instability for convection dominated problems.

The same domain as that used in test problem 3 (figure 4.6) will be used here, except that a permeability of 0.1 is specified. This results in a fluid velocity of 0.002 m/s, making the problem one of convection dominated diffusion-convection. Such problems are uncommon in practice. Fluid velocities within aquifers are relatively small, except in regions close to wells.

Table 4.4 Test Problem 4: Comparison of results

Δt = 10 secs porosity = 0.35

D = $1X10^{-5}$ m^2/s v = 0.002 m/s

x (m)	t = 300 seconds		t = 600 seconds	
	C_{theory}	C_{model}	C_{theory}	C_{model}
0.00	1.000	1.000	1.000	1.000
0.05	1.000	0.999	1.000	1.000
0.10	1.000	1.006	1.000	1.000
0.15	0.905	0.947	1.000	1.000
0.20	0.586	0.699	1.000	0.999
0.25	0.191	0.344	1.000	1.000
0.30	0.024	0.090	0.968	0.988
0.35	0.001	0.001	0.860	0.905
0.40	0.000	-0.005	0.621	0.706
0.45	0.000	0.000	0.322	0.433
0.50	0.000	0.000	0.109	0.194
0.55	0.000	0.000	0.021	0.053
0.60	0.000	0.000	0.003	0.009

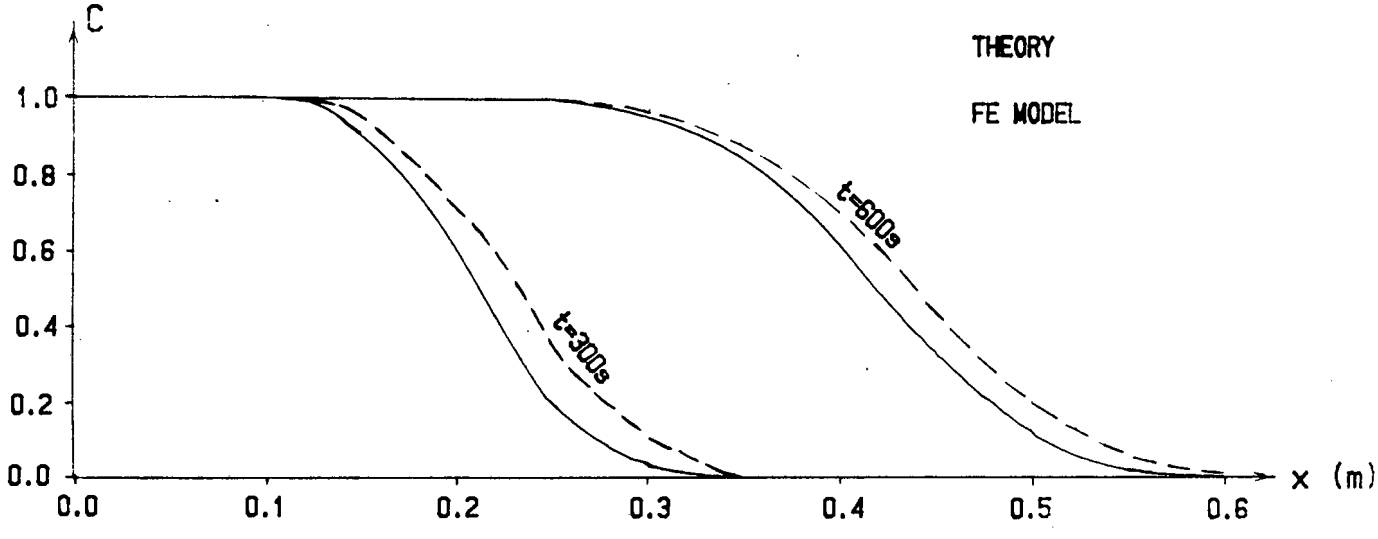


Figure 4.9 Test Problem 4: Concentration curves

The results plotted in Figure 4.9 (and figure 4.7 in the previous example) show that the slope of the dispersion

front is larger than that given by the analytical solution. This is not due to an inability of the method to produce the correct front, but rather due to the failure of the basis functions to simulate the correct boundary condition at x=0. The simulated boundary condition, shown in figure 4.10, implies that, initially, c=1 at x=0 and c=0 at x=Δx. This incorrect distribution, coupled with a high fluid velocity, affects the initial propagation of the contaminant, but its effect, as seen in figure 4.9, diminishes with time. The incorrect simulation of the boundary condition can be overcome by assigning negative values to certain nodes. This causes the initial distribution averaged over the first element to be zero, as shown in figure 4.10.

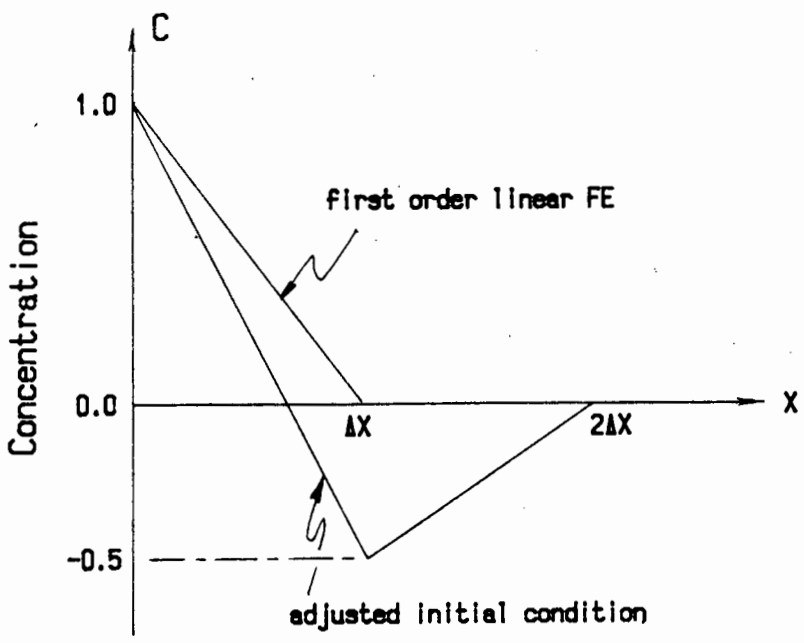


Figure 4.10 Simulated and adjusted initial conditions

The problem was rerun using the adjusted Dirichlet boundary condition and the results, at time t=600 seconds, are shown in table 4.5. The contaminant front is more accurately

approximated than before. In fact, the numerical solution will converge to the analytical solution with time as the oscillations due to the initial negative concentrations diminish.

Table 4.5 Test Problem 4: Comparison of results - adjusted initial boundary condition

Δt = 10 secs porosity = 0.35

$D = 1X10^{-5}$ m^2/s v = 0.002 m/s

x (m)	t = 600 seconds	
	c_{theory}	c_{model}
0.00	1.000	1.000
0.05	1.000	1.000
0.10	1.000	1.000
0.15	1.000	1.000
0.20	1.000	0.998
0.25	1.000	1.001
0.30	0.968	0.977
0.35	0.860	0.847
0.40	0.621	0.578
0.45	0.322	0.271
0.50	0.109	0.056
0.55	0.021	-0.020
0.60	0.003	-0.023

Certain oscillations are still prevalent in the results. These cannot be reduced by reducing the time interval. They are due to the poor approximation of the time derivative which is inherent in the method. Even the use of higher order basis functions cannot reduce these oscillations [9].

Chapter 5

The design and testing of the experimental model

5.1 Introduction

In the past, before scientific computer equipment became readily available, researchers had no option but to undertake, sometimes large-scale, laboratory and field experimentation. Experimental models and predictions were largely used as a basis for design and decision-making. For instance, the Hele-Shaw flow models were often employed to study two-dimensional laminar flow in both soil mechanics and marine engineering. Here, Laplacian flow was simulated by passing fluid between thinly spaced plates. Glass plates were normally used so that flow patterns, made visible through the injection of dyes, could be visualised.

With the advent of powerful numerical methods coupled with the availability of computational resources, experimental methods have suddenly appeared far too expensive. Besides, experimental methods are forced to compete with the very accurate and time-effective computational methods. This requires the use of more sophisticated laboratory equipment which can prove very costly.

In the modelling of aquifers, highly scaled-down experimental models are used. The result is that certain effects, which are regarded as negligible in the actual aquifers, may affect the validity of the results. For instance, Hunt [25] writes "A free surface is usually idealised as a surface of atmospheric pressure that has a zero thickness. In actual fact, however, a free surface consists of a finite-width zone of partially-saturated flow with a thickness that is usually small compared with the model aquifer thickness. This fact is one of the principle reasons why sandbox models are of limited use for modeling free surface flows in the laboratory. Capillarity, or surface tension, often creates a free surface in these models with a thickness that is no longer small when compared with the model aquifer thickness."

Although the computational model produced favourable results as compared to that given by certain available analytical solutions, both computational and analytical methods ignored certain characteristics of the problem. This was done for purposes of simplification. For example, the adsorption of a transported solute on the porous material was neglected. Adsorption gives rise to a decay in the amount of active solute, and may be considerable if a large enough affinity exists between the solute molecules and that of the porous medium. It is for this, and similar reasons, why laboratory experimentation is desired, mainly to verify the validity of the computational model within reasonable bounds.

The purpose of the experimental model will be to simulate a three-dimensional unconfined aquifer with:

(a) a variable potential gradient across the aquifer in order to model convection

(b) a contaminant source

(c) a well in order to see the effect of pumping on a dispersing contaminant

Furthermore, the contaminant concentrations should be measurable from samples of the "groundwater". Previous investigators used various methods to measure concentrations. One of the earliest methods was to inject dyes at the source, but this only provided a visual, rather than a quantitative measure of the solute. A step further was to inject a chemical, into the water, and then to extract the chemical from samples taken using the necessary extraction process. Such extraction processes are employed to ascertain the quality of water. This can be a very expensive method and is not ideal for fieldwork. The use of insoluble particles, visible with ultra-violet detection, also proved successful. A shortcoming of this method is that only dispersion, and not diffusion, is modelled.

The Allen-salt velocity method [2] was also used successfully. It is based on the principle that the electrical conductivity of water is proportional to the concentration of sodium chloride (table salt) in the water. The use of table salt, being readily available, made this a

cost effective method. However, the electrical instrumentation required for the measurement of the conductivity was not always available. This led to the method not becoming popular.

With the advent, in recent times, of more sophisticated and more compact electrical conductivity measurement equipment, it has become possible to measure salt concentration levels very accurately. The reader may refer to Appendix E for more information regarding the conductivity meter as well as the establishment of concentration versus conductivity relationships.

5.2 Design of the experimental model

The reader is referred to the schematic diagram illustrated in figure 5.1. A seepage tank was constructed from 6mm thick clear perspex. The perspex sides were laid into an accurately constructed frame made from angle iron, the inside dimensions being 1200mm x 500mm x 500mm. The perspex sides were joined with a special gluing agent called Tensol. The corners were further sealed with a sealant.

The tank was divided into three sections: the upstream section where the upstream water level will be controlled; the aquifer section into which the aquifer medium (sand) will be placed; and the downstream section where the downstream water level will be controlled. This variation of water levels between the upstream and downstream sections

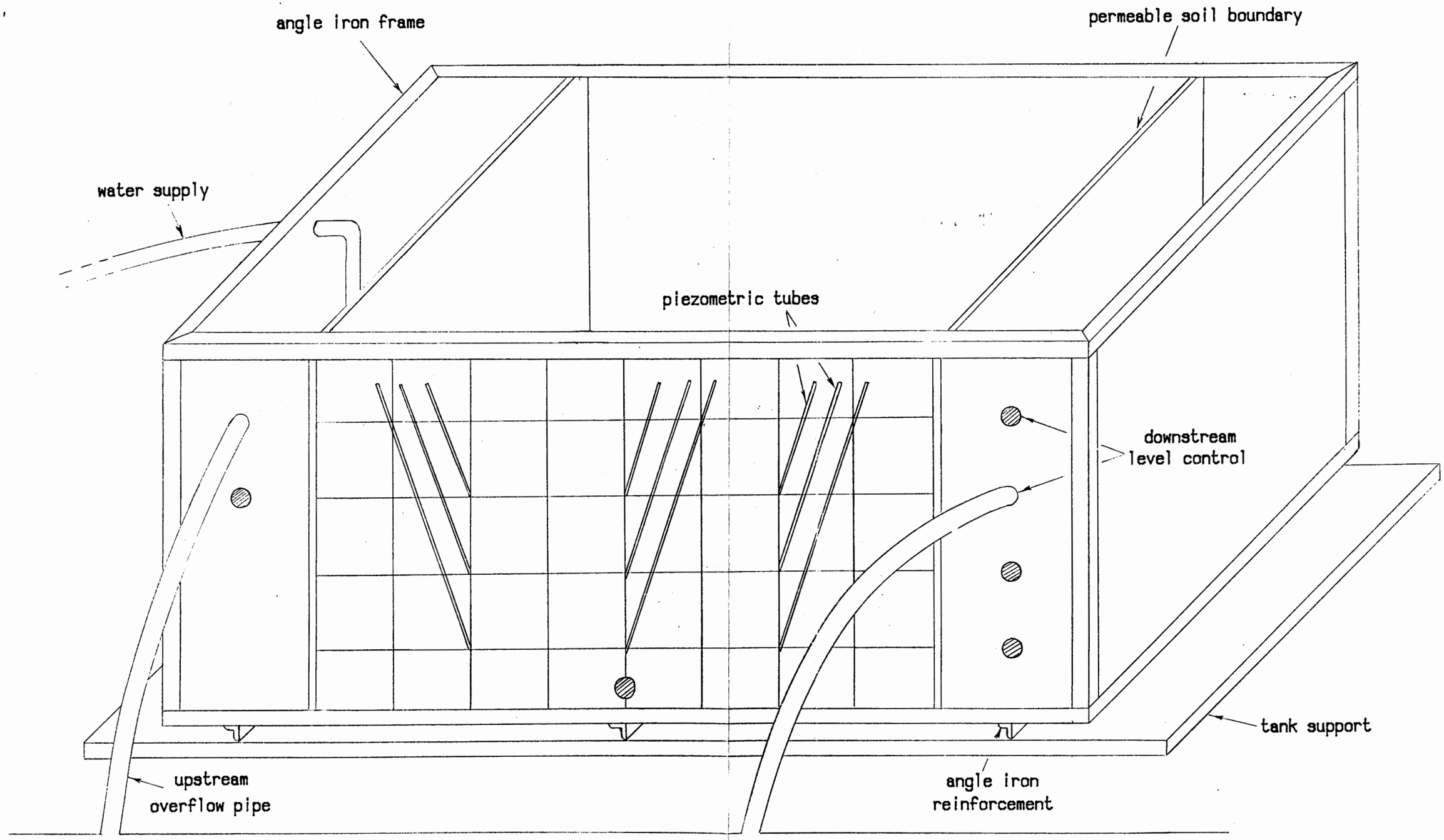


Figure 5.1 Schematic diagram of the seepage tank

will cause seepage through the aquifer to occur. The sections were divided by rigid permeable meshes covered with a geofabric. This combination will ensure the seepage of water through the boundaries of the aquifer section while keeping the aquifer intact. The positions of the meshes may be altered in order to alter the size of the aquifer. For this reason, the upstream and downstream sections were filled with stone to prevent the thrust of the sand from moving the rigid mesh.

Plumbing connections were fastened to the inlet and outlet holes. The holes were machined into the perspex with a milling machine prior to the assembling of the perspex. The inlet hole was connected to the supply tap. The overflow (outlet) openings on both the upstream and downstream sections were led to a drain. The upstream and downstream heads could be varied by changing the overflow connection to a higher or lower opening. Two openings were placed on the upstream section and four on the downstream section. Those openings not used would be closed by stoppers.

Thin glass tubes, of inside diameter 3mm, were glued onto the sides of the tank. The purpose for these were to measure the piezometric pressure head, or the saturated free surface height. Due to capillary effects inside these tubes, the readings would have to be adjusted accordingly.

The tank was first tested against certain effects. Water was allowed to flow into the tank at full supply and the following were checked:

(a) The effective drainage at the overflow openings. The outflow pipes had to be big enough to allow for the simultaneous flow of water and air. Furthermore, the openings had to be big enough otherwise the water level in the tank would rise above the opening.

(b) The extent of turbulence at the inlet end.

(c) Leakage.

Initially, a problem was experienced as regards (a), the thickness of the outflow. The size of the openings had to be increased and the plumbing and pipe connections were changed accordingly.

In order to predict the model's results as accurately as possible, the porous medium had to be carefully chosen. A specially graded sand was obtained from a sand supplier. The supplier's grading analysis showed a grading of between 0.425mm and 0.850mm with an effective particle size of 0.50mm. This data may vary from sample to sample. For this reason, a well-mixed sample of the sand had to be regraded (see Appendix F). Nevertheless, this will allow for a high degree of isotropy. Properties such as the permeability and the porosity could be found for a well-mixed sample and could then be assumed to hold, fairly accurately, for the

entire aquifer. The sand was packed in the aquifer section of the tank. A highly permeable stone was packed in the upstream and downstream sections to prevent the sand body from collapsing sideways.

In practice, an aquifer is normally polluted by polluted water entering the aquifer. Examples of this is the infiltration of leachate from a waste disposal site, and the recharging of the aquifer by a polluted stream. That is, the pollutant seldom enters the aquifer directly as may be the case with the deposition of waste into an unprotected borehole.

Salt, being the contaminant used in the experiment, was allowed to enter the model aquifer in the form of saline water recharging the aquifer via a "borehole". The borehole structure, screened with a geofabric, was movable and could be placed anywhere in the aquifer. In fact, a number of similar boreholes were placed throughout the aquifer. This allowed relatively easy measurement of the saline concentrations of the groundwater. Samples could simply be drawn out of the boreholes using pippettes which have sucking mechanisms attached to them. The conductivity (which is a function of the salinity) could then be measured using a conductivity measuring device.

The saline concentration of the incoming polluted water could easily be kept constant. This was done by preparing, beforehand, a large supply of well-stirred salt-water

mixtures. However, the task of keeping the recharge rate of the polluted water constant was not so straight forward. Figure 5.2 illustrates the apparatus used to produce a constant recharge of polluted water. A tank, referred to as a hydraulic bench, was filled with saline water of known concentration. An electric pump fitted into the hydraulic bench allowed the saline supply water to be raised to a height into a constant head tank. From this tank, the saline water was gravity fed into the aquifer via one of the boreholes. The flowrate, or rate of recharge, could be varied by means of a control valve at the outlet end of the constant head tank.

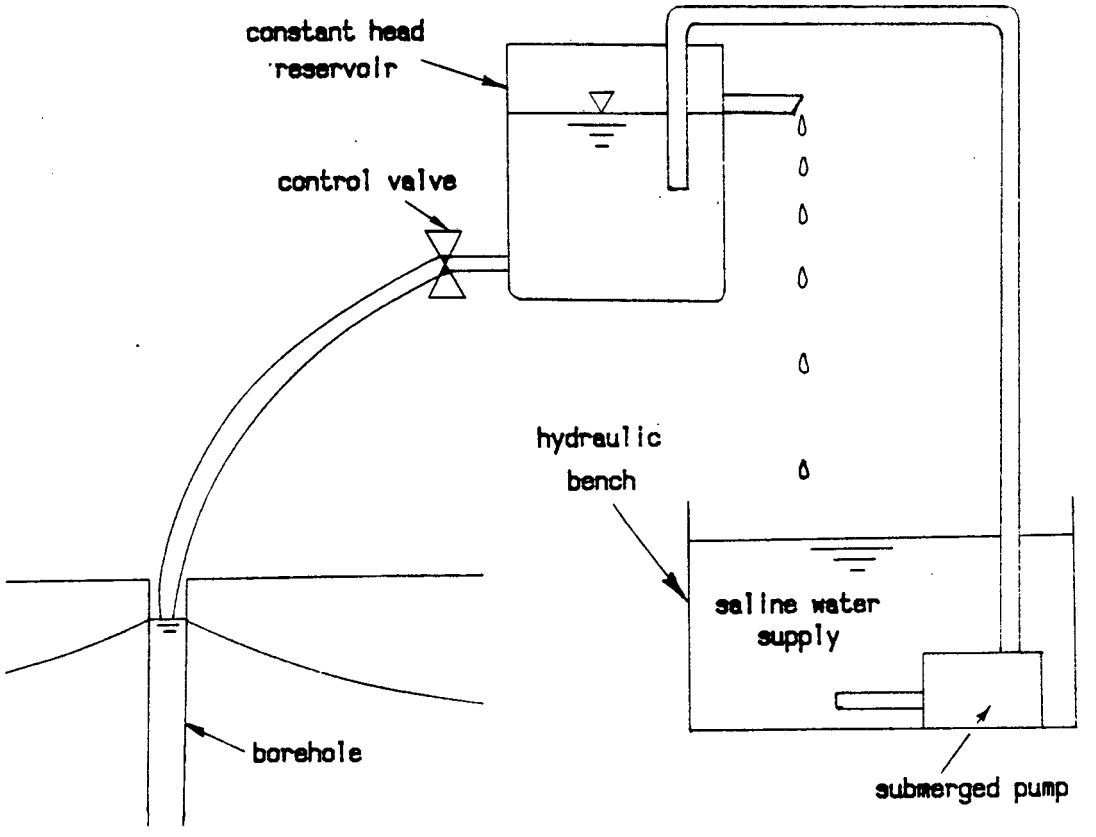


Figure 5.2 The apparatus used to recharge the aquifer with polluted water via a borehole.

A further criterion which the model had to satisfy was that of abstraction, or pumping, of water via a well or borehole. A pipe was laid along the bed of the tank from the borehole to an opening situated low down on one of the sides of the tank, as shown in figure 5.3. The opening was connected to the drain via another pipe. A control valve connected to this pipe allowed the rate of abstraction to be controlled and kept constant. As long as the rate of abstraction does not exceed the rate of recharge, the flow pattern within the aquifer will reach a steady-state condition. It is therefore not necessary to adjust the control valve with time.

Another option for the pumping of water was to maintain a constant water level (constant head) inside the borehole. This would have been easier, but it is not the case in practice.

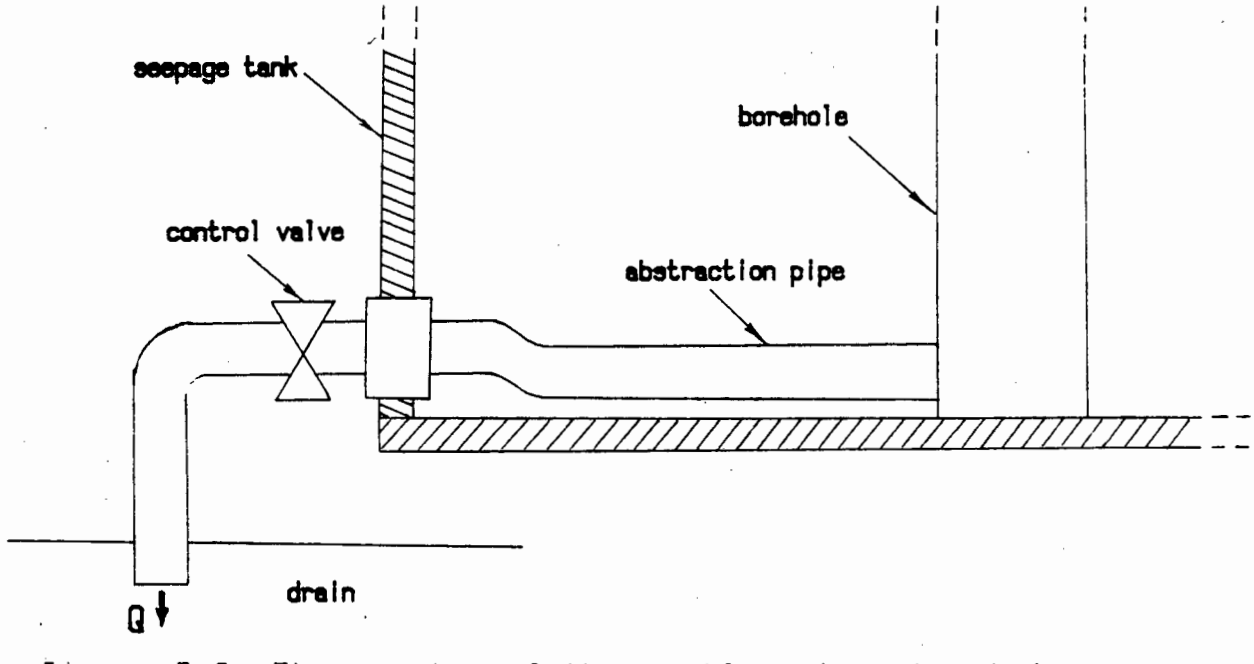


Figure 5.3 The pumping of the aquifer via a borehole

The photographs shown below were taken during the operation of the experimental model.

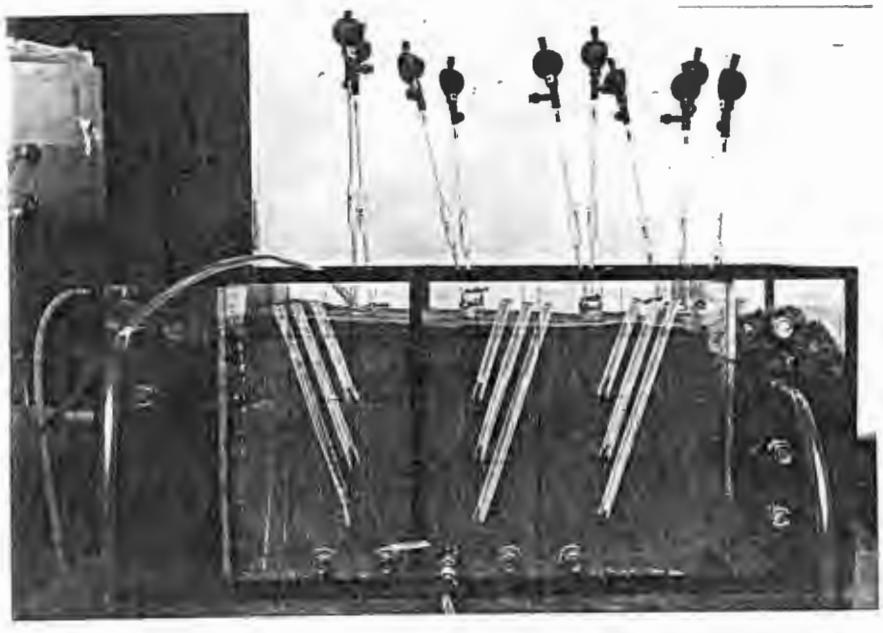


Figure 5.4 A front view of the model. The upstream end is on the left. The well discharge control valve can be seen at the bottom.

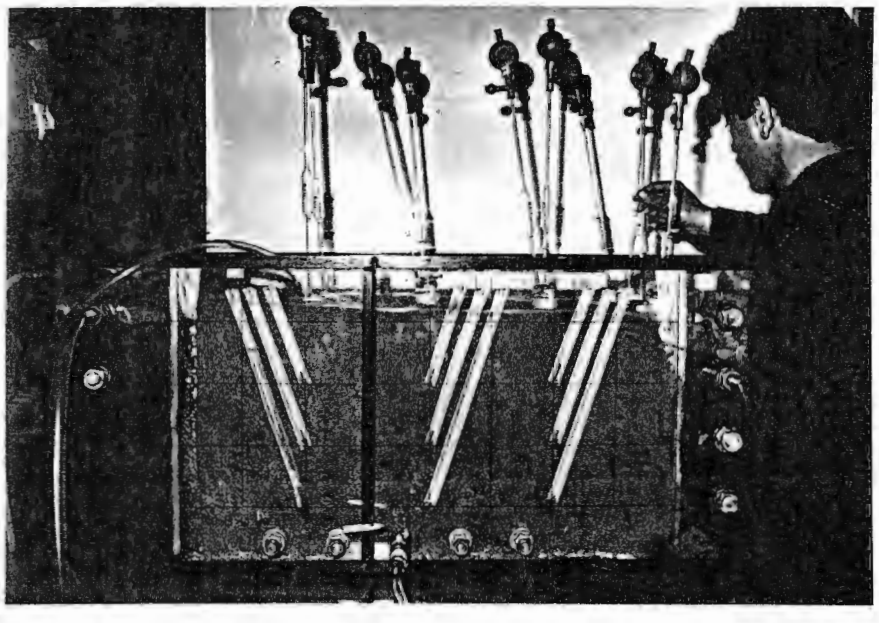


Figure 5.5 The relative size of the model.

Figure 5.6 An aerial view of the model. Pipettes, used for sampling the water, were placed inside the boreholes.

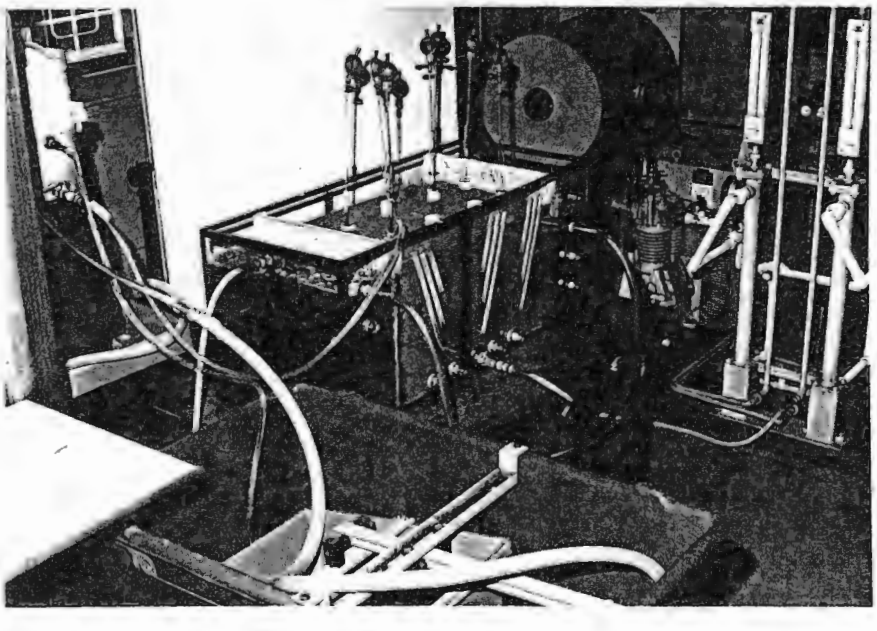


Figure 5.7 The apparatus used to recharge the aquifer model with contaminated water can be seen in the foreground and at the top left hand side. The inlet supply is connected to the rotameter seen in the top right hand corner.

5.3 The measurement of the aquifer properties

In order to compare the results obtained from the experimental model to that obtained from the Finite Element computational model, the nature and values of certain properties of the aquifer are required. Since the exact grain size and shape of the aquifer medium cannot be guaranteed, it is best to experimentally measure all of the properties. Otherwise the properties could be obtained from experimental tabulations produced by other researchers. Furthermore, some of these properties are functions of a number of variables, making it necessary to determine them experimentally. For example, the coefficient of diffusion, D, depends on the aquifer medium as well as the fluid and the chemical properties of the spreading contaminant.

This section serves to introduce the aquifer properties and to discuss the methods to be employed in their measurement.

(a) Coefficient of permeability K

The coefficient of permeability, or hydraulic conductivity, is a measure of the ability of a material to allow the passage of fluid through it. It is defined as the volume flowrate of the fluid through a unit cross-sectional area under a potential gradient of unity. (Units = m/s).

Theoretical relationships, such as those suggested by Fair and Hatch [1] and Kozeny-Carmen [1], can be used to determine the coefficient of permeability. These

relationships, however, require the values of the particle shape factor, the packing factor, the geometric mean diameter of the grains, etc. Since these parameters have to be obtained experimentally, it is easier to experimentally measure K directly.

There are two commonly used laboratory methods for assessing the coefficient of permeability. These are the constant head and the falling head permeameter tests. The constant head permeameter is used for granular materials such as gravels and sands which have high enough porosities, whereas the falling head permeameter is used for fine sands, silts and rock types. In the experimental model, medium to coarse sand will be used as the aquifer medium and, therefore, the constant head permeameter test will be used to determine coefficients of permeability.

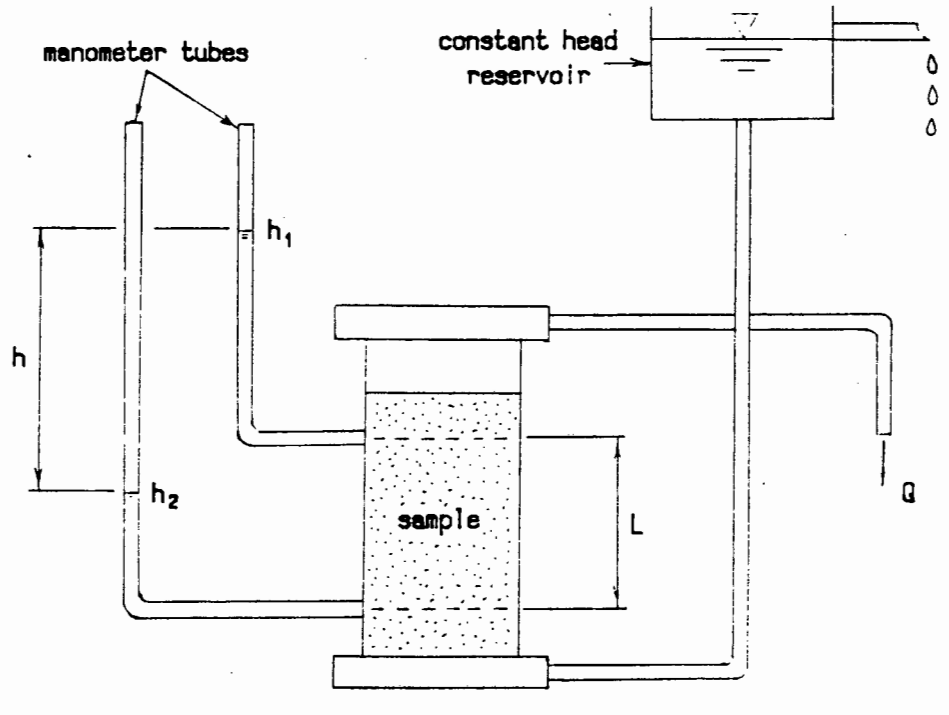


Figure 5.8 The constant head permeameter.

The constant head permeameter test is illustrated in figure 5.8. A sample of the medium is placed in a cylinder of cross-sectional area A and water is allowed to pass through it under a constant head h. The amount of water discharged, Q, in a given time t as well as the hydraulic gradient (i.e. the difference in head h_1-h_2 over a given length L) measured by means of manometer tubes, are obtained. Using Darcy's law applicable to a constant hydraulic gradient

$$\frac{Q}{At} = k\frac{h_1 - h_2}{L} \qquad (5.1)$$

a value for k can be obtained. The value of k, however, depends on the packing of the sand. The packing may vary from dense to loose. In order to estimate, the value of k for the experiment, a more detailed investigation is required. This is given in Appendix F where the value of k for the sand to be used in the experiment is determined.

(b) Porosity n

The porosity of a medium is the percentage pore space per given volume. It is therefore dimensionless and can be expressed as

$$n = V_v/V \qquad (5.2)$$

where V_v is the volume of the voids and V is the entire volume. Besides being a function of the grain size and shape, the porosity also depends on the degree of compaction

of the medium. Variations in the porosity is therefore expected with variations in depth of the aquifer due to gravitational compaction. Whether or not this variation is linear is not known. Intuitively, though, the relationship is thought to be exponential with the porosity becoming relatively constant lower down the aquifer.

The method to be used to experimentally determine the porosity is the standard saturation method. Water is allowed to flow into a sample of known volume. After the sample is saturated, the amount of water that entered the sample is taken equal to the void volume. This gives an effective porosity since the water may not be able to penetrate all the pores. Porosity is closely related to another property called the void ratio, the latter being the ratio of the volume of voids to the volume of solids. By knowing either one, the other could simply be calculated (see Appendix E).

Porosity and permeability are the two most important properties governing the migration of fluid through a porous medium. These properties may vary with the depth of the medium, especially in the case of heterogeneous and anisotropic media.

(c) Specific storage S_S

The specific storage of an aquifer is defined as the volume of recharge/discharge of water per unit bulk volume of the aquifer per unit increase/decrease in potential head.

(Units = 1/m). It is related to the coefficient of storage, or storativity, S, as follows

$$S_s = S/H \qquad (5.3)$$

where H is the saturated thickness of the aquifer. For unconfined aquifers, which will be simulated by the experimental model, the storativity virtually corresponds to the effective porosity [25]. In the case of discharge, it is better to make the storativity equal to the specific yield since all the water is not discharged when the water table is lowered [1].

(d) Specific yield S_y

The measurement of the specific yield is necessary in order to determine the specific storage of an aquifer. It is defined as the volume of water that can be drained by gravity per unit saturated volume of the aquifer. Its measurement in the laboratory can easily be made by simply allowing a saturated sample to drain under gravity and recording the amount of water drained. The ratio of this amount to the volume of the sample is then the specific yield of that medium.

(e) Coefficient of diffusion D

The coefficient of diffusion (or dispersion) is the constant of proportionality in the law of diffusion. (Units = m^2/s). It has a unique value for a particular solid diffusing in a

particular medium, and is found to be dependent upon [6]

(i) the size of the diffusing molecule

(ii) the viscosity of the fluid

(iii) the temperature of the fluid

(iv) the shape and size of the grain comprising the porous medium

(v) the concentration of the solute

(vi) the adsorption of the solute by the porous medium

Theoretical relationships are therefore too complex to use or too simplified resulting in only an approximation of D. It is therefore best to experimentally determine the value of D.

Figure 5.9 illustrates the apparatus that could be used in the measurement of D. A sample of the medium of length L is placed midway inside a cylinder, separated from the bottom section by a permeable structure. The cylinder is then filled with water until the sample is saturated and the water level is at the top end of the sample. A conductivity probe is lowered, through the sample, into the bottom section. A layer of contaminated (saline) water of known concentration is added to the cylinder at time t=0, and allowed to diffuse through the medium. The thickness of the added layer is arbitrary. As the solute diffuses, the conductivity of the water in both the bottom and the top sections is measured against time.

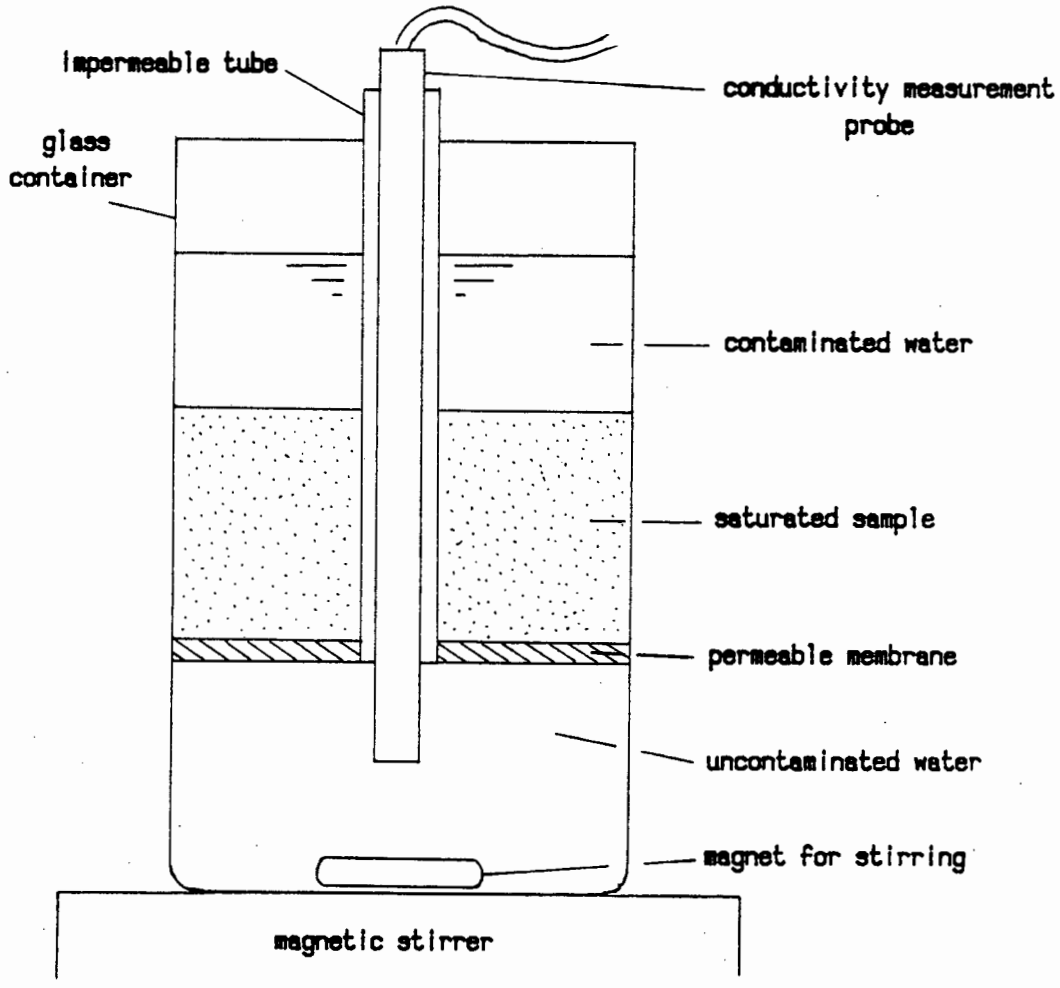


Figure 5.9 Apparatus used to measure the coefficient of diffusion of a solute in water in a porous medium

Steady state diffusion of solute occurs in a medium when the two flux boundaries are held at constant concentrations. The relationship governing the diffusion process is then given by the simple linear law of diffusion [6]

$$\frac{M}{A\Delta t} = D\,\frac{C_0 - C_L}{L} \qquad (5.4)$$

where M is the mass of solute diffusing through the medium of length L in a direction normal to the area A in a time Δt. D is the coefficient of diffusion. The concentration at the boundary where the solute enters the medium is held

constant at C_0 and that at which the solute leaves the medium is held constant at C_L. An important feature of steady state diffusion is the fact that the amount of solute entering the medium is equal to the amount leaving the medium. In the experimental method described earlier, the diffusion is not steady since the concentrations at the boundaries are allowed to vary, as in the case of a transient diffusion problem. However, the transient case does reach a state at some instant, or instances, in time where the percentage difference between the solute entering the medium and that leaving the medium is minimal. This may be described as being, momentarily, a steady state. Although the concentration distribution within the medium will be exponential, the effect at the boundaries is linear.

The curves plotted in figure 5.10 illustrate the variation of flux at the inlet and outlet boundaries. At first, the solute entering is much higher than that leaving. As the medium becomes saturated with solute, the flux at the outlet boundary approaches that at the inlet boundary. Eventually, both will converge to zero as the solute becomes evenly diffused (steady state).

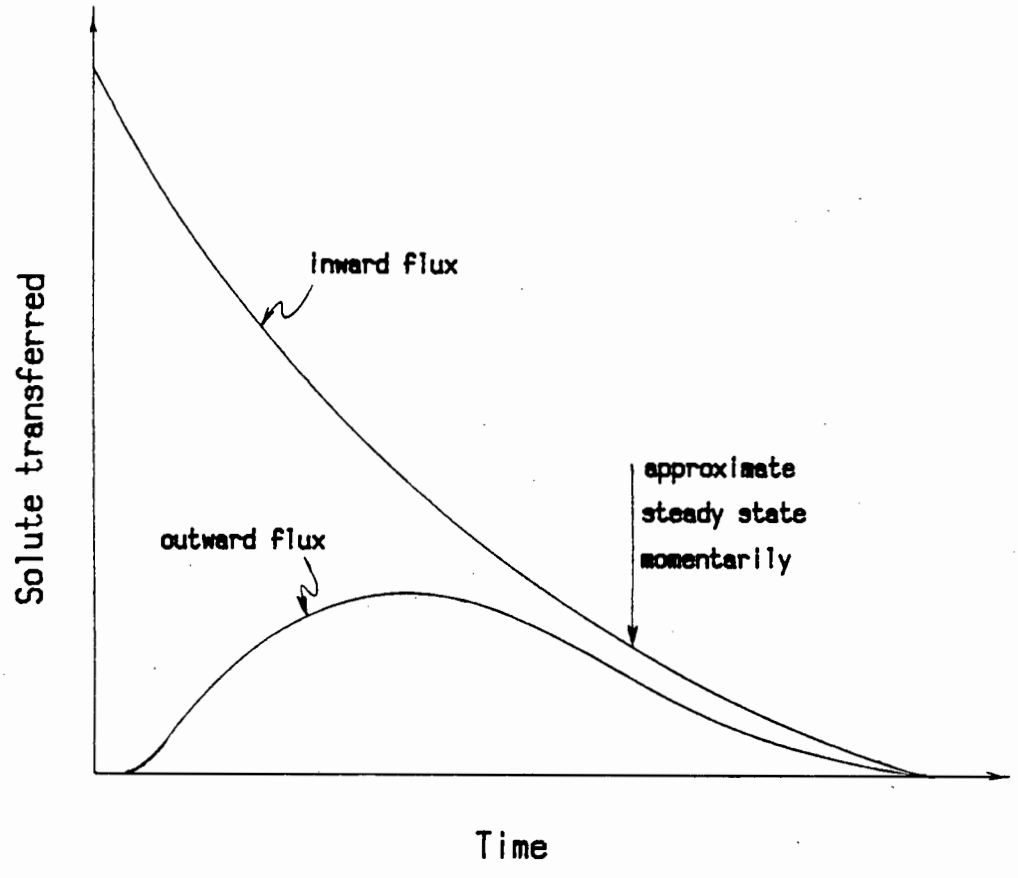


Figure 5.10 Transient solute transfer through a medium.

saturated with solute, the flux at the outlet boundary approaches that at the inlet boundary. Eventually, both will converge to zero as the solute becomes evenly diffused (steady state).

The mass of solute transferred can be calculated by using either the inlet or the outlet boundary conditions. Using the outlet boundary, the mass of solute leaving the medium in a time Δt is therefore given by

$$M = (C_L{}^{t+\Delta t} - C_L{}^{t}) V_d$$

where the subscript d refers to the downstream end. Substituting this into equation 5.4 and approximating the concentrations at the boundary by a finite difference time

derivative approximation, the following relationship results

$$\frac{(C_L^{t+\Delta t} - C_L^{t})V_d}{nA_d\Delta t} = D\left[\frac{(C_0^{t+\Delta t} + C_0^{t})/2 - (C_L^{t+\Delta t} + C_L^{t})/2}{L}\right] \quad \ldots\ldots\ldots \quad (5.5)$$

from which the diffusion coefficient can easily be calculated. The values of concentration are those tabulated at the beginning and end of the time interval within which the momentary steady state occurs. The determination of the coefficient of diffusion for NaCl diffusing in water through a 20/40 Density Grade sand is given in Appendix G.

Note that the temperature of the solvent influences the rate of diffusion and, therefore, the coefficient of diffusion. Changes in the water temperature should therefore be monitored and the value of D adjusted accordingly. This compensation due to temperature variations can be done automatically if the conductivity measuring device has such a compensation facility.

Chapter 6

Comparison and discussion of experimental and computational results

6.1 Introduction

An unconfined aquifer, shown in Figure 6.1, was simulated using the experimental model. The results of two experiments are presented in this chapter. These are compared to the results generated by the computational Finite Element model.

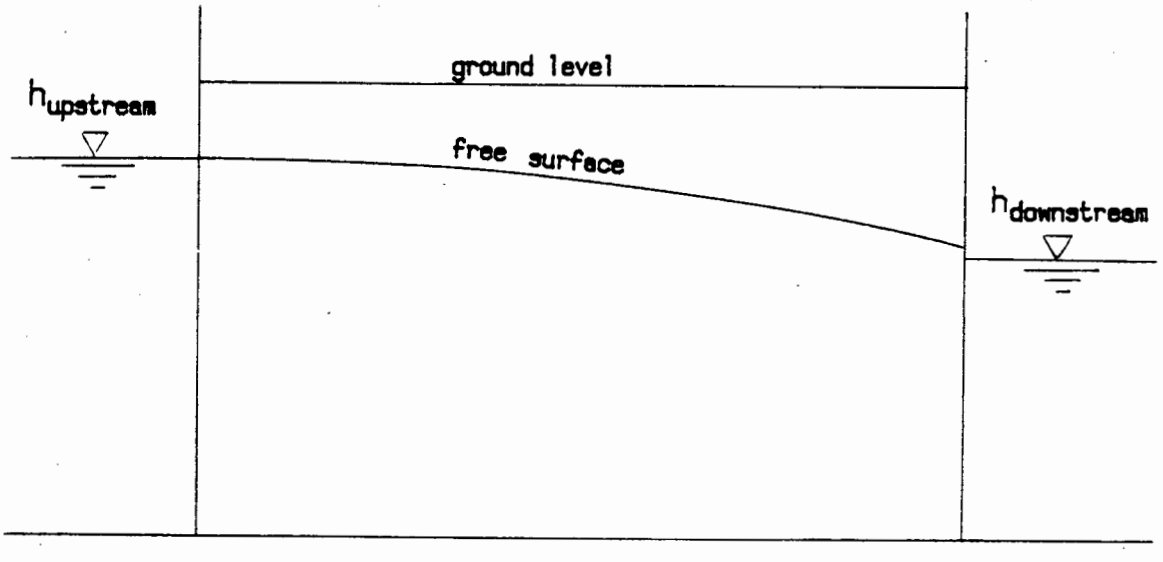


Figure 6.1: The unconfined aquifer simulated in the experimental model.

Ten boreholes were laid into the aquifer as shown in Figure 6.2. These boreholes were used to measure the free surface elevation as well as to obtain samples of the groundwater for contaminant concentration measurements. Two of the boreholes were also used as discharge/recharge wells.

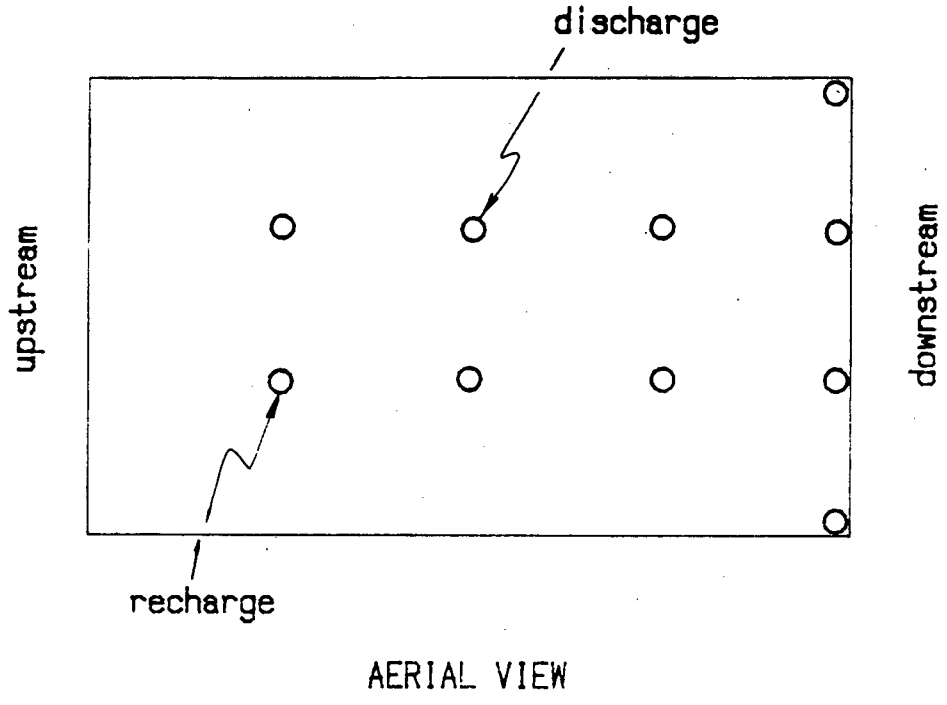


Figure 6.2: Locations of the boreholes in the experimental model

In the first experiment, a relatively high potential gradient was simulated. The purpose was to test the accuracy of the free surface elevation with that of the computational model. In the second experiment, the potential between the upstream and downstream ends was reduced enabling a more realistic modelling of groundwater velocities. In both cases, the aquifer was pumped via one of the boreholes and recharged via another.

The rate of discharge or recharge is termed the external fluid flux, Q, defined as the volume rate of discharge/recharge per unit volume of aquifer. When a volume of Q is specified at a surface node, the volume is applied to all elements to which that node belongs. Thus the volume of Q must be divided by the number of elements adjacent to the well point. Furthermore, the flowrate in m^3/s must be divided by the volume of the element into which it is entering. Since the water is discharged/recharged at one node of the 8-noded element, only one eighth of the elemental volume was considered as the recharge/discharge volume. That is:

$$\text{External fluid flux } Q = \frac{1}{4}\left[\frac{\text{flowrate in } m^3/s}{(\text{elemental volume})/8}\right]$$

The actual coefficient of permeability of the soil is not known. However, the extreme coefficients of permeability, i.e. for a dense soil packing and a loose soil packing, are known from experimentation (see Appendix F). The approach was to run the computational model using these extreme values. This generated two sets of results for the extreme cases of permeability. The experimental aquifer will have a K value between K_{dense} and K_{loose}. This, however, does not necessarily mean that the experimental results should lie in between the two sets of computational results. It would be true if the K value of the experimental aquifer is constant throughout the aquifer, that is, in the case of a consistent

soil compaction or isotropic medium. Variations in the soil packing will result in variations in the value of K. This may produce experimental results that are not bounded by the two sets of computational results.

6.2 Experiment number 1: Free surface drawdown of an unconfined aquifer

In experiment number 1, the following condition prevailed (refer to figure 6.2):

- Upstream potential head h_u = 0.368 M
- Downstream potential head h_d = 0.307 M
- Recharge at node 92 Q_R = 0.541/min ≡ 0.0066 l/s
- Discharge at node 88 Q_D = 0.471/min ≡ 0.0048 l/s

In the computational model, the domain was discretized into 48 elements and 100 nodes, as shown in Figure 6.3. A specific storage coefficient of 1.00 m^{-1} was used with coefficients of permeability of 0.152 m/s and 0.276 m/s. The same value of k was used in all three directions because of the assumed isotropic nature of the porous medium. The free surface patterns obtained as well as the detailed analysis of the input and output data are given in Appendix H.

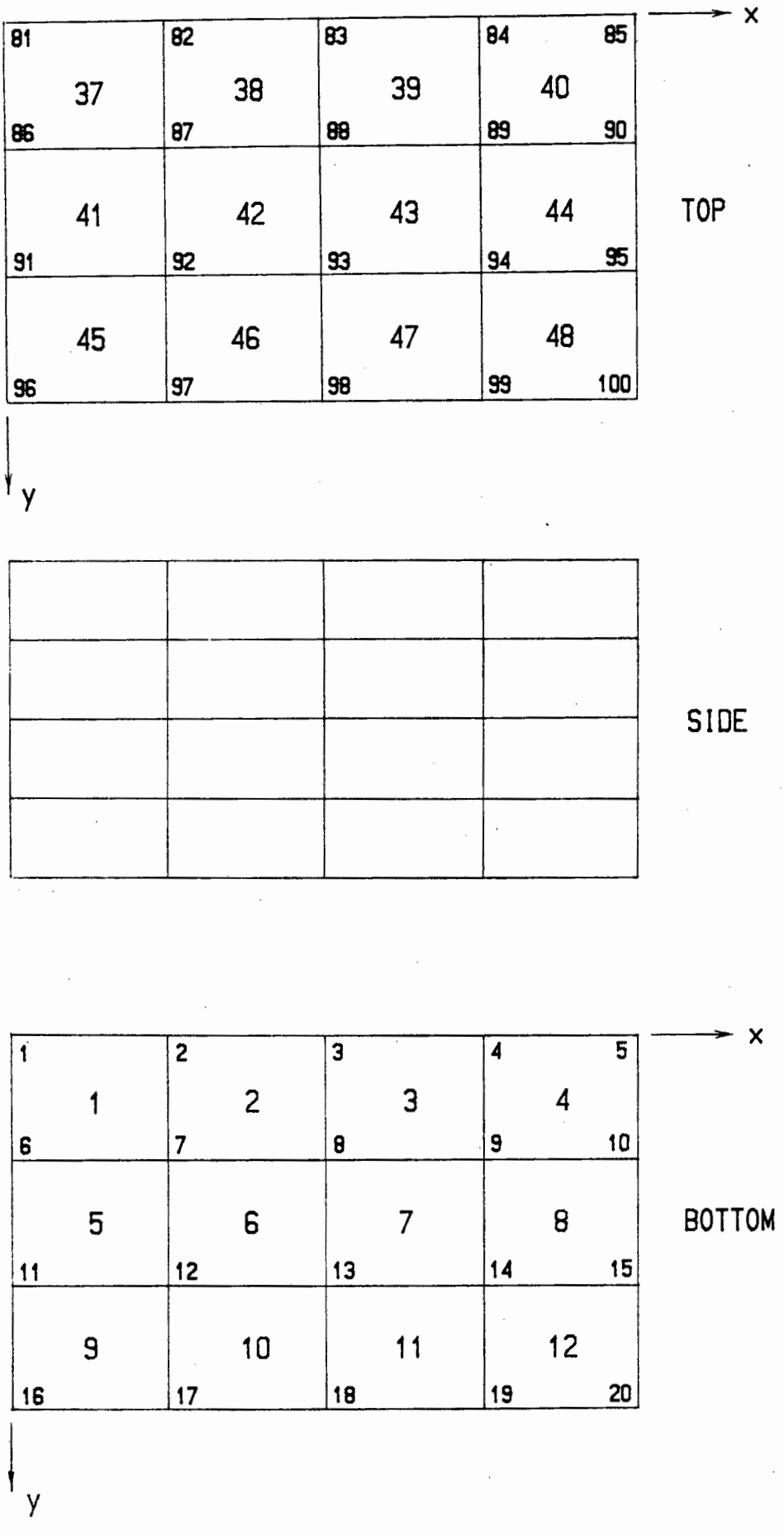


Figure 6.3: Discretization of the aquifer in experiment number 1

A comparison of the free surface elevations obtained from both the experimental and the computational model is shown in Table 6.1.

Table 6.1: Comparison of results for experiment number 1

NODE	POTENTIAL (m)		
	Computational Model k=0.00152 m/s	Experimental Model	Computational Model k=0.00276m/s
87	0.354	0.349	0.353
88	0.312	0.319	0.318
89	0.323	0.318	0.323
92	0.378	0.387	0.368
93	0.335	0.339	0.336
94	0.319	0.321	0.320

Possible causes of error in the experimental results may be ascribed to the following factors:

(a) The porosity, and therefore the coefficient of permeability, of the soil may not be consistent throughout the domain, giving rise to an anisotropic domain.

(b) The 3 cm diameter boreholes have finite areas which are not negligible in comparison with the rest of the aquifer. The boreholes provide a medium with a much higher permeability adding to the anisotropy of the aquifer. In the computational model, on the other hand, the boreholes are modelled as nodal points which do not have finite areas.

(c) Water was supplied to the upstream end of the aquifer from the mains water supply via a rotameter. Although the rotameter was closely monitored and adjusted against fluctuations in the mains water supply pressure, the elevation of the free surface may have experienced minor fluctuations.

(d) The method used to measure the water levels in the boreholes had an estimated tolerance of ± 0.002 m.

Nevertheless, the results are within acceptable limits of accuracy. A computer generated three-dimensional view of the aquifer with its free surface drawdown pattern is shown in figure 6.4

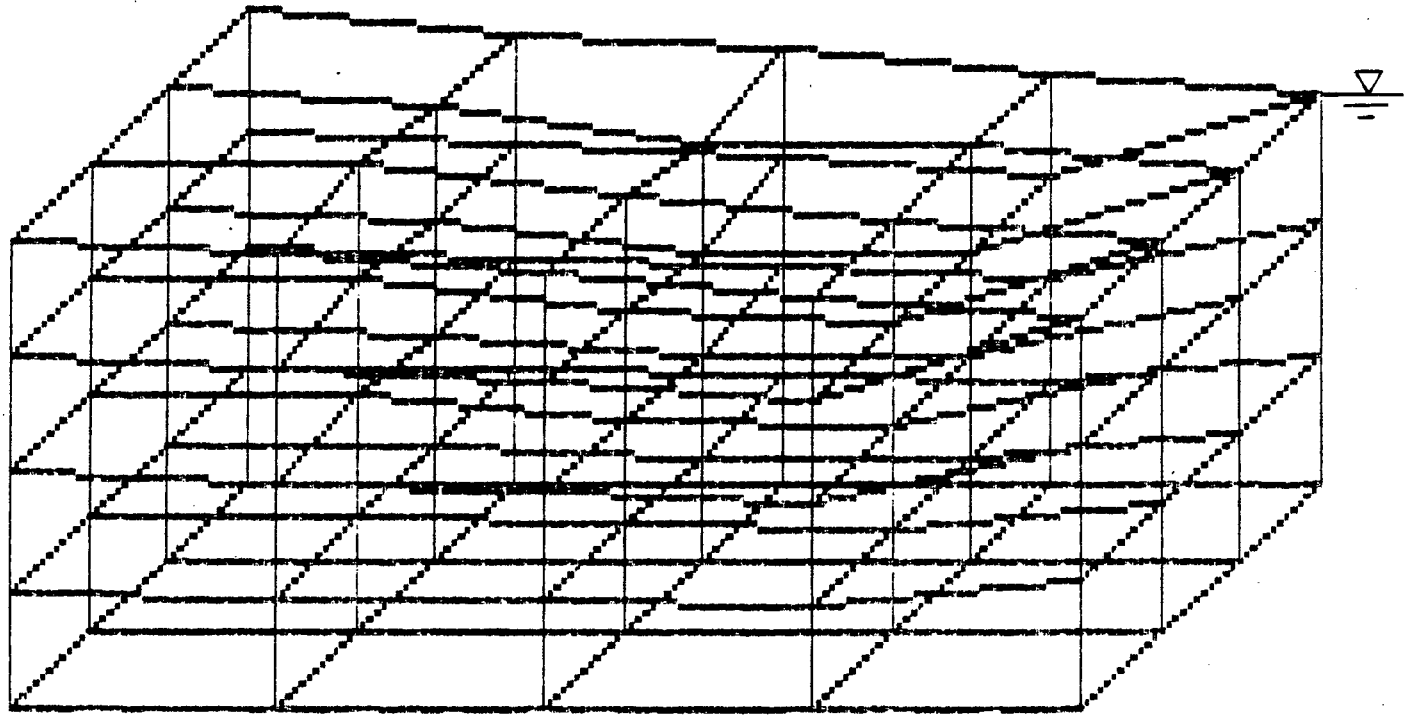


Figure 6.4 The computer generated aquifer domain for experiment number 1 after drawdown (k=0.00152m/s)

6.3 Experiment number 2: Contaminant transport in an unconfined aquifer

The transport of a contaminant in a porous medium is affected by the permeability of the medium in that the permeability determines the magnitudes of the fluid velocities which in turn gives rise to the convection of the contaminant. Furthermore, the contaminant transport is a function of the porosity of the medium which affects the area of diffusion and convection. In the type of medium used, the porosity is a direct function of permeability. This means that the computational model has to be executed twice (as in experiment number 1) for the extreme cases of permeability/porosity.

In the three-dimensional theoretical (computational) model, a contaminant source is defined as an area through which a contaminant passes (contaminant flux). This area therefore has to be non-zero in order for the contaminant to enter the domain (as is the case in practice). It is therefore not possible to model a point source in three-dimensional analysis. This can be seen from problems modelled by Taylor [14].

Although the contaminant entered the aquifer via a well, this cannot be modelled as a point source at a nodal point, but rather as a source spanning the side of an element. (It is possible to model point and line sources in one- and two-dimensional analysis since the sides of the elements are

points and lines respectively.) This also applies to the internal and external fluid fluxes. Although the external fluid fluxes (recharge/discharge at wells) were specified at nodal points, these fluxes are in effect linearly distributed across the surface where their values at adjacent nodes are taken as zero.

In this experiment, the water recharging the aquifer was contaminated with a known concentration of NaCl. This water was pumped from a large supply of saline water prepared beforehand. This enabled a constant contaminant source to be modelled at the recharge well.

The domain was discretized into 43 elements and 102 nodes as shown in figures 6.5a and 6.5b. The following conditions prevailed:

- Upstream potential head h_u = 0.310 M
- Downstream potential head h_d = 0.302 M
- Recharge at node 15 Q_R = 0.541/min ≡ 0.0066 l/s
- Discharge at node 33 Q_D = 0.471/min ≡ 0.0048 l/s

In the computational model, a concentration of unity was specified at the source, whereas in the experimental model, the source concentration had an arbitrary value which remained reasonably constant (a maximum fluctuation of ±4% was recorded over a time period of 16 hours). The electrical conductivity of the borehole samples were measured and recorded in S/cm, and then converted to units of grams of NaCl per litre of water (g/l). These values

were then expressed relative to a unit source concentration by dividing by the value of the source concentration.

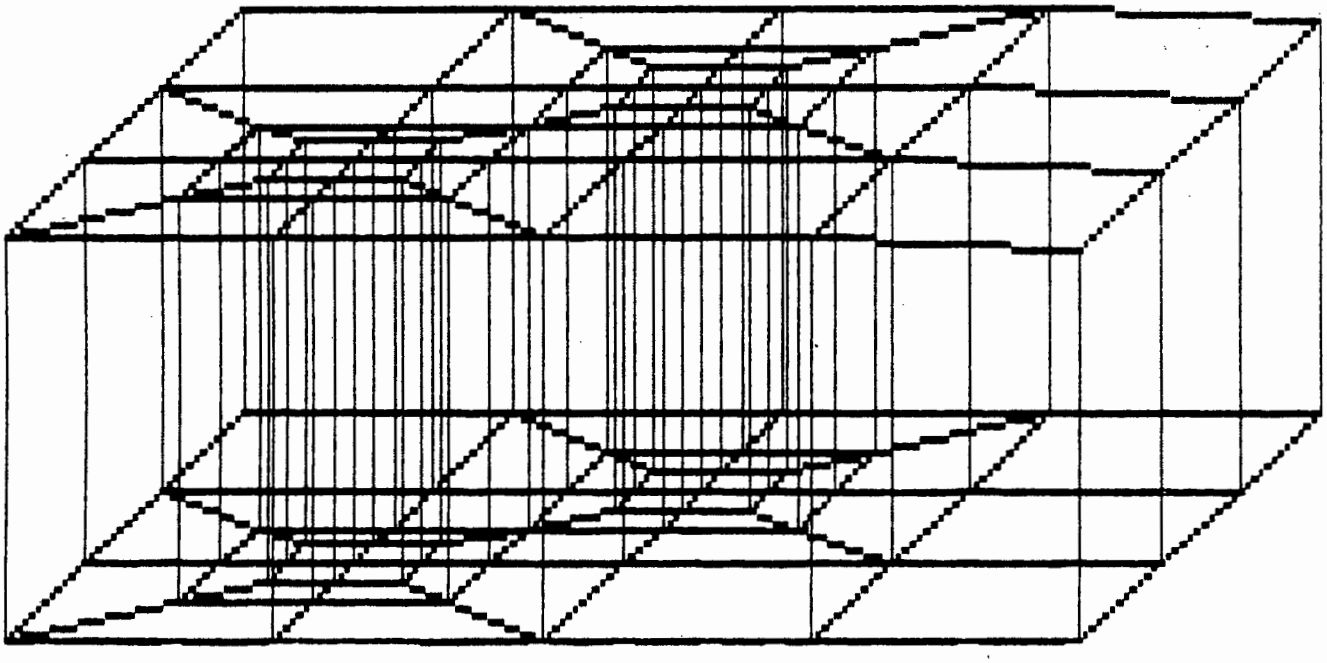


Figure 6.5a The computer generated discretization pattern for experiment number 2

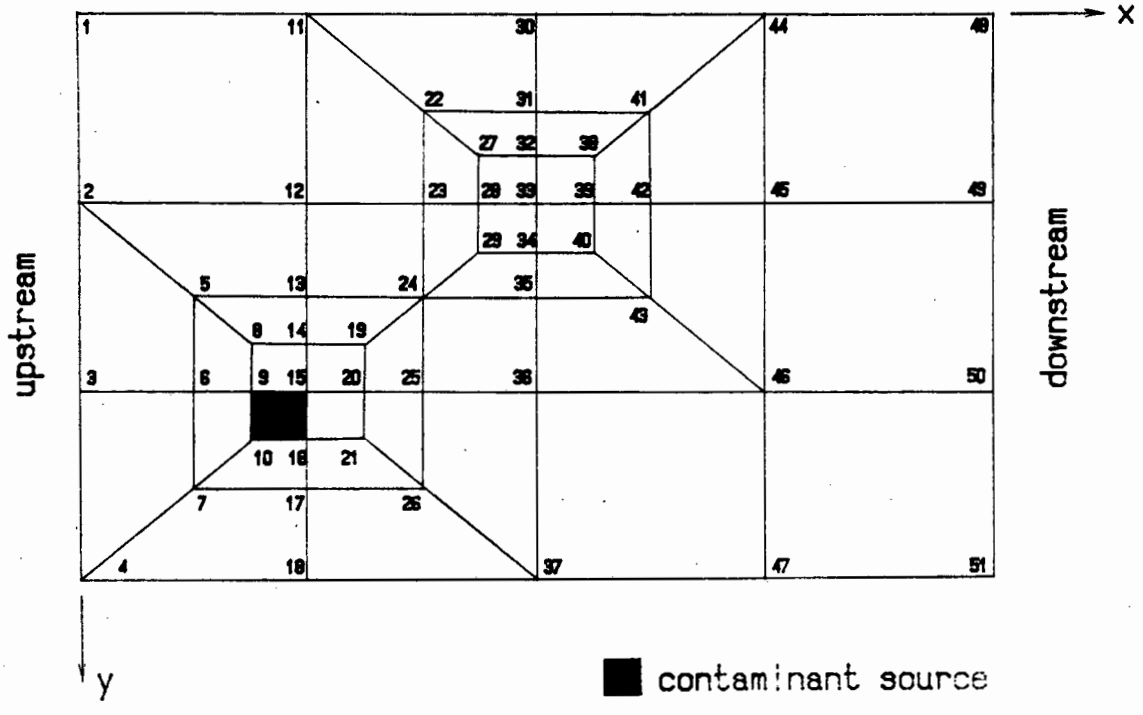


Figure 6.5b The free surface nodes of experiment number 2

A comparison of the results obtained from both the experimental and the computational models is given in table 6.2. The results, based on a coefficient of diffusion of $3x10^{-7}$ m^2/s (as determined in Appendix G), are tabulated only for certain selected nodes where sampling of the groundwater was possible i.e. where the boreholes were laid. The more detailed input and output listings are given in Appendix I.

The contaminant contour maps shown in figures 6.6 and 6.7 show the movement and slope of the contaminant front along the free surface. It is worth noting the tendency of the contaminant to move towards the well. This effect is more pronounced in the lower permeability case because of generally lower fluid velocities in the downstream (x) direction. In the lower permeability case, the potential heads in the recharge area are higher, as expected. This results in higher contaminant concentrations at the upstream nodes because of fluid velocities in that direction.

Table 6.2 Comparison of results for experiment number 2

NODE	CONCENTRATION					
	TIME = 4 HOURS			TIME = 6 HOURS		
	FE MODEL k=0.00152m/s	EXPERIMENTAL MODEL	FE MODEL k=0.00276m/s	FE MODEL k=0.00152m/s	EXPERIMENTAL MODEL	FE MODEL k=0.00276m/s
48						
12						
33						
45						
49						
15	1.00	1.00	1.00	1.00	1.00	1.00
36	0.12	0.13	0.47	0.35	0.42	0.84
46						
50						
51						
	TIME = 8 HOURS			TIME = 10 HOURS		
48						
12						
33						
45						
49						
15	1.00	1.00	1.00	1.00	1.00	1.00
36	0.58	0.70	0.97	0.74	0.83	0.98
46						
50						
51						
	TIME = 12 HOURS			TIME = 14 HOURS		
48				0.00	0.00	0.00
12				0.15	0.30	0.07
33				0.45	0.44	0.30
45				0.08	0.26	0.35
49				0.00	0.05	0.18
15	1.00	1.00	1.00	1.00	1.00	1.00
36	0.85	0.96	0.96	0.92	0.98	0.93
46				0.27	0.22	0.75
50				0.00	0.04	0.40
51				0.00	0.19	0.26
	TIME = 16 HOURS			TIME = 18 HOURS		
48	0.00	0.18	0.00	0.00	0.34	0.00
12	0.14	0.36	0.10	0.13	0.73	0.13
33	0.48	0.46	0.31	0.51	0.50	0.32
45	0.14	0.33	0.39	0.21	0.38	0.41
49	0.01	0.40	0.25	0.03	0.54	0.31
15	1.00	1.00	1.00	1.00	1.00	1.00
36	0.96	0.98	0.91	0.98	0.98	0.91
46	0.38	0.49	0.79	0.49	0.85	0.81
50	0.03	0.31	0.52	0.09	0.56	0.60
51	0.03	0.81	0.35	0.09	0.93	0.41

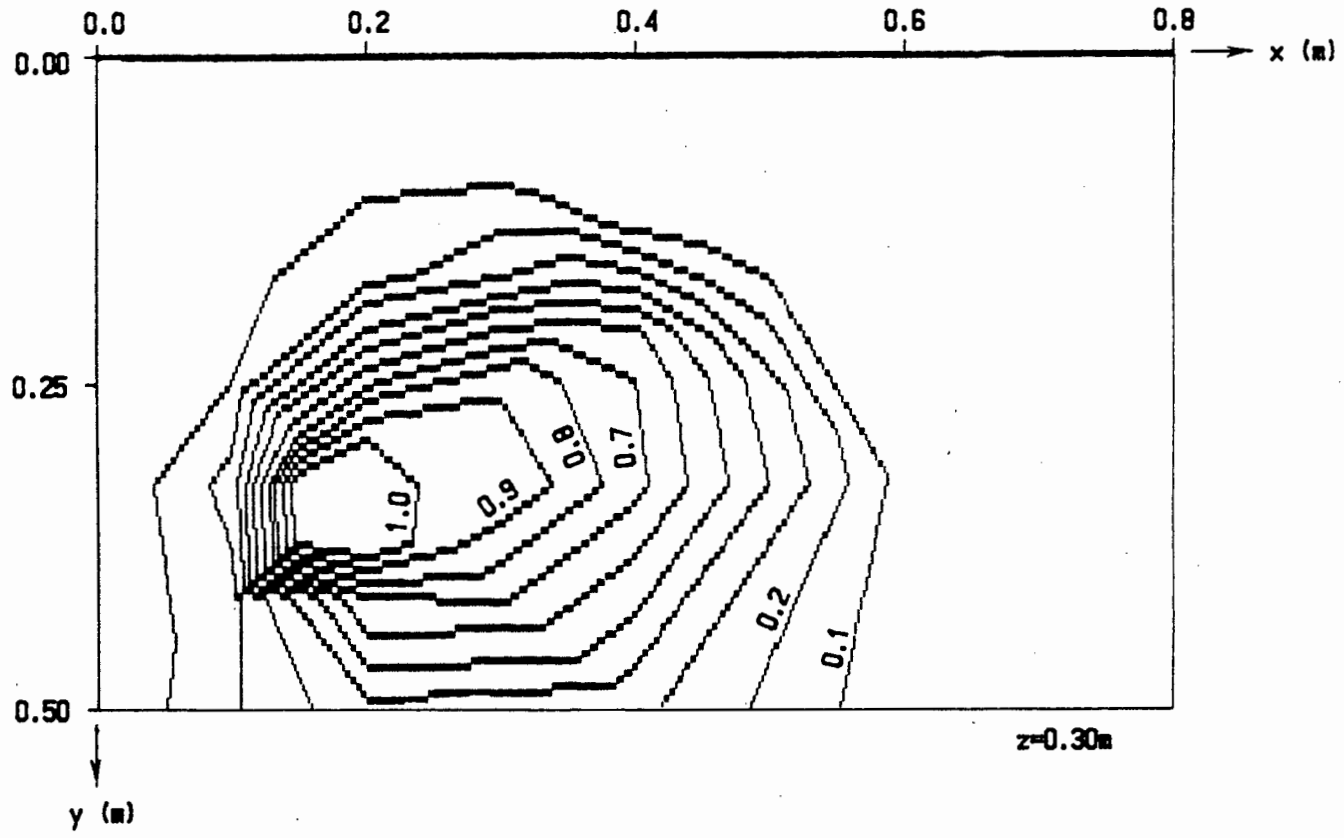


Figure 6.6a Contaminant contour map at t=10 hours for K=0.00152m/s

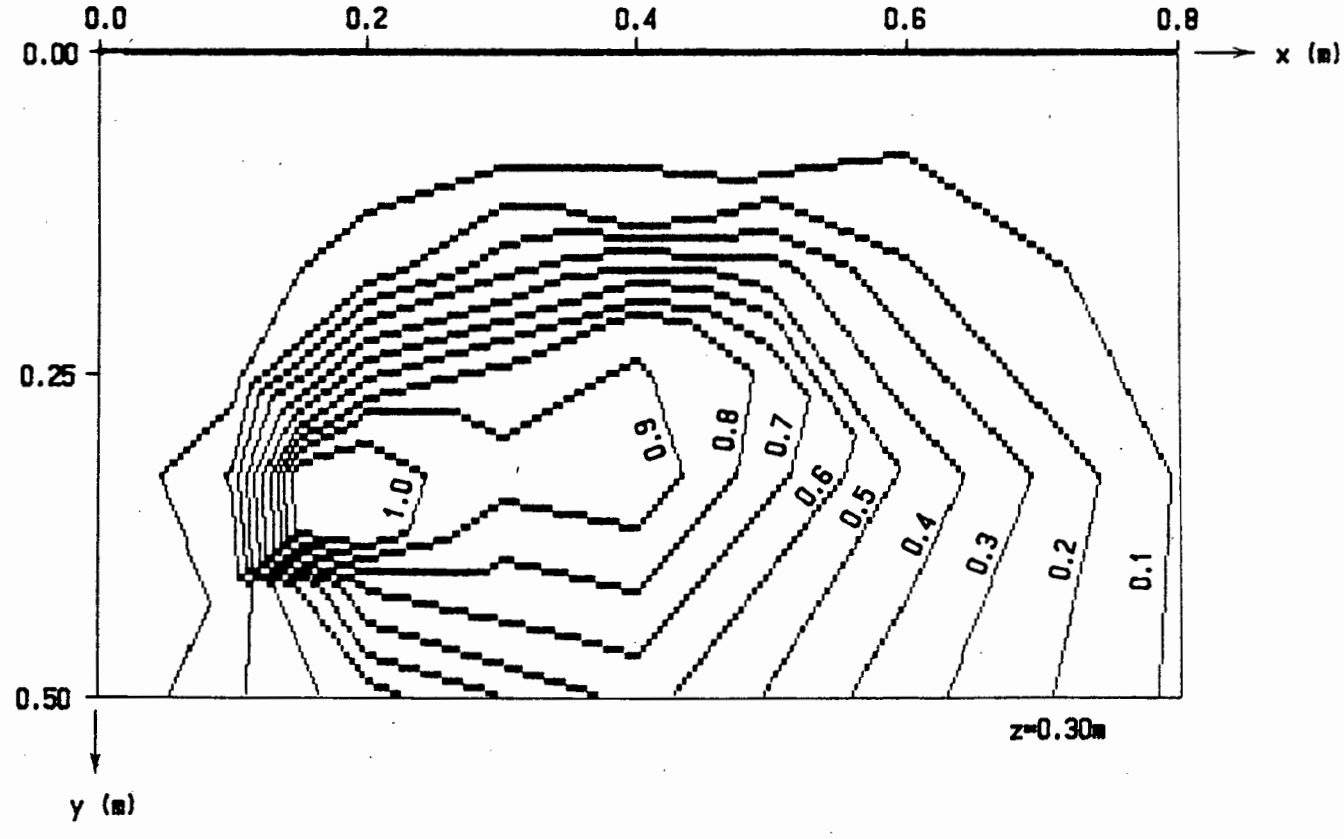


Figure 6.6b Contaminant contour map at t=18 hours for K=0.00152m/s

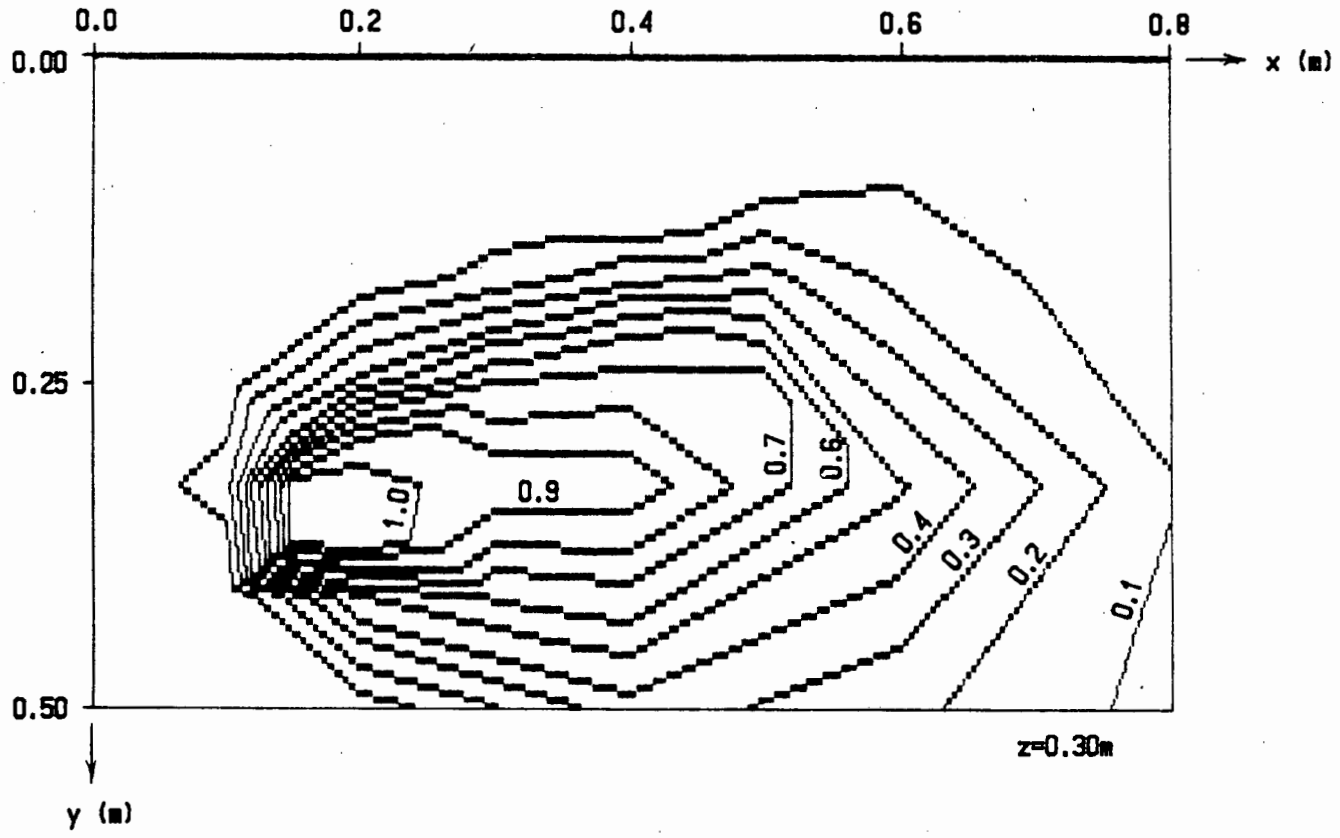


Figure 6.7a Contaminant contour map at t=10 hours for k=0.00276m/s

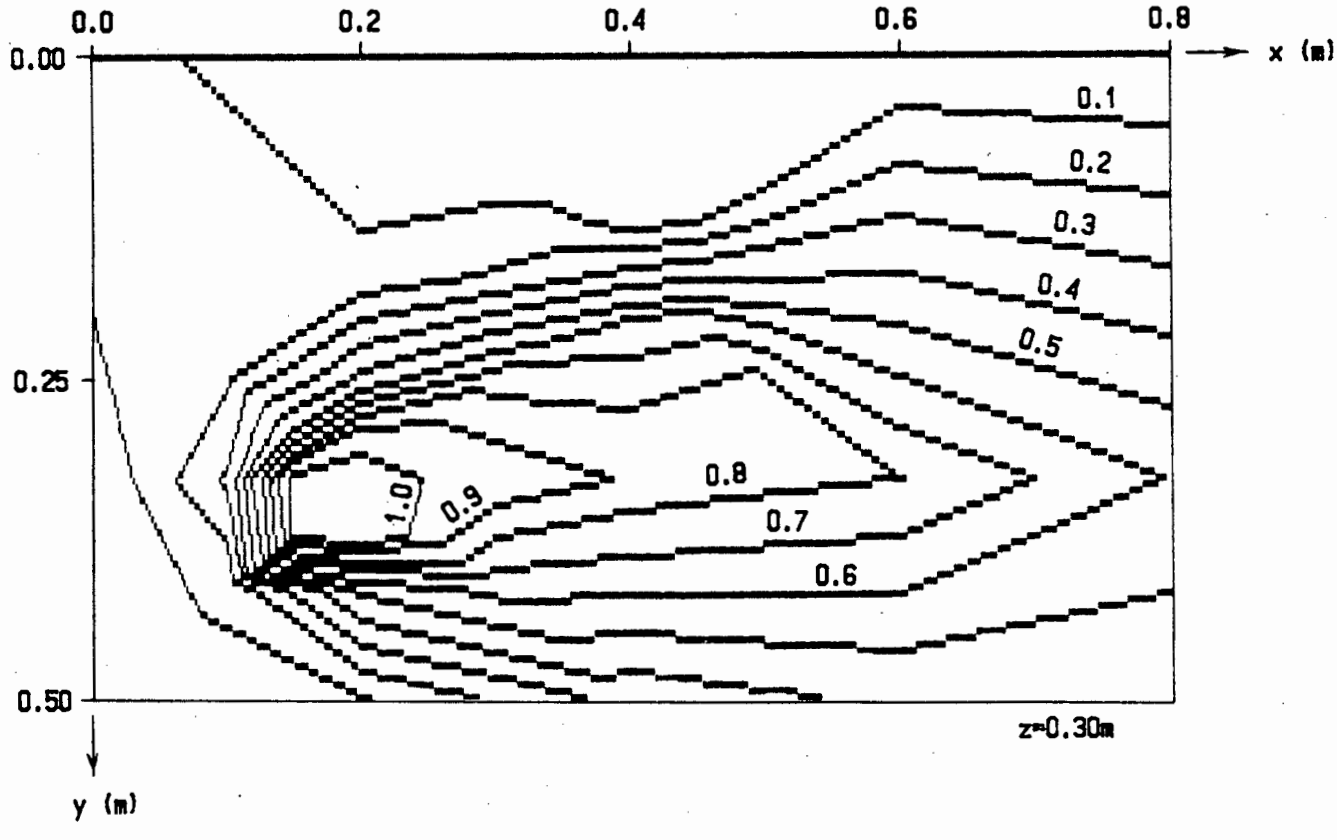


Figure 6.7b Contaminant contour map at t=18 hours for k=0.00276m/s

The following factors may be regarded as possible causes of error in the results:

The Experimental Model

(a) In practice, boreholes are widely spaced and have a negligible area compared to the rest of the aquifer. This is not the case in the experimental model. The boreholes provide a medium with a much higher coefficient of diffusion than that of the rest of the aquifer. This causes a tendency for the contaminant to travel up or down the borehole (depending on the fluid velocities).

(b) The contaminant in the contaminated water leaving the aquifer at the permeable boundaries may diffuse back into the aquifer instead of being washed away through the overflow pipes. This is inevitable, especially in the case of contaminant leaving the lower region of the aquifer. The contaminant would then have to travel up to the surface at the upstream or downstream sections before reaching the overflow pipe. During this period, it may diffuse back into the aquifer under favourable conditions of fluid velocity. This is the reason ascribed to the sudden increase in the concentrations at nodes 12, 49 and 51.

(c) The problem is one of convection-dominated diffusion-convection. Certain oscillations of the solution, although not very pronounced, were therefore experienced, especially in the higher permeability case. This effect can be seen in table 6.2 in the case of node 36.

(d) The contaminant source is not exactly modelled as that in the experimental model. The surface area of the contaminant source is about twice that in the experimental model. Furthermore, the initial conditions of nodal concentrations in the vicinity of the contaminant source are higher than that in the experimental model (see section 4.2.2) because of the linear interpolation of the Finite Element method. This leads to the solution being overestimated, especially in the earlier stages. This effect, however, diminishes with time. This is one of the reasons why most of the nodal concentrations are only compared as from 14 hours onwards. Another reason for this is the initial unsteady state of the flow region.

(e) Contaminant leaves the aquifer via the discharge well at node 33. This is not, and cannot be modelled as a contaminant flux boundary. This is the reason why the computational solution increases more steadily at this node compared to that of the experimental model.

Convection-dominated diffusion-convection problems display steep contaminant fronts, as shown in figures 6.6 and 6.7. This results in sudden increases of nodal concentrations over relatively short periods of time. This effect could produce high discrepancies in the comparison of the experimental and computational results. More improved comparisons are therefore possible with low convection problems.

Chapter 7

Conclusions and recommendations

The Finite Element method provides a useful approach to the understanding and assessment of groundwater contamination problems. The model performed fairly accurately against one-dimensional analytical solutions. Its accuracy in three-dimensional applied problems depends largely on the extent to which the aquifer properties are known as well as the ability to model the initial and boundary conditions correctly.

A drawback of the model was its slow execution time. This is ascribed to the computational facilities used rather than the method itself. The problem of experiment number 2 required an execution time of approximately 30 hours on a 16MHz IBM Compatible AT with a maths coprocessor. Most of the computational time is taken up by the numerical integration procedures used to establish the Finite Element matrices. In the case of free surface problems, these procedures have to be repeated because of variations in the mesh. The slow execution time creates a tendency for the user to employ meshes with less nodal points than might be necessary.

Further scope exists for the improvement of the computational model. Some of the features that could be upgraded or incorporated are as follows:

- The development of an automatic mesh generation scheme. This would reduce the user time required for the preparation and encoding of the mesh data. It would also provide the optimum selection of element sizes and numbers, reducing the risk of employing futile meshes.

- The option of using quadratic (bilinear) elements. This would increase the computational time mainly because of two reasons. Firstly, the numerical integration of the higher order polynomials will take longer. Secondly, the sizes of the Finite Element matrices will increase because of the higher number of nodes per element. For example, the use of the 16-noded Serendipity element will result in a fourfold increase in the size of the Finite Element matrices.

- The improvement in the approximation of the time derivative. This, according to Van Genuchten [9], is the major cause of oscillations in the solution of convection-dominated problems.

The experimental exercise provided useful insight into the behaviour of natural aquifers. Most of the previous Finite Element models have overlooked the practical difficulties encountered in the modelling of aquifers. Extensive time is usually required to establish the properties of the

aquifer. This is complicated by the fact that natural aquifers are seldom homogeneous and isotropic. A major drawback of most experimental models is their dimensional limitations. The existence of unnatural boundaries cause instability in the system which may reduce the correlation between experiment and theory.

The extent of dispersion of contaminants in groundwater is a real problem. This has been verified in theory as well as in experimentation. Many times, the contamination of aquifers is unavoidable or already in progress. A need therefore exists for the development of techniques which will retard or divert the transport of contaminants. One such possibility is the pumping of the aquifer at a point near the contaminant source. This will cause the contaminated water to be attracted to that point whereupon it can be purified and recharged back into the aquifer away from the contaminant source.

Another possibility involves the use of layers of porous media which have higher permeabilities than that of the rest of the aquifer. The layers, whether natural or artificial, are placed at angles to the flow, causing the groundwater to diffract from its original path. Whether the phenomenon of diffraction exists in such cases is only hypothesised. This will be the subject of future research.

References

1. Hamill, L. and Bell, F.G., Groundwater Resource Development, Butterworths (1986)

2. Addison, H., Hydraulic Measurements, Chapman & Hall Ltd (1940)

3. Ven Te Chow, Handbook of Applied Hydrology, McGraw-Hill (1964)

4. Walton, W.C., Groundwater Resource Evaluation, McGraw-Hill (1970)

5. Management of the Water Resources of the RSA, Department of Water Affairs, CTP Book Printers (1986)

6. Crank, J. et al, Diffusion Processes in Environmental Systems, Macmillan (1981)

7. Ehlig, C., Comparison of Numerical Methods for solution of Diffusion-Convection Equation in One and Two Dimensions, Proc. of the First Int. Conference on Finite Elements in Water Resources, pp 1.91-1.102, Pentech Press (1976)

8. Segol G., A Three Dimensional Galerkin Finite Element Model for the analysis of Contaminant Transport in Saturated-Unsaturated Porous Media, Proc. of the First Int. Conference on Finite Elements in Water Resources, pp 2.123-2.145, Pentech Press (1976)

9. Van Genuchten, M.T., On the accuracy of several Numerical Schemes for solving the Convective-Dispersive Equation, Proc. of the First Int. Conference on Finite Elements in Water Resources, pp 1.71-1.90, Pentech Press (1976)

10. Wen-Hsiung Li, Differential Equations of Hydraulic Transients, Dispersion, and Groundwater Flow, Prentice-Hall, New Jersey

11. Varoglu, E. and Finn, W.D.L., A Finite Element Method for the Diffusion-Convection Equation, Proc. of the Second Int. Conference on Finite Elements in Water Resources, pp 4.3-4.20, Pentech Press (1978)

12. Jensen, O.K. and Finlayson, B.A., Solution of the Convection-Diffusion Equation using a Moving Coordinate System, Proc. of the Second Int. Conference on Finite Elements in Water Resources, pp 4.3-4.20, Pentech Press (1978)

13. Zienkiewicz, O.C. and Heinrich, J.C., The Finite Element Method and Convection Problems in Fluid Mechanics, Finite Elements in Fluids, Volume 3, pp 1-22, John Wiley & Sons (1978)

14. Taylor, C., Three Dimensional Groundwater Flow with Convective Dispersion, Finite Elements in Fluids, Volume 3, pp 311-321, John Wiley & Sons (1978)

15. Zienkiewicz, O.C. and Morgan, K., Finite Elements and Approximation, John Wiley & Sons (1983)

16. Smith, I.M., Programming the Finite Element Method, John Wiley & Sons (1982)

17. White, R.E., An Introduction to the Finite Element Method with Applications to Non-linear Problems, John Wiley & Sons (1985)

18. Wait, R. and Mitchell, A.R., Finite Element Analysis and Applications, John Wiley & Sons (1985)

19. Desai, C.S., Elementary Finite Element Method, Prentice-Hall (1979)

20. Reddy, J.N. and Rasmussen, M.L., Advanced Engineering Analysis, John Wiley & Sons (1982)

21. Carey, G.F. and Oden, J.T., Finite Elements: A Second Course, Volume II, Prentice-Hall (1983)

22. Connor, J.J. and Brebbia, C.A., Finite Element Techniques for Fluid Flow, Butterworths (1976)

23. Reddy, B.D., Functional Analysis and Boundary Value Problems: An Introductory Treatment, Pitman (1986)

24. Huyakorn, P. and Taylor, C., Finite Element Models for Coupled Groundwater Flow and Convective Dispersion, Proc. of the First Int. Conference on Finite Elements in Water Resources, pp 1.131-1.151, Pentech Press (1976)

25. Hunt, B., Mathematical Analysis of Groundwater Resources, Butterworths (1983)

26. Desai, C.S., Finite Element Methods for Flow in Porous Media, Finite Elements in Fluids, Volume 1, pp 157-182, John Wiley & Sons (1978)

27. Neuman, S.P., Galerkin Approach to Saturated-Unsaturated Flow in Porous Media, Finite Elements in Fluids, Volume 1, pp 201-217, John Wiley & Sons (1978)

28. Kreyszig, E., Advanced Engineering Mathematics, Fifth Edition, John Wiley & Sons (1983)

29. France, P.W., Finite Element Analysis of Two- and Three-Dimensional Unconfined Seepage Problems, Proc. of the First Int. Conference on Finite Elements in Water Resources, pp 2.71-1.85, Pentech Press (1976)

30. Mercer, J.W. and Faust, C.R., The Application of Finite Element Techniques to Immiscible Flow in Porous Media, Proc. of the First Int. Conference on Finite Elements in Water Resources, pp 1.21-1.57, Pentech Press (1976)

31. Lam, D.C.L., Comparison of Finite Element and Finite Difference Methods for Nearshore Advection-Diffusion Transport Model, Proc. of the First Int. Conference on Finite Elements in Water Resources, pp 1.115-1.129, Pentech Press (1976)

32. Nwaogazie, Ify L., Revised SOTRAN - A 2-D Finite Element Aquifer Flow-Transport Program, Software for Engineering Workstations, Volume 4, Computational Mechanics Publications (1988).

33. Pinder, G.F. and Page, R.H., Finite Element Simulation of Salt Water Intrusion on the South Fork of Long Island, Proc. of the First Int. Conference on Finite Elements in Water Resources, pp 2.51-2.69, Pentech Press (1976)

34. Considine, A. and Ross, A., Handbook of Process Instrumentation.

APPENDIX A Derivation of the Finite Element form of the Diffusion-Convection Equation in Three Dimensions.

In the Galerkin Finite Element formulation of the diffusion-convection equation, the dispersion corrected scheme of Van Genuchten [9], shown in equation (A1) will be used.

$$\frac{c^{t+\Delta t}-c^{t}}{\Delta t} = \frac{1}{2}\left[\frac{\partial c}{\partial t} - \frac{\Delta t}{6} \frac{\partial^2 c}{\partial t^2} \right]^{t+\Delta t} + \frac{1}{2}\left[\frac{\partial c}{\partial t} + \frac{\Delta t}{6} \frac{\partial^2 c}{\partial t^2} \right]^{t} \qquad (A.1)$$

In order to substitute for the time derivatives in equation (A.1), an estimate of $\frac{\partial^2 c}{\partial t^2}$ is required. Differentiating equation (3.1) with respect to time and neglecting higher order derivatives

$$\frac{\partial^2 c}{\partial t^2} = \frac{\partial}{\partial t}\left[\frac{\partial c}{\partial t}\right]$$

$$= n^2\left[v_x\frac{\partial}{\partial t} + v_y\frac{\partial}{\partial t} + v_z\frac{\partial}{\partial t}\right]\left[v_x\frac{\partial c}{\partial t} + v_y\frac{\partial c}{\partial t} + v_z\frac{\partial c}{\partial t}\right]$$

$$= n^2\left[v_x^2\frac{\partial^2 c}{\partial x^2} + 2v_xv_y\frac{\partial^2 c}{\partial x\partial y} + 2v_xv_z\frac{\partial^2 c}{\partial x\partial z} + v_y^2\frac{\partial^2 c}{\partial y^2} + 2v_yv_z\frac{\partial^2 c}{\partial y\partial z} + v_z^2\frac{\partial^2 c}{\partial z^2}\right]$$

..... (A.2)

Substituting equations (3.1) and (A.2) into equation (A.1) gives

$$\frac{c^{t+\Delta t}-c^{t}}{\Delta t} = \frac{n}{2}\left[D_{xx1}\frac{\partial^2 c}{\partial x^2} + D_{yy1}\frac{\partial^2 c}{\partial y^2} + D_{zz1}\frac{\partial^2 c}{\partial z^2} + 2D_{xy1}\frac{\partial^2 c}{\partial x \partial y}\right.$$

$$\left. + 2D_{xz1}\frac{\partial^2 c}{\partial x \partial z} + 2D_{yz1}\frac{\partial^2 c}{\partial y \partial z} - v_x\frac{\partial c}{\partial x} - v_y\frac{\partial c}{\partial y} - v_z\frac{\partial c}{\partial z}\right]^{t+\Delta t}$$

$$+ \frac{n}{2}\left[D_{xx2}\frac{\partial^2 c}{\partial x^2} + D_{yy2}\frac{\partial^2 c}{\partial y^2} + D_{zz2}\frac{\partial^2 c}{\partial z^2} + 2D_{xy2}\frac{\partial^2 c}{\partial x \partial y}\right.$$

$$\left. + 2D_{xz2}\frac{\partial^2 c}{\partial x \partial z} + 2D_{yz2}\frac{\partial^2 c}{\partial y \partial z} - v_x\frac{\partial c}{\partial x} - v_y\frac{\partial c}{\partial y} - v_z\frac{\partial c}{\partial z}\right]^{t} \qquad (A.3)$$

where

$$D_{xx1} = D_x - v_x^2\frac{\Delta t n}{6} \qquad D_{xx2} = D_x + v_x^2\frac{\Delta t n}{6}$$

$$D_{yy1} = D_y - v_y^2\frac{\Delta t n}{6} \qquad D_{yy2} = D_y + v_y^2\frac{\Delta t n}{6}$$

$$D_{zz1} = D_z - v_z^2\frac{\Delta t n}{6} \qquad D_{zz2} = D_z + v_z^2\frac{\Delta t n}{6}$$

$$D_{xy1} = -v_x v_y\frac{\Delta t n}{6} \qquad D_{xy2} = +v_x v_y\frac{\Delta t n}{6}$$

$$D_{xz1} = -v_x v_z\frac{\Delta t n}{6} \qquad D_{xz2} = +v_x v_z\frac{\Delta t n}{6}$$

$$D_{yz1} = -v_y v_z\frac{\Delta t n}{6} \qquad D_{yz2} = +v_y v_z\frac{\Delta t n}{6}$$

Rearranging equation (A. 3)

$$\frac{n}{2}\left[- D_{xx1}\frac{\partial^2 c}{\partial x^2} - D_{yy1}\frac{\partial^2 c}{\partial y^2} - D_{zz1}\frac{\partial^2 c}{\partial z^2} - 2D_{xy1}\frac{\partial^2 c}{\partial x \partial y} \right.$$

$$\left. - 2D_{xz1}\frac{\partial^2 c}{\partial x \partial z} - 2D_{yz1}\frac{\partial^2 c}{\partial y \partial z} + v_x\frac{\partial c}{\partial x} + v_y\frac{\partial c}{\partial y} + v_z\frac{\partial c}{\partial z} + 2/n\frac{c}{\Delta t} \right]^{t+\Delta t}$$

$$= \frac{n}{2}\left[D_{xx2}\frac{\partial^2 c}{\partial x^2} + D_{yy2}\frac{\partial^2 c}{\partial y^2} + D_{zz2}\frac{\partial^2 c}{\partial z^2} + 2D_{xy2}\frac{\partial^2 c}{\partial x \partial y} \right.$$

$$\left. + 2D_{xz2}\frac{\partial^2 c}{\partial x \partial z} + 2D_{yz2}\frac{\partial^2 c}{\partial y \partial z} - v_x\frac{\partial c}{\partial x} - v_y\frac{\partial c}{\partial y} - v_z\frac{\partial c}{\partial z} + 2/n\frac{c}{\Delta t} \right]^{t} \qquad (A.4)$$

Using the approximation

$$c(x, y, z, t) = \sum_{j=1}^{n} c_j(t)\ \phi_j(x, y, z) \qquad (A.5)$$

the Galerkin Finite Element method can now be applied to equation (A. 4) resulting in the equation

$$c_i^{t+\Delta t}\int_\Omega \frac{n}{2}\left[- D_{xx1}\frac{\partial^2 \phi_i}{\partial x^2} - D_{yy1}\frac{\partial^2 \phi_i}{\partial y^2} - D_{zz1}\frac{\partial^2 \phi_i}{\partial z^2} - 2D_{xy1}\frac{\partial^2 \phi_i}{\partial x \partial y} - 2D_{xz1}\frac{\partial^2 \phi_i}{\partial x \partial z}\right.$$

$$\left. - 2D_{yz1}\frac{\partial^2 \phi_i}{\partial y \partial z} + v_x\frac{\partial \phi_i}{\partial x} + v_y\frac{\partial \phi_i}{\partial y} + v_z\frac{\partial \phi_i}{\partial z} + 2/n\frac{\phi_i}{\Delta t}\right] \phi_j \, d\Omega$$

$$= c_i^{t}\int_\Omega \frac{n}{2}\left[D_{xx2}\frac{\partial^2 \phi_i}{\partial x^2} + D_{yy2}\frac{\partial^2 \phi_i}{\partial y^2} + D_{zz2}\frac{\partial^2 \phi_i}{\partial z^2} + 2D_{xy2}\frac{\partial^2 \phi_i}{\partial x \partial y} + 2D_{xz2}\frac{\partial^2 \phi_i}{\partial x \partial z}\right.$$

$$\left. + 2D_{yz2}\frac{\partial^2 \phi_i}{\partial y \partial z} - v_x\frac{\partial \phi_i}{\partial x} - v_y\frac{\partial \phi_i}{\partial y} - v_z\frac{\partial \phi_i}{\partial z} + 2/n\frac{\phi_i}{\Delta t}\right] \phi_j \, d\Omega \qquad (A.6)$$

where ϕ_i, i=1,2,.....,n, are the basis or trial functions which are similar to those used in the Finite Element formulation of the groundwater flow equation. Integrating the second order terms by parts gives

$$c_i^{t+\Delta t}\left[\int_\Omega \frac{n}{2}\left[D_{xx1}\frac{\partial\phi_i}{\partial x}\frac{\partial\phi_j}{\partial x}+D_{yy1}\frac{\partial\phi_i}{\partial y}\frac{\partial\phi_j}{\partial y}+D_{zz1}\frac{\partial\phi_i}{\partial z}\frac{\partial\phi_j}{\partial z}+2D_{xy1}\frac{\partial\phi_i}{\partial x}\frac{\partial\phi_j}{\partial y}\right.\right.$$

$$\left.+2D_{xz1}\frac{\partial\phi_i}{\partial x}\frac{\partial\phi_j}{\partial z}+2D_{yz1}\frac{\partial\phi_i}{\partial y}\frac{\partial\phi_j}{\partial z}+v_x\frac{\partial\phi_i}{\partial x}\phi_j+v_y\frac{\partial\phi_i}{\partial y}\phi_j+v_z\frac{\partial\phi_i}{\partial z}\phi_j+2/n\frac{\phi_i}{\Delta t}\phi_j\right]d\Omega$$

$$-\int_\Gamma \frac{n}{2}\left[D_{xx1}\frac{\partial\phi_i}{\partial x}\phi_j\nu_{xx}+D_{yy1}\frac{\partial\phi_i}{\partial y}\phi_j\nu_{yy}+D_{zz1}\frac{\partial\phi_i}{\partial z}\phi_j\nu_{zz}\right.$$

$$\left.\left.2D_{xy1}\frac{\partial\phi_i}{\partial x}\phi_j\nu_{xy}+2D_{xz1}\frac{\partial\phi_i}{\partial x}\phi_j\nu_{xz}+2D_{yz1}\frac{\partial\phi_i}{\partial y}\phi_j\nu_{yz}\right]d\Gamma\right]$$

$$=c_i^{t}\left[\int_\Omega \frac{n}{2}\left[-D_{xx2}\frac{\partial\phi_i}{\partial x}\frac{\partial\phi_j}{\partial x}-D_{yy2}\frac{\partial\phi_i}{\partial y}\frac{\partial\phi_j}{\partial y}-D_{zz2}\frac{\partial\phi_i}{\partial z}\frac{\partial\phi_j}{\partial z}-2D_{xy2}\frac{\partial\phi_i}{\partial x}\frac{\partial\phi_j}{\partial y}\right.\right.$$

$$\left.-2D_{xz2}\frac{\partial\phi_i}{\partial x}\frac{\partial\phi_j}{\partial z}-2D_{yz2}\frac{\partial\phi_i}{\partial y}\frac{\partial\phi_j}{\partial z}-v_x\frac{\partial\phi_i}{\partial x}\phi_j-v_y\frac{\partial\phi_i}{\partial y}\phi_j-v_z\frac{\partial\phi_i}{\partial z}\phi_j-2/n\frac{\phi_i}{\Delta t}\phi_j\right]d\Omega$$

$$+\int_\Gamma \frac{n}{2}\left[D_{xx2}\frac{\partial\phi_i}{\partial x}\phi_j\nu_{xx}+D_{yy2}\frac{\partial\phi_i}{\partial y}\phi_j\nu_{yy}+D_{zz2}\frac{\partial\phi_i}{\partial z}\phi_j\nu_{zz}\right.$$

$$\left.\left.2D_{xy2}\frac{\partial\phi_i}{\partial x}\phi_j\nu_{xy}+2D_{xz2}\frac{\partial\phi_i}{\partial x}\phi_j\nu_{xz}+2D_{yz2}\frac{\partial\phi_i}{\partial y}\phi_j\nu_{yz}\right]d\Gamma\right] \qquad (A.7)$$

Adding the integrals over the boundary equation (A.7) may be rewritten as

$$c_i^{t+\Delta t}\left[\int_\Omega \frac{n}{2}\left[D_{xx1}\frac{\partial\phi_i}{\partial x}\frac{\partial\phi_j}{\partial x}+ D_{yy1}\frac{\partial\phi_i}{\partial y}\frac{\partial\phi_j}{\partial y}+ D_{zz1}\frac{\partial\phi_i}{\partial z}\frac{\partial\phi_j}{\partial z}+ 2D_{xy1}\frac{\partial\phi_i}{\partial x}\frac{\partial\phi_j}{\partial y}\right.\right.$$

$$\left.\left.+ 2D_{xz1}\frac{\partial\phi_i}{\partial x}\frac{\partial\phi_j}{\partial z}+ 2D_{yz1}\frac{\partial\phi_i}{\partial y}\frac{\partial\phi_j}{\partial z}+ v_x\frac{\partial\phi_i}{\partial x}\phi_j + v_y\frac{\partial\phi_i}{\partial y}\phi_j + v_z\frac{\partial\phi_i}{\partial z}\phi_j + 2/n\frac{\phi_i}{\Delta t}\phi_j\right] d\Omega\right.$$

$$= c_i^{t}\left[\int_\Omega \frac{n}{2}\left[- D_{xx2}\frac{\partial\phi_i}{\partial x}\frac{\partial\phi_j}{\partial x}- D_{yy2}\frac{\partial\phi_i}{\partial y}\frac{\partial\phi_j}{\partial y}- D_{zz2}\frac{\partial\phi_i}{\partial z}\frac{\partial\phi_j}{\partial z}- 2D_{xy2}\frac{\partial\phi_i}{\partial x}\frac{\partial\phi_j}{\partial y}\right.\right.$$

$$\left.\left.- 2D_{xz2}\frac{\partial\phi_i}{\partial x}\frac{\partial\phi_j}{\partial z}- 2D_{yz2}\frac{\partial\phi_i}{\partial y}\frac{\partial\phi_j}{\partial z}- v_x\frac{\partial\phi_i}{\partial x}\phi_j - v_y\frac{\partial\phi_i}{\partial y}\phi_j - v_z\frac{\partial\phi_i}{\partial z}\phi_j - 2/n\frac{\phi_i}{\Delta t}\phi_j\right] d\Omega\right.$$

$$+ \int_\Gamma n\left[D_x\frac{\partial\phi_i}{\partial x}\phi_j\nu_x + D_y\frac{\partial\phi_i}{\partial y}\phi_j\nu_y + D_z\frac{\partial\phi_i}{\partial z}\phi_j\nu_z\right] d\Gamma \qquad (A.8)$$

The natural boundary condition from equation (3.3) may now be substituted into the above equation. In matrix form, the Finite Element formulation becomes

$$[M] = \int_{\Omega} n\left[v_x[\phi_x][\phi]^T + v_y[\phi_y][\phi]^T + v_z[\phi_z][\phi]^T\right] d\Omega$$

$$[N] = \int_{\Omega} [\phi][\phi]^T \, d\Omega$$

$$\underset{\sim}{g} = \int_{\Gamma} n f_n[\phi] \, d\Gamma \qquad \text{(A.10)}$$

APPENDIX B Simplifying the FE formulation for groundwater flow.

Consider the FE formulation governing the flow of groundwater

$$\{[A] + [B]/\Delta t\}\underset{\sim}{h}^{t+\Delta t} = \{[B]/\Delta t\}\underset{\sim}{h}^{t} - \underset{\sim}{Q} - \underset{\sim}{q} \qquad (A.1)$$

This equation can be written in expanded matrix form as follows

$$\left[\left[\begin{array}{cccc} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ a_{31} & a_{32} & \cdots & a_{3n} \\ \vdots & & & \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{array}\right] + \left[\begin{array}{cccc} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ b_{31} & b_{32} & \cdots & b_{3n} \\ \vdots & & & \\ b_{n1} & b_{n2} & \cdots & b_{nn} \end{array}\right] \frac{1}{\Delta t}\right]\left[\begin{array}{c} h_1 \\ h_2 \\ h_3 \\ \vdots \\ h_n \end{array}\right]^{t+\Delta t}$$

$$= \frac{1}{\Delta t}\left[\begin{array}{cccc} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & b_{2n} \\ b_{31} & b_{32} & \cdots & b_{3n} \\ \vdots & & & \\ b_{n1} & b_{n2} & \cdots & b_{nn} \end{array}\right]\left[\begin{array}{c} h_1 \\ h_2 \\ h_3 \\ \vdots \\ h_n \end{array}\right]^{t} - \left[\begin{array}{c} Q_1 \\ Q_2 \\ Q_3 \\ \vdots \\ Q_n \end{array}\right] - \left[\begin{array}{c} q_1 \\ q_2 \\ q_3 \\ \vdots \\ q_n \end{array}\right] \qquad (A.2)$$

The matrices [A] and [B] on the left hand side can be added. Furthermore, since $\underset{\sim}{h}^{t}$ is known, the system simplifies to

$$\begin{bmatrix} k_{11} & k_{12} & \cdots & k_{1n} \\ k_{21} & k_{22} & \cdots & k_{2n} \\ k_{31} & k_{32} & \cdots & k_{3n} \\ \vdots & & & \\ k_{n1} & k_{n2} & \cdots & k_{nn} \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ \vdots \\ h_n \end{bmatrix}^{t+\Delta t} = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ \vdots \\ e_n \end{bmatrix} - \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \\ \vdots \\ Q_n \end{bmatrix} - \begin{bmatrix} q_1 \\ q_2 \\ q_3 \\ \vdots \\ q_n \end{bmatrix} \qquad (A.3)$$

Assume that nodes 1 and 2 constitute part of a boundary where the potential heads are specified as constant and equal to H_1 and H_2. Then

$$\begin{bmatrix} k_{11} & k_{12} & \cdots & k_{1n} \\ k_{21} & k_{22} & \cdots & k_{2n} \\ k_{31} & k_{32} & \cdots & k_{3n} \\ \vdots & & & \\ k_{n1} & k_{n2} & \cdots & k_{nn} \end{bmatrix} \begin{bmatrix} H_1 \\ H_2 \\ h_3 \\ \vdots \\ h_n \end{bmatrix}^{t+\Delta t} = \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ \vdots \\ e_n \end{bmatrix} - \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \\ \vdots \\ Q_n \end{bmatrix} - \begin{bmatrix} q_1 \\ q_2 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \qquad (A.4)$$

The elements q_1 and q_2, together with h_3, h_4, ..., h_n are the unknowns in the system. By writing the system as

$$k_{11}H_1 + k_{12}H_2 + k_{13}h_3 + \cdots + k_{1n}h_n = e_1 - Q_1 - q_1$$

$$k_{21}H_1 + k_{22}H_2 + k_{23}h_3 + \cdots + k_{2n}h_n = e_2 - Q_2 - q_2$$

$$k_{31}H_1 + k_{32}H_2 + k_{33}h_3 + \cdots + k_{3n}h_n = e_3 - Q_3$$

$$\vdots$$

$$k_{n1}H_1 + k_{n2}H_2 + k_{n3}h_3 + \cdots + k_{nn}h_n = e_n - Q_n$$

and rearranging as

$$
\begin{aligned}
q_1 \quad &+ k_{13}h_3 + \ldots\ldots + k_{1n}h_n = -(k_{11}H_1 + k_{12}H_2) + e_1 - Q_1 \\
q_2 \; &+ k_{23}h_3 + \ldots\ldots + k_{2n}h_n = -(k_{21}H_1 + k_{22}H_2) + e_2 - Q_2 \\
&k_{33}h_3 + \ldots\ldots + k_{3n}h_n = -(k_{31}H_1 + k_{32}H_2) + e_3 - Q_3 \\
&\vdots \\
&k_{n3}h_3 + \ldots\ldots + k_{nn}h_n = -(k_{n1}H_1 + k_{n2}H_2) + e_n - Q_n
\end{aligned}
$$

the following matrix form can be deduced

$$
\begin{bmatrix} r_1 & 0 & \ldots\ldots & k_{1n} \\ 0 & r_2 & \ldots\ldots & k_{2n} \\ 0 & 0 & \ldots\ldots & k_{3n} \\ \vdots & & & \\ 0 & 0 & \ldots\ldots & k_{nn} \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ h_3 \\ \vdots \\ h_n \end{bmatrix}^{t+\Delta t} = \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \\ k_{31} & k_{32} \\ \vdots & \\ k_{n1} & k_{n2} \end{bmatrix} \begin{bmatrix} H_1 \\ H_2 \end{bmatrix} + \begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ \vdots \\ e_n \end{bmatrix} - \begin{bmatrix} Q_1 \\ Q_2 \\ Q_3 \\ \vdots \\ Q_n \end{bmatrix} \qquad (A.4)
$$

where r_1 and r_2 are the coefficients of the unknowns q_1 and q_2. All the terms on the right hand side can be combined to form one vector, giving

$$
\begin{bmatrix} r_1 & 0 & \ldots\ldots & k_{1n} \\ 0 & r_2 & \ldots\ldots & k_{2n} \\ 0 & 0 & \ldots\ldots & k_{3n} \\ \vdots & & & \\ 0 & 0 & \ldots\ldots & k_{nn} \end{bmatrix} \begin{bmatrix} q_1 \\ q_2 \\ h_3 \\ \vdots \\ h_n \end{bmatrix}^{t+\Delta t} = \begin{bmatrix} s_1 \\ s_2 \\ s_3 \\ \vdots \\ s_n \end{bmatrix} \qquad (A.5)
$$

which is of the form

$$[U]\underset{\sim}{u} = \underset{\sim}{s}$$

The above form can easily be reduced to echelon form by the Gauss elimination method. A similar procedure can be applied to the diffusion-convection equation.

APPENDIX C Program listings of AQUIFEM

Program listings of all the programs comprising AQUIFEM is given below. Interested readers may make modifications to the programs if required.

Two different compilers were used. The first one was DBaseIV, a product of Ashton Tate (USA), which was used to construct the data capturing and input error trapping routines as well as the menu selection routines. The second one, Turbo BASIC, a product of Borland Inc (USA), was used to write the FE processing and output routines.

The software was compiled and run on an IBM Compatible AT (80286) with a maths coprocessor (80287).

Filename: AQUIFEM.PRG Type: DBaseIV procedure file

Purpose : Controls the execution of ALL other subprograms, including the Finite Element execution program MAIN.FEM. In addition, it has incorporated within it the data entry and data checking routines. Some of these routines are held, for ease of programming, in a library file called AQUILIB.PRG.

```
**********AQUIFEM.PRG***BEGIN**************************************************

**********SET UP
CLEAR ALL
SET SCORE OFF
SET STATUS OFF
SET DATE BRIT
SET BELL OFF
SET TALK OFF
SET HEADING OFF
SET CLOCK TO 0,68
SET CLOCK ON
SET HELP OFF
SET ESCAPE OFF
SET SAFE OFF
```

```
SET PROCEDURE TO AQUILIB
DO START
DO INIT

PROCEDURE START
   SET CLOCK OFF
   CLEAR
   @ 3,10 SAY "AQUIFEM"
   @ 5,10 SAY "PC VERSION 1.0"
   @ 9,10 SAY "Written by Nawaz Mahomed"
   @12,10 SAY "This software is licenced to be used by the Dept of Mech Eng,"
   @13,10 SAY "Peninsula Technikon"
   @16,10 SAY "Licence number 10000001-PT"
   @20,10 SAY "Copyright (1989) PENINSULA TECHNIKON "
   @22,10 SAY "AQUIFEM is a Peninsula Technikon trademark"
   READ
   CLEAR
   SET CLOCK ON
RETURN

PROCEDURE INIT
   PUBLIC CHOICE1,CHOICE2
   STORE "NO" TO CHOICE1
   STORE "YES" TO CHOICE2
   ON KEY LABEL F1 DO HELPER
   ON KEY LABEL F3 DO CODES
   DO DEFPOP
   ACTIVATE POPUP MAIN
RETURN

*******************************************************************************

**********MAIN

PROCEDURE MAIN
   PUBLIC PROG,MESG,REC,RECBOT,RECTOP,PRN,CHOICE,CHEK
   STORE SPACE(4)  TO CHEK
   STORE SPACE(1)  TO CHOICE
   STORE SPACE(60) TO MESG
   STORE RECNO()   TO REC
   STORE RECNO()   TO RECBOT
   STORE RECNO()   TO RECTOP
   STORE SPACE(1)  TO PRN
   STORE SPACE(10) TO PROG
   DO VARLIST
   DO CASE
      CASE BAR()=4
           CLEAR
           DO INTRO
      CASE BAR()=6
           CLEAR
           USE FEMDATA1 IN 1
           USE FEMDATA2 IN 2
           USE FEMDATA3 IN 3
```

```
            USE FEMDATA4 IN 4
            ACTIVATE POPUP EDITDATA
            DO DATACHEK
            IF CHEK="OKAY"
               @20,1 SAY "PLEASE WAIT. PROCESSING DATA....."
               RUN ERASE FEMDATA?.DB2
               SELECT 1
               EXPORT TO FEMDATA1 TYPE DBASE!!
               SELECT 2
               EXPORT TO FEMDATA2 TYPE DBASE!!
               SELECT 3
               EXPORT TO FEMDATA3 TYPE DBASE!!
               SELECT 4
               EXPORT TO FEMDATA4 TYPE DBASE!!
               CLOSE DATABASES
               USE COUNTER
               ZAP
               APPEND BLANK
               DO ASSN5
               DO PLACE5
               EXPORT TO COUNTER TYPE DBASE!!
               CLOSE DATA
               CLEAR
               MESG="DO YOU WANT A PRINTOUT OF THE DATA FOR        CHECKING?"
               DO CHOICE
               IF CHOICE="Y"
                  MESG="IS THE PRINTER CONNECTED?"
                  DO CHOICE
                  IF CHOICE="Y"
                     RUN CONVERTP
                     CLEAR
                  ELSE
                     MESG="AQUIFEM HAS ABORTED DATA PRINTOUT"
                     DO MESG WITH MESG
                     RUN CONVERT
                  ENDIF
               ELSE
                  RUN CONVERT
               ENDIF
            ELSE
               MESG="WARNING! DATA NOT PROCESSED"
               DO MESG WITH MESG
            ENDIF
         CASE BAR()=8
            CLEAR
            ACTIVATE POPUP PROGRUN
         CASE BAR()=10
            CLEAR
            ACTIVATE POPUP OUTPUT
         CASE BAR()=12
            ! CD\
            QUIT
      ENDCASE
   RETURN
```

```
**********INTRO**********************************************************************

PROCEDURE INTRO
   PUBLIC U
   U=1
   STORE "INTRO " TO INTRO
   DO WHILE LASTKEY()<>27
      IF LASTKEY()=-8
         U=U-1
         IF U<1
            U=1
         ENDIF
      ELSE
         IF LASTKEY()=-9
            U=U+1
            IF U>3
               U=3
            ENDIF
         ENDIF
      ENDIF
      INTRO="INTRO"+SUBSTR(STR(U),10,1)
      CLEAR
      SET MESSAGE TO "¦ F9 - PREVIOUS PAGE ¦ F10 - NEXT PAGE ¦ ESC - EXIT ¦"
      DO &INTRO
      WAIT " "
   ENDDO
   SET MESSAGE TO " "
   CLEAR
RETURN

PROCEDURE INTRO1
   TEXT
                                AQUIFEM

This  program uses the Galerkin Finite Element procedure  to

solve  the  three  dimensional coupled  groundwater  flow  -

diffusion-convection problem.

The domain could have one or more of the following characteristics:

(a) an unconfined aquifer where the free surface is not known a priori

(b) pumping from wells

(c) constant potential head boundaries

(d) constant or instantaneous contaminant sources

The  program could also be used to solve any  other  problem
```

```
that  fall  within  the above domain such  as  free  surface

seepage through a dam wall.

ENDTEXT
RETURN

PROCEDURE INTRO2
TEXT

AQUIFEM was wriiten and developed at the Peninsula Technikon

in the Department of Mechanical Engineering.

ENDTEXT
RETURN


**********EDITDATA*************************************************************

PROCEDURE EDITDATA
   DO CASE
      CASE BAR()=3
           MESG="ARE YOU SURE ???"
           DO CHOICE
           IF CHOICE="Y"
              SELECT 1
              ZAP
              APPEND BLANK
              SELECT 2
              ZAP
              SELECT 3
              ZAP
              SELECT 4
              ZAP
           ENDIF
      CASE BAR()=4
           SELECT 1
           ACTIVATE WINDOW INFO1
           DO ASSN1
           DO SCREEN1
           IF LASTKEY()<>27
              DO PLACE1
           ELSE
              MESG="NO CHANGES AFFECTED"
              DO MESG WITH MESG
           ENDIF
           DEACTIVATE WINDOW INFO1
      CASE BAR()=5
           SELECT 1
           IF VAL(ELEM)>0
              ELEMV=VAL(ELEM)
              MAXV=VAL(MAX)
              SELECT 2
```

```
      ACTIVATE WINDOW INFO2
      IF EV=ELEMV
         EV=0
      ENDIF
      DO WHILE EV<>ELEMV
            EV=EV+1
         IF EV>RECCOUNT()
            APPEND BLANK
         ELSE
               GOTO RECORD EV
         ENDIF
         DO ASSN2
         DO SCREEN2
         IF LASTKEY()<>27
            DO PLACE2
            IF LASTKEY()=18 .OR. LASTKEY()=5
               EV=EV-2
               IF EV<0
                  EV=0
               ENDIF
            ENDIF
         ELSE
            MESG="DO YOU WANT TO ABORT THE OPERATION?"
            DO CHOICE
            IF CHOICE="Y"
               EV=ELEMV
            ELSE
               EV=EV-1
            ENDIF
         ENDIF
      ENDDO
      DEACTIVATE WINDOW INFO2
   ELSE
      MESG="YOU MUST FIRST SPECIFY THE DOMAIN PARAMETERS"
      DO MESG WITH MESG
   ENDIF
CASE BAR()=6
   SELECT 1
   IF VAL(ELEM)>0
      ELEMV=VAL(ELEM)
      SELECT 3
      ACTIVATE WINDOW INFO3
      IF EV=ELEMV
         EV=0
      ENDIF
      DO WHILE EV<>ELEMV
         EV=EV+1
         IF EV>RECCOUNT()
            APPEND BLANK
         ELSE
            GOTO RECORD EV
         ENDIF
         DO ASSN3
         DO SCREEN3
```

```
            IF LASTKEY()<>27
               DO PLACE3
               IF LASTKEY()=18 .OR. LASTKEY()=5
                  EV=EV-2
                  IF EV<0
                   EV=0
                  ENDIF
               ENDIF
            ELSE
               MESG="DO YOU WANT TO ABORT THE OPERATION?"
               DO CHOICE
               IF CHOICE="Y"
                  EV=ELEMV
               ELSE
                  EV=EV-1
               ENDIF
            ENDIF
         ENDDO
         DEACTIVATE WINDOW INFO3
      ELSE
         MESG="YOU MUST FIRST SPECIFY THE DOMAIN PARAMETERS"
         DO MESG WITH MESG
      ENDIF
CASE BAR()=7
      SELECT 1
      IF VAL(ELEM)>0
         MAXV=VAL(MAX)
         SELECT 4
         ACTIVATE WINDOW INFO4
         IF NV=MAXV
            NV=0
         ENDIF
         DO WHILE NV<>MAXV
            NV=NV+1
            IF NV>RECCOUNT()
               APPEND BLANK
            ELSE
               GOTO RECORD NV
            ENDIF
            DO ASSN4
            DO SCREEN4
            IF LASTKEY()<>27
               DO PLACE4
               IF LASTKEY()=18 .OR. LASTKEY()=5
                  NV=NV-2
                  IF NV<0
                   NV=0
                  ENDIF
               ENDIF
            ELSE
               MESG="DO YOU WANT TO ABORT THE OPERATION?"
               DO CHOICE
               IF CHOICE="Y"
                  NV=MAXV
```

```
                ELSE
                   NV=NV-1
                ENDIF
             ENDIF
          ENDDO
          DEACTIVATE WINDOW INFO4
       ELSE
          MESG="YOU MUST FIRST SPECIFY THE DOMAIN PARAMETERS"
          DO MESG WITH MESG
       ENDIF
   ENDCASE
RETURN

PROCEDURE DATACHEK
   USE FEMDATA1 IN 1
   USE FEMDATA2 IN 2
   USE FEMDATA3 IN 3
   USE FEMDATA4 IN 4
   SELECT 1
   DO ASSN1
   CHEK=" "
   DO WHILE CHEK=" "
      IF ELEMV=0 .OR. MAXV<=ELEMV
         MESG="DOMAIN PARAMETERS INCORRECTLY SPECIFIED"
         DO MESG WITH MESG
         CHEK="NOOK"
      ELSE
         SELECT 2
         IF RECCOUNT()<ELEMV
            MESG="ELEMENT-NODAL CONFIGURATIONS MISSING"
            DO MESG WITH MESG
            CHEK="NOOK"
         ELSE
            IF RECCOUNT()>ELEMV
               MESG="TOO MANY ELEMENT-NODAL CONFIGURATIONS -       ERASE REST?"
               DO CHOICE
               IF CHOICE="Y"
                  GO RECORD ELEMV
                  SKIP
                  DELETE REST
                  PACK
               ENDIF
            ELSE
               SELECT 3
               IF RECCOUNT()<ELEMV
                  MESG="AQUIFER ELEMENTAL PROPERTIES MISSING"
                  DO MESG WITH MESG
                  CHEK="NOOK"
               ELSE
                  IF RECCOUNT()>ELEMV
                     MESG="TOO MANY AQUIFER ELEMENTAL PROPERTIES -       ERASE REST?"
                     DO CHOICE
                     IF CHOICE="Y"
                        GO RECORD ELEMV
```

```
                        SKIP
                        DELETE REST
                        PACK
                     ENDIF
                  ELSE
                     SELECT 4
                     IF RECCOUNT()<MAXV
                        MESG="NODAL VARIABLES - RECORDS MISSING"
                        DO MESG WITH MESG
                        CHEK="NOOK"
                     ELSE
                        IF RECCOUNT()>MAXV
                           MESG="NODAL VARIABLES - TOO MANY RECORDS - ERASE    REST?"
                           DO CHOICE
                           IF CHOICE="Y"
                              GO RECORD MAXV
                              SKIP
                              DELETE REST
                              PACK
                              CHEK="OKAY"
                           ENDIF
                        ELSE
                           CHEK="OKAY"
                        ENDIF
                     ENDIF
                  ENDIF
               ENDIF
            ENDIF
         ENDIF
      ENDIF
   ENDDO
RETURN

**********PROGRUN**************************************************************

PROCEDURE PROGRUN
   USE COUNTER
   ACTIVATE WINDOW INFO1
   DO ASSN5
   DO SCREEN5
   IF LASTKEY()<>27
      DO PLACE5
      EXPORT TO COUNTER TYPE DBASEII
   ELSE
      MESG="RUN CANCELLED - NO CHANGES EFFECTED"
      DO MESG WITH MESG
   ENDIF
   DEACTIVATE WINDOW INFO1
   CLOSE DATA
   IF MESG<>"RUN CANCELLED - NO CHANGES EFFECTED"
      DO CASE
         CASE BAR()=3
         CASE BAR()=4
         CASE BAR()=5
```

```
            CLEAR
            RUN MAIN
            CLEAR
      ENDCASE
   ELSE
      MESG=" "
   ENDIF
RETURN

**********OUTPUT**************************************************************

PROCEDURE OUTPUT
   USE COUNTER
   IF VAL(STEPS)>0
      CHEK="OKAY"
   ELSE
      CHECK=" "
   ENDIF
   CLOS DATA
   IF CHEK="OKAY"
      DO CASE
    CASE BAR()=4
         MESG="SEND OUTPUT TO PRINTER?"
         DO CHOICE
         IF CHOICE="Y"
            DO STEPS
            RUN OUTPUT2P
            CLEAR
         ELSE
            DO STEPS
            RUN OUTPUT2
            CLEAR
         ENDIF
    CASE BAR()=6
         USE CONTOUR
         ACTIVATE WINDOW INFO1
         DO ASSN7
         DO SCREEN7
         IF LASTKEY()<>27
            DO PLACE7
            EXPORT TO CONTOUR TYPE DBASE11
            DEACTIVATE WINDOW INFO1
            CLOS DATA
            DO STEPS
            RUN CONTOUR
         ELSE
            MESG="NO CHANGES EFFECTED - OUTPUT CANCELLED"
            DO MESG WITH MESG
            DEACTIVATE WINDOW INFO1
            CLOSE DATA
         ENDIF
         CLEAR
      ENDCASE
   ELSE
```

```
      IF BAR()=4 .OR. BAR()=6
         MESG="NO RESULTS AS YET"
         DO MESG WITH MESG
         CLEAR
      ENDIF
   ENDIF
   DO CASE
      CASE BAR()=3
           MESG="IS THE PRINTER CONNECTED?"
           DO CHOICE
           IF CHOICE="Y"
              RUN OUTPUT1
              CLEAR
           ELSE
              MESG="AQUIFEM HAS ABORTED THE OPERATION"
              DO MESG WITH MESG
           ENDIF
      CASE BAR()=5
           USE GRAPH
           ACTIVATE WINDOW INFO1
           DO ASSN6
           DO SCREEN6
           IF LASTKEY()<>27
              DO PLACE6
              EXPORT TO GRAPH TYPE DBASEII
              DEACTIVATE WINDOW INFO1
              DO STEPS
              CLOSE DATA
              RUN DRW
           ELSE
              MESG="NO CHANGES EFFECTED - OUTPUT CANCELLED"
              DO MESG WITH MESG
              DEACTIVATE WINDOW INFO1_CLOSE DATA
           ENDIF
           CLEAR
   ENDCASE
RETURN

PROCEDURE STEPS
   USE STEPS IN 1
   USE COUNTER IN 2
   SELECT 2
   DO ASSN5
   SELECT 1
   ACTIVATE WINDOW INFO5
   @1,1 SAY "ENTER STEP RANGE"
   @3,1 SAY "ENTER STARTING STEP NUMBER "   GET SSTEPV  PICTURE  "9999"  VALID SSTEPV<=STEPSV
   @5,1 SAY "ENTER  ENDING  STEP NUMBER "   GET ESTEPV  PICTURE  "9999"  VALID ESTEPV<=STEPSV
   READ
   REPLACE SSTEP WITH SUBSTR(STR(SSTEPV),7,4)+","
   REPLACE ESTEP WITH SUBSTR(STR(ESTEPV),7,4)+","
   DEACTIVATE WINDOW INFO5
   EXPORT TO STEPS TYPE DBASEII_CLOSE DATA
RETURN
```

```
**********AQUIFEM.PRG***END**********************************************************


Filename: AQUILIB.PRG          Type: DBaseIV procedure file

Purpose : Contains the data entry and data checking  library
          routines used by AQUIFEM.PRG.


**********AQUILIB.PRG***BEGIN*********************************************

**********LIBRARY

PROCEDURE DEFPOP
   DEFINE POPUP MAIN FROM  6,30 TO 20,50 MESSAGE "¦ Use Arrow Keys to move and Return to select_
                                              ¦ F1 for help ¦"
   DEFINE BAR  2 OF MAIN     PROMPT "     MAIN MENU      " MESSAGE "" SKIP
   DEFINE BAR  4 OF MAIN     PROMPT "    INTRODUCTION    " MESSAGE ""
   DEFINE BAR  6 OF MAIN     PROMPT "     EDIT DATA      " MESSAGE ""
   DEFINE BAR  8 OF MAIN     PROMPT "        RUN         " MESSAGE ""
   DEFINE BAR 10 OF MAIN     PROMPT "  OUTPUT DISPLAY    " MESSAGE ""
   DEFINE BAR 12 OF MAIN     PROMPT "    EXIT TO DOS     " MESSAGE ""
   ON SELECTION POPUP MAIN DO MAIN

   DEFINE POPUP EDITDATA FROM  2,57 TO 10,78 MESSAGE ""
   DEFINE BAR  1 OF EDITDATA PROMPT "EDIT DATA           " MESSAGE "" SKIP
   DEFINE BAR  2 OF EDITDATA PROMPT "====================" MESSAGE "" SKIP
   DEFINE BAR  3 OF EDITDATA PROMPT "CLEAR PREVIOUS DATA " MESSAGE ""
   DEFINE BAR  4 OF EDITDATA PROMPT "DOMAIN PARAMETERS   " MESSAGE ""
   DEFINE BAR  5 OF EDITDATA PROMPT "ELEMENT/NODE CONFIG " MESSAGE ""
   DEFINE BAR  6 OF EDITDATA PROMPT "AQUIFER PROPERTIES  " MESSAGE ""
   DEFINE BAR  7 OF EDITDATA PROMPT "NODAL VARIABLES     " MESSAGE ""
   ON SELECTION POPUP EDITDATA DO EDITDATA

   DEFINE POPUP PROGRUN FROM  2,37 TO  8,78 MESSAGE ""
   DEFINE BAR  1 OF PROGRUN  PROMPT "          PROGRAM EXECUTION             " MESSAGE "" SKIP
   DEFINE BAR  2 OF PROGRUN  PROMPT "========================================" MESSAGE "" SKIP
   DEFINE BAR  3 OF PROGRUN  PROMPT "========================================" MESSAGE "" SKIP
   DEFINE BAR  4 OF PROGRUN  PROMPT "========================================" MESSAGE "" SKIP
   DEFINE BAR  5 OF PROGRUN  PROMPT "COUPLED GROUNDWATER FLOW AND DIFFUSION  " MESSAGE ""
   ON SELECTION POPUP PROGRUN DO PROGRUN

   DEFINE POPUP OUTPUT      FROM  2,37 TO  9,78 MESSAGE ""
   DEFINE BAR  1 OF OUTPUT   PROMPT "                 OUTPUT                 " MESSAGE "" SKIP
   DEFINE BAR  2 OF OUTPUT   PROMPT "========================================" MESSAGE "" SKIP
   DEFINE BAR  3 OF OUTPUT   PROMPT "PRINT DATA FOR RUN                       " MESSAGE ""
   DEFINE BAR  4 OF OUTPUT   PROMPT "PRINT NODAL VARIABLES PER TIME STEP      " MESSAGE ""
   DEFINE BAR  5 OF OUTPUT   PROMPT "VIEW MESH MOVEMENT                       " MESSAGE ""
   DEFINE BAR  6 OF OUTPUT   PROMPT "VIEW CONTAMINANT TRANSPORT               " MESSAGE ""
   ON SELECTION POPUP OUTPUT DO OUTPUT

   DEFINE POPUP DUMMY   FROM 2,3 TO 10,27
   DEFINE BAR 1 OF DUMMY    PROMPT "                    "       MESSAGE "" SKIP
```

```
   DEFINE BAR 2 OF DUMMY    PROMPT "                    "        MESSAGE "" SKIP
   DEFINE BAR 3 OF DUMMY    PROMPT "                    "        MESSAGE "" SKIP
   DEFINE BAR 4 OF DUMMY    PROMPT "                    "        MESSAGE "" SKIP
   DEFINE BAR 5 OF DUMMY    PROMPT "                    "        MESSAGE "" SKIP
   DEFINE BAR 6 OF DUMMY    PROMPT "                    "        MESSAGE "" SKIP
   DEFINE BAR 7 OF DUMMY    PROMPT "                    "        MESSAGE "" SKIP
   ON SELECTION POPUP DUMMY DO DUMMY

   DEFINE WINDOW CRITERIA FROM 15, 3 TO 19,50
   DEFINE WINDOW HELPER   FROM  4,10 TO 16,70
   DEFINE WINDOW CODES    FROM  2, 3 TO 10,68
   DEFINE WINDOW INFO1    FROM  2, 3 TO 22,50
   DEFINE WINDOW INFO2    FROM  2, 3 TO 22,50
   DEFINE WINDOW INFO3    FROM  2, 3 TO 22,50
   DEFINE WINDOW INFO4    FROM  2, 3 TO 22,50
   DEFINE WINDOW INFO5    FROM  2, 3 TO 10,50
   DEFINE WINDOW CHOICE   FROM  1, 3 TO  7,50

   DEFINE MENU CHOICE

   DEFINE PAD CHOICE1 OF CHOICE AT 3,13 PROMPT CHOICE1
   DEFINE PAD CHOICE2 OF CHOICE AT 3,28 PROMPT CHOICE2
   ON SELECTION PAD CHOICE1 OF CHOICE DO C1
   ON SELECTION PAD CHOICE2 OF CHOICE DO C2

RETURN

PROCEDURE MESG
   PARAMETERS MESG
   ACTIVATE WINDOW CRITERIA
   @1,1 SAY MESG
   SET ESCAPE OFF
   WAIT " "
   SET ESCAPE ON
   DEACTIVATE WINDOW CRITERIA
RETURN

PROCEDURE HELPER
   ACTIVATE WINDOW HELPER
   DO CASE
   OTHERWISE
      TEXT
   Move through the menus using the UP and DOWN
   arrow keys.   To make a selection, place the
   cursor on your choice and press RETURN.

   To exit from a menu, press ESC.




   F3 - explanation of data codes.
```

```
      ENDTEXT
      READ
   ENDCASE
   DEACTIVATE WINDOW HELPER
RETURN

PROCEDURE CODES
   ACTIVATE WINDOW CODES
      @ 0, 1 SAY "                                                    "
      @ 1, 1 SAY "                                                    "
      @ 2, 1 SAY "                                                    "
      @ 3, 1 SAY "                                                    "
      @ 4, 1 SAY "                                                    "
      @ 5, 1 SAY "                                                    "
      @ 6, 1 SAY "                                                    "
      READ
   DEACTIVATE WINDOW CODES
RETURN

PROCEDURE IFERR
   CLEAR
   ? ERROR()
   ? "RECORD ERROR NUMBER AND REPORT TO NATIONAL MANAGEMENT SYSTEMS"
   READ
   RUN CD\
   QUIT
RETURN

PROCEDURE PRNON
   MESG="SEND OUTPUT TO PRINTER? "
   DO CHOICE
   IF CHOICE="Y"
      PRN="Y"
      MESG="MAKE SURE THE PRINTER IS CONNECTED"
      DO MESG WITH MESG
      SET CONSOLE OFF
      SET PRINT ON
   ELSE
      CLEAR
   ENDIF
RETURN

PROCEDURE PRNOFF
   IF UPPER(PRN)="Y"
      PRN="N"
      SET PRINT OFF
      SET CONSOLE ON
      EJECT
   ELSE
      READ
      CLEAR
   ENDIF
RETURN
```

```
PROCEDURE DUMMY

RETURN

PROCEDURE CHOICE
   CHOICE="N"
   ACTIVATE WINDOW CHOICE
   @1, 2 SAY MESG
   ACTIVATE MENU CHOICE
   PROCEDURE C
      IF BAR()=1
         CHOICE="N"
      ELSE
         CHOICE="Y"
      ENDIF
      DEACTIVATE POPUP
      DEACTIVATE WINDOW CHOICE
   RETURN
   PROCEDURE C1
      CHOICE="N"
      DEACTIVATE WINDOW CHOICE
      DEACTIVATE MENU
   RETURN
   PROCEDURE C2
      CHOICE="Y"
      DEACTIVATE WINDOW CHOICE
      DEACTIVATE MENU
   RETURN
RETURN

*****************************************************************************

PROCEDURE VARLIST
   PUBLIC ELEMV,MAXV,HV,STEPSV,STEPSVP,TIMEV,EV,N1V,N2V,N3V,N4V,N5V,N6V,N7V,N8V
   PUBLIC KXV,KYV,KUV,SV,PV,DXV,DYV,DUV
   PUBLIC NV,XV,YV,UV,OV,BSV,QV,CV,CSV,SSV,CFV
   PUBLIC XORIGINV,YORIGINV,XSCALEV,YSCALEV,ZFACTORV,ANGLEV
   PUBLIC SSTEPV,ESTEPV
   PUBLIC PLANEV,COORDV,HIGHESTV,INTERVALV
   STORE 11    TO ELEMV
   STORE 111   TO MAXV
   STORE 1111  TO HV
   STORE 111   TO STEPSV
   STORE 111   TO STEPSVP
   SSTEPV=0
   ESTEPV=0
   STORE 111111 TO TIMEV
   STORE 11    TO EV
   STORE 111   TO N1V
   STORE 111   TO N2V
   STORE 111   TO N3V
   STORE 111   TO N4V
   STORE 111   TO N5V
   STORE 111   TO N6V
```

```
STORE 111   TO N7V
STORE 111   TO N8V
STORE SPACE(6)  TO KXV
STORE SPACE(6)  TO KYV
STORE SPACE(6)  TO KUV
STORE SPACE(6)  TO SV
STORE 1.1111    TO PV
STORE SPACE(6)  TO DXV
STORE SPACE(6)  TO DYV
STORE SPACE(6)  TO DUV
STORE 111       TO NV
STORE 1111.111  TO XV
STORE 1111.111  TO YV
STORE 1111.111  TO UV
STORE 1111.111  TO OV
STORE SPACE(2)  TO BSV
STORE SPACE(6)  TO QV
STORE 111.111   TO CV
STORE SPACE(2)  TO CSV
STORE SPACE(2)  TO SSV
STORE SPACE(6)  TO CFV
STORE SPACE(1)  TO PLANEV
STORE 11111.11  TO COORDV
STORE 11.11     TO HIGHESTV
STORE 11.11     TO INTERVALV
EV=0
N1V=0
N2V=0
N3V=0
N4V=0
N5V=0
N6V=0
N7V=0
N8V=0
ELEMV=0
MAXV=0
HV=0
PV=0
STEPSV=0
STEPSVP=0
TIMEV=0
NV=0
XV=0
YV=0
UV=0
HV=0
QV=0
CV=0
XORIGINV=100
YORIGINV=100
XSCALEV=200
YSCALEV=100
ZFACTORV=1.0
ANGLEV=0.78
```

```
   COORDV=0
   HIGHESTV=0
   INTERVALV=0
RETURN

PROCEDURE SCREEN1
   @ 1, 3 SAY "ENTER THE AQUIFER DOMAIN PARAMETERS"
   @10, 3 SAY "NUMBER OF ELEMENTS              " GET ELEMV  PICTURE  "99"    RANGE 1,99
   @13, 3 SAY "NUMBER OF NODES                 "  GET MAXV   PICTURE  "999"   RANGE 8,100_
              MESSAGE "NUMBER OF NODES MUST NOT EXCEED 100"
   READ
RETURN

PROCEDURE SCREEN2
   @ 1, 3 SAY "ENTER THE ELEMENT-NODAL CONFIGURATIONS"
   @ 3, 3 SAY "ELEMENT NUMBER "
   @ 3,19 SAY EV
   @10, 3 SAY "NODE 1 " GET N1V    PICTURE "999"   RANGE 0,MAXV
   @11, 3 SAY "NODE 2 " GET N2V    PICTURE "999"   RANGE 0,MAXV
   @12, 3 SAY "NODE 3 " GET N3V    PICTURE "999"   RANGE 0,MAXV
   @13, 3 SAY "NODE 4 " GET N4V    PICTURE "999"   RANGE 0,MAXV
   @14, 3 SAY "NODE 5 " GET N5V    PICTURE "999"   RANGE 0,MAXV
   @15, 3 SAY "NODE 6 " GET N6V    PICTURE "999"   RANGE 0,MAXV
   @16, 3 SAY "NODE 7 " GET N7V    PICTURE "999"   RANGE 0,MAXV
   @17, 3 SAY "NODE 8 " GET N8V    PICTURE "999"   RANGE 0,MAXV
   READ
RETURN

PROCEDURE SCREEN3
   @ 1, 3 SAY "ENTER AQUIFER CHARACTERISTICS PER ELEMENT"
   @ 3, 3 SAY "ELEMENT NUMBER "
   @ 3,19 SAY EV
   @ 8, 3 SAY "PERMEABILITY    SPECIFIC     DISPERSION  "
   @ 9, 3 SAY "COEFFICIENTS    STORAGE      COEFFICIENTS"
   @10, 3 SAY "    m/s           1/m            m)/s    "
   @12, 3 SAY "KX "  GET KXV   PICTURE "999E-9"
   @14, 3 SAY "KY "  GET KYV   PICTURE "999E-9"
   @16, 3 SAY "KU "  GET KUV   PICTURE "999E-9"
   @12,19 SAY "S "   GET SV    PICTURE "999E-9"
   @14,18 SAY "POROSITY"
   @16,19 SAY "P "   GET PV    PICTURE "9.9999"
   @12,34 SAY "DX "  GET DXV   PICTURE "999E-9"
   @14,34 SAY "DY "  GET DYV   PICTURE "999E-9"
   @16,34 SAY "DU "  GET DUV   PICTURE "999E-9"
   READ
RETURN

PROCEDURE SCREEN4
   @ 1, 3 SAY "ENTER NODAL VARIABLES"
   @ 3, 3 SAY "NODE NUMBER "
   @ 3,19 SAY NV
   @ 5, 3 SAY "X COORDINATE (m)          " GET XV   PICTURE "9999.999"
   @ 6, 3 SAY "Y COORDINATE (m)          " GET YV   PICTURE "9999.999"
   @ 7, 3 SAY "U COORDINATE (m)          " GET UV   PICTURE "9999.999"
```

```
   @ 9, 3 SAY "POTENTIAL (m)              " GET OV  PICTURE "9999.999"
   @10, 3 SAY "CONSTANT POTENTIAL NODE? (Y/N)      " GET BSV_
          VALID BSV="Y" .OR. BSV="N" .OR. BSV="S"
   @11, 3 SAY "IS THIS A FREE SURFACE? (Y/N)       " GET SSV_
          VALID SSV="Y" .OR. SSV="N"
   @12, 3 SAY "APPLIED NODAL FLUX (1/s) " GET QV   PICTURE "X99E-9"
   @14, 3 SAY "CONCENTRATION              " GET CV   PICTURE "999.999"
   @15, 3 SAY "ENTER BOUNDARY TYPE? (C/F/N)        " GET CSV_
          VALID CSV="C" .OR. CSV="N" .OR. CSV="F"
   @17, 3 SAY "NODAL CONC FLUX (kg/m}/s)"
   @17,29 SAY CFV  PICTURE "X99E-9"
   READ
   IF CSV="F"
   @17, 3 SAY "NODAL CONC FLUX (kg/m}/s)" GET CFV  PICTURE "X99E-9"
   ENDIF
   READ
RETURN

PROCEDURE SCREEN5
   @ 1, 3 SAY "SPECIFY THE TIME PARAMETERS"
   @ 7, 3 SAY "TIME INTERVAL                  " GET HV   PICTURE "9999"  VALID HV>0
   @11, 3 SAY "NUMBER OF TIME STEPS FOR RUN  " GET STEPSV  PICTURE "999" RANGE STEPSVP+1,999
   @15, 3 SAY "NOTE:"
   @15, 9 SAY SUBSTR(STR(STEPSVP),8,3)
   @15,13 SAY "STEPS HAVE ALREADY BEEN EXECUTED"
   @17, 3 SAY "TOTAL TIME ELAPSED = "
   @17,24 SAY SUBSTR(STR(TIMEV),5,6)
   @17,31 SAY "SECONDS"
   READ
RETURN

PROCEDURE SCREEN6
   @ 1, 3 SAY "SPECIFY THE MESH DISPLAY PARAMETERS"
   @ 6, 3 SAY "X-ORIGIN COORDINATE " GET XORIGINV  PICTURE "999"  RANGE 1,639
   @ 8, 3 SAY "Y-ORIGIN COORDINATE " GET YORIGINV  PICTURE "999"  RANGE 1,479
   @10, 3 SAY "X-SCALE             " GET XSCALEV   PICTURE "999"  RANGE 1,999
   @12, 3 SAY "Y-SCALE             " GET YSCALEV   PICTURE "999"  RANGE 1,999
   @14, 3 SAY "Z-FACTOR            " GET ZFACTORV  PICTURE "9.9"
   @16, 3 SAY "ANGLE  (radians)    " GET ANGLEV    PICTURE "9.99"
   READ
RETURN
PROCEDURE SCREEN7
   @ 1, 3 SAY "SPECIFY THE CONTOUR DISPLAY PARAMETERS"
   @ 3, 3 SAY "CONTOUR PLANE X,Y,U " GET PLANEV    PICTURE "X"_
              VALID PLANEV="X" .OR. PLANEV="Y" .OR. PLANEV="U"
   @ 5, 3 SAY "PLANE COORDINATE    " GET COORDV    PICTURE "99999.99"
   @ 7, 3 SAY "X-ORIGIN COORDINATE " GET XORIGINV  PICTURE "999"      RANGE 1,639
   @ 9, 3 SAY "Y-ORIGIN COORDINATE " GET YORIGINV  PICTURE "999"      RANGE 1,479
   @11, 3 SAY "X-SCALE             " GET XSCALEV   PICTURE "999"      RANGE 1,999
   @13, 3 SAY "Y-SCALE             " GET YSCALEV   PICTURE "999"      RANGE 1,999
   @15, 3 SAY "HIGHEST CONTOUR     " GET HIGHESTV  PICTURE "99.99"
   @17, 3 SAY "CONTOUR INTERVAL    " GET INTERVALV PICTURE "99.99"
   READ
RETURN
```

```
PROCEDURE PLACE1
   REPLACE ELEM  WITH SUBSTR(STR(ELEMV),9,2)+","
   REPLACE MAX   WITH SUBSTR(STR(MAXV),8,3)+","
RETURN

PROCEDURE PLACE2
   REPLACE E  WITH SUBSTR(STR(EV),9,2)+","
   REPLACE N1 WITH SUBSTR(STR(N1V),8,3)+","
   REPLACE N2 WITH SUBSTR(STR(N2V),8,3)+","
   REPLACE N3 WITH SUBSTR(STR(N3V),8,3)+","
   REPLACE N4 WITH SUBSTR(STR(N4V),8,3)+","
   REPLACE N5 WITH SUBSTR(STR(N5V),8,3)+","
   REPLACE N6 WITH SUBSTR(STR(N6V),8,3)+","
   REPLACE N7 WITH SUBSTR(STR(N7V),8,3)+","
   REPLACE N8 WITH SUBSTR(STR(N8V),8,3)+","
RETURN

PROCEDURE PLACE3
   REPLACE E  WITH SUBSTR(STR(EV),9,2)+","
   REPLACE KX WITH KXV+","
   REPLACE KY WITH KYV+","
   REPLACE KU WITH KUV+","
   REPLACE S  WITH SV+","
   REPLACE P  WITH STR(PV,6,4)+","
   REPLACE DX WITH DXV+","
   REPLACE DY WITH DYV+","
   REPLACE DU WITH DUV+","
RETURN

PROCEDURE PLACE4
    REPLACE N WITH SUBSTR(STR(NV),8,3)+","
    REPLACE X WITH STR(XV,8,3)+","
    REPLACE Y WITH STR(YV,8,3)+","
    REPLACE U WITH STR(UV,8,3)+","
    REPLACE O WITH STR(OV,8,3)+","
    REPLACE BS WITH BSV+","
    REPLACE Q WITH QV+","
    REPLACE C WITH STR(CV,7,3)+","
    REPLACE CS WITH CSV+","
    REPLACE SS WITH SSV+","
    REPLACE CF WITH CFV+","
RETURN

PROCEDURE PLACE5
   REPLACE H      WITH SUBSTR(STR(HV),7,4)+","
   REPLACE STEPS  WITH SUBSTR(STR(STEPSV),8,3)+","
   REPLACE STEPSP WITH SUBSTR(STR(STEPSVP),8,3)+","
   REPLACE TIME   WITH SUBSTR(STR(TIMEV+HV*(STEPSV-STEPSVP)),5,6)+","
RETURN

PROCEDURE PLACE6
   REPLACE XORIGIN WITH SUBSTR(STR(XORIGINV),8,3)+","
   REPLACE YORIGIN WITH SUBSTR(STR(YORIGINV),8,3)+","
   REPLACE XSCALE  WITH SUBSTR(STR(XSCALEV),8,3)+","
```

```
   REPLACE YSCALE  WITH SUBSTR(STR(YSCALEV),8,3)+","
   REPLACE ZFACTOR WITH STR(ZFACTORV,3,1)+","
   REPLACE ANGLE   WITH STR(ANGLEV,4,2)+","
RETURN

PROCEDURE PLACE7
   REPLACE PLANE   WITH PLANEV+","
   REPLACE COORD   WITH STR(COORDV,8,2)+","
   REPLACE XORIGIN WITH SUBSTR(STR(XORIGINV),8,3)+","
   REPLACE YORIGIN WITH SUBSTR(STR(YORIGINV),8,3)+","
   REPLACE XSCALE  WITH SUBSTR(STR(XSCALEV),8,3)+","
   REPLACE YSCALE  WITH SUBSTR(STR(YSCALEV),8,3)+","
   REPLACE HIGHEST WITH STR(HIGHESTV,5,2)+","
   REPLACE INTERVAL WITH STR(INTERVALV,5,2)+","
RETURN
PROCEDURE ASSN1
   ELEMV=VAL(ELEM)
   MAXV=VAL(MAX)
RETURN

PROCEDURE ASSN2
   N1V=VAL(N1)
   N2V=VAL(N2)
   N3V=VAL(N3)
   N4V=VAL(N4)
   N5V=VAL(N5)
   N6V=VAL(N6)
   N7V=VAL(N7)
   N8V=VAL(N8)
RETURN

PROCEDURE ASSN3
   KXV=SUBSTR((KX),1,6)
   KYV=SUBSTR((KY),1,6)
   KUV=SUBSTR((KU),1,6)
   SV =SUBSTR((S),1,6)
   PV =VAL(P)
   DXV=SUBSTR((DX),1,6)
   DYV=SUBSTR((DY),1,6)
   DUV=SUBSTR((DU),1,6)
RETURN

PROCEDURE ASSN4
   XV=VAL(X)
   YV=VAL(Y)
   UV=VAL(U)
   OV=VAL(O)
   BSV=SUBSTR(BS,1,1)
   QV=SUBSTR((Q),1,6)
   CV=VAL(C)
   CSV=SUBSTR(CS,1,1)
   SSV=SUBSTR(SS,1,1)
   CFV=SUBSTR((CF),1,6)
RETURN
```

```
PROCEDURE ASSN5
   HV=VAL(H)
   STEPSV=VAL(STEPS)
   STEPSVP=VAL(STEPS)
   TIMEV=VAL(TIME)
RETURN

PROCEDURE ASSN6
   XORIGINV=VAL(XORIGIN)
   YORIGINV=VAL(YORIGIN)
   XSCALEV =VAL(XSCALE)
   YSCALEV =VAL(YSCALE)
   ZFACTORV=VAL(ZFACTOR)
   ANGLEV  =VAL(ANGLE)
RETURN

PROCEDURE ASSN7
   PLANEV=SUBSTR(PLANE,1,1)
   COORDV=VAL(COORD)
   XORIGINV=VAL(XORIGIN)
   YORIGINV=VAL(YORIGIN)
   XSCALEV =VAL(XSCALE)
   YSCALEV =VAL(YSCALE)
   HIGHESTV=VAL(HIGHEST)
   INTERVALV=VAL(INTERVAL)
RETURN

**********AQUILIB.PRG***END**************************************************
```

Filename: FEMDATA1.DBF Type: DBaseIV database file

Purpose : Stores the domain parameters viz. the number of elements and the number of nodes.

Structure for database: FEMDATA1.DBF

Field	Field Name	Type	Width	Dec	Index
1	ELEM	Character	3		N
2	MAX	Character	4		N

Filename: FEMDATA2.DBF Type: DBaseIV database file

Purpose : Stores the element nodal configurations.

Structure for database: FEMDATA2.DBF

Field	Field Name	Type	Width	Dec	Index
1	E	Character	3		N
2	N1	Character	4		N
3	N2	Character	4		N
4	N3	Character	4		N
5	N4	Character	4		N
6	N5	Character	4		N
7	N6	Character	4		N
8	N7	Character	4		N
9	N8	Character	4		N

Filename: FEMDATA3.DBF Type: DBaseIV database file

Purpose : Stores the elemental properties viz. the permeabilities, the coefficients of diffusion, etc.

Structure for database: FEMDATA3.DBF

Field	Field Name	Type	Width	Dec	Index
1	E	Character	3		N
2	KX	Character	7		N
3	KY	Character	7		N
4	KU	Character	7		N
5	S	Character	7		N
6	P	Character	7		N
7	DX	Character	7		N
8	DY	Character	7		N
9	DU	Character	7		N

Filename: FEMDATA4.DBF Type: DBaseIV database file

Purpose : Stores the initial values of the nodal variables viz. the spatial coordinates, the values of concentration, etc.

Structure for database: FEMDATA4.DBF

Field	Field Name	Type	Width	Dec	Index
1	N	Character	4		N
2	X	Character	8		N
3	Y	Character	8		N
4	U	Character	8		N
5	O	Character	8		N
6	BS	Character	2		N
7	Q	Character	8		N
8	C	Character	8		N
9	CS	Character	2		N
10	SS	Character	2		N
11	CF	Character	8		n

Filename: CONVERT.FEM Type: Turbo BASIC program file

Purpose : Converts the FEMDATA DBaseIV database files into one ASCII file for use by the Finite Element execution program MAIN.FEM.

```
REM   ***CONVERT.FEM***BEGIN***

CLS
OPEN "O",#1,"DATA"
OPEN "O",#2,"MESH"
OPEN "I",#3,"FEMDATA1.DB2"
OPEN "I",#4,"FEMDATA2.DB2"
OPEN "I",#5,"FEMDATA3.DB2"
OPEN "I",#6,"FEMDATA4.DB2"
OPEN "O",#7,"HISTOOO"
WRITE #7,0,0
I=0
DO WHILE DUMMY=0
   INPUT #3,DUMMY
   I=I+1
LOOP
CLOSE #3
OPEN "I",#3,"FEMDATA1.DB2"
FOR J=1 TO I-1
    INPUT #3,DUMMY
NEXT

I=0
DO WHILE DUMMY=0
   INPUT #4,DUMMY
   I=I+1
LOOP
CLOSE #4
OPEN "I",#4,"FEMDATA2.DB2"
FOR J=1 TO I-1
    INPUT #4,DUMMY
NEXT

I=0
DO WHILE DUMMY=0
   INPUT #5,DUMMY
   I=I+1
LOOP
CLOSE #5
OPEN "I",#5,"FEMDATA3.DB2"
FOR J=1 TO I-1
    INPUT #5,DUMMY
NEXT

I=0
DO WHILE DUMMY=0
   INPUT #6,DUMMY
```

```
   I=I+1
LOOP
CLOSE #6
OPEN "I",#6,"FEMDATA4.DB2"
FOR J=1 TO I-1
    INPUT #6,DUMMY
NEXT

INPUT #3,ELEM,MAX
WRITE #1,ELEM,MAX

DIM X(MAX),Y(MAX),U(MAX),H(MAX),B$(MAX),Q(MAX),S$(MAX),C(MAX),C$(MAX),CF(MAX)

FOR NODE=1 TO MAX
    INPUT #6,DUMMYN,X(NODE),Y(NODE),U(NODE),H(NODE),B$(NODE),Q(NODE),C(NODE),C$(NODE),_
          S$(NODE),CF(NODE)
    WRITE #7,DUMMYN,X(NODE),Y(NODE),U(NODE),H(NODE),C(NODE)
NEXT NODE

FOR ELEMENT=1 TO ELEM
    INPUT #5,DUMMYE,KX,KY,KU,S,P,DX,DY,DU
    WRITE #1,KX,KY,KU,S,P,DX,DY,DU
    INPUT #4,DUMMYE,N(1),N(2),N(3),N(4),N(5),N(6),N(7),N(8)
    FOR I=1 TO 8
        WRITE #1,H(N(I)),B$(N(I)),Q(N(I)),S$(N(I)),C(N(I)),C$(N(I)),CF(N(I))
        WRITE #2,N(I),X(N(I)),Y(N(I)),U(N(I))
    NEXT I
    WRITE #7,0,0,0
NEXT ELEMENT

CLOSE 1,2,3,4,5,6,7

REM    ***CONVERT.FEM***END*****
```

Filename: STEPS.DBF Type: DBaseIV database file

Purpose : Stores the range of steps to be used in the execution of DRW.FEM, CONTOUR.FEM and OUTPUT2.FEM.

Structure for database: STEPS.DBF

Field	Field Name	Type	Width	Dec	Index
1	SSTEP	Character	5		N
2	ESTEP	Character	5		N

Filename: GRAPH.DBF Type: DBaseIV database file

Purpose : Stores the parameters governing the graphical output of the Finite Element mesh.

Structure for database: GRAPH.DBF

Field	Field Name	Type	Width	Dec	Index
1	XORIGIN	Character	4		N
2	YORIGIN	Character	4		N
3	XSCALE	Character	4		N
4	YSCALE	Character	4		N
5	ZFACTOR	Character	4		N
6	ANGLE	Character	6		N

Filename: CONTOUR.DBF Type: DBaseIV database file

Purpose : Stores the parameters governing the graphical output of the concentration contours.

Structure for database: CONTOUR.DBF

Field	Field Name	Type	Width	Dec	Index
1	PLANE	Character	2		N
2	COORD	Character	9		N
3	XORIGIN	Character	4		N
4	YORIGIN	Character	4		N
5	XSCALE	Character	4		N
6	YSCALE	Character	4		N
7	HIGHEST	Character	6		N
8	INTERVAL	Character	6		N

Filename: DRW.FEM Type: Turbo BASIC program file

Purpose : Graphically traces the mesh per time step in 3D.

```
REM   ***DRW.FEM***BEGIN***

OPEN "I",#1,"DATA"
OPEN "I",#2,"COUNTER.DB2"
INPUT#1,ELEM,MAX
INPUT #2,DUMMY,DUMMY,DUMMY
INPUT#2,STEPSP,H,STEPS,T
CLOSE #1,#2
DIM N(MAX),X(MAX),Y(MAX),U(MAX)

OPEN "I",#1,"GRAPH.DB2"
I=0
DO WHILE DUMMY=0
   INPUT #1,DUMMY
   I=I+1
LOOP
CLOSE #1
OPEN "I",#1,"GRAPH.DB2"
FOR J=1 TO I-1
    INPUT #1,DUMMY
NEXT
INPUT#1,XORIGIN,YORIGIN,XSCALE,YSCALE,ZFACTOR,ANGLE
CLOSE #1

I=0
OPEN "I",#1,"STEPS.DB2"
DO WHILE NOT EOF(1)
   I=I+1
   INPUT#1,DUMMY(I)
LOOP
ESTEP=DUMMY(I)
SSTEP=DUMMY(I-1)
CLOSE #1

CLS

SCREEN 2
FOR DRWSTEP=SSTEP TO ESTEP
   OPEN "I",#3,"FEMDATA4.DB2"
   I=0
   DO WHILE DUMMY=0
      INPUT #3,DUMMY
      I=I+1
   LOOP
CLOSE #3
OPEN "I",#3,"FEMDATA4.DB2"
FOR J=1 TO I-1
    INPUT #3,DUMMY
    NEXT
```

```
HIST$="HIST"+MID$(STR$(DRWSTEP),2,3)
IF DRWSTEP<>0 THEN DRW=1
IF DRW=0 THEN
   OPEN "I",#2,"HISTOOO"
   DRWSTEP=DRWSTEP-1
   DRW=1
ELSE
   OPEN "I",#2,HIST$
END IF
INPUT #2,TIME,ITERATE
FOR NODE=1 TO MAX
    INPUT #2,N,X(NODE),Y(NODE),U(NODE),H(NODE),C
    INPUT #3,N,X,Y,U,H,B$,Q,C,C$,S$
    IF S$="Y" THEN
       U(NODE)=H(NODE)
    END IF
NEXT NODE
CLOSE #3
OPEN "I",#1,"FEMDATA2.DB2"
I=0
DO WHILE DUMMY=0
   INPUT #1,DUMMY
   I=I+1
LOOP
CLOSE #1
OPEN "I",#1,"FEMDATA2.DB2"
FOR J=1 TO I-1
    INPUT #1,DUMMY
NEXT
FOR ELEMENT=1 TO ELEM
    INPUT #1,E
    FOR NODE=1 TO 8
        INPUT #1,N(NODE)
    NEXT NODE
    FOR SNODE=1 TO 8
        XSTART=X(N(SNODE))-Y(N(SNODE))*COS(ANGLE)*ZFACTOR
        YSTART=-U(N(SNODE))+Y(N(SNODE))*SIN(ANGLE)*ZFACTOR
        FOR NODE=SNODE TO 8
            IF SNODE=1 THEN
               IF NODE=2 OR NODE=4 OR NODE=5 THEN
                  XEND=X(N(NODE))-Y(N(NODE))*COS(ANGLE)*ZFACTOR
                  YEND=-U(N(NODE))+Y(N(NODE))*SIN(ANGLE)*ZFACTOR
                  LINE (XSTART*XSCALE+XORIGIN,YSTART*YSCALE+YORIGIN)-_
                       (XEND*XSCALE+XORIGIN,YEND*YSCALE+YORIGIN)
               END IF
            ELSE
               IF SNODE=2 THEN
                  IF NODE=3 OR NODE=6 THEN
                     XEND=X(N(NODE))-Y(N(NODE))*COS(ANGLE)*ZFACTOR
                     YEND=-U(N(NODE))+Y(N(NODE))*SIN(ANGLE)*ZFACTOR
                     LINE (XSTART*XSCALE+XORIGIN,YSTART*YSCALE+YORIGIN)-_
                          (XEND*XSCALE+XORIGIN,YEND*YSCALE+YORIGIN)
                  END IF
               ELSE
```

```
                    IF SNODE=3 THEN
                       IF NODE=4 OR NODE=7 THEN
                          XEND=X(N(NODE))-Y(N(NODE))*COS(ANGLE)*ZFACTOR
                          YEND=-U(N(NODE))+Y(N(NODE))*SIN(ANGLE)*ZFACTOR
                          LINE (XSTART*XSCALE+XORIGIN,YSTART*YSCALE+YORIGIN)-_
                               (XEND*XSCALE+XORIGIN,YEND*YSCALE+YORIGIN)
                       END IF
                    ELSE
                       IF SNODE=4 THEN
                          IF NODE=8 THEN
                             XEND=X(N(NODE))-Y(N(NODE))*COS(ANGLE)*ZFACTOR
                             YEND=-U(N(NODE))+Y(N(NODE))*SIN(ANGLE)*ZFACTOR
                             LINE (XSTART*XSCALE+XORIGIN,YSTART*YSCALE+YORIGIN)-_
                                  (XEND*XSCALE+XORIGIN,YEND*YSCALE+YORIGIN)
                          END IF
                       ELSE
                        IF SNODE=5 THEN
                           IF NODE=6 OR NODE=8 THEN
                                 XEND=X(N(NODE))-Y(N(NODE))*COS(ANGLE)*ZFACTOR
                                 YEND=-U(N(NODE))+Y(N(NODE))*SIN(ANGLE)*ZFACTOR
                                 LINE (XSTART*XSCALE+XORIGIN,YSTART*YSCALE+YORIGIN)-_
                                      (XEND*XSCALE+XORIGIN,YEND*YSCALE+YORIGIN)
                             END IF
                           ELSE
                              IF SNODE=6 THEN
                                 IF NODE=7 THEN
                                    XEND=X(N(NODE))-Y(N(NODE))*COS(ANGLE)*ZFACTOR
                                    YEND=-U(N(NODE))+Y(N(NODE))*SIN(ANGLE)*ZFACTOR
                                    LINE (XSTART*XSCALE+XORIGIN,YSTART*YSCALE+YORIGIN)-_
                                         (XEND*XSCALE+XORIGIN,YEND*YSCALE+YORIGIN)
                                 END IF
                              ELSE
                                 IF SNODE=7 THEN
                                    IF NODE=8 THEN
                                       XEND=X(N(NODE))-Y(N(NODE))*COS(ANGLE)*ZFACTOR
                                       YEND=-U(N(NODE))+Y(N(NODE))*SIN(ANGLE)*ZFACTOR
                                       LINE (XSTART*XSCALE+XORIGIN,YSTART*YSCALE+YORIGIN)-_
                                            (XEND*XSCALE+XORIGIN,YEND*YSCALE+YORIGIN)
                                    END IF
                                 END IF
                              END IF
                           END IF
                        END IF
                       END IF
                    END IF
                 END IF
              END IF
           NEXT NODE
        NEXT SNODE
     NEXT ELEMENT
     CLOSE #1,#2
     IF DRWSTEP>=0 THEN
        LOCATE 23,1:PRINT "STEP NO.";DRWSTEP;"  PRESS ANY KEY TO CONTINUE (C FOR MULTI DISPLAY)"
     ELSE
        LOCATE 23,1:PRINT "MESH FROM INPUT DATA;PRESS ANY KEY TO CONTINUE (C FOR MULTI DISPLAY)"
```

```
    END IF
    A$=""
    WHILE A$="":A$=INKEY$:WEND
    IF A$<>"C" THEN
       CLS
    END IF
NEXT DRWSTEP
CLOSE

REM    ***DRW.FEM***END*****
```

```
Filename: CONTOUR.FEM        Type: Turbo BASIC program file

Purpose : Graphically traces the concentration contours  per
          time step in any two-dimensional plane.


REM   ***CONTOUR.FEM***BEGIN***

CLS
OPEN "I",#1,"DATA"
INPUT#1,ELEM,MAX
CLOSE #1
DIM N(MAX),X(MAX),Y(MAX),U(MAX),C(MAX),ELEMENT(ELEM*4),SIDES(MAX)

OPEN "I",#1,"COUNTER.DB2"
DO WHILE NOT EOF(1)
   I=I+1
   INPUT#1,DUMMY(I)
LOOP
TIMEV=DUMMY(I)
STEPS=DUMMY(I-1)
H=DUMMY(I-2)
STEPSP=DUMMY(I-3)
CLOSE #1

OPEN "I",#1,"CONTOUR.DB2"
I=0
DO WHILE DUMMY=0
   INPUT #1,DUMMY
   I=I+1
LOOP
CLOSE #1
OPEN "I",#1,"CONTOUR.DB2"
FOR J=1 TO I-2
    INPUT #1,DUMMY
NEXT
INPUT#1,PLANE$,COORD,XORIGIN,YORIGIN,XSCALE,YSCALE,HIGHEST,INTERVAL
CLOSE #1
CLS

OPEN "I",#1,"STEPS.DB2"
DO WHILE NOT EOF(1)
   I=I+1
   INPUT#1,DUMMY(I)
LOOP
ESTEP=DUMMY(I)
SSTEP=DUMMY(I-1)
CLOSE #1

CLS
SCREEN 2
```

```
FOR DRWSTEP=SSTEP TO ESTEP

   HIST$="HIST"+MID$(STR$(DRWSTEP),2,3)
   IF DRW=0 AND SSTEP=0 THEN
      OPEN "I",#2,"HISTOOO"
      IF STEPS<>0 THEN DRWSTEP=DRWSTEP-1
   ELSE
      OPEN "I",#2,HIST$
   END IF
   DRW=1
   INPUT #2,TIME,ITERATE
   MAXX=0:MAXY=0
   FOR NODE=1 TO MAX
       IF PLANE$="U" THEN
          INPUT #2,N,X(NODE),Y(NODE),U(NODE),0,C(NODE)
       ELSE
          IF PLANE$="X" THEN
             INPUT #2,N,U(NODE),Y(NODE),X(NODE),0,C(NODE)
             X(NODE)=X(NODE)
          ELSE
             INPUT #2,N,X(NODE),U(NODE),Y(NODE),0,C(NODE)
             Y(NODE)=Y(NODE)
          END IF
       END IF
       IF ABS(X(NODE))>ABS(MAXX) THEN MAXX=X(NODE)
       IF ABS(Y(NODE))>ABS(MAXY) THEN MAXY=Y(NODE)
   NEXT NODE
   LINE (XORIGIN,YORIGIN)-(MAXX*XSCALE+XORIGIN,YORIGIN)
   LINE (XORIGIN,YORIGIN)-(XORIGIN,MAXY*YSCALE+YORIGIN)
   CLOSE #2

   OPEN "O",#5,"CONTOUR1"
   CONTOUR1=0
   OPEN "I",#1,"FEMDATA2.DB2"
   I=0
   DO WHILE DUMMY=0
      INPUT #1,DUMMY
      I=I+1
   LOOP
   CLOSE #1
   OPEN "I",#1,"FEMDATA2.DB2"
   FOR J=1 TO I-1
   INPUT #1,DUMMY
   NEXT

   U=COORD
   FOR ELEMENT=1 TO ELEM
       INPUT #1,E
       FOR NODE=1 TO 8
           INPUT #1,N(NODE)
       NEXT NODE
       NN(1)=N(1):NN(2)=N(2):NN(3)=N(3):NN(4)=N(4):NN(5)=N(5):NN(6)=N(6):NN(7)=N(7):NN(8)=N(8)
       IF PLANE$="X" THEN
       N(1)=NN(1):N(2)=NN(5):N(3)=NN(8):N(4)=NN(4):N(5)=NN(2):N(6)=NN(6):N(7)=NN(7):N(8)=NN(3)
```

```
    END IF
    IF PLANE$="Y" THEN
    N(1)=NN(5):N(2)=NN(6):N(3)=NN(2):N(4)=NN(1):N(5)=NN(8):N(6)=NN(7):N(7)=NN(3):N(8)=NN(4)
    END IF
    FOR I=1 TO 8
        IF U(N(I))>U THEN
           IF I<=4 THEN C=+4 ELSE C=-4
           IF I<4 OR I>4 AND I<8 THEN A=+1 ELSE A=-1
           IF I>1 AND I<4 OR I>5 AND I<8 THEN B=-1
           IF I=1 OR I=5 THEN B=+3
           IF I=4 OR I=8 THEN B=-3
           FOR J=1 TO 8
               IF J<>I THEN
                  IF PLANE$="U" THEN
                     IF U(N(J))<=U THEN
                        IF J=I+A OR J=I+B OR J=I+C THEN
                           C=C(N(I))+(C(N(J))-C(N(I)))*(U(N(I))-U)/(U(N(I))-U(N(J)))
                           X=X(N(I))+(X(N(J))-X(N(I)))*(U(N(I))-U)/(U(N(I))-U(N(J)))
                           Y=Y(N(I))+(Y(N(J))-Y(N(I)))*(U(N(I))-U)/(U(N(I))-U(N(J)))
                           WRITE #5,C,X,Y,ELEMENT
                           CONTOUR1=CONTOUR1+1
                        END IF
                     END IF
                  ELSE
                     IF PLANE$="X" THEN
                        IF U(N(J))<=U THEN
                           IF J=I+A OR J=I+B OR J=I+C THEN
                              C=C(N(I))+(C(N(J))-C(N(I)))*(U(N(I))-U)/(U(N(I))-U(N(J)))
                              X=X(N(I))+(X(N(J))-X(N(I)))*(U(N(I))-U)/(U(N(I))-U(N(J)))
                              Y=Y(N(I))+(Y(N(J))-Y(N(I)))*(U(N(I))-U)/(U(N(I))-U(N(J)))
                              WRITE #5,C,X,Y,ELEMENT
                              CONTOUR1=CONTOUR1+1
                           END IF
                        END IF
                     ELSE
                        IF U(N(J))<=U THEN
                           IF J=I+A OR J=I+B OR J=I+C THEN
                              C=C(N(I))+(C(N(J))-C(N(I)))*(U(N(I))-U)/(U(N(I))-U(N(J)))
                              X=X(N(I))+(X(N(J))-X(N(I)))*(U(N(I))-U)/(U(N(I))-U(N(J)))
                              Y=Y(N(I))+(Y(N(J))-Y(N(I)))*(U(N(I))-U)/(U(N(I))-U(N(J)))
                              WRITE #5,C,X,Y,ELEMENT
                              CONTOUR1=CONTOUR1+1
                           END IF
                        END IF
                     END IF
                  END IF
               END IF
           NEXT
        END IF
    NEXT
NEXT
CLOSE #1,#5
```

```
FOR CONTOUR=HIGHEST TO -.001 STEP -INTERVAL
    IF CONTOUR<=0 THEN
       C=0.001
    ELSE
       C=CONTOUR
    END IF
    OPEN "I",#5,"CONTOUR1"
    FOR I=1 TO CONTOUR1
        INPUT #5,C(I),X(I),Y(I),ELEMENT(I)
    NEXT
    CLOSE #5

    I=0
    OPEN "O",#6,"CONTOUR2"
    CONTOUR2=0
    DO WHILE I<CONTOUR1
       IF C(I+1)>=C AND C(I+2)<=C OR C(I+1)<=C AND C(I+2)>=C THEN
          IF C(I+2)>=C AND C(I+3)<=C OR C(I+2)<=C AND C(I+3)>=C THEN
             IF C(I+2)>C AND C(I+4)<C OR C(I+2)<C AND C(I+4)>C THEN
                X=X(I+1)+(X(I+2)-X(I+1))*(C(I+1)-C)/(C(I+1)-C(I+2))
                Y=Y(I+1)+(Y(I+2)-Y(I+1))*(C(I+1)-C)/(C(I+1)-C(I+2))
                WRITE #6,C,X,Y,ELEMENT(I+1),12
                X=X(I+2)+(X(I+3)-X(I+2))*(C(I+2)-C)/(C(I+2)-C(I+3))
                Y=Y(I+2)+(Y(I+3)-Y(I+2))*(C(I+2)-C)/(C(I+2)-C(I+3))
                WRITE #6,C,X,Y,ELEMENT(I+2),12
                CONTOUR2=CONTOUR2+2
             END IF
          END IF
       END IF
       IF C(I+1)>=C AND C(I+2)<=C OR C(I+1)<=C AND C(I+2)>=C THEN
          IF C(I+4)>=C AND C(I+1)<=C OR C(I+4)<=C AND C(I+1)>=C THEN
             IF C(I+1)>C AND C(I+3)<C OR C(I+1)<C AND C(I+3)>C THEN
                X=X(I+1)+(X(I+2)-X(I+1))*(C(I+1)-C)/(C(I+1)-C(I+2))
                Y=Y(I+1)+(Y(I+2)-Y(I+1))*(C(I+1)-C)/(C(I+1)-C(I+2))
                WRITE #6,C,X,Y,ELEMENT(I+1),14
                X=X(I+4)+(X(I+1)-X(I+4))*(C(I+4)-C)/(C(I+4)-C(I+1))
                Y=Y(I+4)+(Y(I+1)-Y(I+4))*(C(I+4)-C)/(C(I+4)-C(I+1))
                WRITE #6,C,X,Y,ELEMENT(I+4),14
                CONTOUR2=CONTOUR2+2
             END IF
          END IF
       END IF
       IF C(I+1)>=C AND C(I+2)<=C OR C(I+1)<=C AND C(I+2)>=C THEN
          IF C(I+3)>=C AND C(I+4)<=C OR C(I+3)<=C AND C(I+4)>=C THEN
             IF C(I+1)>C AND C(I+3)<C OR C(I+1)<C AND C(I+3)>C THEN
                IF C(I+2)>C AND C(I+4)<C OR C(I+2)<C AND C(I+4)>C THEN
                   X=X(I+1)+(X(I+2)-X(I+1))*(C(I+1)-C)/(C(I+1)-C(I+2))
                   Y=Y(I+1)+(Y(I+2)-Y(I+1))*(C(I+1)-C)/(C(I+1)-C(I+2))
                   WRITE #6,C,X,Y,ELEMENT(I+1),13
                   X=X(I+3)+(X(I+4)-X(I+3))*(C(I+3)-C)/(C(I+3)-C(I+4))
                   Y=Y(I+3)+(Y(I+4)-Y(I+3))*(C(I+3)-C)/(C(I+3)-C(I+4))
                   WRITE #6,C,X,Y,ELEMENT(I+3),13
                   CONTOUR2=CONTOUR2+2
                END IF
```

```
            END IF
         END IF
      END IF
      IF C(I+2)>=C AND C(I+3)<=C OR C(I+2)<=C AND C(I+3)>=C THEN
         IF C(I+4)>=C AND C(I+1)<=C OR C(I+4)<=C AND C(I+1)>=C THEN
            IF C(I+2)>C AND C(I+4)<C OR C(I+2)<C AND C(I+4)>C THEN
               IF C(I+3)>C AND C(I+1)<C OR C(I+3)<C AND C(I+1)>C THEN
                  X=X(I+2)+(X(I+3)-X(I+2))*(C(I+2)-C)/(C(I+2)-C(I+3))
                  Y=Y(I+2)+(Y(I+3)-Y(I+2))*(C(I+2)-C)/(C(I+2)-C(I+3))
                  WRITE #6,C,X,Y,ELEMENT(I+2),24
                  X=X(I+4)+(X(I+1)-X(I+4))*(C(I+4)-C)/(C(I+4)-C(I+1))
                  Y=Y(I+4)+(Y(I+1)-Y(I+4))*(C(I+4)-C)/(C(I+4)-C(I+1))
                  WRITE #6,C,X,Y,ELEMENT(I+4),24
                  CONTOUR2=CONTOUR2+2
               END IF
            END IF
         END IF
      END IF
      IF C(I+2)>=C AND C(I+3)<=C OR C(I+2)<=C AND C(I+3)>=C THEN
         IF C(I+3)>=C AND C(I+4)<=C OR C(I+3)<=C AND C(I+4)>=C THEN
            IF C(I+1)>C AND C(I+3)<C OR C(I+1)<C AND C(I+3)>C THEN
               X=X(I+2)+(X(I+3)-X(I+2))*(C(I+2)-C)/(C(I+2)-C(I+3))
               Y=Y(I+2)+(Y(I+3)-Y(I+2))*(C(I+2)-C)/(C(I+2)-C(I+3))
               WRITE #6,C,X,Y,ELEMENT(I+2),23
               X=X(I+3)+(X(I+4)-X(I+3))*(C(I+3)-C)/(C(I+3)-C(I+4))
               Y=Y(I+3)+(Y(I+4)-Y(I+3))*(C(I+3)-C)/(C(I+3)-C(I+4))
               WRITE #6,C,X,Y,ELEMENT(I+3),23
               CONTOUR2=CONTOUR2+2
            END IF
         END IF
      END IF
      IF C(I+3)>=C AND C(I+4)<=C OR C(I+3)<=C AND C(I+4)>=C THEN
         IF C(I+4)>=C AND C(I+1)<=C OR C(I+4)<=C AND C(I+1)>=C THEN
            IF C(I+2)>C AND C(I+4)<C OR C(I+2)<C AND C(I+4)>C THEN
               X=X(I+3)+(X(I+4)-X(I+3))*(C(I+3)-C)/(C(I+3)-C(I+4))
               Y=Y(I+3)+(Y(I+4)-Y(I+3))*(C(I+3)-C)/(C(I+3)-C(I+4))
               WRITE #6,C,X,Y,ELEMENT(I+3),34
               X=X(I+4)+(X(I+1)-X(I+4))*(C(I+4)-C)/(C(I+4)-C(I+1))
               Y=Y(I+4)+(Y(I+1)-Y(I+4))*(C(I+4)-C)/(C(I+4)-C(I+1))
               WRITE #6,C,X,Y,ELEMENT(I+4),34
               CONTOUR2=CONTOUR2+2
            END IF
         END IF
      END IF
      I=I+4
LOOP
CLOSE #6

OPEN "I",#6,"CONTOUR2"
FOR I=1 TO CONTOUR2
    INPUT #6,C(I),X(I),Y(I),ELEMENT(I),SIDES(I)
NEXT
CLOSE #6
```

```
        FOR I=1 TO CONTOUR2
            FOR J=I+1 TO CONTOUR2
                IF ELEMENT(J)=ELEMENT(I) AND SIDES(J)=SIDES(I) THEN
                   LINE (X(I)*XSCALE+XORIGIN,Y(I)*YSCALE+YORIGIN)-_
                        (X(J)*XSCALE+XORIGIN,Y(J)*YSCALE+YORIGIN)
                   LIN=1
                END IF
            NEXT
        NEXT
        IF LIN=1 THEN
           LIN=0
           PRINT USING "#.###";C
        END IF
    NEXT CONTOUR

    IF DRWSTEP>=0 THEN
       LOCATE 23,1:PRINT "STEP NO. ";DRWSTEP;"  PRESS ANY KEY TO CONTINUE (C FOR MULTI DISPLAY)"
    ELSE
       LOCATE 23,1:PRINT "INITIAL CONCENTRATION;PRESS ANY KEY TO CONTINUE (C FOR MULTI DISPLAY)"
    END IF
    A$=""
    WHILE A$="":A$=INKEY$:WEND
    IF A$<>"C" THEN
       CLS
    END IF
NEXT DRWSTEP
CLOSE

REM    ***CONTOUR.FEM***END*****
```

Filename: OUTPUT1.FEM Type: Turbo BASIC program file

Purpose : Generates a copy of the data used.

```
REM   ***OUTPUT1.FEM***BEGIN***

CLS
OPEN "I",#3,"FEMDATA1.DB2"
OPEN "I",#4,"FEMDATA2.DB2"
OPEN "I",#5,"FEMDATA3.DB2"
OPEN "I",#6,"FEMDATA4.DB2"
I=0
DO WHILE DUMMY=0
   INPUT #3,DUMMY
   I=I+1
LOOP
CLOSE #3
OPEN "I",#3,"FEMDATA1.DB2"
FOR J=1 TO I-1
    INPUT #3,DUMMY
NEXT

I=0
DO WHILE DUMMY=0
   INPUT #4,DUMMY
   I=I+1
LOOP
CLOSE #4
OPEN "I",#4,"FEMDATA2.DB2"
FOR J=1 TO I-1
    INPUT #4,DUMMY
NEXT

I=0
DO WHILE DUMMY=0
   INPUT #5,DUMMY
   I=I+1
LOOP
CLOSE #5
OPEN "I",#5,"FEMDATA3.DB2"
FOR J=1 TO I-1
    INPUT #5,DUMMY
NEXT

I=0
DO WHILE DUMMY=0
   INPUT #6,DUMMY
   I=I+1
LOOP
CLOSE #6
OPEN "I",#6,"FEMDATA4.DB2"
FOR J=1 TO I-1
    INPUT #6,DUMMY
```

```
NEXT
INPUT #3,ELEM,MAX
LPRINT "AQUIFEM DATA FOR RUN"
LPRINT
LPRINT "NUMBER OF ELEMENTS   = ";ELEM
LPRINT "NUMBER OF NODES      = ";MAX
LPRINT

DIM X(MAX),Y(MAX),U(MAX),H(MAX),B$(MAX),Q(MAX),S$(MAX),C(MAX),C$(MAX),CF(MAX)

LPRINT "N       X          Y          U          H      B$  Q           C       C$   S$ CF"
LPRINT
FOR NODE=1 TO MAX
    INPUT #6,DUMMYN,X(NODE),Y(NODE),U(NODE),H(NODE),B$(NODE),Q(NODE),C(NODE),C$(NODE),S$(NODE),_
          CF(NODE)
    LPRINT USING "####";DUMMYN;
    LPRINT USING " #####.#### ";X(NODE);Y(NODE);U(NODE);
    LPRINT USING "  ####.####  ";H(NODE);
    LPRINT B$(NODE);
    LPRINT USING "####.########";Q(NODE);
    LPRINT USING "  ##.####";C(NODE);
    LPRINT "   ";C$(NODE);"    ";S$(NODE);
    LPRINT USING " ##.########";CF(NODE);
NEXT NODE

LPRINT
LPRINT "E   KX        KY        KU        S     P     DX          DY          DU"
LPRINT
FOR ELEMENT=1 TO ELEM
    INPUT #5,DUMMYE,KX,KY,KU,S,P,DX,DY,DU
    LPRINT USING "##   ";DUMMYE;
    LPRINT USING "#.########  ";KX;KY;KU;
    LPRINT USING "#.####  ";S;P;
    LPRINT USING "#.##########  ";DX;DY;DU
    INPUT #4,DUMMYE,N(1),N(2),N(3),N(4),N(5),N(6),N(7),N(8)
    LPRINT "    NODES:  ";
    LPRINT USING "#### ";N(1);N(2);N(3);N(4);N(5);N(6);N(7);N(8)
NEXT ELEMENT

CLOSE 3,4,5,6

LPRINT
LPRINT "Please check whether the data is correct."
LPRINT "AQUIFEM may err in the conversion of data"
LPRINT "from one database type to another."
LPRINT
LPRINT "THANK YOU"

REM   ***OUTPUT1.FEM***END*****
```

```
Filename: OUTPUT2.FEM          Type: Turbo BASIC program file

Purpose : Generates the results of the Finite Element
          procedure per time step.


REM   ***OUTPUT2.FEM***BEGIN***

OPEN "I",#1,"COUNTER.DB2"
OPEN "I",#2,"FEMDATA1.DB2"
OPEN "I",#3,"STEPS.DB2"
DO WHILE NOT EOF(1)
   I=I+1
   INPUT#1,DUMMY(I)
LOOP
TIMEV=DUMMY(I)
STEPS=DUMMY(I-1)
H=DUMMY(I-2)
STEPSP=DUMMY(I-3)
I=0
DO WHILE DUMMY=0
   INPUT #2,DUMMY
   I=I+1
LOOP
CLOSE #2
OPEN "I",#2,"FEMDATA1.DB2"
FOR J=1 TO I-1
    INPUT #2,DUMMY
NEXT
INPUT#2,ELEM,MAX
DO WHILE NOT EOF(3)
   I=I+1
   INPUT#3,DUMMY(I)
LOOP
ESTEP=DUMMY(I)
SSTEP=DUMMY(I-1)
CLOSE #1,#2,#3

PRINT "AQUIFEM RESULTS"
PRINT
PRINT "NUMBER OF ELEMENTS   = ";
PRINT USING "#####";ELEM
PRINT "NUMBER OF NODES      = ";
PRINT USING "#####";MAX
PRINT
FOR COUNTER=SSTEP TO ESTEP
    HIST$="HIST"+MID$(STR$(COUNTER),2,3)
    OPEN "I",#1,HIST$
    INPUT#1,TIME,ITERATE
    PRINT "STEP NUMBER ";
    PRINT USING "#####";COUNTER;
    PRINT "        TIME";
    PRINT USING " #########";TIME;
    PRINT " SECS       NUMBER OF ITERATIONS";
```

```
    PRINT USING "#####"; ITERATE
    PRINT
    PRINT "NODE    X-COORD      Y-COORD      U-COORD    POTENTIAL HEAD   CONCENTRATION"
    PRINT
    FOR NODE=1 TO MAX
        INPUT#1,N,X,Y,U,H,C
        PRINT USING "#####"; N;
        PRINT USING "   ######.#### "; X; Y; U;
        PRINT USING "       ####.####    "; H;
        PRINT USING "        ####.####"; C
        A$=""
        WHILE A$="":A$=INKEY$:WEND
    NEXT NODE
    PRINT
    PRINT "ELEMENT    X-VELOCITY       Y-VELOCITY       U-VELOCITY"
    PRINT
    FOR ELEMENT=1 TO ELEM
        INPUT #1,VX,VY,VU
        PRINT USING "#####"; ELEMENT;
        PRINT USING "       ###.##########"; VX; VY; VU
    NEXT ELEMENT
    PRINT
    CLOSE #1
NEXT COUNTER

REM   ***OUTPUT2.FEM***END*****
```

Filename: MAIN.FEM Type: Turbo BASIC program file

Purpose : Controls the Finite Element execution procedure.

```
REM   ***MAIN.FEM***BEGIN***

$INCLUDE "JACFN.FEM"
$INCLUDE "INTERFN.FEM"
$INCLUDE "SUB.FEM"
CLS
$DYNAMIC
OPEN "I",#1,"COUNTER.DB2"
DO WHILE NOT EOF(1)
   I=I+1
   INPUT#1,DUMMY(I)
LOOP
TIMEV=DUMMY(I)
STEPS=DUMMY(I-1)
H=DUMMY(I-2)
STEPSP=DUMMY(I-3)
IF STEPSP=0 THEN
   TIME=0
ELSE
   TIME=TIMEV-H*(STEPS-STEPSP)+H
END IF
CLOSE #1
OPEN "I",#1,"DATA"
INPUT#1,ELEM,MAX
CLOSE #1
DIM KX(ELEM),KY(ELEM),KU(ELEM),S(ELEM),P(ELEM)
DIM DX(ELEM),DY(ELEM),DU(ELEM)
DIM DXX1(ELEM),DYY1(ELEM),DUU1(ELEM) ',DXY1(ELEM),DYU1(ELEM),DUX1(ELEM)
DIM DXX2(ELEM),DYY2(ELEM),DUU2(ELEM) ',DXY2(ELEM),DYU2(ELEM),DUX2(ELEM)
DIM N(MAX),X(MAX),Y(MAX),U(MAX)
DIM H(MAX),B$(MAX),Q(MAX),S$(MAX)
DIM C(MAX),C$(MAX),CF(MAX)
DIM ELEMENT(ELEM)                                    'MESH.FEM
DIM VX(ELEM),VY(ELEM),VU(ELEM)                       'MESH.FEM
DIM CON$(MAX),CONC$(MAX)

DIM KG(MAX,MAX),LG(MAX,MAX),DG(MAX),RG(MAX)          'GLOBALGF.FEM
DIM MG(MAX,MAX),NG(MAX,MAX),GG(MAX),GGG(MAX)         'GLOBALCD.FEM

DIM A(MAX)                                           'GEMGF.FEM
DIM B(MAX)                                           'GEMCD.FEM

DIM E(MAX),K(MAX,MAX),KH(MAX),KK(MAX,MAX+1)          'MEMORY CHECK
ERASE E,K,KH,KK

NEWCOUNT=1
IF STEPSP=0 THEN
   COUNTER=STEPSP
ELSE
```

```
   COUNTER=STEPSP+1
END IF
IF COUNTER=0 THEN
   FOR E=1 TO ELEM:ELEMENT(E)=1:NEXT
ELSE
   HIST$="HIST"+MID$(STR$(COUNTER-1),2,3)
   OPEN "I",#1,HIST$
   INPUT#1,DUMMY,DUMMY
   FOR I=1 TO MAX
       INPUT#1,DUMMY,DUMMY,DUMMY,DUMMY,DUMMY,DUMMY
   NEXT
   FOR E=1 TO ELEM
       INPUT#1,VX(E),VY(E),VU(E)
   NEXT
   CLOSE #1
END IF

$SEGMENT

MAINSTART:

   HIST$="HIST"+MID$(STR$(COUNTER),2,3)
   OPEN "O",#5,HIST$
   ITERATE=0

   ITERATION:

      ITER=0
      ITERATE=ITERATE+1
      OPEN "I",#1,"DATA"
      OPEN "I",#2,"MESH"
      IF COUNTER>0 OR COUNTER=0 AND ITERATE>1 THEN
         OPEN "I",#3,"MATVEC"
      END IF
      OPEN "O",#4,"MATVECTM"
      LOCATE 5,1:PRINT SPACE$(80)
      LOCATE 5,1:PRINT "STEP NUMBER";COUNTER;"      ITERATION NUMBER";ITERATE
      INPUT#1,ELEM,MAX
      FOR E=1 TO ELEM
          INPUT#1,KX(E),KY(E),KU(E),S(E),P(E),DX(E),DY(E),DU(E)
          FOR I=1 TO 8
              INPUT#2,N(I),X(N(I)),Y(N(I)),U(N(I))
              INPUT#1,H(N(I)),B$(N(I)),Q(N(I)),S$(N(I)),C(N(I)),C$(N(I)),CF(N(I))
              IF COUNTER<>0 OR ITERATE <>1 THEN
                 FOR J=1 TO 8
                     INPUT#3,OO(I,J),OXOX(I,J),OYOY(I,J),OUOU(I,J),OOX(I,J),OOY(I,J),OOU(I,J)
                 NEXT
                 INPUT#3,OB(I),O(I),OC(I)
              END IF
          NEXT
          IF ELEMENT(E)>0 THEN
             RESTORE GFDATA:
             $INCLUDE "MATGF.FEM"
             $INCLUDE "VECGF.FEM"
```

```
    END IF
    IF COUNTER=1 OR COUNTER>1 AND ELEMENT(E)>0 THEN
       RESTORE CDDATA:
       $INCLUDE "MATCD.FEM"
       $INCLUDE "VECCD.FEM"
    END IF
    ELEMENT(E)=0
    FOR I=1 TO 8
        FOR J=1 TO 8
            WRITE#4,00(I,J),0X0X(I,J),0Y0Y(I,J),0U0U(I,J),00X(I,J),00Y(I,J),00U(I,J)
        NEXT
        WRITE#4,0B(I),0(I),0C(I)
    NEXT
    IF COUNTER=0 THEN
       $INCLUDE "GLOBALGF.FEM"
    END IF
    IF COUNTER=1 OR COUNTER>1 AND NEWCOUNT=1 THEN
       $INCLUDE "GLOBALCD.FEM"
    END IF
NEXT
CLOSE #1,#2,#3,#4

OPEN "O",#3,"MATVEC"
OPEN "I",#4,"MATVECTM
FOR E=1 TO ELEM
    FOR I=1 TO 8
        FOR J=1 TO 8
            INPUT#4,00,0X0X,0Y0Y,0U0U,00X,00Y,00U  ',0X0Y,0Y0U,0U0X
            WRITE#3,00,0X0X,0Y0Y,0U0U,00X,00Y,00U  ',0X0Y,0Y0U,0U0X
        NEXT
        INPUT#4,0B,0,0C
        WRITE#3,0B,0,0C
    NEXT
NEXT
CLOSE #3,#4
KILL "MATVECTM"

CLS
IF COUNTER=0 THEN
   DIM E(MAX),K(MAX,MAX),KH(MAX),KK(MAX,MAX+1)
   $INCLUDE "MATRIXGF.FEM"
   ERASE E,K,KH,KK
   $INCLUDE "MESH.FEM"
END IF
IF ITER=0 THEN
   IF COUNTER>0 THEN
      DIM F(MAX),M(MAX,MAX),MH(MAX),MM(MAX,MAX+1)
      $INCLUDE "MATRIXCD.FEM"
      ERASE F,M,MH,MM
   END IF
END IF
OPEN "I",#1,"DATA"
OPEN "O",#6,"TEMPDATA"
OPEN "I",#2,"MESH"
```

```
        INPUT#1,ELEM,MAX
        WRITE#6,ELEM,MAX
        FOR E=1 TO ELEM
            INPUT#1,KX(E),KY(E),KU(E),S(E),P(E),DX(E),DY(E),DU(E)
            WRITE#6,KX(E),KY(E),KU(E),S(E),P(E),DX(E),DY(E),DU(E)
            FOR I=1 TO 8
                INPUT#2,N(I),X,Y,U
                INPUT#1,HOLD,B$,Q,S$,COLD,C$,CF
                WRITE#6,H(N(I)),B$,Q,S$,C(N(I)),C$,CF
            NEXT
        NEXT
        CLOSE #1,#6,#2

        OPEN "O",#1,"DATA"
        OPEN "I",#6,"TEMPDATA"
        INPUT#6,ELEM,MAX
        WRITE#1,ELEM,MAX
        FOR E=1 TO ELEM
            INPUT#6,KX(E),KY(E),KU(E),S(E),P(E)
            INPUT#6,DX(E),DY(E),DU(E)
            WRITE#1,KX(E),KY(E),KU(E),S(E),P(E)
            WRITE#1,DX(E),DY(E),DU(E)
            FOR I=1 TO 8
                INPUT#6,O,B$,Q,S$,C,C$,CF
                WRITE#1,O,B$,Q,S$,C,C$,CF
            NEXT
        NEXT E
        CLOSE #1,#6
        KILL "TEMPDATA"

        CLS
        IF ITER>O THEN GOTO ITERATION

        WRITE#5,TIME,ITERATE
        FOR NODE=1 TO MAX
        WRITE#5,NODE,X(NODE),Y(NODE),U(NODE),H(NODE),C(NODE)
    NEXT
    FOR E=1 TO ELEM
        WRITE#5,VX(E),VY(E),VU(E)
    NEXT
    CLOSE #5
    CLS
    TIME=TIME+H
    COUNTER=COUNTER+1
    NEWCOUNT=NEWCOUNT+1

    IF COUNTER<=STEPS THEN GOTO MAINSTART

REM   ***MAIN.FEM***END*****
```

```
Filename: INTERFN.FEM          Type: Turbo BASIC program file

Purpose : Defines the interpolation functions.


REM   ***INTERFN.FEM***BEGIN***

DEF FN OO(Z,E,W)=(1/8*(1-ZR*Z)*(1-ER*E)*(1-WR*W))*(1/8*(1-ZC*Z)*(1-EC*E)*(1-WC*W))*FN J(Z,E,W)

DEF FN DZDX(Z,E,W)= (FN DYDE(Z,E,W)*FN DUDW(Z,E,W)-FN DUDE(Z,E,W)*FN DYDW(Z,E,W))
DEF FN DZDY(Z,E,W)=-(FN DXDE(Z,E,W)*FN DUDW(Z,E,W)-FN DUDE(Z,E,W)*FN DXDW(Z,E,W))
DEF FN DZDU(Z,E,W)= (FN DXDE(Z,E,W)*FN DYDW(Z,E,W)-FN DYDE(Z,E,W)*FN DXDW(Z,E,W))
DEF FN DEDX(Z,E,W)=-(FN DYDZ(Z,E,W)*FN DUDW(Z,E,W)-FN DUDZ(Z,E,W)*FN DYDW(Z,E,W))
DEF FN DEDY(Z,E,W)= (FN DXDZ(Z,E,W)*FN DUDW(Z,E,W)-FN DUDZ(Z,E,W)*FN DXDW(Z,E,W))
DEF FN DEDU(Z,E,W)=-(FN DXDZ(Z,E,W)*FN DYDW(Z,E,W)-FN DYDZ(Z,E,W)*FN DXDW(Z,E,W))
DEF FN DWDX(Z,E,W)= (FN DYDZ(Z,E,W)*FN DUDE(Z,E,W)-FN DUDZ(Z,E,W)*FN DYDE(Z,E,W))
DEF FN DWDY(Z,E,W)=-(FN DXDZ(Z,E,W)*FN DUDE(Z,E,W)-FN DUDZ(Z,E,W)*FN DXDE(Z,E,W))
DEF FN DWDU(Z,E,W)= (FN DXDZ(Z,E,W)*FN DYDE(Z,E,W)-FN DYDZ(Z,E,W)*FN DXDE(Z,E,W))

DEF FN D01DX(Z,E,W)=(FN D01DZ(Z,E,W)*FN DZDX(Z,E,W)+FN D01DE(Z,E,W)*FN DEDX(Z,E,W)+_
    FN D01DW(Z,E,W)*FN DWDX(Z,E,W))/FN J(Z,E,W)
DEF FN D01DY(Z,E,W)=(FN D01DZ(Z,E,W)*FN DZDY(Z,E,W)+FN D01DE(Z,E,W)*FN DEDY(Z,E,W)+_
    FN D01DW(Z,E,W)*FN DWDY(Z,E,W))/FN J(Z,E,W)
DEF FN D01DU(Z,E,W)=(FN D01DZ(Z,E,W)*FN DZDU(Z,E,W)+FN D01DE(Z,E,W)*FN DEDU(Z,E,W)+_
    FN D01DW(Z,E,W)*FN DWDU(Z,E,W))/FN J(Z,E,W)
DEF FN D02DX(Z,E,W)=(FN D02DZ(Z,E,W)*FN DZDX(Z,E,W)+FN D02DE(Z,E,W)*FN DEDX(Z,E,W)+_
    FN D02DW(Z,E,W)*FN DWDX(Z,E,W))/FN J(Z,E,W)
DEF FN D02DY(Z,E,W)=(FN D02DZ(Z,E,W)*FN DZDY(Z,E,W)+FN D02DE(Z,E,W)*FN DEDY(Z,E,W)+_
    FN D02DW(Z,E,W)*FN DWDY(Z,E,W))/FN J(Z,E,W)
DEF FN D02DU(Z,E,W)=(FN D02DZ(Z,E,W)*FN DZDU(Z,E,W)+FN D02DE(Z,E,W)*FN DEDU(Z,E,W)+_
    FN D02DW(Z,E,W)*FN DWDU(Z,E,W))/FN J(Z,E,W)
DEF FN D03DX(Z,E,W)=(FN D03DZ(Z,E,W)*FN DZDX(Z,E,W)+FN D03DE(Z,E,W)*FN DEDX(Z,E,W)+_
    FN D03DW(Z,E,W)*FN DWDX(Z,E,W))/FN J(Z,E,W)
DEF FN D03DY(Z,E,W)=(FN D03DZ(Z,E,W)*FN DZDY(Z,E,W)+FN D03DE(Z,E,W)*FN DEDY(Z,E,W)+_
    FN D03DW(Z,E,W)*FN DWDY(Z,E,W))/FN J(Z,E,W)
DEF FN D03DU(Z,E,W)=(FN D03DZ(Z,E,W)*FN DZDU(Z,E,W)+FN D03DE(Z,E,W)*FN DEDU(Z,E,W)+_
    FN D03DW(Z,E,W)*FN DWDU(Z,E,W))/FN J(Z,E,W)
DEF FN D04DX(Z,E,W)=(FN D04DZ(Z,E,W)*FN DZDX(Z,E,W)+FN D04DE(Z,E,W)*FN DEDX(Z,E,W)+_
    FN D04DW(Z,E,W)*FN DWDX(Z,E,W))/FN J(Z,E,W)
DEF FN D04DY(Z,E,W)=(FN D04DZ(Z,E,W)*FN DZDY(Z,E,W)+FN D04DE(Z,E,W)*FN DEDY(Z,E,W)+_
    FN D04DW(Z,E,W)*FN DWDY(Z,E,W))/FN J(Z,E,W)
DEF FN D04DU(Z,E,W)=(FN D04DZ(Z,E,W)*FN DZDU(Z,E,W)+FN D04DE(Z,E,W)*FN DEDU(Z,E,W)+_
    FN D04DW(Z,E,W)*FN DWDU(Z,E,W))/FN J(Z,E,W)
DEF FN D05DX(Z,E,W)=(FN D05DZ(Z,E,W)*FN DZDX(Z,E,W)+FN D05DE(Z,E,W)*FN DEDX(Z,E,W)+_
    FN D05DW(Z,E,W)*FN DWDX(Z,E,W))/FN J(Z,E,W)
DEF FN D05DY(Z,E,W)=(FN D05DZ(Z,E,W)*FN DZDY(Z,E,W)+FN D05DE(Z,E,W)*FN DEDY(Z,E,W)+_
    FN D05DW(Z,E,W)*FN DWDY(Z,E,W))/FN J(Z,E,W)
DEF FN D05DU(Z,E,W)=(FN D05DZ(Z,E,W)*FN DZDU(Z,E,W)+FN D05DE(Z,E,W)*FN DEDU(Z,E,W)+_
    FN D05DW(Z,E,W)*FN DWDU(Z,E,W))/FN J(Z,E,W)
DEF FN D06DX(Z,E,W)=(FN D06DZ(Z,E,W)*FN DZDX(Z,E,W)+FN D06DE(Z,E,W)*FN DEDX(Z,E,W)+_
    FN D06DW(Z,E,W)*FN DWDX(Z,E,W))/FN J(Z,E,W)
DEF FN D06DY(Z,E,W)=(FN D06DZ(Z,E,W)*FN DZDY(Z,E,W)+FN D06DE(Z,E,W)*FN DEDY(Z,E,W)+_
    FN D06DW(Z,E,W)*FN DWDY(Z,E,W))/FN J(Z,E,W)
DEF FN D06DU(Z,E,W)=(FN D06DZ(Z,E,W)*FN DZDU(Z,E,W)+FN D06DE(Z,E,W)*FN DEDU(Z,E,W)+_
```

```
    FN DO6DW(Z,E,W)*FN DWDU(Z,E,W))/FN J(Z,E,W)
DEF FN DO7DX(Z,E,W)=(FN DO7DZ(Z,E,W)*FN DZDX(Z,E,W)+FN DO7DE(Z,E,W)*FN DEDX(Z,E,W)+_
    FN DO7DW(Z,E,W)*FN DWDX(Z,E,W))/FN J(Z,E,W)
DEF FN DO7DY(Z,E,W)=(FN DO7DZ(Z,E,W)*FN DZDY(Z,E,W)+FN DO7DE(Z,E,W)*FN DEDY(Z,E,W)+_
    FN DO7DW(Z,E,W)*FN DWDY(Z,E,W))/FN J(Z,E,W)
DEF FN DO7DU(Z,E,W)=(FN DO7DZ(Z,E,W)*FN DZDU(Z,E,W)+FN DO7DE(Z,E,W)*FN DEDU(Z,E,W)+_
    FN DO7DW(Z,E,W)*FN DWDU(Z,E,W))/FN J(Z,E,W)
DEF FN DO8DX(Z,E,W)=(FN DO8DZ(Z,E,W)*FN DZDX(Z,E,W)+FN DO8DE(Z,E,W)*FN DEDX(Z,E,W)+_
    FN DO8DW(Z,E,W)*FN DWDX(Z,E,W))/FN J(Z,E,W)
DEF FN DO8DY(Z,E,W)=(FN DO8DZ(Z,E,W)*FN DZDY(Z,E,W)+FN DO8DE(Z,E,W)*FN DEDY(Z,E,W)+_
    FN DO8DW(Z,E,W)*FN DWDY(Z,E,W))/FN J(Z,E,W)
DEF FN DO8DU(Z,E,W)=(FN DO8DZ(Z,E,W)*FN DZDU(Z,E,W)+FN DO8DE(Z,E,W)*FN DEDU(Z,E,W)+_
    FN DO8DW(Z,E,W)*FN DWDU(Z,E,W))/FN J(Z,E,W)

DEF FN OXOX(Z,E,W)=(SZR/8*(1-EZR*E)*(1-WZR*W)*FN DZDX(Z,E,W)+SER/8*(1-ZER*Z)*(1-WER*W)*_
                    FN DEDX(Z,E,W)+SWR/8*(1-ZWR*Z)*(1-EWR*E)*FN DWDX(Z,E,W))*_
                    (SZC/8*(1-EZC*E)*(1-WZC*W)*FN DZDX(Z,E,W)+SEC/8*(1-ZEC*Z)*(1-WEC*W)*_
                    FN DEDX(Z,E,W)+SWC/8*(1-ZWC*Z)*(1-EWC*E)*FN DWDX(Z,E,W))/FN J(Z,E,W)

DEF FN OYOY(Z,E,W)=(SZR/8*(1-EZR*E)*(1-WZR*W)*FN DZDY(Z,E,W)+SER/8*(1-ZER*Z)*(1-WER*W)*_
                    FN DEDY(Z,E,W)+SWR/8*(1-ZWR*Z)*(1-EWR*E)*FN DWDY(Z,E,W))*_
                    (SZC/8*(1-EZC*E)*(1-WZC*W)*FN DZDY(Z,E,W)+SEC/8*(1-ZEC*Z)*(1-WEC*W)*_
                    FN DEDY(Z,E,W)+SWC/8*(1-ZWC*Z)*(1-EWC*E)*FN DWDY(Z,E,W))/FN J(Z,E,W)

DEF FN OUOU(Z,E,W)=(SZR/8*(1-EZR*E)*(1-WZR*W)*FN DZDU(Z,E,W)+SER/8*(1-ZER*Z)*(1-WER*W)*_
                    FN DEDU(Z,E,W)+SWR/8*(1-ZWR*Z)*(1-EWR*E)*FN DWDU(Z,E,W))*_
                    (SZC/8*(1-EZC*E)*(1-WZC*W)*FN DZDU(Z,E,W)+SEC/8*(1-ZEC*Z)*(1-WEC*W)*_
                    FN DEDU(Z,E,W)+SWC/8*(1-ZWC*Z)*(1-EWC*E)*FN DWDU(Z,E,W))/FN J(Z,E,W)



DEF FN OOX(Z,E,W)=(1/8*(1-ZR*Z)*(1-ER*E)*(1-WR*W))*(SZC/8*(1-EZC*E)*(1-WZC*W)*FN DZDX(Z,E,W)_
    +SEC/8*(1-ZEC*Z)*(1-WEC*W)*FN DEDX(Z,E,W)+SWC/8*(1-ZWC*Z)*(1-EWC*E)*FN DWDX(Z,E,W))
DEF FN OOY(Z,E,W)=(1/8*(1-ZR*Z)*(1-ER*E)*(1-WR*W))*(SZC/8*(1-EZC*E)*(1-WZC*W)*FN DZDY(Z,E,W)_
    +SEC/8*(1-ZEC*Z)*(1-WEC*W)*FN DEDY(Z,E,W)+SWC/8*(1-ZWC*Z)*(1-EWC*E)*FN DWDY(Z,E,W))
DEF FN OOU(Z,E,W)=(1/8*(1-ZR*Z)*(1-ER*E)*(1-WR*W))*(SZC/8*(1-EZC*E)*(1-WZC*W)*FN DZDU(Z,E,W)_
    +SEC/8*(1-ZEC*Z)*(1-WEC*W)*FN DEDU(Z,E,W)+SWC/8*(1-ZWC*Z)*(1-EWC*E)*FN DWDU(Z,E,W))

REM THE BOUNDARY FLUX INTEGRAL FN

DEF FN DEDW(Z,E,W)=(FN DEDX(Z,E,W)*FN DXDW(Z,E,W)+FN DEDY(Z,E,W)*FN DYDW(Z,E,W)+FN DEDU(Z,E,W)*_
    FN DUDW(Z,E,W))/FN J(Z,E,W)
DEF FN DWDE(Z,E,W)=(FN DWDX(Z,E,W)*FN DXDE(Z,E,W)+FN DWDY(Z,E,W)*FN DYDE(Z,E,W)+FN DWDU(Z,E,W)*_
    FN DUDE(Z,E,W))/FN J(Z,E,W)
DEF FN DZDE(Z,E,W)=(FN DZDX(Z,E,W)*FN DXDE(Z,E,W)+FN DZDY(Z,E,W)*FN DYDE(Z,E,W)+FN DZDU(Z,E,W)*_
    FN DUDE(Z,E,W))/FN J(Z,E,W)
DEF FN DEDZ(Z,E,W)=(FN DEDX(Z,E,W)*FN DXDZ(Z,E,W)+FN DEDY(Z,E,W)*FN DYDZ(Z,E,W)+FN DEDU(Z,E,W)*_
    FN DUDZ(Z,E,W))/FN J(Z,E,W)
DEF FN DWDZ(Z,E,W)=(FN DWDX(Z,E,W)*FN DXDZ(Z,E,W)+FN DWDY(Z,E,W)*FN DYDZ(Z,E,W)+FN DWDU(Z,E,W)*_
    FN DUDZ(Z,E,W))/FN J(Z,E,W)
DEF FN DZDW(Z,E,W)=(FN DZDX(Z,E,W)*FN DXDW(Z,E,W)+FN DZDY(Z,E,W)*FN DYDW(Z,E,W)+FN DZDU(Z,E,W)*_
    FN DUDW(Z,E,W))/FN J(Z,E,W)

DEF FN Z(Z,E,W)=(FN DXDE(Z,E,W)^2+FN DYDE(Z,E,W)^2+FN DUDE(Z,E,W)^2)5*(FN DXDW(Z,E,W)^2+_
```

```
    FN DYDW(Z,E,W)^2+FN DUDW(Z,E,W)^2)5
DEF FN E(Z,E,W)=(FN DXDZ(Z,E,W)^2+FN DYDZ(Z,E,W)^2+FN DUDZ(Z,E,W)^2)5*(FN DXDW(Z,E,W)^2+_
    FN DYDW(Z,E,W)^2+FN DUDW(Z,E,W)^2)5
DEF FN W(Z,E,W)=(FN DXDE(Z,E,W)^2+FN DYDE(Z,E,W)^2+FN DUDE(Z,E,W)^2)5*(FN DXDZ(Z,E,W)^2+_
    FN DYDZ(Z,E,W)^2+FN DUDZ(Z,E,W)^2)5

DEF FN OZ(Z,E,W)=(1/8*(1-ZR*Z)*(1-ER*E)*(1-WR*W))*FN Z(Z,E,W)
DEF FN OE(Z,E,W)=(1/8*(1-ZR*Z)*(1-ER*E)*(1-WR*W))*FN E(Z,E,W)
DEF FN OW(Z,E,W)=(1/8*(1-ZR*Z)*(1-ER*E)*(1-WR*W))*FN W(Z,E,W)

REM THE APPLIED FLUX INTEGRAL FN

DEF FN O(Z,E,W)=(1/8*(1-ZR*Z)*(1-ER*E)*(1-WR*W))*FN J(Z,E,W)

REM   ***INTERFN.FEM***END*****
```

Filename: JACFN.FEM Type: Turbo BASIC program file

Purpose : Defines the functions leading up to, and including, the Jacobian.

```
REM   ***JACFN.FEM***BEGIN***

DEF FN D01DZ(Z,E,W)=-1/8*(1-E)*(1-W):DEF FN D01DE(Z,E,W)=-1/8*(1-Z)*(1-W):_
DEF FN D01DW(Z,E,W)=-1/8*(1-Z)*(1-E)
DEF FN D02DZ(Z,E,W)= 1/8*(1-E)*(1-W):DEF FN D02DE(Z,E,W)=-1/8*(1+Z)*(1-W):_
DEF FN D02DW(Z,E,W)=-1/8*(1+Z)*(1-E)
DEF FN D03DZ(Z,E,W)= 1/8*(1+E)*(1-W):DEF FN D03DE(Z,E,W)= 1/8*(1+Z)*(1-W):_
DEF FN D03DW(Z,E,W)=-1/8*(1+Z)*(1+E)
DEF FN D04DZ(Z,E,W)=-1/8*(1+E)*(1-W):DEF FN D04DE(Z,E,W)= 1/8*(1-Z)*(1-W):_
DEF FN D04DW(Z,E,W)=-1/8*(1-Z)*(1+E)
DEF FN D05DZ(Z,E,W)=-1/8*(1-E)*(1+W):DEF FN D05DE(Z,E,W)=-1/8*(1-Z)*(1+W):_
DEF FN D05DW(Z,E,W)= 1/8*(1-Z)*(1-E)
DEF FN D06DZ(Z,E,W)= 1/8*(1-E)*(1+W):DEF FN D06DE(Z,E,W)=-1/8*(1+Z)*(1+W):_
DEF FN D06DW(Z,E,W)= 1/8*(1+Z)*(1-E)
DEF FN D07DZ(Z,E,W)= 1/8*(1+E)*(1+W):DEF FN D07DE(Z,E,W)= 1/8*(1+Z)*(1+W):_
DEF FN D07DW(Z,E,W)= 1/8*(1+Z)*(1+E)
DEF FN D08DZ(Z,E,W)=-1/8*(1+E)*(1+W):DEF FN D08DE(Z,E,W)= 1/8*(1-Z)*(1+W):_
DEF FN D08DW(Z,E,W)= 1/8*(1-Z)*(1+E)


DEF FN DXDZ(Z,E,W)=X(N(1))*FN D01DZ(Z,E,W)+X(N(2))*FN D02DZ(Z,E,W)+X(N(3))*_
    FN D03DZ(Z,E,W)+X(N(4))*FN D04DZ(Z,E,W)+X(N(5))*FN D05DZ(Z,E,W)+X(N(6))*_
    FN D06DZ(Z,E,W)+X(N(7))*FN D07DZ(Z,E,W)+X(N(8))*FN D08DZ(Z,E,W)
DEF FN DXDE(Z,E,W)=X(N(1))*FN D01DE(Z,E,W)+X(N(2))*FN D02DE(Z,E,W)+X(N(3))*_
    FN D03DE(Z,E,W)+X(N(4))*FN D04DE(Z,E,W)+X(N(5))*FN D05DE(Z,E,W)+X(N(6))*_
    FN D06DE(Z,E,W)+X(N(7))*FN D07DE(Z,E,W)+X(N(8))*FN D08DE(Z,E,W)
DEF FN DXDW(Z,E,W)=X(N(1))*FN D01DW(Z,E,W)+X(N(2))*FN D02DW(Z,E,W)+X(N(3))*_
    FN D03DW(Z,E,W)+X(N(4))*FN D04DW(Z,E,W)+X(N(5))*FN D05DW(Z,E,W)+X(N(6))*_
    FN D06DW(Z,E,W)+X(N(7))*FN D07DW(Z,E,W)+X(N(8))*FN D08DW(Z,E,W)

DEF FN DYDZ(Z,E,W)=Y(N(1))*FN D01DZ(Z,E,W)+Y(N(2))*FN D02DZ(Z,E,W)+Y(N(3))*_
    FN D03DZ(Z,E,W)+Y(N(4))*FN D04DZ(Z,E,W)+Y(N(5))*FN D05DZ(Z,E,W)+Y(N(6))*_
    FN D06DZ(Z,E,W)+Y(N(7))*FN D07DZ(Z,E,W)+Y(N(8))*FN D08DZ(Z,E,W)
DEF FN DYDE(Z,E,W)=Y(N(1))*FN D01DE(Z,E,W)+Y(N(2))*FN D02DE(Z,E,W)+Y(N(3))*_
    FN D03DE(Z,E,W)+Y(N(4))*FN D04DE(Z,E,W)+Y(N(5))*FN D05DE(Z,E,W)+Y(N(6))*_
    FN D06DE(Z,E,W)+Y(N(7))*FN D07DE(Z,E,W)+Y(N(8))*FN D08DE(Z,E,W)
DEF FN DYDW(Z,E,W)=Y(N(1))*FN D01DW(Z,E,W)+Y(N(2))*FN D02DW(Z,E,W)+Y(N(3))*_
    FN D03DW(Z,E,W)+Y(N(4))*FN D04DW(Z,E,W)+Y(N(5))*FN D05DW(Z,E,W)+Y(N(6))*_
    FN D06DW(Z,E,W)+Y(N(7))*FN D07DW(Z,E,W)+Y(N(8))*FN D08DW(Z,E,W)

DEF FN DUDZ(Z,E,W)=U(N(1))*FN D01DZ(Z,E,W)+U(N(2))*FN D02DZ(Z,E,W)+U(N(3))*_
    FN D03DZ(Z,E,W)+U(N(4))*FN D04DZ(Z,E,W)+U(N(5))*FN D05DZ(Z,E,W)+U(N(6))*_
    FN D06DZ(Z,E,W)+U(N(7))*FN D07DZ(Z,E,W)+U(N(8))*FN D08DZ(Z,E,W)
DEF FN DUDE(Z,E,W)=U(N(1))*FN D01DE(Z,E,W)+U(N(2))*FN D02DE(Z,E,W)+U(N(3))*_
    FN D03DE(Z,E,W)+U(N(4))*FN D04DE(Z,E,W)+U(N(5))*FN D05DE(Z,E,W)+U(N(6))*_
    FN D06DE(Z,E,W)+U(N(7))*FN D07DE(Z,E,W)+U(N(8))*FN D08DE(Z,E,W)
DEF FN DUDW(Z,E,W)=U(N(1))*FN D01DW(Z,E,W)+U(N(2))*FN D02DW(Z,E,W)+U(N(3))*_
    FN D03DW(Z,E,W)+U(N(4))*FN D04DW(Z,E,W)+U(N(5))*FN D05DW(Z,E,W)+U(N(6))*_
```

```
    FN D06DW(Z,E,W)+U(N(7))*FN D07DW(Z,E,W)+U(N(8))*FN D08DW(Z,E,W)

DEF FN J(Z,E,W)=FN DXDZ(Z,E,W)*(FN DYDE(Z,E,W)*FN DUDW(Z,E,W)-FN DUDE(Z,E,W)*FN DYDW(Z,E,W))-_
    FN DXDE(Z,E,W)*(FN DYDZ(Z,E,W)*FN DUDW(Z,E,W)-FN DUDZ(Z,E,W)*FN DYDW(Z,E,W))+_
    FN DXDW(Z,E,W)*(FN DYDZ(Z,E,W)*FN DUDE(Z,E,W)-FN DUDZ(Z,E,W)*FN DYDE(Z,E,W))

REM    ***JACFN.FEM***END*****
```

```
Filename: SUB.FEM            Type: Turbo BASIC program file

Purpose : Guass quadrature subroutines


REM   ***SUB.FEM***BEGIN***

GOTO SUBFIN

GLOE:
IN=0
FOR II=-.774597 TO .774597 STEP .774597
    IF II=0 THEN WI=8/9 ELSE WI=5/9
    FOR KK=-.774597 TO .774597 STEP .774597
        IF KK=0 THEN WK=8/9 ELSE WK=5/9
        IN=IN+WI*WK*FN OE(II,CONS,KK)
    NEXT
NEXT
RETURN

GLOZ:
IN=0
FOR JJ=-.774597 TO .774597 STEP .774597
    IF JJ=0 THEN WJ=8/9 ELSE WJ=5/9
    FOR KK=-.774597 TO .774597 STEP .774597
        IF KK=0 THEN WK=8/9 ELSE WK=5/9
        IN=IN+WJ*WK*FN OZ(CONS,JJ,KK)
    NEXT
NEXT
RETURN

GLOW:
IN=0
FOR II=-.774597 TO .774597 STEP .774597
    IF II=0 THEN WI=8/9 ELSE WI=5/9
    FOR JJ=-.774597 TO .774597 STEP .774597
        IF JJ=0 THEN WJ=8/9 ELSE WJ=5/9
        IN=IN+WI*WJ*FN OW(II,JJ,CONS)
    NEXT
NEXT
RETURN
GLO:
IN=0
FOR II=-.774597 TO .774597 STEP .774597
    IF II=0 THEN WI=8/9 ELSE WI=5/9
    FOR JJ=-.774597 TO .774597 STEP .774597
        IF JJ=0 THEN WJ=8/9 ELSE WJ=5/9
        FOR KK=-.774597 TO .774597 STEP .774597
            IF KK=0 THEN WK=8/9 ELSE WK=5/9
            IN=IN+WI*WJ*WK*FN O(II,JJ,KK)
        NEXT
    NEXT
NEXT
RETURN
```

```
GLOO:
IN=0
FOR II=-.774597 TO .774597 STEP .774597
    IF II=0 THEN WI=8/9 ELSE WI=5/9
    FOR JJ=-.774597 TO .774597 STEP .774597
        IF JJ=0 THEN WJ=8/9 ELSE WJ=5/9
        FOR KK=-.774597 TO .774597 STEP .774597
            IF KK=0 THEN WK=8/9 ELSE WK=5/9
            IN=IN+WI*WJ*WK*FN OO(II,JJ,KK)
        NEXT
    NEXT
NEXT
RETURN

GLOXOX:
IN=0
FOR II=-.774597 TO .774597 STEP .774597
    IF II=0 THEN WI=8/9 ELSE WI=5/9
    FOR JJ=-.774597 TO .774597 STEP .774597
        IF JJ=0 THEN WJ=8/9 ELSE WJ=5/9
        FOR KK=-.774597 TO .774597 STEP .774597
            IF KK=0 THEN WK=8/9 ELSE WK=5/9
            IN=IN+WI*WJ*WK*FN OXOX(II,JJ,KK)
        NEXT
    NEXT
NEXT
RETURN

GLOYOY:
IN=0
FOR II=-.774597 TO .774597 STEP .774597
    IF II=0 THEN WI=8/9 ELSE WI=5/9
    FOR JJ=-.774597 TO .774597 STEP .774597
        IF JJ=0 THEN WJ=8/9 ELSE WJ=5/9
        FOR KK=-.774597 TO .774597 STEP .774597
            IF KK=0 THEN WK=8/9 ELSE WK=5/9
            IN=IN+WI*WJ*WK*FN OYOY(II,JJ,KK)
        NEXT
    NEXT
NEXT
RETURN
GLOUOU:
IN=0
FOR II=-.774597 TO .774597 STEP .774597
    IF II=0 THEN WI=8/9 ELSE WI=5/9
    FOR JJ=-.774597 TO .774597 STEP .774597
        IF JJ=0 THEN WJ=8/9 ELSE WJ=5/9
        FOR KK=-.774597 TO .774597 STEP .774597
            IF KK=0 THEN WK=8/9 ELSE WK=5/9
            IN=IN+WI*WJ*WK*FN OUOU(II,JJ,KK)
        NEXT
    NEXT
NEXT
RETURN
```

```
GLOOX:
IN=0
FOR II=-.774597 TO .774597 STEP .774597
    IF II=0 THEN WI=8/9 ELSE WI=5/9
    FOR JJ=-.774597 TO .774597 STEP .774597
        IF JJ=0 THEN WJ=8/9 ELSE WJ=5/9
        FOR KK=-.774597 TO .774597 STEP .774597
            IF KK=0 THEN WK=8/9 ELSE WK=5/9
            IN=IN+WI*WJ*WK*FN OOX(II,JJ,KK)
        NEXT KK
    NEXT JJ
NEXT II
RETURN

GLOOY:
IN=0
FOR II=-.774597 TO .774597 STEP .774597
    IF II=0 THEN WI=8/9 ELSE WI=5/9
    FOR JJ=-.774597 TO .774597 STEP .774597
        IF JJ=0 THEN WJ=8/9 ELSE WJ=5/9
        FOR KK=-.774597 TO .774597 STEP .774597
            IF KK=0 THEN WK=8/9 ELSE WK=5/9
            IN=IN+WI*WJ*WK*FN OOY(II,JJ,KK)
        NEXT KK
    NEXT JJ
NEXT II
RETURN

GLOOU:
IN=0
FOR II=-.774597 TO .774597 STEP .774597
    IF II=0 THEN WI=8/9 ELSE WI=5/9
    FOR JJ=-.774597 TO .774597 STEP .774597
        IF JJ=0 THEN WJ=8/9 ELSE WJ=5/9
        FOR KK=-.774597 TO .774597 STEP .774597
            IF KK=0 THEN WK=8/9 ELSE WK=5/9
            IN=IN+WI*WJ*WK*FN OOU(II,JJ,KK)
        NEXT KK
    NEXT JJ
NEXT II
RETURN

REM    ***SUB.FEM***END*****
```

```
Filename: MATGF.FEM          Type: Turbo BASIC program file

Purpose : Calculates the stiffness matrices of the FE
          groundwater flow formulation.


REM   ***MATGF.FEM***BEGIN***

FOR I=1 TO 8
    READ ZR,ER,WR
    FOR J=I TO 8
        READ ZC,EC,WC
        GOSUB GLOO
        OO(I,J)=IN
        OO(J,I)=IN
        LOCATE 22,1:PRINT SPACE$(80)
        LOCATE 22,1:PRINT "ELEMENT";E;"- CALCULATING MATRIX OO ";I;J
    NEXT
NEXT

FOR I=1 TO 8
    READ SZR,EZR,WZR,SER,ZER,WER,SWR,ZWR,EWR
    FOR J=I TO 8
        READ SZC,EZC,WZC,SEC,ZEC,WEC,SWC,ZWC,EWC
        GOSUB GLOXOX
        OXOX(I,J)=IN
        OXOX(J,I)=IN
        LOCATE 22,1:PRINT SPACE$(80)
        LOCATE 22,1:PRINT "ELEMENT";E;"- CALCULATING MATRIX OXOX ";I;J
        GOSUB GLOYOY
        OYOY(I,J)=IN
        OYOY(J,I)=IN
        LOCATE 22,1:PRINT SPACE$(80)
        LOCATE 22,1:PRINT "ELEMENT";E;"- CALCULATING MATRIX OYOY ";I;J
        GOSUB GLOUOU
        OUOU(I,J)=IN
        OUOU(J,I)=IN
        LOCATE 22,1:PRINT SPACE$(80)
        LOCATE 22,1:PRINT "ELEMENT";E;"- CALCULATING MATRIX OUOU ";I;J
   NEXT
NEXT

GFDATA:

DATA  1, 1, 1, 1, 1, 1,-1, 1, 1,-1,-1, 1, 1,-1, 1, 1, 1,-1,-1, 1,-1,-1,-1,-1, 1,-1,-1
DATA -1, 1, 1,         -1, 1, 1,-1,-1, 1, 1,-1, 1, 1, 1,-1,-1, 1,-1,-1,-1,-1, 1,-1,-1
DATA -1,-1, 1,                   -1,-1, 1, 1,-1, 1, 1, 1,-1,-1, 1,-1,-1,-1,-1, 1,-1,-1
DATA  1,-1, 1,                            1,-1, 1, 1, 1,-1,-1, 1,-1,-1,-1,-1, 1,-1,-1
DATA  1, 1,-1,                                    1, 1,-1,-1, 1,-1,-1,-1,-1, 1,-1,-1
DATA -1, 1,-1,                                           -1, 1,-1,-1,-1,-1, 1,-1,-1
DATA -1,-1,-1,                                                 -1,-1,-1, 1,-1,-1
DATA  1,-1,-1,                                                       1,-1,-1
```

```
DATA -1, 1, 1,-1, 1, 1,-1, 1, 1,-1, 1, 1,-1, 1, 1,-1, 1, 1, 1, 1, 1,-1,-1, 1,-1,-1, 1,_
      1,-1, 1, 1,-1, 1,-1,-1,-1,-1,-1, 1, 1, 1, 1,-1, 1,-1,-1, 1,-1,-1, 1,-1, 1, 1, 1,_
      1, 1,-1,-1,-1,-1, 1,-1, 1, 1,-1,-1, 1,-1,-1, 1,-1,-1,-1,-1,-1,-1, 1, 1,-1, 1, 1,-1
DATA  1, 1, 1,-1,-1, 1,-1,-1, 1,                                1, 1, 1,-1,-1, 1,-1,-1, 1,_
      1,-1, 1, 1,-1, 1,-1,-1,-1,-1,-1, 1, 1, 1, 1,-1, 1,-1,-1, 1,-1,-1, 1,-1, 1, 1, 1,_
      1, 1,-1,-1,-1,-1, 1,-1, 1, 1,-1,-1, 1,-1,-1, 1,-1,-1,-1,-1,-1,-1, 1, 1,-1, 1, 1,-1
DATA  1,-1, 1, 1,-1, 1,-1,-1,-1,                                                        _
      1,-1, 1, 1,-1, 1,-1,-1,-1,-1,-1, 1, 1, 1, 1,-1, 1,-1,-1, 1,-1,-1, 1,-1, 1, 1, 1,_
      1, 1,-1,-1,-1,-1, 1,-1, 1, 1,-1,-1, 1,-1,-1, 1,-1,-1,-1,-1,-1,-1, 1, 1,-1, 1, 1,-1
DATA -1,-1, 1, 1, 1, 1,-1, 1,-1,
                                -1,-1, 1, 1, 1, 1,-1, 1,-1,-1, 1,-1,-1, 1,-1, 1, 1, 1,_
      1, 1,-1,-1,-1,-1, 1,-1, 1, 1,-1,-1, 1,-1,-1, 1,-1,-1,-1,-1,-1,-1, 1, 1,-1, 1, 1,-1
DATA -1, 1,-1,-1, 1,-1, 1, 1, 1,                                                        _
                                                              -1, 1,-1,-1, 1,-1, 1, 1, 1,_
      1, 1,-1,-1,-1,-1, 1,-1, 1, 1,-1,-1, 1,-1,-1, 1,-1,-1,-1,-1,-1,-1, 1, 1,-1, 1, 1,-1
DATA  1, 1,-1,-1,-1,-1, 1,-1, 1,                                                        _
                                                                                        _
      1, 1,-1,-1,-1,-1, 1,-1, 1, 1,-1,-1, 1,-1,-1, 1,-1,-1,-1,-1,-1,-1, 1, 1,-1, 1, 1,-1
DATA  1,-1,-1, 1,-1,-1, 1,-1,-1,                                                        _
                                                                                        _
                                  1,-1,-1, 1,-1,-1, 1,-1,-1,-1,-1,-1, 1, 1,-1, 1, 1,-1
DATA -1,-1,-1, 1, 1,-1, 1, 1,-1,                                                        _
                                                                                        _
                                                              -1,-1,-1, 1, 1,-1, 1, 1,-1

REM   ***MATGF.FEM***END*****
```

```
Filename: VECGF.FEM          Type: Turbo BASIC program file

Purpose : Calculates the boundary integral vector of the  FE
          groundwater flow formulation.


REM   ***VECGF.FEM***BEGIN***

REM
REM   OB ->  ∫[O] dΓ
REM

REM
REM    O ->  ∫[O] dΩ
REM

FOR I=1 TO 8
    OB(I)=0
NEXT I

LOCATE 22,1:PRINT SPACE$(80)
LOCATE 22,1:PRINT "ELEMENT";E;"- CALCULATING VECTOR OB"
FOR I=1 TO 8
    IF B$(N(I))="S" THEN
       B$(N(I))="Y"
       CON$(N(I))="S"
    END IF
NEXT

IF B$(N(3))="Y" AND B$(N(4))="Y" AND B$(N(7))="Y" AND B$(N(8))="Y" THEN
   CONS=1
   FOR I=1 TO 8
       READ ZR,ER,WR
       GOSUB GLOE
       OB(I)=IN
   NEXT
ELSE
   IF B$(N(1))="Y" AND B$(N(2))="Y" AND B$(N(5))="Y" AND B$(N(6))="Y" THEN
      CONS=-1
      FOR I=1 TO 8
          READ ZR,ER,WR
          GOSUB GLOE
          OB(I)=IN
      NEXT
   ELSE
      IF B$(N(2))="Y" AND B$(N(6))="Y" AND B$(N(7))="Y" AND B$(N(3))="Y" THEN
         CONS=1
         FOR I=1 TO 8
             READ ZR,ER,WR
             GOSUB GLOZ
             OB(I)=IN
         NEXT
      ELSE
         IF B$(N(1))="Y" AND B$(N(5))="Y" AND B$(N(8))="Y" AND B$(N(4))="Y" THEN
```

```
            CONS=-1
            FOR I=1 TO 8
                READ ZR,ER,WR
                GOSUB GLOZ
                OB(I)=IN
            NEXT
         ELSE
            IF B$(N(5))="Y" AND B$(N(6))="Y" AND B$(N(7))="Y" AND B$(N(8))="Y" THEN
               CONS=1
               FOR I=1 TO 8
                   READ ZR,ER,WR
                   GOSUB GLOW
                   OB(I)=IN
               NEXT
            ELSE
               IF B$(N(1))="Y" AND B$(N(2))="Y" AND B$(N(3))="Y" AND B$(N(4))="Y" THEN
                  CONS=-1
                  FOR I=1 TO 8
                      READ ZR,ER,WR
                      GOSUB GLOW
                      OB(I)=IN
                  NEXT
               ELSE
                  FOR I=1 TO 8
                      READ ZR,ER,WR
                  NEXT
               END IF
            END IF
         END IF
      END IF
   END IF
END IF

FOR I=1 TO 8
    IF CON$(N(I))="S" THEN
       B$(N(I))="S"
       CON$(N(I))=""
    END IF
NEXT
FOR I=1 TO 8
    LOCATE 22,1:PRINT "ELEMENT";E;"- CALCULATING VECTOR O";:PRINT
    READ ZR,ER,WR
    GOSUB GLO
    O(I)=IN
NEXT

DATA  1, 1, 1
DATA -1, 1, 1
DATA -1,-1, 1
DATA  1,-1, 1
DATA  1, 1,-1
DATA -1, 1,-1
DATA -1,-1,-1
DATA  1,-1,-1
```

```
DATA  1, 1, 1
DATA -1, 1, 1
DATA -1,-1, 1
DATA  1,-1, 1
DATA  1, 1,-1
DATA -1, 1,-1
DATA -1,-1,-1
DATA  1,-1,-1

REM   ***VECGF.FEM***END*****
```

```
Filename: MATCD.FEM          Type: Turbo BASIC program file

Purpose : Calculates the stiffness matrices of the FE
          diffusion-convection formulation.

REM   ***MATCD.FEM***BEGIN***

FOR I=1 TO 8
    READ ZR,ER,WR
    FOR J=1 TO 8
        READ SZC,EZC,WZC,SEC,ZEC,WEC,SWC,ZWC,EWC
        LOCATE 22,1:PRINT SPACE$(80)
        LOCATE 22,1:PRINT "ELEMENT";E;"- CALCULATING MATRIX OOX ";I;J
        GOSUB GLOOX
        OOX(I,J)=IN
        LOCATE 22,1:PRINT SPACE$(80)
        LOCATE 22,1:PRINT "ELEMENT";E;"- CALCULATING MATRIX OOY ";I;J
        GOSUB GLOOY
        OOY(I,J)=IN
        LOCATE 22,1:PRINT SPACE$(80)
        LOCATE 22,1:PRINT "ELEMENT";E;"- CALCULATING MATRIX OOU ";I;J
        GOSUB GLOOU
        OOU(I,J)=IN
    NEXT J
NEXT I

CDDATA:

DATA  1, 1, 1, -1, 1, 1,-1, 1, 1,-1, 1, 1,  1, 1, 1,-1,-1, 1,-1,-1, 1,_
      1,-1, 1, 1,-1, 1,-1,-1,-1, -1,-1, 1, 1, 1, 1,-1, 1,-1, -1, 1,-1,-1, 1,-1, 1, 1, 1,_
      1, 1,-1,-1,-1,-1, 1,-1, 1,  1,-1,-1, 1,-1,-1, 1,-1,-1, -1,-1,-1, 1, 1,-1, 1, 1,-1
DATA -1, 1, 1, -1, 1, 1,-1, 1, 1,-1, 1, 1,  1, 1, 1,-1,-1, 1,-1,-1, 1,_
      1,-1, 1, 1,-1, 1,-1,-1,-1, -1,-1, 1, 1, 1, 1,-1, 1,-1, -1, 1,-1,-1, 1,-1, 1, 1, 1,_
      1, 1,-1,-1,-1,-1, 1,-1, 1,  1,-1,-1, 1,-1,-1, 1,-1,-1, -1,-1,-1, 1, 1,-1, 1, 1,-1
DATA -1,-1, 1, -1, 1, 1,-1, 1, 1,-1, 1, 1,  1, 1, 1,-1,-1, 1,-1,-1, 1,_
      1,-1, 1, 1,-1, 1,-1,-1,-1, -1,-1, 1, 1, 1, 1,-1, 1,-1, -1, 1,-1,-1, 1,-1, 1, 1, 1,_
      1, 1,-1,-1,-1,-1, 1,-1, 1,  1,-1,-1, 1,-1,-1, 1,-1,-1, -1,-1,-1, 1, 1,-1, 1, 1,-1
DATA  1,-1, 1, -1, 1, 1,-1, 1, 1,-1, 1, 1,  1, 1, 1,-1,-1, 1,-1,-1, 1,_
      1,-1, 1, 1,-1, 1,-1,-1,-1, -1,-1, 1, 1, 1, 1,-1, 1,-1, -1, 1,-1,-1, 1,-1, 1, 1, 1,_
      1, 1,-1,-1,-1,-1, 1,-1, 1,  1,-1,-1, 1,-1,-1, 1,-1,-1, -1,-1,-1, 1, 1,-1, 1, 1,-1
DATA  1, 1,-1, -1, 1, 1,-1, 1, 1,-1, 1, 1,  1, 1, 1,-1,-1, 1,-1,-1, 1,_
      1,-1, 1, 1,-1, 1,-1,-1,-1, -1,-1, 1, 1, 1, 1,-1, 1,-1, -1, 1,-1,-1, 1,-1, 1, 1, 1,_
      1, 1,-1,-1,-1,-1, 1,-1, 1,  1,-1,-1, 1,-1,-1, 1,-1,-1, -1,-1,-1, 1, 1,-1, 1, 1,-1
DATA -1, 1,-1, -1, 1, 1,-1, 1, 1,-1, 1, 1,  1, 1, 1,-1,-1, 1,-1,-1, 1,_
      1,-1, 1, 1,-1, 1,-1,-1,-1, -1,-1, 1, 1, 1, 1,-1, 1,-1, -1, 1,-1,-1, 1,-1, 1, 1, 1,_
      1, 1,-1,-1,-1,-1, 1,-1, 1,  1,-1,-1, 1,-1,-1, 1,-1,-1, -1,-1,-1, 1, 1,-1, 1, 1,-1
DATA -1,-1,-1, -1, 1, 1,-1, 1, 1,-1, 1, 1,  1, 1, 1,-1,-1, 1,-1,-1, 1,_
      1,-1, 1, 1,-1, 1,-1,-1,-1, -1,-1, 1, 1, 1, 1,-1, 1,-1, -1, 1,-1,-1, 1,-1, 1, 1, 1,_
      1, 1,-1,-1,-1,-1, 1,-1, 1,  1,-1,-1, 1,-1,-1, 1,-1,-1, -1,-1,-1, 1, 1,-1, 1, 1,-1
DATA  1,-1,-1, -1, 1, 1,-1, 1, 1,-1, 1, 1,  1, 1, 1,-1,-1, 1,-1,-1, 1,_
      1,-1, 1, 1,-1, 1,-1,-1,-1, -1,-1, 1, 1, 1, 1,-1, 1,-1, -1, 1,-1,-1, 1,-1, 1, 1, 1,_
      1, 1,-1,-1,-1,-1, 1,-1, 1,  1,-1,-1, 1,-1,-1, 1,-1,-1, -1,-1,-1, 1, 1,-1, 1, 1,-1

REM   ***MATCD.FEM***END*****
```

```
Filename: VECCD.FEM          Type: Turbo BASIC program file

Purpose : Calculates the boundary integral  vector of the FE
          diffusion-convection formulation.


REM   ***VECCD.FEM***BEGIN***

REM
REM   OC ->  ∫[O]dΓ
REM

FOR I=1 TO 8
    OC(I)=0
NEXT I

LOCATE 22,1:PRINT SPACE$(80)
LOCATE 22,1:PRINT "ELEMENT";E;"- CALCULATING VECTOR OC"

FOR I=1 TO 8
    IF C$(N(I))="F" THEN
       C$(N(I))="C"
       CONC$(N(I))="F"
    END IF
NEXT

IF C$(N(3))="C" AND C$(N(4))="C" AND C$(N(7))="C" AND C$(N(8))="C" THEN
   CONS=1
   FOR I=1 TO 8
       READ ZR,ER,WR
       GOSUB GLOE
       OC(I)=IN
   NEXT
ELSE
   IF C$(N(1))="C" AND C$(N(2))="C" AND C$(N(5))="C" AND C$(N(6))="C" THEN
      CONS=-1
      FOR I=1 TO 8
          READ ZR,ER,WR
          GOSUB GLOE
          OC(I)=IN
      NEXT
   ELSE
      IF C$(N(2))="C" AND C$(N(6))="C" AND C$(N(7))="C" AND C$(N(3))="C" THEN
         CONS=1
         FOR I=1 TO 8
             READ ZR,ER,WR
             GOSUB GLOZ
             OC(I)=IN
         NEXT
      ELSE
         IF C$(N(1))="C" AND C$(N(5))="C" AND C$(N(8))="C" AND C$(N(4))="C" THEN
            CONS=-1
            FOR I=1 TO 8
                READ ZR,ER,WR
```

```
                GOSUB GLOZ
                OC(I)=IN
            NEXT
        ELSE
            IF C$(N(5))="C" AND C$(N(6))="C" AND C$(N(7))="C" AND C$(N(8))="C" THEN
               CONS=1
               FOR I=1 TO 8
                   READ ZR,ER,WR
                   GOSUB GLOW
                   OC(I)=IN
               NEXT
            ELSE
               IF C$(N(1))="C" AND C$(N(2))="C" AND C$(N(3))="C" AND C$(N(4))="C" THEN
                  CONS=-1
                  FOR I=1 TO 8
                      READ ZR,ER,WR
                      GOSUB GLOW
                      OC(I)=IN
                  NEXT
               ELSE
                  FOR I=1 TO 8
                      READ ZR,ER,WR
                  NEXT
               END IF
            END IF
        END IF
    END IF
END IF

FOR I=1 TO 8
    IF CONC$(N(I))="F" THEN
       C$(N(I))="F"
       CONC$(N(I))=""
    END IF
NEXT

DATA  1, 1, 1
DATA -1, 1, 1
DATA -1,-1, 1
DATA  1,-1, 1
DATA  1, 1,-1
DATA -1, 1,-1
DATA -1,-1,-1
DATA  1,-1,-1

REM   ***VECCD.FEM***END*****
```

Filename: GLOBALGF.FEM Type: Turbo BASIC program file

Purpose : Globally assembles the FE groundwater flow formulation.

```
REM   ***GLOBALGF.FEM***BEGIN***    GLOBAL POSITIONING

REM KE(I,J)->ELEMENT (Kx[m/s][m/s]+Ky[m/s][m/s]+Ku[mu][mu])+S[m][m]/H
REM KG(I,J)->GLOBAL  (Kx[m/s][m/s]+Ky[m/s][m/s]+Ku[mu][mu])+S[m][m]/H
REM LE(I,J)->ELEMENT S[m][m]/H
REM LG(I,J)->GLOBAL  S[m][m]/H
REM DE(I)  ->ELEMENT Q[m]
REM DG(I)  ->GLOBAL  Q[m]
REM RE(I)  ->ELEMENT r[m]
REM RG(I)  ->GLOBAL  r[m]

REM N->TOTAL NO. OF NODES

LOCATE 22,1:PRINT SPACE$(80)
LOCATE 22,1:PRINT "ELEMENT";E;"- ASSEMBLING GLOBALLY INTO GROUNDWATER FLOW EQUATION"

FOR I=1 TO 8
    IF N(I)>N THEN N=N(I)
NEXT
IF COUNTER=0 THEN HH=600 ELSE HH=1
BBB=1000:AAA=0
FOR I=1 TO 8
    RE(I)=-OB(I)
    DE(I)=Q(N(I))*O(I):O(I)=0
    FOR J=1 TO 8
        KE(I,J)=(KX(E)*OXOX(I,J)+KY(E)*OYOY(I,J)+KU(E)*OUOU(I,J))+S(E)*OO(I,J)/H/HH
        LE(I,J)=S(E)*OO(I,J)/H/HH
    NEXT
    IF N(I)>AAA THEN AAA=N(I)
    IF N(I)<BBB THEN BBB=N(I)
NEXT

FOR K=1 TO 8
    FOR L=1 TO 8
        FOR I=BBB TO AAA
            IF I=N(K) AND L=1 THEN RG(I)=RG(I)+RE(K):DG(I)=DG(I)+DE(K)
            FOR J=BBB TO AAA
                IF I=N(K) AND J=N(L) THEN KG(I,J)=KG(I,J)+KE(K,L):LG(I,J)=LG(I,J)+LE(K,L)
            NEXT
        NEXT
    NEXT
NEXT

REM   ***GLOBALGF.FEM***END*****
```

```
Filename: GLOBALCD.FEM        Type: Turbo BASIC program file

Purpose : Globally assembles the FE diffusion-convection formulation.


REM   ***GLOBALCD.FEM***BEGIN***    GLOBAL POSITIONING

REM MG(I,J)->GLOBAL  .5{(Dx[mx][mx]+Dy[my][my]+Du[mu][mu])+
REM                     (Vx[mx][m]+Vy[my][m]+Vu[mu][m])+2[m][m]/H}
REM NG(I,J)->GLOBAL  .5{-(Dx[mx][mx]+Dy[my][my]+Du[mu][mu])
REM                     -(Vx[mx][m]+Vy[my][m]+Vu[mu][m])+2[m][m]/H}
REM ME(I,J)->ELEMENT .5{(Dx.....
REM NE(I,J)->ELEMENT .5{-Dx.....
REM GE(I)  ->ELEMENT f[m]
REM GG(I)  ->GLOBAL  f[m]

LOCATE 22,1:PRINT SPACE$(80)
LOCATE 22,1:PRINT "ELEMENT";E;"- ASSEMBLING GLOBALLY INTO DIFFUSION-CONVECTION EQUATION"

DXX1(E)=DX(E)-H/6*(VX(E))^2*P(E):DXX2(E)=DX(E)+H/6*(VX(E))^2*P(E)
DYY1(E)=DY(E)-H/6*(VY(E))^2*P(E):DYY2(E)=DY(E)+H/6*(VY(E))^2*P(E)
DUU1(E)=DU(E)-H/6*(VU(E))^2*P(E):DUU2(E)=DU(E)+H/6*(VU(E))^2*P(E)

FOR I=1 TO 8
    IF N(I)>N THEN N=N(I)
NEXT
BBB=1000:AAA=0

FOR I=1 TO 8
    GE(I)=-P(E)*OC(I)
    OC(I)=0
    FOR J=1 TO 8
        ME(I,J)=.5*( (DXX1(E)*OXOX(I,J)*P(E)+DYY1(E)*OYOY(I,J)*P(E)+DUU1(E)*OUOU(I,J)*P(E))+_
                     (VX(E)*P(E)*OOX(I,J)+VY(E)*P(E)*OOY(I,J)+VU(E)*P(E)*OOU(I,J))+2*OO(I,J)/H)
        NE(I,J)=.5*(-(DXX2(E)*OXOX(I,J)*P(E)+DYY2(E)*OYOY(I,J)*P(E)+DUU2(E)*OUOU(I,J)*P(E))-_
                     (VX(E)*P(E)*OOX(I,J)+VY(E)*P(E)*OOY(I,J)+VU(E)*P(E)*OOU(I,J))+2*OO(I,J)/H)
    NEXT
    IF N(I)>AAA THEN AAA=N(I)
    IF N(I)<BBB THEN BBB=N(I)
NEXT

FOR K=1 TO 8
    FOR L=1 TO 8
        FOR I=BBB TO AAA
            IF I=N(K) AND L=1 THEN GG(I)=GG(I)+GE(K)
               IF C$(I)="C" THEN
                  GG(I)=GG(I)+GE(K)
               ELSE
                  IF C$(I)="F" THEN
                     GGG(I)=GGG(I)+GE(K)*-CF(I)
                  END IF
               END IF
            END IF
            FOR J=BBB TO AAA
```

```
                IF I=N(K) AND J=N(L) THEN MG(I,J)=MG(I,J)+ME(K,L):NG(I,J)=NG(I,J)+NE(K,L)
            NEXT
        NEXT
    NEXT
NEXT

REM   ***GLOBALCD.FEM***END*****
```

```
Filename: MATRIXGF.FEM        Type: Turbo BASIC program file

Purpose : Simplifies the FE groundwater flow formulation
          into a system of linear algebraic equations.


REM   ***MATRIXGF.FEM***BEGIN***

REM   TIME->TOTAL TIME ELAPSED
REM   KK(I,J) -> REORGANISED KG(I,J)
REM   K(I,J)  -> FILTERED KG(I,J)
REM   KH(I)   -> VECTOR K(I,J)*H(J)

LOCATE 22,1:PRINT "CALCULATING THE NODAL POTENTIALS AT TIME";TIME;"SECS                    "

MATRIXGFSTART:

FOR I=1 TO N
    IF RG(I)<>0 THEN
       FOR J=1 TO N
           K(J,I)=KG(J,I)
           IF I=J THEN
              KK(J,I)=-RG(I)
           ELSE
              KK(J,I)=0
           END IF
       NEXT
    ELSE
       FOR J=1 TO N
           KK(J,I)=KG(J,I)
       NEXT
    END IF
NEXT

FOR I= 1 TO N
    E(I)=0
    FOR J=1 TO N
        E(I)=E(I)+LG(I,J)*H(J)
    NEXT
NEXT

FOR I=1 TO N
    KH(I)=0
    FOR J=1 TO N
        KH(I)=KH(I)-K(I,J)*H(J)
    NEXT
    KK(I,N+1)=KH(I)-DG(I)+E(I)
NEXT

FOR I=1 TO N
    DG(I)=0
    RG(I)=0
    FOR J=1 TO N
        KG(I,J)=0
```

```
        LG(I,J)=0
    NEXT
NEXT

$INCLUDE "GEMGF.FEM"

FOR I=1 TO N
    IF B$(I)<>"Y" AND B$(I)<>"S" THEN H(I)=A(I)
NEXT

REM    ***MATRIXGF.FEM***END*****
```

```
REM    ***GEMGF.FEM***BEGIN***

W=0:W1=0:W2=0:W3=0

GFELIMA:
FOR I=1 TO N
    W2=KK(I,I)
    IF W2=0 THEN
       W3=W3+1
    ELSE
       FOR J=1 TO N+1
           KK(I,J)=KK(I,J)/W2
       NEXT
    END IF
    FOR K=1 TO N
        IF K<>I THEN
        W=KK(K,I)
        FOR J=1 TO N+1
            KK(K,J)=KK(K,J)-W*KK(I,J)
        NEXT
        END IF
    NEXT
NEXT
FOR I=1 TO N
    FOR J=1 TO N
        W1=W1+KK(I,J)
    NEXT
    IF W1=0 THEN GOTO GEMGFFIN
    W1=0
NEXT
IF W3=0 THEN GOTO GEMGFFIN
W3=W3-1
GOTO GFELIMA

GEMGFFIN:

FOR I=1 TO N
    A(I)=KK(I,N+1)
NEXT

REM    ***GEMGF.FEM***END*****
```

```
Filename: MESH.FEM              Type: Turbo BASIC program file

Purpose : Adjusts the mesh in the case of an unconfined
          aquifer, and controls the convergence tolerance.
          Also reorders the calculation of the siffness
          matrices if the mesh changes beyond the specified
          tolerance.


REM   ***MESH.FEM***BEGIN***

LOCATE 22,1:PRINT SPACE$(80)
LOCATE 22,1:PRINT "ADJUSTING THE MESH                                    "

FOR I=1 TO N
    IF S$(I)="Y" THEN
       X=X(I):Y=Y(I):U=U(I)
       IF ABS((U(I)-H(I))/U(I))>.02 THEN ITER=ITER+1
       U(I)=H(I)
       FOR J=1 TO N
           IF J<>I AND B$(J)<>"Y" THEN
              IF X(J)=X AND Y(J)=Y THEN
                 U(J)=U(J)/U*H(I)
              END IF
           END IF
       NEXT J
    END IF
NEXT I

OPEN "I",#2,"MESH"
OPEN "O",#3,"TEMPMESH"
FOR E=1 TO ELEM
    ELEMENT(E)=0
    FOR I=1 TO 8
        INPUT #2,N(I),X(N(I)),Y(N(I)),U
        IF U>0 THEN
           IF ABS((U-U(N(I)))/U)>.02 THEN
              ELEMENT(E)=ELEMENT(E)+1
              WRITE #3,N(I),X(N(I)),Y(N(I)),U(N(I))
           ELSE
              WRITE #3,N(I),X(N(I)),Y(N(I)),U
           END IF
        ELSE
           WRITE #3,N(I),X(N(I)),Y(N(I)),U
        END IF
    NEXT I
NEXT E
CLOSE #2,#3

OPEN "O",#2,"MESH"
OPEN "I",#3,"TEMPMESH"

LOCATE 22,1:PRINT SPACE$(80)
LOCATE 22,1:PRINT "CALCULATING ELEMENT VELOCITIES AT TIME";TIME;"SECS"
```

```
FOR E=1 TO ELEM
    FOR I=1 TO 8
        INPUT #3,N(I),X(N(I)),Y(N(I)),U(N(I))
       WRITE #2,N(I),X(N(I)),Y(N(I)),U(N(I))
    NEXT I
    VX=FN D01DX(0,0,0)*H(N(1))+FN D02DX(0,0,0)*H(N(2))+FN D03DX(0,0,0)*H(N(3))+_
       FN D04DX(0,0,0)*H(N(4))+FN D05DX(0,0,0)*H(N(5))+FN D06DX(0,0,0)*H(N(6))+_
       FN D07DX(0,0,0)*H(N(7))+FN D08DX(0,0,0)*H(N(8))
    VX(E)=-KX(E)*VX
    VY=FN D01DY(0,0,0)*H(N(1))+FN D02DY(0,0,0)*H(N(2))+FN D03DY(0,0,0)*H(N(3))+_
       FN D04DY(0,0,0)*H(N(4))+FN D05DY(0,0,0)*H(N(5))+FN D06DY(0,0,0)*H(N(6))+_
       FN D07DY(0,0,0)*H(N(7))+FN D08DY(0,0,0)*H(N(8))
    VY(E)=-KY(E)*VY
    VU=FN D01DU(0,0,0)*H(N(1))+FN D02DU(0,0,0)*H(N(2))+FN D03DU(0,0,0)*H(N(3))+_
       FN D04DU(0,0,0)*H(N(4))+FN D05DU(0,0,0)*H(N(5))+FN D06DU(0,0,0)*H(N(6))+_
       FN D07DU(0,0,0)*H(N(7))+FN D08DU(0,0,0)*H(N(8))
    VU(E)=-KU(E)*VU
NEXT E
CLOSE #2,#3

KILL "TEMPMESH"

REM   ***MESH.FEM***END*****
```

```
Filename: MATRIXCD.FEM        Type: Turbo BASIC program file

Purpose : Simplifies the FE diffusion-convection formulation
          into a system of linear algebraic equations.


REM   ***MATRIXCD.FEM***BEGIN***

REM   MM(I,J) -> REORGANISED MG(I,J)
REM   M(I,J)  -> FILTERED MG(I,J)
REM   MH(I)   -> VECTOR M(I,J)*C(J)
REM   F(I)    -> VECTOR NG(I,J)*H(J)

LOCATE 22,1:PRINT "CALCULATING THE NODAL CONCENTRATIONS AT TIME";TIME;"SECS                "

MATRIXCDSTART:

FOR I=1 TO N
    IF GG(I)<>0 THEN
       FOR J=1 TO N
       M(J,I)=MG(J,I)
           IF I=J THEN
              MM(J,I)=-GG(I)
           ELSE
              MM(J,I)=0
           END IF
       NEXT
    ELSE
       FOR J=1 TO N
           MM(J,I)=MG(J,I)
       NEXT
    END IF
NEXT

FOR I=1 TO N
    F(I)=0
    FOR J=1 TO N
        F(I)=F(I)+NG(I,J)*C(J)
    NEXT
NEXT

FOR I=1 TO N
    MH(I)=0
    FOR J=1 TO N
        MH(I)=MH(I)-M(I,J)*C(J)
    NEXT
    MM(I,N+1)=MH(I)+F(I)+GGG(I)
NEXT

FOR I=1 TO N
    'GG(I)=0
    'GGG(I)=0
    'FOR J=1 TO N
    '    MG(I,J)=0
```

```
    '     NG(I,J)=0
    'NEXT
NEXT

$INCLUDE "GEMCD.FEM"

FOR I=1 TO N
    IF C$(I)<>"C" THEN
       C(I)=B(I)
    END IF
NEXT

REM   ***MATRIXCD.FEM***END*****
```

```
Filename: GEMCD.FEM          Type: Turbo BASIC program file

Purpose : Solves the nodal values of concentration using
          the Gaussian elimination method.


REM   ***GEMCD.FEM***BEGIN***

W=0:W1=0:W2=0:W3=0

CDELIMA:
FOR I=1 TO N
    W2=MM(I,I)
    IF W2=0 THEN
       W3=W3+1
    ELSE
       FOR J=1 TO N+1
           MM(I,J)=MM(I,J)/W2
       NEXT
    END IF
    FOR K=1 TO N
        IF K<>I THEN
        W=MM(K,I)
        FOR J=1 TO N+1
            MM(K,J)=MM(K,J)-W*MM(I,J)
        NEXT
        END IF
    NEXT
NEXT
FOR I=1 TO N
    FOR J=1 TO N
        W1=W1+MM(I,J)
    NEXT
    IF W1=0 THEN GOTO GEMCDFIN
    W1=0
NEXT
IF W3=0 THEN GOTO GEMCDFIN
W3=W3-1
GOTO CDELIMA

GEMCDFIN:

FOR I=1 TO N
    B(I)=MM(I,N+1)
NEXT

REM   ***GEMCD.FEM***END*****
```

APPENDIX D Model verification - input and output listings for test problems 2 and 3

Test Problem 2: An unconfined aquifer

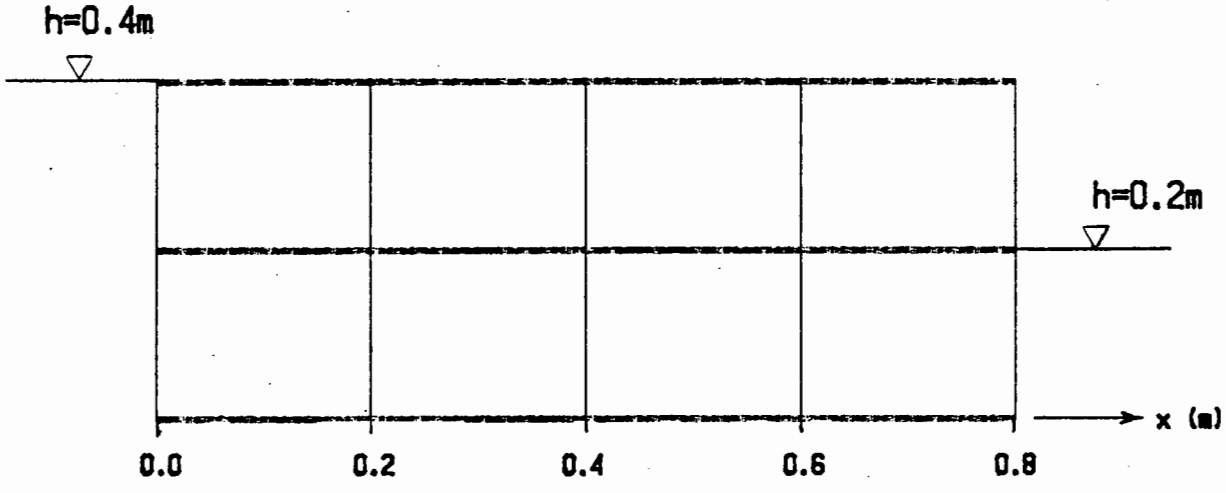


Figure D.1 Test Problem 2: An unconfined aquifer - before FE process

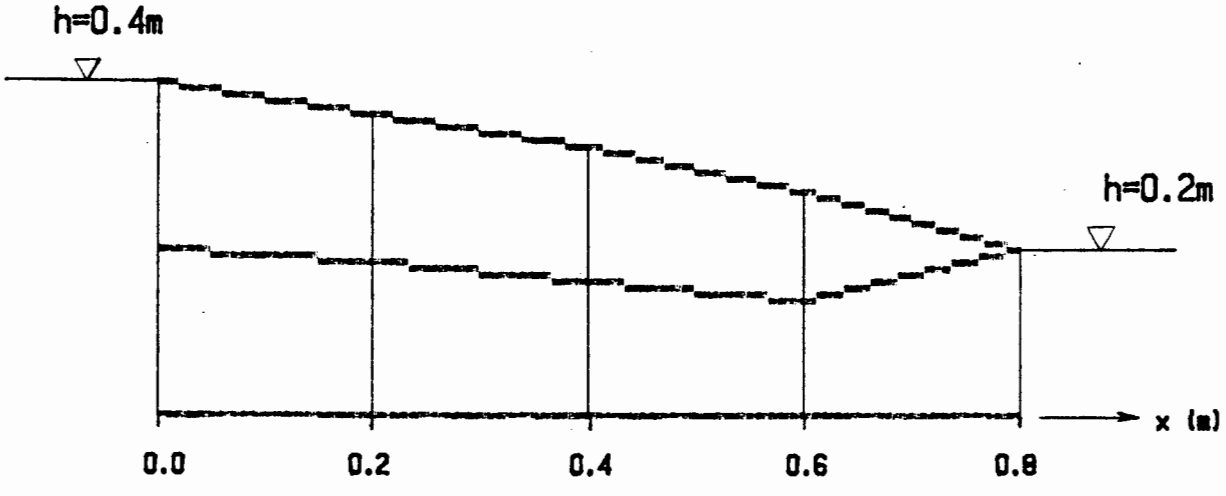


Figure D.2 Test Problem 2: An unconfined aquifer - after FE process

AQUIFEM DATA FOR RUN TEST - PROBLEM 2

NUMBER OF ELEMENTS = 8
NUMBER OF NODES = 30

N	X	Y	U	H	B$	Q	C	C$	S$	CF
1	0.000	0.000	0.000	0.400	Y	0.000000	0.000	N	N	0.000000
2	0.000	0.200	0.000	0.400	Y	0.000000	0.000	N	N	0.000000
3	0.200	0.000	0.000	0.400	N	0.000000	0.000	N	N	0.000000
4	0.200	0.200	0.000	0.400	N	0.000000	0.000	N	N	0.000000
5	0.400	0.000	0.000	0.400	N	0.000000	0.000	N	N	0.000000
6	0.400	0.200	0.000	0.400	N	0.000000	0.000	N	N	0.000000
7	0.600	0.000	0.000	0.400	N	0.000000	0.000	N	N	0.000000
8	0.600	0.200	0.000	0.400	N	0.000000	0.000	N	N	0.000000
9	0.800	0.000	0.000	0.200	Y	0.000000	0.000	N	N	0.000000
10	0.800	0.200	0.000	0.200	Y	0.000000	0.000	N	N	0.000000
11	0.000	0.000	0.200	0.400	Y	0.000000	0.000	N	N	0.000000
12	0.000	0.200	0.200	0.400	Y	0.000000	0.000	N	N	0.000000
13	0.200	0.000	0.200	0.400	N	0.000000	0.000	N	N	0.000000
14	0.200	0.200	0.200	0.400	N	0.000000	0.000	N	N	0.000000
15	0.400	0.000	0.200	0.400	N	0.000000	0.000	N	N	0.000000
16	0.400	0.200	0.200	0.400	N	0.000000	0.000	N	N	0.000000
17	0.600	0.000	0.200	0.400	N	0.000000	0.000	N	N	0.000000
18	0.600	0.200	0.200	0.400	N	0.000000	0.000	N	N	0.000000
19	0.800	0.000	0.200	0.200	Y	0.000000	0.000	N	N	0.000000
20	0.800	0.200	0.200	0.200	Y	0.000000	0.000	N	N	0.000000
21	0.000	0.000	0.400	0.400	Y	0.000000	0.000	N	Y	0.000000
22	0.000	0.200	0.400	0.400	Y	0.000000	0.000	N	Y	0.000000
23	0.200	0.000	0.400	0.400	N	0.000000	0.000	N	Y	0.000000
24	0.200	0.200	0.400	0.400	N	0.000000	0.000	N	Y	0.000000
25	0.400	0.000	0.400	0.400	N	0.000000	0.000	N	Y	0.000000
26	0.400	0.200	0.400	0.400	N	0.000000	0.000	N	Y	0.000000
27	0.600	0.000	0.400	0.400	N	0.000000	0.000	N	Y	0.000000
28	0.600	0.200	0.400	0.400	N	0.000000	0.000	N	Y	0.000000
29	0.800	0.000	0.400	0.400	N	0.000000	0.000	N	Y	0.000000
30	0.800	0.200	0.400	0.400	N	0.000000	0.000	N	Y	0.000000

E	KX	KY	KU	S	P	DX	DY	DU
1	0.001000	0.001000	0.001000	1.500	1.000	0.00001000	0.00001000	0.00001000
	NODES:	1	3	4	2	11	13	14	12
2	0.001000	0.001000	0.001000	1.500	1.000	0.00001000	0.00001000	0.00001000
	NODES:	3	5	6	2	13	15	16	14
3	0.001000	0.001000	0.001000	1.500	1.000	0.00001000	0.00001000	0.00001000
	NODES:	5	7	8	6	15	17	18	16
4	0.001000	0.001000	0.001000	1.500	1.000	0.00001000	0.00001000	0.00001000
	NODES:	7	9	10	8	17	19	20	18
5	0.001000	0.001000	0.001000	1.500	1.000	0.00001000	0.00001000	0.00001000
	NODES:	11	13	14	12	21	23	24	22
6	0.001000	0.001000	0.001000	1.500	1.000	0.00001000	0.00001000	0.00001000
	NODES:	13	15	16	14	23	25	26	24
7	0.001000	0.001000	0.001000	1.500	1.000	0.00001000	0.00001000	0.00001000
	NODES:	15	17	18	16	25	27	28	26
8	0.001000	0.001000	0.001000	1.500	1.000	0.00001000	0.00001000	0.00001000
	NODES:	17	19	20	18	27	29	30	28

AQUIFEM RESULTS - TEST PROBLEM 2

NUMBER OF ELEMENTS = 8
NUMBER OF NODES = 30

STEP NUMBER 0 TIME 0 SECS NUMBER OF ITERATIONS 5

NODE	X-COORD	Y-COORD	U-COORD	POTENTIAL HEAD	CONCENTRATION
1	0.000	0.000	0.000	0.400	0.000
2	0.000	0.200	0.000	0.400	0.000
3	0.200	0.000	0.000	0.358	0.000
4	0.200	0.200	0.000	0.358	0.000
5	0.400	0.000	0.000	0.313	0.000
6	0.400	0.200	0.000	0.313	0.000
7	0.600	0.000	0.000	0.261	0.000
8	0.600	0.200	0.000	0.261	0.000
9	0.800	0.000	0.000	0.200	0.000
10	0.800	0.200	0.000	0.200	0.000
11	0.000	0.000	0.200	0.400	0.000
12	0.000	0.200	0.200	0.400	0.000
13	0.200	0.000	0.190	0.360	0.000
14	0.200	0.200	0.190	0.360	0.000
15	0.400	0.000	0.160	0.315	0.000
16	0.400	0.200	0.160	0.315	0.000
17	0.600	0.000	0.140	0.264	0.000
18	0.600	0.200	0.140	0.264	0.000
19	0.800	0.000	0.200	0.200	0.000
20	0.800	0.200	0.200	0.200	0.000
21	0.000	0.000	0.400	0.400	0.000
22	0.000	0.200	0.400	0.400	0.000
23	0.200	0.000	0.370	0.365	0.000
24	0.200	0.200	0.370	0.365	0.000
25	0.400	0.000	0.330	0.322	0.000
26	0.400	0.200	0.330	0.322	0.000
27	0.600	0.000	0.280	0.271	0.000
28	0.600	0.200	0.280	0.271	0.000
29	0.800	0.000	0.200	0.203	0.000
30	0.800	0.200	0.200	0.203	0.000

ELEMENT	X-VELOCITY	Y-VELOCITY	U-VELOCITY
1	0.0002059	-0.0000000	-0.0000040
2	0.0002233	-0.0000000	-0.0000110
3	0.0002585	-0.0000000	-0.0000179
4	0.0003119	-0.0000000	-0.0000091
5	0.0001881	-0.0000000	-0.0000131
6	0.0002122	-0.0000000	-0.0000337
7	0.0002461	-0.0000000	-0.0000479
8	0.0003290	-0.0000000	-0.0000732

Test Problem 3: One - dimensional diffusion convection

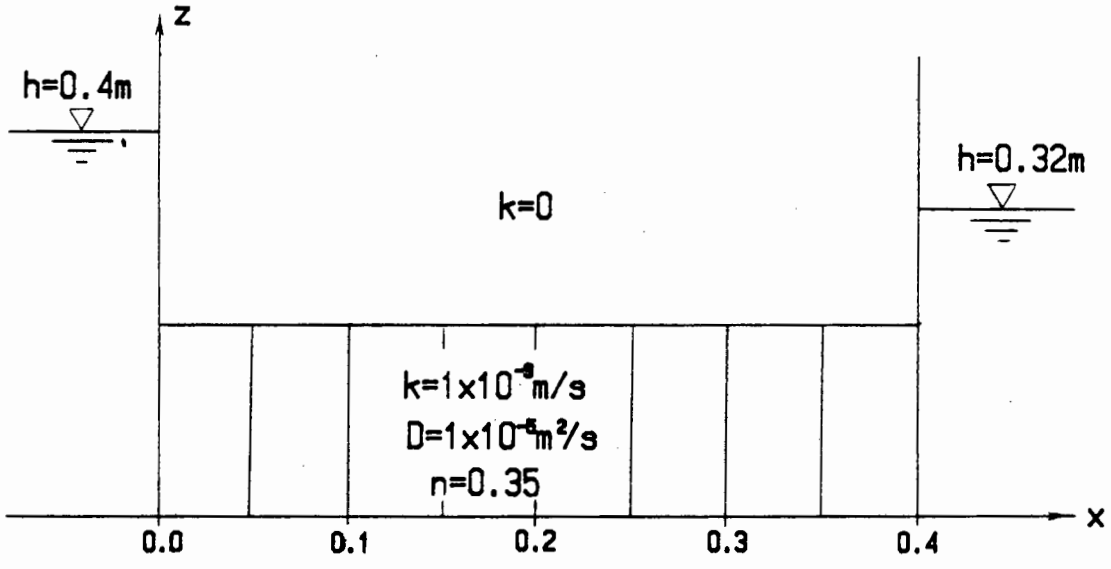


Figure D.3 Test Problem 3: Contaminant transport domain

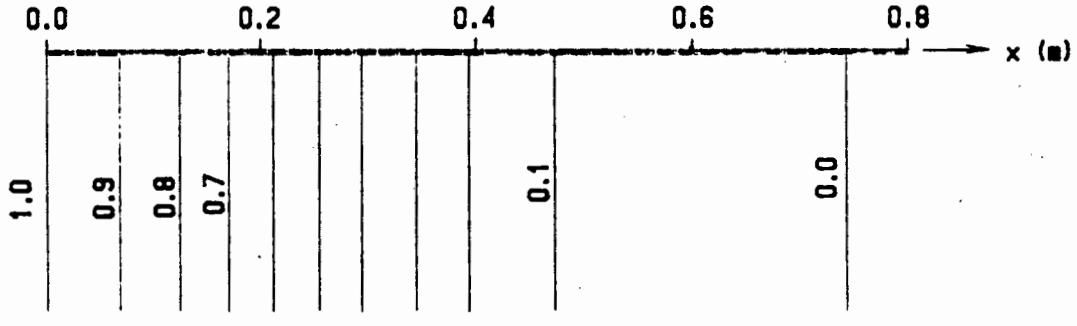


Figure D.4 Test Problem 3: Concentration contour map at time t=1200 seconds

AQUIFEM DATA FOR RUN - TEST PROBLEM 3

NUMBER OF ELEMENTS = 8
NUMBER OF NODES = 36

N	X	Y	U	H	B$	Q	C	C$	S$	CF
1	0.000	0.000	0.000	0.400	Y	0.000000	1.000	C	N	0.000000
2	0.000	0.200	0.000	0.400	Y	0.000000	1.000	C	N	0.000000
3	0.050	0.000	0.000	0.390	N	0.000000	0.000	N	N	0.000000
4	0.050	0.200	0.000	0.390	N	0.000000	0.000	N	N	0.000000

N	X	Y	U	H	B$	Q	C	C$	S$	CF
5	0.100	0.000	0.000	0.380	N	0.000000	0.000	N	N	0.000000
6	0.100	0.200	0.000	0.380	N	0.000000	0.000	N	N	0.000000
7	0.150	0.000	0.000	0.370	N	0.000000	0.000	N	N	0.000000
8	0.150	0.200	0.000	0.370	N	0.000000	0.000	N	N	0.000000
9	0.200	0.000	0.000	0.360	N	0.000000	0.000	N	N	0.000000
10	0.200	0.200	0.000	0.360	N	0.000000	0.000	N	N	0.000000
11	0.250	0.000	0.000	0.350	N	0.000000	0.000	N	N	0.000000
12	0.250	0.200	0.000	0.350	N	0.000000	0.000	N	N	0.000000
13	0.300	0.000	0.000	0.340	N	0.000000	0.000	N	N	0.000000
14	0.300	0.200	0.000	0.340	N	0.000000	0.000	N	N	0.000000
15	0.350	0.000	0.000	0.330	N	0.000000	0.000	N	N	0.000000
16	0.350	0.200	0.000	0.330	N	0.000000	0.000	N	N	0.000000
17	0.400	0.000	0.000	0.320	Y	0.000000	0.000	N	N	0.000000
18	0.400	0.200	0.000	0.320	Y	0.000000	0.000	N	N	0.000000
19	0.000	0.000	0.200	0.400	Y	0.000000	1.000	C	N	0.000000
20	0.000	0.200	0.200	0.400	Y	0.000000	1.000	C	N	0.000000
21	0.050	0.000	0.200	0.390	N	0.000000	0.000	N	N	0.000000
22	0.050	0.200	0.200	0.390	N	0.000000	0.000	N	N	0.000000
23	0.100	0.000	0.200	0.380	N	0.000000	0.000	N	N	0.000000
24	0.100	0.200	0.200	0.380	N	0.000000	0.000	N	N	0.000000
25	0.150	0.000	0.200	0.370	N	0.000000	0.000	N	N	0.000000
26	0.150	0.200	0.200	0.370	N	0.000000	0.000	N	N	0.000000
27	0.200	0.000	0.200	0.360	N	0.000000	0.000	N	N	0.000000
28	0.200	0.200	0.200	0.360	N	0.000000	0.000	N	N	0.000000
29	0.250	0.000	0.200	0.350	N	0.000000	0.000	N	N	0.000000
30	0.250	0.200	0.200	0.350	N	0.000000	0.000	N	N	0.000000
31	0.300	0.000	0.200	0.340	N	0.000000	0.000	N	N	0.000000
32	0.300	0.200	0.200	0.340	N	0.000000	0.000	N	N	0.000000
33	0.350	0.000	0.200	0.330	N	0.000000	0.000	N	N	0.000000
34	0.350	0.200	0.200	0.330	N	0.000000	0.000	N	N	0.000000
35	0.400	0.000	0.200	0.320	Y	0.000000	0.000	N	N	0.000000
36	0.400	0.200	0.200	0.320	Y	0.000000	0.000	N	N	0.000000

E	KX	KY	KU	S	P	DX	DY	DU
1	0.001000	0.001000	0.001000	1.500	0.350	0.00001000	0.00001000	0.00001000
	NODES:	1	3	4	2	19	21	22	20
2	0.001000	0.001000	0.001000	1.500	0.350	0.00001000	0.00001000	0.00001000
	NODES:	3	5	6	4	21	23	24	22
3	0.001000	0.001000	0.001000	1.500	0.350	0.00001000	0.00001000	0.00001000
	NODES:	5	7	8	6	23	25	26	24
4	0.001000	0.001000	0.001000	1.500	0.350	0.00001000	0.00001000	0.00001000
	NODES:	7	9	10	8	25	27	28	26
5	0.001000	0.001000	0.001000	1.500	0.350	0.00001000	0.00001000	0.00001000
	NODES:	9	11	12	10	27	29	30	28
6	0.001000	0.001000	0.001000	1.500	0.350	0.00001000	0.00001000	0.00001000
	NODES:	11	13	14	12	29	31	32	30
7	0.001000	0.001000	0.001000	1.500	0.350	0.00001000	0.00001000	0.00001000
	NODES:	13	15	16	14	31	33	34	32
8	0.001000	0.001000	0.001000	1.500	0.350	0.00001000	0.00001000	0.00001000
	NODES:	15	17	18	16	33	35	36	34

AQUIFEM RESULTS - TEST PROBLEM 3

NUMBER OF ELEMENTS = 8
NUMBER OF NODES = 36

STEP NUMBER 20 TIME 1200 SECS NUMBER OF ITERATIONS 1

NODE	X-COORD	Y-COORD	U-COORD	POTENTIAL HEAD	CONCENTRATION
1	0.000	0.000	0.000	0.400	1.000
2	0.000	0.200	0.000	0.400	1.000
3	0.050	0.000	0.000	0.390	0.852
4	0.050	0.200	0.000	0.390	0.852
5	0.100	0.000	0.000	0.380	0.626
7	0.150	0.000	0.000	0.370	0.382
8	0.150	0.200	0.000	0.370	0.382
9	0.200	0.000	0.000	0.360	0.186
10	0.200	0.200	0.000	0.360	0.186
11	0.250	0.000	0.000	0.350	0.068
12	0.250	0.200	0.000	0.350	0.068
13	0.300	0.000	0.000	0.340	0.017
14	0.300	0.200	0.000	0.340	0.017
15	0.350	0.000	0.000	0.330	0.002
16	0.350	0.200	0.000	0.330	0.002
17	0.400	0.000	0.000	0.320	-0.000
18	0.400	0.200	0.000	0.320	-0.000
19	0.000	0.000	0.200	0.400	1.000
20	0.000	0.200	0.200	0.400	1.000
21	0.050	0.000	0.200	0.390	0.852
22	0.050	0.200	0.200	0.390	0.852
23	0.100	0.000	0.200	0.380	0.626
24	0.100	0.200	0.200	0.380	0.626
25	0.150	0.000	0.200	0.370	0.382
26	0.150	0.200	0.200	0.370	0.382
27	0.200	0.000	0.200	0.360	0.186
28	0.200	0.200	0.200	0.360	0.186
29	0.250	0.000	0.200	0.350	0.068
30	0.250	0.200	0.200	0.350	0.068
31	0.300	0.000	0.200	0.340	0.017
32	0.300	0.200	0.200	0.340	0.017
33	0.350	0.000	0.200	0.330	0.002
34	0.350	0.200	0.200	0.330	0.002
35	0.400	0.000	0.200	0.320	-0.000
36	0.400	0.200	0.200	0.320	-0.000

ELEMENT	X-VELOCITY	Y-VELOCITY	U-VELOCITY
1	0.0002000	-0.0000000	0.0000000
2	0.0002000	-0.0000000	0.0000000
3	0.0002000	0.0000000	-0.0000000
4	0.0002000	0.0000000	-0.0000000
5	0.0002000	0.0000000	-0.0000000
6	0.0002000	0.0000000	-0.0000000
7	0.0002000	0.0000000	-0.0000000
8	0.0002000	0.0000000	0.0000000

AQUIFEM RESULTS TEST - PROBLEM 3

NUMBER OF ELEMENTS = 8
NUMBER OF NODES = 36

STEP NUMBER 30 TIME 1800 SECS NUMBER OF ITERATIONS 1

NODE	X-COORD	Y-COORD	U-COORD	POTENTIAL HEAD	CONCENTRATION
1	0.000	0.000	0.000	0.400	1.000
2	0.000	0.200	0.000	0.400	1.000
3	0.050	0.000	0.000	0.390	0.915
4	0.050	0.200	0.000	0.390	0.915
5	0.100	0.000	0.000	0.380	0.769
7	0.150	0.000	0.000	0.370	0.583
8	0.150	0.200	0.000	0.370	0.583
9	0.200	0.000	0.000	0.360	0.385
10	0.200	0.200	0.000	0.360	0.385
11	0.250	0.000	0.000	0.350	0.220
12	0.250	0.200	0.000	0.350	0.220
13	0.300	0.000	0.000	0.340	0.105
14	0.300	0.200	0.000	0.340	0.105
15	0.350	0.000	0.000	0.330	0.041
16	0.350	0.200	0.000	0.330	0.041
17	0.400	0.000	0.000	0.320	0.020
18	0.400	0.200	0.000	0.320	0.020
19	0.000	0.000	0.200	0.400	1.000
20	0.000	0.200	0.200	0.400	1.000
21	0.050	0.000	0.200	0.390	0.915
22	0.050	0.200	0.200	0.390	0.915
23	0.100	0.000	0.200	0.380	0.769
24	0.100	0.200	0.200	0.380	0.769
25	0.150	0.000	0.200	0.370	0.580
26	0.150	0.200	0.200	0.370	0.580
27	0.200	0.000	0.200	0.360	0.385
28	0.200	0.200	0.200	0.360	0.385
29	0.250	0.000	0.200	0.350	0.220
30	0.250	0.200	0.200	0.350	0.220
31	0.300	0.000	0.200	0.340	0.105
32	0.300	0.200	0.200	0.340	0.105
33	0.350	0.000	0.200	0.330	0.041
34	0.350	0.200	0.200	0.330	0.041
35	0.400	0.000	0.200	0.320	0.020
36	0.400	0.200	0.200	0.320	0.020

ELEMENT	X-VELOCITY	Y-VELOCITY	U-VELOCITY
1	0.0002000	-0.0000000	0.0000000
2	0.0002000	-0.0000000	0.0000000
3	0.0002000	0.0000000	-0.0000000
4	0.0002000	0.0000000	-0.0000000
5	0.0002000	0.0000000	-0.0000000
6	0.0002000	0.0000000	-0.0000000
7	0.0002000	0.0000000	-0.0000000
8	0.0002000	0.0000000	0.0000000

APPENDIX E The conductivity meter

The Model CG 858 Conductivity Meter, manufactured by Schott Geräte of West Germany, has the measuring accuracy of a high class laboratory instrument. Being portable, it is ideal for both laboratory and field measurements. It has a measuring range of 0.1 µS/cm to 19.9 mS/cm. The conductivity probe, or conductivity measuring cell, consists of platinum (Pt1000) electrodes.

CG 858
Digital Portable
Conductivity Meter

with liquid crystal display (LCD)
with three conductivity measuring ranges
with optimum frequency matching
with temperature measuring range
with automatic or manual temperature compensation

Figure E.1 The portable conductivity meter

The concentration of sodium chloride (table salt) in a solution is a function of the electrical conductivity of the solution. In fact, the presence of any ionic substances in

a water sample will affect the conductivity of that sample. The only reason why NaCl, and not another ionic substance such as KCl or AgCl, was chosen is that NaCl is readily available as well as being the cheapest ionic substance. Solutions of ionic substances follow a uniform pattern of change of conductance with concentration. This pattern is almost linear at low concentrations, rising gradually to a maximum, and then falling as the concentration is further increased [34].

In general, the conductivity of a solution varies with the temperature of the solution. This variation is about 2½% per ·C. A family of conductivity versus concentration curves for NaCl at different temperatures is given in reference [34]. Although the temperature of water from the domestic water supply only varies by a few degrees at most over a period of 24 hours, the corresponding variations in electrical conductivity may be significant. The conductivity meter has a temperature compensation facility which uses a temperature of 25·C as its reference. The temperature compensation control knob is set to the actual temperature of the measured solution.

Water processed for domestic use was found to have an electrical conductivity, probably due to the addition of purification chemicals. This electrical conductivity is of the order of 0.10 mS/cm. On the other hand, the addition of 1 gram of NaCl to a water solution was found to cause an increase in electrical conductivity of about 1.50 mS/cm.

The main objective here is to establish a relationship between electrical conductivity and contaminant (salt) concentration. Conductivity tests were carried out in order to establish this relationship. A measuring cylinder was filled with a sample of domestic (tap) water of known volume. The electrical conductivity of the sample was measured in order to adjust the conductivity readings to be taken after the addition of salt. Salt was added in amounts measured to within 1/100th of a gram and the solution was well stirred with a magnetic stirrer. After each addition of salt, the electrical conductivity was measured and tabulated as shown in table E.1.

The results showed a steady, though minimal, decrease in the slope as the concentration was increased. This behaviour is in accordance with that given in reference [34]. After a suitable number of tests, a final conductivity-salinity curve, based on the average of the individual tests, was produced. This curve, shown in figure E.2, can now be used to read off values of concentration for a given value of electrical conductivity. Note that the curve is shifted along the conductivity axis to adjust for the initial value of electrical conductivity already existent in the uncontaminated (zero-salinity) water.

Table E.1 Conductivity test number 1

Volume of water = 1000ml

Salt concentration (grams)	Conductivity (mS/cm)
0.00	0.09
1.00	1.99
2.00	3.78
3.00	5.50
4.00	7.16
5.00	8.79
6.00	10.40
7.00	11.96
8.00	13.49
9.00	15.01
10.00	16.50
11.00	17.96
12.00	19.39

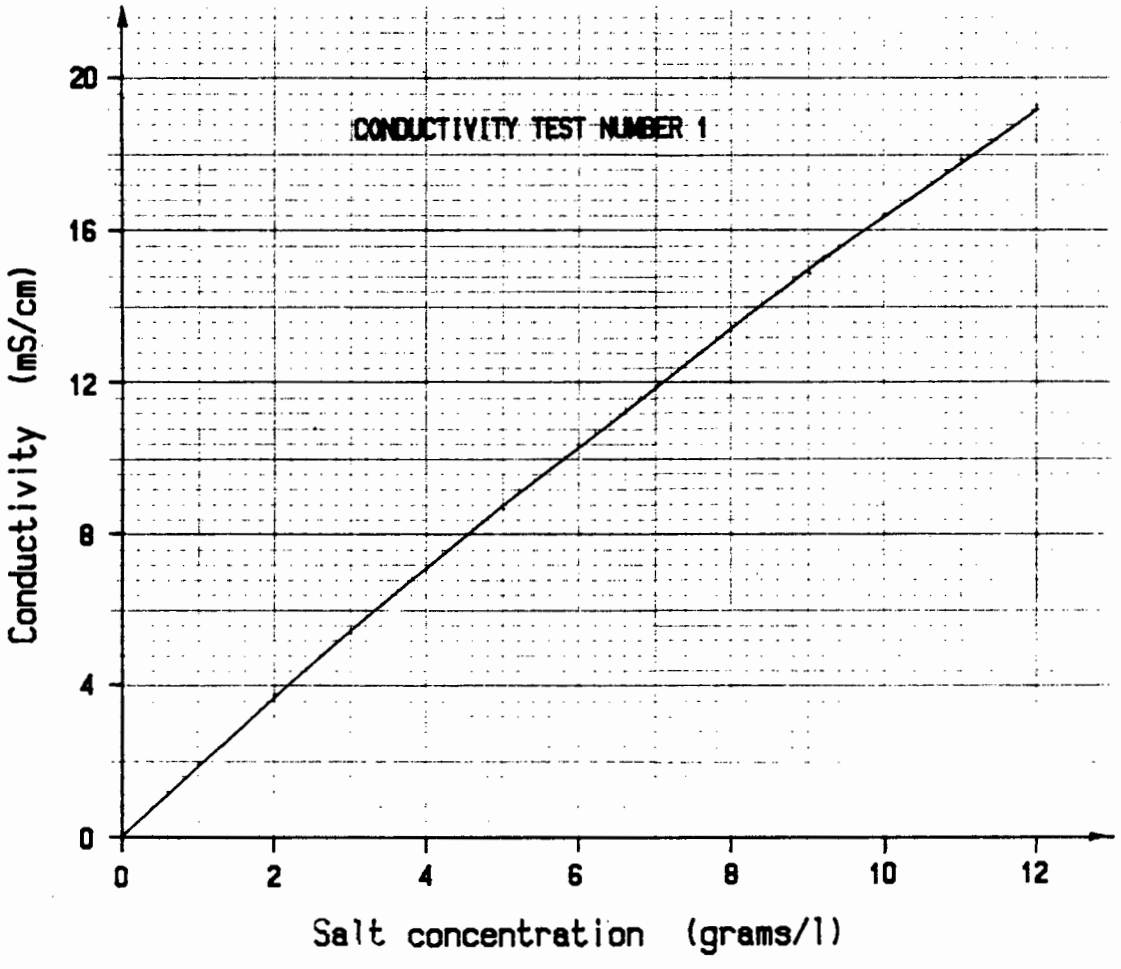


Figure E.2 Plot of conductivity versus salinity for the Schott-Geräte Model CG858 conductivity meter

APPENDIX F Evaluation of the coefficient of permeability of a 20/40 Density sand

Sand was obtained from a sand supplier, the grade size being given as 0.850-0.425 mm. Although the grading data was provided by the supplier, it was best thought to redo the analysis with a sample from the actual sand supplied. The aim here was to establish the grading curve from which the D_{10} size, to be used later in the determination of the coefficient of permeability, could be obtained.

The grading analysis is shown in table F.1. A 500μm sieve was not available for use. The corresponding grading curve is shown plotted on a linear graph as shown in figure F.1. From the curve, the D_{10} size of the sample is found to be 0.49 mm.

Table F.1 Grading analysis for a 20/40 Density sand (0.850-0.425 mm)

APERTURE IN MICROMETERS (μm)	U.S. MESH	% MASS RETAINED	% MASS PASSED
1000	18		
850	20	0.0	100.0
710	25	0.4	99.6
600	30	2.0	97.6
500	35	----	-----
425	40	96.4	1.2
355	45	0.6	0.6
300	50	0.2	0.4
212	70	0.2	0.2
<212	>70	0.2	

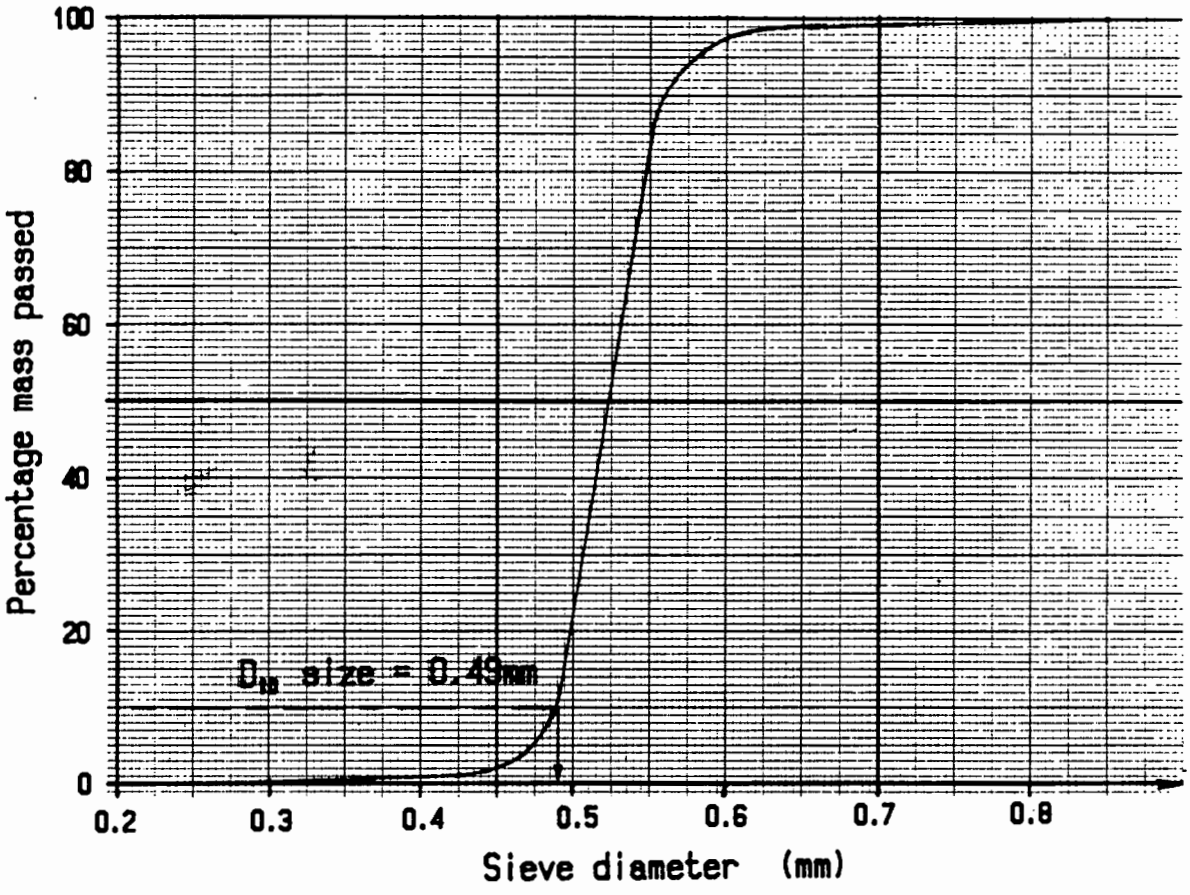


Figure F.1 Grading curve for 20/40 Density sand (0.850-0.425 mm)

A dry sample of the sand of mass md was poured into the permeameter cylinder through water. The sample was then densely compacted. The mass of the saturated sample, m_s, was measured and the following properties of the sand calculated:

(a) Void Ratio e:

Mass of dry sand sample $m_d = 1.671$ kg

Volume of solids $V_s = m_d/SG_s$

$= 0.631 \times 10^{-3}$ m^3

(where SG_s is the specific gravity of sand taken as 2.65)

Volume of saturated sample V = sample height h x A

$= 0.258 \times 0.00385$

$= 0.993 \times 10^{-3}$ m^3

(where A = permeameter area)

Void ratio $e = V_v/V_s$ (E. 1)

$= (V - V_s)/V_s$

$= 0.574$ .... e_{dense}

(b) Porosity n:

Porosity $n = V_v/V$ (E. 2)

$= 0.365$ .... n_{dense}

In the permeameter test, the flowrate through the sample and the corresponding hydraulic gradient across a length L of the sample were recorded. The flowrate was increased in stages and the measurements were repeated. At a certain flowrate, the packing became unstable. The flowrate was

Table F.2 Permeameter test data for 20/40 Density sand (0.850-0.425 mm)

Note: The length L of the sample = 20cm
The permeameter area A = 38.5 cm²

	Overall sample height h' (cm)	Hydraulic head h (cm)	Flowrate Q cm^3/s	Hydraulic gradient i=h/L	Seepage velocity v=Q/A cm/s
	25.8	3.7	1.017	0.185	0.0264
D	25.8	5.4	1.761	0.270	0.0457
E	25.8	8.2	2.400	0.410	0.0623
N	25.8	10.4	3.033	0.520	0.0788
S	25.8	13.0	3.783	0.650	0.0983
E	25.8	15.0	4.350	0.750	0.1130
	25.8	16.9	4.844	0.845	0.1258
	25.8	19.3	5.583	0.965	0.1450
	28.1	16.8	9.511	0.840	0.2470
	28.1	15.7	8.800	0.785	0.2286
L	28.0	13.6	7.500	0.680	0.1948
O	28.0	11.8	6.417	0.590	0.1667
O	28.0	10.5	5.700	0.525	0.1481
S	27.9	8.3	4.450	0.415	0.1156
E	27.9	7.2	3.833	0.360	0.0996
	27.9	4.5	2.333	0.225	0.0606
	27.9	2.3	1.139	0.115	0.0296

then gradually reduced until stability was reached (loose packing). The flowrate was further decreased in stages, and the corresponding hydraulic gradients recorded. The results of the test data is shown in table E.2.

From Darcy's Law, it can be deduced that

$$v = ki$$

Thus, the hydraulic gradient i was plotted against the seepage velocity v and the values of k (k_{dense} and k_{loose})

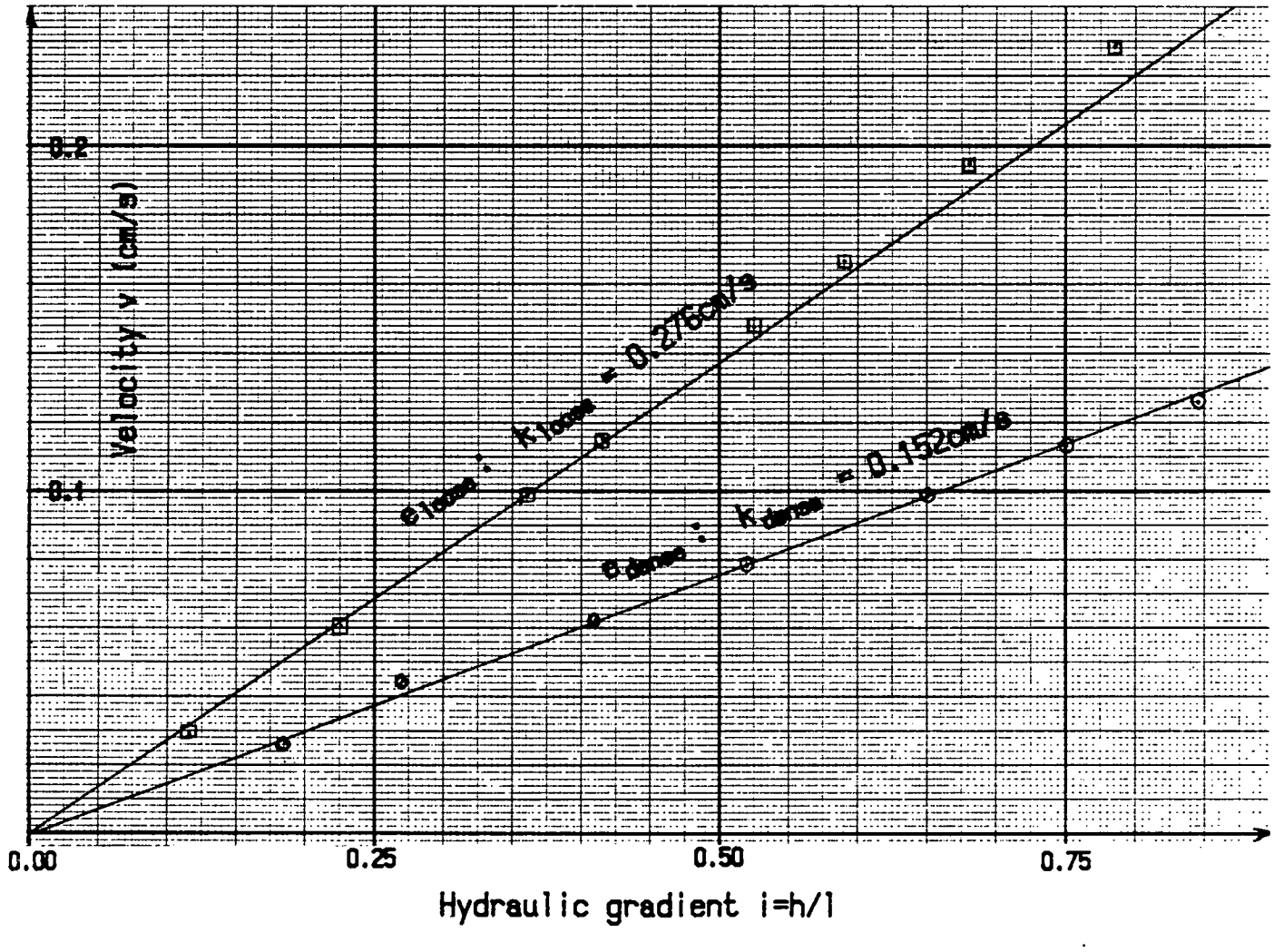


Figure F.2 Plot of hydraulic gradient versus seepage velocity for 20/40 Density sand (0.850-0.425 mm)

estimated from the slopes of the graphs shown in figure F.2. The coefficient of permeability will lie between

$$k_{dense} = 0.152 \text{ cm/s} \quad \text{and} \quad k_{loose} = 0.276 \text{ cm/s}$$

depending on the compaction of the sand.

Since the packing changed from dense to loose, the void ratio and, hence, the porosity will change likewise. These values are therefore recalculated, this time for the loose packing.

(a) Void Ratio e:

Mass of dry sand sample $m_d = 1.671$ kg

Volume of solids $V_s = m_d/SG_s$

$= 0.631 \times 10^{-3}$ m^3

Volume of saturated sample V = sample height h x A

$= 0.279 \times 0.00385$

$= 1.074 \times 10^{-3}$ m^3

Void ratio $e = V_v/V_s$

$= (V - V_s)/V_s$

$= 0.702 \quad \ldots\ e_{loose}$

(b) Porosity n:

Porosity $n = V_v/V$

$= 0.412 \quad \ldots\ n_{loose}$

Estimates of the coefficient of permeability can be found otherwise. Using the D_{10} size of the sample, K can be approximated as follows:

$$K \approx 100\ D_{10}^2 \qquad (E.3)$$
$$\approx 100(0.049^2)$$
$$\approx 0.240\ cm/s$$

This value lies within the K_{dense}-K_{loose} range found from the permeameter test. Furthermore, a method exists for checking the values of K found from the permeameter test. Using the void ratio, e, the value of K is said to be proportional to $e^3/(1+e)$ so that

$$K \approx Ce^3/(1+e) \qquad (E.4)$$

where C is the constant of proportionality. This implies that

$$\frac{kd(1 + ed)}{ed3} \approx \frac{kl(1 + el)}{el3}$$

i.e. $$0.0127 \approx 0.0136$$

where the subscripts d and l indicate "dense" and "loose" respectively. The constant, C, is fairly consistent over the range, showing a variation of only 7 percent.

APPENDIX G Evaluation of the Coefficient of Diffusion for NaCl diffusing into a 20/40 Density sand

A number of tests were run using the method and apparatus described in section 5.3(e). Figure G.1 shows the dimensions used in the calculation of the coefficient of diffusion. These dimensions are required to determine the area of diffusion and the volume of solution (to calculate solute concentrations)

Equation (5.5) can be rearranged to give

$$D = \frac{LV_d}{n\Delta t A_d}\left[\frac{C_L{}^{t+\Delta t} - C_L{}^{t}}{(C_0{}^{t+\Delta t} + C_0{}^{t})/2 - (C_L{}^{t+\Delta t} + C_L{}^{t})/2}\right] \quad \ldots\ldots\ldots \text{(G.1)}$$

The above formula was used to estimate the diffusion

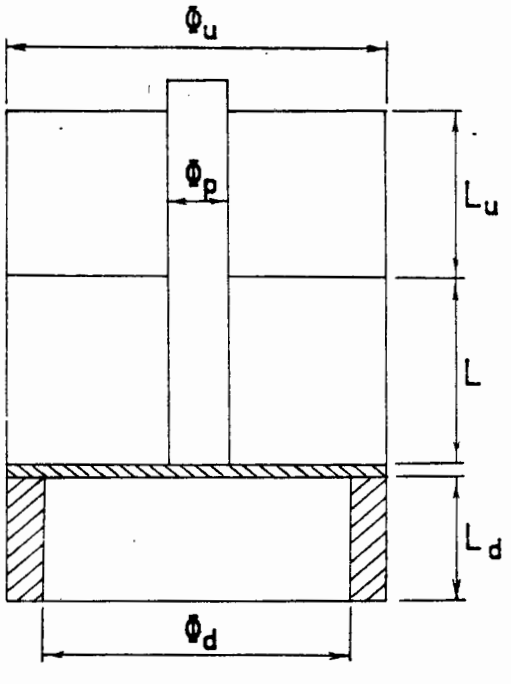


Figure G.1 Diffusion test dimensions

coefficient from the test data. During the diffusion experiments undertaken, the electrical conductivity of the solution at both the upstream and downstream ends were measured at regular time intervals. The conductivity readings were then converted to actual salt content using the graph of figure E.2. This is necessary since the conductivity/concentration relationship is not always linear. The aim is to determine instances where the amount of solute entering the porous medium is equal to that leaving the porous medium. The results obtained from one of the test runs is tabulated in table G.1.

The change in concentrations (ΔC_0 and ΔC_L) are calculated. Since (in this case) the upstream (u) and downstream (d) volumes are different, the value of ΔC_L has to be multiplied by a factor equal to V_d/V_u. This results in $\Delta C_L'$ which can be interpreted as ΔC_L relative to ΔC_0 (or changes in the actual solute content).

According to table G.1, the solute transferred during the 90-120 minute interval is exactly equal. However, the concentration curves at this stage is more exponential than linear, whereas equation (G.1) is based on the linear law of diffusion.

The results obtained in the 210-225 and 225-240 minute intervals also show approximate equal solute transfer. The values of D obtained are approximately equal. These values comply well with those of other similar tests.

Table G.1 Results of diffusion test number 4

$L = 0.065m$ $\phi_p = 0.020m$ $V_d = .000556$
$L_u = 0.025m$ $\phi_u = 0.125m$ $V_u = .000299$
$L_d = 0.065m$ $\phi_d = 0.102m$ $V_d/V_u = 1.87$

porosity n=0.412 (loose packing)

TIME	C_O		ΔC_O	C_L		ΔC_L	$\Delta C_L'$	D
	mS/cm	g/l	g/l	mS/cm	g/l	g/l	g/l	m^2/s
0	18.85	11.70		0.00	0.00			
			1.85			0.10	0.19	0.6E-7
30	16.20	9.85		0.13	0.10			
			1.20			0.35	0.65	2.3E-7
60	14.44	8.65		0.84	0.45			
			0.90			0.40	0.75	3.2E-7
90	13.08	7.75		1.60	0.85			
			0.65			0.35	0.65	3.3E-7
120	12.10	7.10		2.17	1.20			
			0.50			0.25	0.48	2.7E-7
150	11.30	6.60		2.64	1.45			
			0.45			0.20	0.37	2.5E-7
180	10.57	6.15		3.03	1.65			
			0.40			0.15	0.28	2.1E-7
210	9.94	5.75		3.35	1.80			
			0.35			0.20	0.37	3.2E-7
225	9.69	5.60		3.48	1.90			
			0.20			0.10	0.19	3.4E-7
240	9.42	5.40		3.62	2.00			

Figure G.2 shows a plot of the solute transferred versus time for both the upstream and downstream ends. Theoretically, the amount of solute leaving the medium cannot exceed that entering the medium. This may seem to happen when the two curves overlap. However, it is simply due to experimental error.

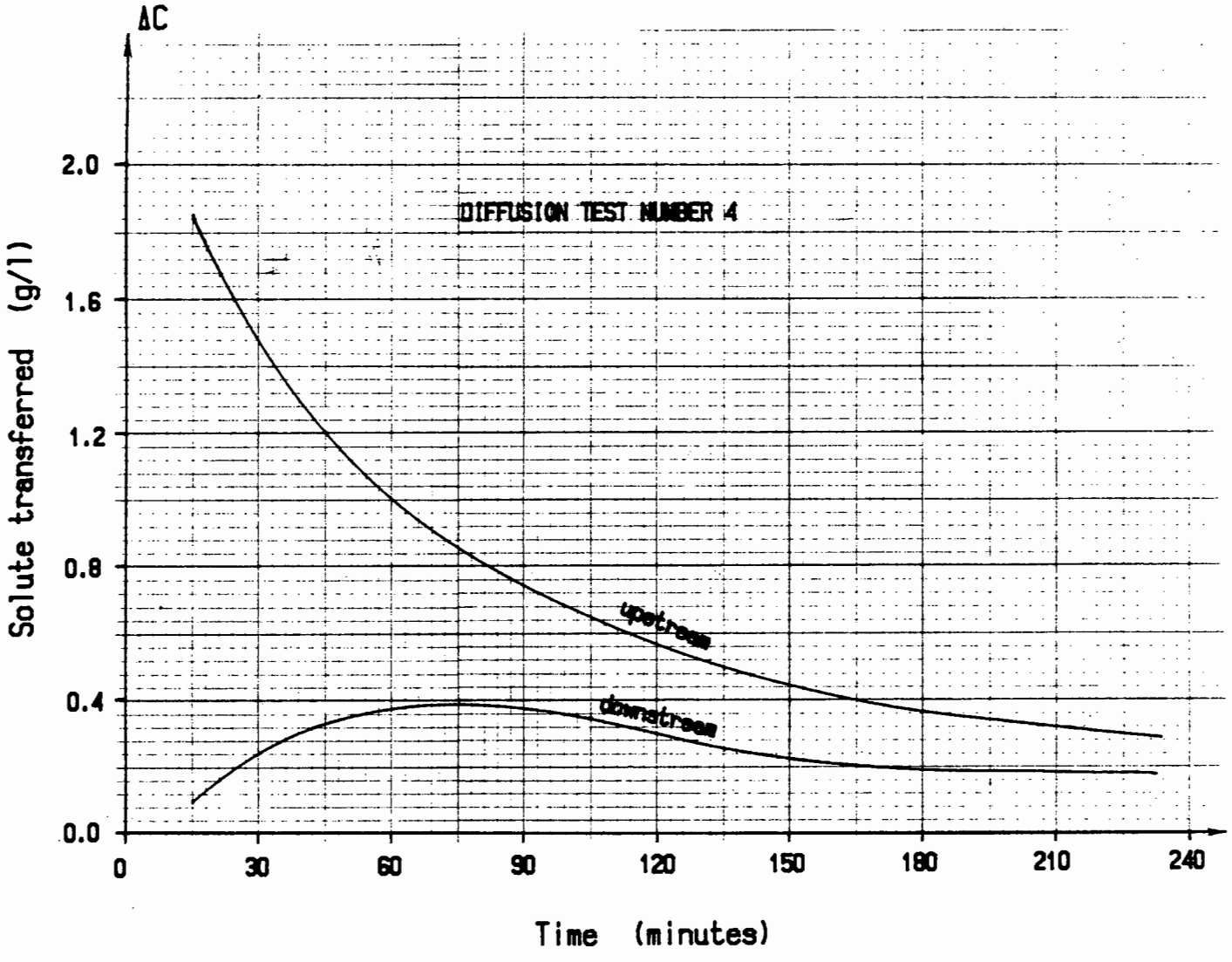


Figure G.2 Solute transfer (ΔC) curves for diffusion test number 4

For the contaminant transport model, a value of D equal to 3.0×10^{-7} will be used. The use of a slightly lower value is justified since the sand packing in the transport model is not entirely loose.

APPENDIX H Input and output listings for experiment number 1

AQUIFEM DATA FOR RUN - EXPERIMENT NO. 1 (K=0.00276m/s)

NUMBER OF ELEMENTS = 48
NUMBER OF NODES = 100

N	X	Y	U	H	B$	Q	C	C$	S$	CF
1	0.000	0.000	0.000	0.368	Y	0.000000	0.000	N	N	0.000000
2	0.200	0.000	0.000	0.368	N	0.000000	0.000	N	N	0.000000
3	0.400	0.000	0.000	0.368	N	0.000000	0.000	N	N	0.000000
4	0.600	0.000	0.000	0.368	N	0.000000	0.000	N	N	0.000000
5	0.800	0.000	0.000	0.307	Y	0.000000	0.000	N	N	0.000000
6	0.000	0.170	0.000	0.368	Y	0.000000	0.000	N	N	0.000000
7	0.200	0.170	0.000	0.368	N	0.000000	0.000	N	N	0.000000
8	0.400	0.170	0.000	0.368	N	0.000000	0.000	N	N	0.000000
9	0.600	0.170	0.000	0.368	N	0.000000	0.000	N	N	0.000000
10	0.800	0.170	0.000	0.307	Y	0.000000	0.000	N	N	0.000000
11	0.000	0.330	0.000	0.368	Y	0.000000	0.000	N	N	0.000000
12	0.200	0.330	0.000	0.368	N	0.000000	0.000	N	N	0.000000
13	0.400	0.330	0.000	0.368	N	0.000000	0.000	N	N	0.000000
14	0.600	0.330	0.000	0.368	N	0.000000	0.000	N	N	0.000000
15	0.800	0.330	0.000	0.307	Y	0.000000	0.000	N	N	0.000000
16	0.000	0.500	0.000	0.368	Y	0.000000	0.000	N	N	0.000000
17	0.200	0.500	0.000	0.368	N	0.000000	0.000	N	N	0.000000
18	0.400	0.500	0.000	0.368	N	0.000000	0.000	N	N	0.000000
19	0.600	0.500	0.000	0.368	N	0.000000	0.000	N	N	0.000000
20	0.800	0.500	0.000	0.307	Y	0.000000	0.000	N	N	0.000000
21	0.000	0.000	0.092	0.368	Y	0.000000	0.000	N	N	0.000000
22	0.200	0.000	0.092	0.368	N	0.000000	0.000	N	N	0.000000
23	0.400	0.000	0.092	0.368	N	0.000000	0.000	N	N	0.000000
24	0.600	0.000	0.092	0.368	N	0.000000	0.000	N	N	0.000000
25	0.800	0.000	0.102	0.307	Y	0.000000	0.000	N	N	0.000000
26	0.000	0.170	0.092	0.368	Y	0.000000	0.000	N	N	0.000000
27	0.200	0.170	0.092	0.368	N	0.000000	0.000	N	N	0.000000
28	0.400	0.170	0.092	0.368	N	0.000000	0.000	N	N	0.000000
29	0.600	0.170	0.092	0.368	N	0.000000	0.000	N	N	0.000000
30	0.800	0.170	0.102	0.307	Y	0.000000	0.000	N	N	0.000000
31	0.000	0.330	0.092	0.368	Y	0.000000	0.000	N	N	0.000000
32	0.200	0.330	0.092	0.368	N	0.000000	0.000	N	N	0.000000
33	0.400	0.330	0.092	0.368	N	0.000000	0.000	N	N	0.000000
34	0.600	0.330	0.092	0.368	N	0.000000	0.000	N	N	0.000000
35	0.800	0.330	0.102	0.307	Y	0.000000	0.000	N	N	0.000000
36	0.000	0.500	0.092	0.368	Y	0.000000	0.000	N	N	0.000000
37	0.200	0.500	0.092	0.368	N	0.000000	0.000	N	N	0.000000
38	0.400	0.500	0.092	0.368	N	0.000000	0.000	N	N	0.000000
39	0.600	0.500	0.092	0.368	N	0.000000	0.000	N	N	0.000000
40	0.800	0.500	0.102	0.307	Y	0.000000	0.000	N	N	0.000000
41	0.000	0.000	0.184	0.368	Y	0.000000	0.000	N	N	0.000000
42	0.200	0.000	0.184	0.368	N	0.000000	0.000	N	N	0.000000
43	0.400	0.000	0.184	0.368	N	0.000000	0.000	N	N	0.000000
44	0.600	0.000	0.184	0.368	N	0.000000	0.000	N	N	0.000000

N	X	Y	U	H	B$	Q	C	C$	S$	CF
45	0.800	0.000	0.204	0.307	Y	0.000000	0.000	N	N	0.000000
46	0.000	0.170	0.184	0.368	Y	0.000000	0.000	N	N	0.000000
47	0.200	0.170	0.184	0.368	N	0.000000	0.000	N	N	0.000000
48	0.400	0.170	0.184	0.368	N	0.000000	0.000	N	N	0.000000
49	0.600	0.170	0.184	0.368	N	0.000000	0.000	N	N	0.000000
50	0.800	0.170	0.204	0.307	Y	0.000000	0.000	N	N	0.000000
51	0.000	0.330	0.184	0.368	Y	0.000000	0.000	N	N	0.000000
52	0.200	0.330	0.184	0.368	N	0.000000	0.000	N	N	0.000000
53	0.400	0.330	0.184	0.368	N	0.000000	0.000	N	N	0.000000
54	0.600	0.330	0.184	0.368	N	0.000000	0.000	N	N	0.000000
55	0.800	0.330	0.204	0.307	Y	0.000000	0.000	N	N	0.000000
56	0.000	0.500	0.184	0.368	Y	0.000000	0.000	N	N	0.000000
57	0.200	0.500	0.184	0.368	N	0.000000	0.000	N	N	0.000000
58	0.400	0.500	0.184	0.368	N	0.000000	0.000	N	N	0.000000
59	0.600	0.500	0.184	0.368	N	0.000000	0.000	N	N	0.000000
60	0.800	0.500	0.204	0.307	Y	0.000000	0.000	N	N	0.000000
61	0.000	0.000	0.276	0.368	Y	0.000000	0.000	N	N	0.000000
62	0.200	0.000	0.276	0.368	N	0.000000	0.000	N	N	0.000000
63	0.400	0.000	0.276	0.368	N	0.000000	0.000	N	N	0.000000
64	0.600	0.000	0.276	0.368	N	0.000000	0.000	N	N	0.000000
65	0.800	0.000	0.307	0.307	Y	0.000000	0.000	N	N	0.000000
66	0.000	0.170	0.276	0.368	Y	0.000000	0.000	N	N	0.000000
67	0.200	0.170	0.276	0.368	N	0.000000	0.000	N	N	0.000000
68	0.400	0.170	0.276	0.368	N	0.000000	0.000	N	N	0.000000
69	0.600	0.170	0.276	0.368	N	0.000000	0.000	N	N	0.000000
70	0.800	0.170	0.307	0.307	Y	0.000000	0.000	N	N	0.000000
71	0.000	0.330	0.276	0.368	Y	0.000000	0.000	N	N	0.000000
72	0.200	0.330	0.276	0.368	N	0.000000	0.000	N	N	0.000000
73	0.400	0.330	0.276	0.368	N	0.000000	0.000	N	N	0.000000
74	0.600	0.330	0.276	0.368	N	0.000000	0.000	N	N	0.000000
75	0.800	0.330	0.307	0.307	Y	0.000000	0.000	N	N	0.000000
76	0.000	0.500	0.276	0.368	Y	0.000000	0.000	N	N	0.000000
77	0.200	0.500	0.276	0.368	N	0.000000	0.000	N	N	0.000000
78	0.400	0.500	0.276	0.368	N	0.000000	0.000	N	N	0.000000
79	0.600	0.500	0.276	0.368	N	0.000000	0.000	N	N	0.000000
80	0.800	0.500	0.307	0.307	Y	0.000000	0.000	N	N	0.000000
81	0.000	0.000	0.368	0.368	Y	0.000000	0.000	N	N	0.000000
82	0.200	0.000	0.368	0.368	N	0.000000	0.000	N	N	0.000000
83	0.400	0.000	0.368	0.368	N	0.000000	0.000	N	N	0.000000
84	0.600	0.000	0.368	0.368	N	0.000000	0.000	N	N	0.000000
85	0.800	0.000	0.368	0.307	N	0.000000	0.000	N	N	0.000000
86	0.000	0.170	0.368	0.368	Y	0.000000	0.000	N	N	0.000000
87	0.200	0.170	0.368	0.368	N	0.000000	0.000	N	N	0.000000
88	0.400	0.170	0.368	0.368	N	0.000000	0.000	N	N	0.000000
89	0.600	0.170	0.368	0.368	N	0.000000	0.000	N	N	0.000000
90	0.800	0.170	0.368	0.307	N	0.000000	0.000	N	N	0.000000
91	0.000	0.330	0.368	0.368	Y	0.000000	0.000	N	N	0.000000
92	0.200	0.330	0.368	0.368	N	0.000000	0.000	N	N	0.000000
93	0.400	0.330	0.368	0.368	N	0.000000	0.000	N	N	0.000000
94	0.600	0.330	0.368	0.368	N	0.000000	0.000	N	N	0.000000
95	0.800	0.330	0.368	0.307	N	0.000000	0.000	N	N	0.000000
96	0.000	0.500	0.368	0.368	Y	0.000000	0.000	N	N	0.000000

N	X	Y	U	H	B$	Q	C	C$	S$	CF
97	0.200	0.500	0.368	0.368	N	0.000000	0.000	N	N	0.000000
98	0.400	0.500	0.368	0.368	N	0.000000	0.000	N	N	0.000000
99	0.600	0.500	0.368	0.368	N	0.000000	0.000	N	N	0.000000
100	0.800	0.500	0.368	0.307	Y	0.000000	0.000	N	N	0.000000

E	KX	KY	KU	S	P	DX	DY	DU
1	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	1	2	7	6	21	22	27	26
2	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	2	3	8	7	22	23	28	27
3	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	3	4	9	8	23	24	29	28
4	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	4	5	10	9	24	25	30	29
5	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	6	7	12	11	26	27	32	31
6	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	7	8	13	12	27	28	33	32
7	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	8	9	14	13	28	29	34	33
8	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	9	10	15	14	29	30	35	34
9	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	11	12	17	16	31	32	37	36
10	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	12	13	18	17	32	33	38	37
11	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	13	14	19	18	33	34	39	38
12	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	14	15	20	19	34	35	40	39
13	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	21	22	27	26	41	42	47	46
14	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	22	23	28	27	42	43	48	47
15	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	23	24	29	28	43	44	49	48
16	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	24	25	30	29	44	45	50	49
17	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	26	27	32	31	46	47	52	51
18	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	27	28	33	32	47	48	53	52
19	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	28	29	34	33	48	49	54	53
20	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	29	30	35	34	49	50	55	54
21	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	31	32	37	36	51	52	57	56
22	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	32	33	38	37	52	53	58	57

E	KX	KY	KU	S	P	DX	DY	DU
23	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	33	34	39	38	53	54	59	58
24	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	34	35	40	39	54	55	60	59
25	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	41	42	47	46	61	62	67	66
26	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	42	43	48	47	62	63	68	67
27	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	43	44	49	48	63	64	69	68
28	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	44	45	50	49	64	65	70	69
29	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	46	47	52	51	66	67	72	71
30	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	47	48	53	52	67	68	73	72
31	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	48	49	54	53	68	69	74	73
32	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	49	50	55	54	69	70	75	74
33	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	51	52	57	56	71	72	77	76
34	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	52	53	58	57	72	73	78	77
35	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	53	54	59	58	73	74	79	78
36	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	54	55	60	59	74	75	80	79
37	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	61	62	67	66	81	82	87	86
38	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	62	63	68	67	82	83	88	87
39	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	63	64	69	68	83	84	89	88
40	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	64	65	70	69	84	85	90	89
41	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	66	67	72	71	86	87	92	91
42	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	67	68	73	72	87	88	93	92
43	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	68	69	74	73	88	89	94	93
44	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	69	70	75	74	89	90	95	94
45	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	71	72	77	76	91	92	97	96
46	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	72	73	78	77	92	93	98	97
47	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	73	74	79	78	93	94	99	98
48	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	74	75	80	79	94	95	100	99

AQUIFEM RESULTS - EXPERIMENT NO. 1 (k=0.00276m/s)

NUMBER OF ELEMENTS = 48
NUMBER OF NODES = 100

STEP NUMBER 0 TIME 0 SECS NUMBER OF ITERATIONS 2

NODE	X-COORD	Y-COORD	U-COORD	POTENTIAL HEAD	CONCENTRATION
1	0.000	0.000	0.000	0.368	0.000
2	0.200	0.000	0.000	0.352	0.000
3	0.400	0.000	0.000	0.337	0.000
4	0.600	0.000	0.000	0.322	0.000
5	0.800	0.000	0.000	0.307	0.000
6	0.000	0.170	0.000	0.368	0.000
7	0.200	0.170	0.000	0.353	0.000
8	0.400	0.170	0.000	0.337	0.000
9	0.600	0.170	0.000	0.322	0.000
10	0.800	0.170	0.000	0.307	0.000
11	0.000	0.330	0.000	0.368	0.000
12	0.200	0.330	0.000	0.353	0.000
13	0.400	0.330	0.000	0.338	0.000
14	0.600	0.330	0.000	0.322	0.000
15	0.800	0.330	0.000	0.307	0.000
16	0.000	0.500	0.000	0.368	0.000
17	0.200	0.500	0.000	0.353	0.000
18	0.400	0.500	0.000	0.338	0.000
19	0.600	0.500	0.000	0.322	0.000
20	0.800	0.500	0.000	0.307	0.000
21	0.000	0.000	0.092	0.368	0.000
22	0.200	0.000	0.088	0.352	0.000
23	0.400	0.000	0.084	0.337	0.000
24	0.600	0.000	0.080	0.322	0.000
25	0.800	0.000	0.102	0.307	0.000
26	0.000	0.170	0.092	0.368	0.000
27	0.200	0.170	0.088	0.353	0.000
28	0.400	0.170	0.080	0.337	0.000
29	0.600	0.170	0.081	0.322	0.000
30	0.800	0.170	0.102	0.307	0.000
31	0.000	0.330	0.092	0.368	0.000
32	0.200	0.330	0.092	0.353	0.000
33	0.400	0.330	0.084	0.338	0.000
34	0.600	0.330	0.080	0.322	0.000
35	0.800	0.330	0.102	0.307	0.000
36	0.000	0.500	0.092	0.368	0.000
37	0.200	0.500	0.088	0.354	0.000
38	0.400	0.500	0.084	0.338	0.000
39	0.600	0.500	0.080	0.322	0.000
40	0.800	0.500	0.102	0.307	0.000
41	0.000	0.000	0.184	0.368	0.000
42	0.200	0.000	0.175	0.352	0.000
43	0.400	0.000	0.169	0.336	0.000
44	0.600	0.000	0.159	0.321	0.000
45	0.800	0.000	0.204	0.307	0.000
46	0.000	0.170	0.184	0.368	0.000

NODE	X-COORD	Y-COORD	U-COORD	POTENTIAL HEAD	CONCENTRATION
47	0.200	0.170	0.177	0.353	0.000
48	0.400	0.170	0.159	0.336	0.000
49	0.600	0.170	0.161	0.322	0.000
50	0.800	0.170	0.204	0.307	0.000
51	0.000	0.330	0.184	0.368	0.000
52	0.200	0.330	0.184	0.354	0.000
53	0.400	0.330	0.168	0.338	0.000
54	0.600	0.330	0.160	0.322	0.000
55	0.800	0.330	0.204	0.307	0.000
56	0.000	0.500	0.184	0.368	0.000
57	0.200	0.500	0.176	0.354	0.000
58	0.400	0.500	0.169	0.338	0.000
59	0.600	0.500	0.161	0.322	0.000
60	0.800	0.500	0.204	0.307	0.000
61	0.000	0.000	0.276	0.368	0.000
62	0.200	0.000	0.263	0.352	0.000
63	0.400	0.000	0.253	0.337	0.000
64	0.600	0.000	0.239	0.322	0.000
65	0.800	0.000	0.307	0.307	0.000
66	0.000	0.170	0.276	0.368	0.000
67	0.200	0.170	0.265	0.353	0.000
68	0.400	0.170	0.239	0.337	0.000
69	0.600	0.170	0.242	0.322	0.000
70	0.800	0.170	0.307	0.307	0.000
71	0.000	0.330	0.276	0.368	0.000
72	0.200	0.330	0.276	0.353	0.000
73	0.400	0.330	0.252	0.338	0.000
74	0.600	0.330	0.240	0.322	0.000
75	0.800	0.330	0.307	0.307	0.000
76	0.000	0.500	0.276	0.368	0.000
77	0.200	0.500	0.264	0.353	0.000
78	0.400	0.500	0.253	0.338	0.000
79	0.600	0.500	0.241	0.322	0.000
80	0.800	0.500	0.307	0.307	0.000
81	0.000	0.000	0.368	0.368	0.000
82	0.200	0.000	0.350	0.352	0.000
83	0.400	0.000	0.337	0.340	0.000
84	0.600	0.000	0.319	0.321	0.000
85	0.800	0.000	0.310	0.307	0.000
86	0.000	0.170	0.368	0.368	0.000
87	0.200	0.170	0.353	0.355	0.000
88	0.400	0.170	0.380	0.323	0.000
89	0.600	0.170	0.323	0.324	0.000
90	0.800	0.170	0.308	0.307	0.000
91	0.000	0.330	0.368	0.368	0.000
92	0.200	0.330	0.368	0.367	0.000
93	0.400	0.330	0.336	0.339	0.000
94	0.600	0.330	0.320	0.322	0.000
95	0.800	0.330	0.309	0.307	0.000
96	0.000	0.500	0.368	0.368	0.000
97	0.200	0.500	0.352	0.353	0.000
98	0.400	0.500	0.338	0.340	0.000
99	0.600	0.500	0.321	0.323	0.000
100	0.800	0.500	0.309	0.307	0.000

ELEMENT	X-VELOCITY	Y-VELOCITY	U-VELOCITY
1	0.0002132	-0.0000022	0.0000000
2	0.0002172	-0.0000046	0.0000025
3	0.0002082	-0.0000037	0.0000036
4	0.0002032	-0.0000013	0.0000009
5	0.0002078	-0.0000044	-0.0000015
6	0.0002159	-0.0000101	-0.0000006
7	0.0002113	-0.0000087	0.0000016
8	0.0002067	-0.0000030	0.0000005
9	0.0002024	-0.0000022	-0.0000029
10	0.0002141	-0.0000053	-0.0000040
11	0.0002147	-0.0000048	-0.0000012
12	0.0002106	-0.0000017	-0.0000001
13	0.0002129	-0.0000034	-0.0000006
14	0.0002229	-0.0000058	0.0000088
15	0.0002047	-0.0000036	0.0000123
16	0.0002015	-0.0000013	0.0000021
17	0.0002040	-0.0000073	-0.0000066
18	0.0002220	-0.0000161	-0.0000022
19	0.0002097	-0.0000127	0.0000065
20	0.0002058	-0.0000038	0.0000012
21	0.0001959	-0.0000026	-0.0000110
22	0.0002181	-0.0000079	-0.0000145
23	0.0002160	-0.0000075	-0.0000039
24	0.0002110	-0.0000023	-0.0000006
25	0.0002104	-0.0000069	-0.0000045
26	0.0002405	-0.0000031	0.0000193
27	0.0001919	-0.0000023	0.0000257
28	0.0002007	-0.0000017	0.0000003
29	0.0001899	-0.0000172	-0.0000233
30	0.0002455	-0.0000374	-0.0000055
31	0.0001998	-0.0000250	0.0000193
32	0.0002060	-0.0000045	-0.0000008
33	0.0001777	0.0000017	-0.0000279
34	0.0002303	-0.0000095	-0.0000345
35	0.0002182	-0.0000143	-0.0000079
36	0.0002127	-0.0000032	-0.0000020
37	0.0002008	-0.0000175	-0.0000150
38	0.0002842	0.0000540	0.0000365
39	0.0001559	0.0000578	0.0000439
40	0.0002082	-0.0000110	-0.0000205
41	0.0001335	-0.0000590	-0.0000951
42	0.0003380	-0.0001406	-0.0000234
43	0.0001520	-0.0000779	0.0000643
44	0.0002183	0.0000022	-0.0000307
45	0.0001341	0.0000562	-0.0000644
46	0.0002585	0.0000427	-0.0000807
47	0.0002233	-0.0000185	-0.0000173
48	0.0002150	-0.0000049	-0.0000086

APPENDIX I Input and output listings for experiment number 2

AQUIFEM DATA FOR RUN - EXPERIMENT NO. 2 (k=0.00276m/s)

NUMBER OF ELEMENTS = 43
NUMBER OF NODES = 102

N	X	Y	U	H	B$	Q	C	C$	S$	CF
1	0.000	0.000	0.310	0.310	Y	0.000000	0.000	N	Y	0.000000
2	0.000	0.170	0.310	0.310	Y	0.000000	0.000	N	Y	0.000000
3	0.000	0.330	0.310	0.310	Y	0.000000	0.000	N	Y	0.000000
4	0.000	0.500	0.310	0.310	Y	0.000000	0.000	N	Y	0.000000
5	0.100	0.250	0.310	0.310	N	0.000000	0.000	N	Y	0.000000
6	0.100	0.330	0.310	0.310	N	0.000000	0.000	N	Y	0.000000
7	0.100	0.415	0.310	0.310	N	0.000000	0.000	N	Y	0.000000
8	0.150	0.290	0.310	0.310	N	0.000000	0.000	N	Y	0.000000
9	0.150	0.330	0.310	0.310	N	0.000000	1.000	C	Y	0.000000
10	0.150	0.373	0.310	0.310	N	0.000000	1.000	C	Y	0.000000
11	0.200	0.000	0.310	0.310	N	0.000000	0.000	N	Y	0.000000
12	0.200	0.170	0.310	0.310	N	0.000000	0.000	N	Y	0.000000
13	0.200	0.250	0.310	0.310	N	0.000000	0.000	N	Y	0.000000
14	0.200	0.290	0.310	0.310	N	0.000000	0.000	N	Y	0.000000
15	0.200	0.330	0.310	0.310	N	-0.005600	1.000	C	Y	0.000000
16	0.200	0.373	0.310	0.310	N	0.000000	1.000	C	Y	0.000000
17	0.200	0.415	0.310	0.310	N	0.000000	0.000	N	Y	0.000000
18	0.200	0.500	0.310	0.310	N	0.000000	0.000	N	Y	0.000000
19	0.250	0.290	0.310	0.310	N	0.000000	0.000	N	Y	0.000000
20	0.250	0.330	0.310	0.310	N	0.000000	0.000	N	Y	0.000000
21	0.250	0.373	0.310	0.310	N	0.000000	0.000	N	Y	0.000000
22	0.300	0.085	0.310	0.310	N	0.000000	0.000	N	Y	0.000000
23	0.300	0.170	0.310	0.310	N	0.000000	0.000	N	Y	0.000000
24	0.300	0.250	0.310	0.310	N	0.000000	0.000	N	Y	0.000000
25	0.300	0.330	0.310	0.310	N	0.000000	0.000	N	Y	0.000000
26	0.300	0.415	0.310	0.310	N	0.000000	0.000	N	Y	0.000000
27	0.350	0.128	0.310	0.310	N	0.000000	0.000	N	Y	0.000000
28	0.350	0.170	0.310	0.310	N	0.000000	0.000	N	Y	0.000000
29	0.350	0.210	0.310	0.310	N	0.000000	0.000	N	Y	0.000000
30	0.400	0.000	0.310	0.310	N	0.000000	0.000	N	Y	0.000000
31	0.400	0.085	0.310	0.310	N	0.000000	0.000	N	Y	0.000000
32	0.400	0.128	0.310	0.310	N	0.000000	0.000	N	Y	0.000000
33	0.400	0.170	0.310	0.310	N	0.003500	0.000	N	Y	0.000000
34	0.400	0.210	0.310	0.310	N	0.000000	0.000	N	Y	0.000000
35	0.400	0.250	0.310	0.310	N	0.000000	0.000	N	Y	0.000000
36	0.400	0.330	0.310	0.310	N	0.000000	0.000	N	Y	0.000000
37	0.400	0.500	0.310	0.310	N	0.000000	0.000	N	Y	0.000000
38	0.450	0.128	0.310	0.310	N	0.000000	0.000	N	Y	0.000000
39	0.450	0.170	0.310	0.310	N	0.000000	0.000	N	Y	0.000000
40	0.450	0.210	0.310	0.310	N	0.000000	0.000	N	Y	0.000000
41	0.500	0.085	0.310	0.310	N	0.000000	0.000	N	Y	0.000000
42	0.500	0.170	0.310	0.310	N	0.000000	0.000	N	Y	0.000000
43	0.500	0.250	0.310	0.310	N	0.000000	0.000	N	Y	0.000000
44	0.600	0.000	0.310	0.310	N	0.000000	0.000	N	Y	0.000000

N	X	Y	U	H	B$	Q	C	C$	S$	CF
45	0.600	0.170	0.310	0.310	N	0.000000	0.000	N	Y	0.000000
46	0.600	0.330	0.310	0.310	N	0.000000	0.000	N	Y	0.000000
47	0.600	0.500	0.310	0.310	N	0.000000	0.000	N	Y	0.000000
48	0.800	0.000	0.302	0.302	Y	0.000000	0.000	N	Y	0.000000
49	0.800	0.170	0.302	0.302	Y	0.000000	0.000	N	Y	0.000000
50	0.800	0.330	0.302	0.302	Y	0.000000	0.000	N	Y	0.000000
51	0.800	0.500	0.302	0.302	Y	0.000000	0.000	N	Y	0.000000
52	0.000	0.000	0.000	0.310	Y	0.000000	0.000	N	N	0.000000
53	0.000	0.170	0.000	0.310	Y	0.000000	0.000	N	N	0.000000
54	0.000	0.330	0.000	0.310	Y	0.000000	0.000	N	N	0.000000
55	0.000	0.500	0.000	0.310	Y	0.000000	0.000	N	N	0.000000
56	0.100	0.250	0.000	0.310	N	0.000000	0.000	N	N	0.000000
57	0.100	0.330	0.000	0.310	N	0.000000	0.000	N	N	0.000000
58	0.100	0.415	0.000	0.310	N	0.000000	0.000	N	N	0.000000
59	0.150	0.290	0.000	0.310	N	0.000000	0.000	N	N	0.000000
60	0.150	0.330	0.000	0.310	N	0.000000	1.000	C	N	0.000000
61	0.150	0.373	0.000	0.310	N	0.000000	1.000	C	N	0.000000
62	0.200	0.000	0.000	0.310	N	0.000000	0.000	N	N	0.000000
63	0.200	0.170	0.000	0.310	N	0.000000	0.000	N	N	0.000000
64	0.200	0.250	0.000	0.310	N	0.000000	0.000	N	N	0.000000
65	0.200	0.290	0.000	0.310	N	0.000000	0.000	N	N	0.000000
66	0.200	0.330	0.000	0.310	N	-0.005600	1.000	C	N	0.000000
67	0.200	0.373	0.000	0.310	N	0.000000	1.000	C	N	0.000000
68	0.200	0.415	0.000	0.310	N	0.000000	0.000	N	N	0.000000
69	0.200	0.500	0.000	0.310	N	0.000000	0.000	N	N	0.000000
70	0.250	0.290	0.000	0.310	N	0.000000	0.000	N	N	0.000000
71	0.250	0.330	0.000	0.310	N	0.000000	0.000	N	N	0.000000
72	0.250	0.373	0.000	0.310	N	0.000000	0.000	N	N	0.000000
73	0.300	0.085	0.000	0.310	N	0.000000	0.000	N	N	0.000000
74	0.300	0.170	0.000	0.310	N	0.000000	0.000	N	N	0.000000
75	0.300	0.250	0.000	0.310	N	0.000000	0.000	N	N	0.000000
76	0.300	0.330	0.000	0.310	N	0.000000	0.000	N	N	0.000000
77	0.300	0.415	0.000	0.310	N	0.000000	0.000	N	N	0.000000
78	0.350	0.128	0.000	0.310	N	0.000000	0.000	N	N	0.000000
79	0.350	0.170	0.000	0.310	N	0.000000	0.000	N	N	0.000000
80	0.350	0.210	0.000	0.310	N	0.000000	0.000	N	N	0.000000
81	0.400	0.000	0.000	0.310	N	0.000000	0.000	N	N	0.000000
82	0.400	0.085	0.000	0.310	N	0.000000	0.000	N	N	0.000000
83	0.400	0.128	0.000	0.310	N	0.000000	0.000	N	N	0.000000
84	0.400	0.170	0.000	0.310	N	0.003500	0.000	N	N	0.000000
85	0.400	0.210	0.000	0.310	N	0.000000	0.000	N	N	0.000000
86	0.400	0.250	0.000	0.310	N	0.000000	0.000	N	N	0.000000
87	0.400	0.330	0.000	0.310	N	0.000000	0.000	N	N	0.000000
88	0.400	0.500	0.000	0.310	N	0.000000	0.000	N	N	0.000000
89	0.450	0.128	0.000	0.310	N	0.000000	0.000	N	N	0.000000
90	0.450	0.170	0.000	0.310	N	0.000000	0.000	N	N	0.000000
91	0.450	0.210	0.000	0.310	N	0.000000	0.000	N	N	0.000000
92	0.500	0.085	0.000	0.310	N	0.000000	0.000	N	N	0.000000
93	0.500	0.170	0.000	0.310	N	0.000000	0.000	N	N	0.000000
94	0.500	0.250	0.000	0.310	N	0.000000	0.000	N	N	0.000000
95	0.600	0.000	0.000	0.310	N	0.000000	0.000	N	N	0.000000
96	0.600	0.170	0.000	0.310	N	0.000000	0.000	N	N	0.000000

N	X	Y	U	H	B$	Q	C	C$	S$	CF
97	0.600	0.330	0.000	0.310	N	0.000000	0.000	N	N	0.000000
98	0.600	0.500	0.000	0.310	N	0.000000	0.000	N	N	0.000000
99	0.800	0.000	0.000	0.302	Y	0.000000	0.000	N	N	0.000000
100	0.800	0.170	0.000	0.302	Y	0.000000	0.000	N	N	0.000000
101	0.800	0.330	0.000	0.302	Y	0.000000	0.000	N	N	0.000000
102	0.800	0.500	0.000	0.302	Y	0.000000	0.000	N	N	0.000000

E	KX	KY	KU	S	P	DX	DY	DU
1	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	52	62	63	53	1	11	12	2
2	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	53	63	64	56	2	12	13	5
3	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	53	56	57	54	2	5	6	3
4	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	54	57	58	55	3	6	7	4
5	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	58	68	69	55	7	17	18	4
6	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	56	64	65	59	5	13	14	8
7	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	56	59	60	57	5	8	9	6
8	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	57	60	61	58	6	9	10	7
9	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	61	67	68	58	10	16	17	7
10	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	59	65	66	60	8	14	15	9
11	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	60	66	67	61	9	15	16	10
12	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	65	70	71	66	14	19	20	15
13	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	66	71	72	67	15	20	21	16
14	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	64	75	70	65	13	24	19	14
15	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	70	75	76	71	19	24	25	20
16	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	71	76	77	72	20	25	26	21
17	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	67	72	77	68	16	21	26	17
18	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	62	81	82	73	11	30	31	22
19	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	62	73	74	63	11	22	23	12
20	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	63	74	75	64	12	23	24	13
21	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	75	86	87	76	24	35	36	25

E	KX	KY	KU	S	P	DX	DY	DU
22	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	76	87	88	77	25	36	37	26
23	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	68	77	88	69	17	26	37	18
24	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	73	82	83	78	22	31	32	27
25	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	73	78	79	74	22	27	28	23
26	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	74	79	80	75	23	28	29	24
27	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	80	85	86	75	29	34	35	24
28	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	78	83	84	79	27	32	33	28
29	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	79	84	85	80	28	33	34	29
30	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	83	89	90	84	32	38	39	33
31	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	84	90	91	85	33	39	40	34
32	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	82	92	89	83	31	41	38	32
33	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	89	92	93	90	38	41	42	39
34	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	90	93	94	91	39	42	43	40
35	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	85	91	94	86	34	40	43	35
36	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	81	95	92	82	30	44	41	31
37	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	92	95	96	93	41	44	45	42
38	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	93	96	97	94	42	45	46	43
39	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	86	94	97	87	35	43	46	36
40	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	87	97	98	88	36	46	47	37
41	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	95	99	100	96	44	48	49	45
42	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	96	100	101	97	45	49	50	46
43	0.002760	0.002760	0.002760	1.000	0.412	0.00000030	0.00000030	0.00000030
	NODES:	97	101	102	98	46	50	51	47

AQUIFEM RESULTS - EXPERIMENT NO. 2 (k=0.00276m/s)

NUMBER OF ELEMENTS = 43
NUMBER OF NODES = 102

STEP NUMBER 72 TIME 64800 SECS NUMBER OF ITERATIONS 1

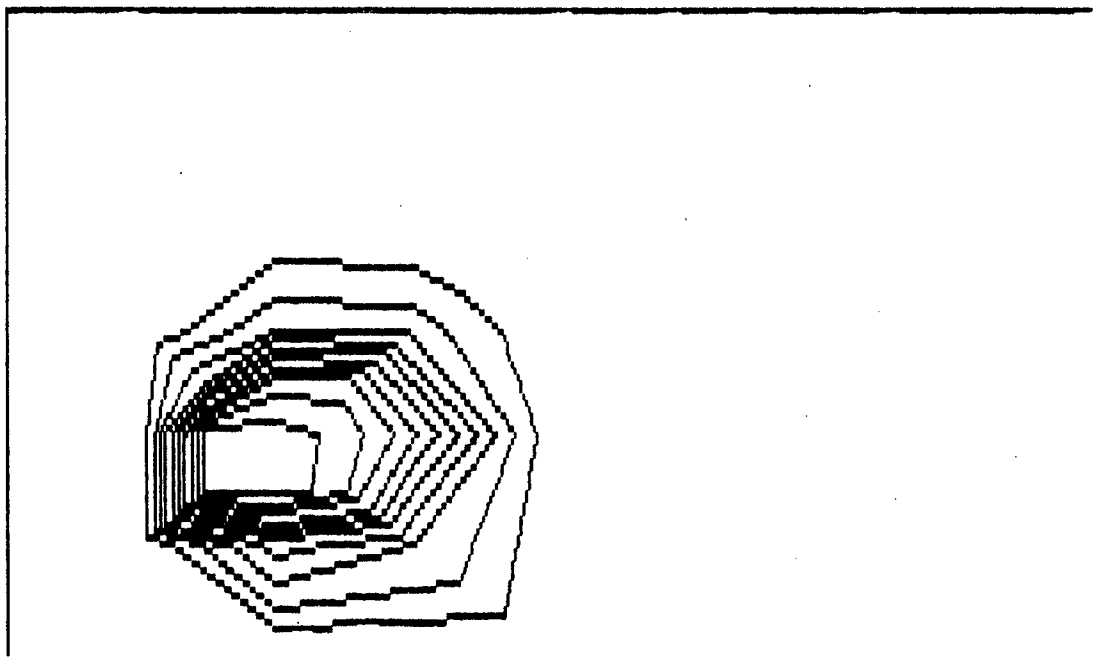
NODE	X-COORD	Y-COORD	U-COORD	POTENTIAL HEAD	CONCENTRATION
1	0.000	0.000	0.310	0.310	0.149
2	0.000	0.170	0.310	0.310	0.128
3	0.000	0.330	0.310	0.310	0.510
4	0.000	0.500	0.310	0.310	-0.020
5	0.100	0.250	0.310	0.309	0.193
6	0.100	0.330	0.310	0.310	0.325
7	0.100	0.415	0.310	0.310	0.131
8	0.150	0.290	0.310	0.310	0.595
9	0.150	0.330	0.310	0.310	1.000
10	0.150	0.373	0.310	0.310	1.000
11	0.200	0.000	0.308	0.308	0.003
12	0.200	0.170	0.308	0.308	0.128
13	0.200	0.250	0.310	0.309	0.523
14	0.200	0.290	0.310	0.309	0.923
15	0.200	0.330	0.312	0.312	1.000
16	0.200	0.373	0.310	0.309	1.000
17	0.200	0.415	0.310	0.309	0.547
18	0.200	0.500	0.310	0.309	0.096
19	0.250	0.290	0.310	0.309	0.940
20	0.250	0.330	0.310	0.309	1.035
21	0.250	0.373	0.310	0.309	0.977
22	0.300	0.085	0.307	0.307	0.043
23	0.300	0.170	0.307	0.307	0.211
24	0.300	0.250	0.307	0.307	0.754
25	0.300	0.330	0.308	0.308	1.066
26	0.300	0.415	0.308	0.308	0.642
27	0.350	0.128	0.306	0.306	0.121
28	0.350	0.170	0.306	0.306	0.267
29	0.350	0.210	0.306	0.306	0.529
30	0.400	0.000	0.306	0.306	0.053
31	0.400	0.085	0.306	0.306	0.061
32	0.400	0.128	0.305	0.305	0.072
33	0.400	0.170	0.304	0.304	0.322
34	0.400	0.210	0.305	0.305	0.629
35	0.400	0.250	0.306	0.306	0.762
36	0.400	0.330	0.306	0.306	0.894
37	0.400	0.500	0.306	0.306	0.347
38	0.450	0.128	0.305	0.305	0.090
39	0.450	0.170	0.305	0.305	0.368
40	0.450	0.210	0.305	0.305	0.662
41	0.500	0.085	0.305	0.305	0.013
42	0.500	0.170	0.305	0.305	0.378
43	0.500	0.250	0.305	0.305	0.818
44	0.600	0.000	0.304	0.304	-0.079
45	0.600	0.170	0.304	0.304	0.405

NODE	X-COORD	Y-COORD	U-COORD	POTENTIAL HEAD	CONCENTRATION
46	0.600	0.330	0.304	0.304	0.808
47	0.600	0.500	0.304	0.304	0.427
48	0.800	0.000	0.302	0.302	-0.009
49	0.800	0.170	0.302	0.302	0.312
50	0.800	0.330	0.302	0.302	0.597
51	0.800	0.500	0.302	0.302	0.412
52	0.000	0.000	0.000	0.310	0.088
53	0.000	0.170	0.000	0.310	0.009
54	0.000	0.330	0.000	0.310	0.082
55	0.000	0.500	0.000	0.310	-0.027
56	0.100	0.250	0.000	0.309	0.009
57	0.100	0.330	0.000	0.309	0.093
58	0.100	0.415	0.000	0.309	-0.004
59	0.150	0.290	0.000	0.308	0.160
60	0.150	0.330	0.000	0.308	0.110
61	0.150	0.373	0.000	0.308	-0.046
62	0.200	0.000	0.000	0.308	-0.006
63	0.200	0.170	0.000	0.308	0.024
64	0.200	0.250	0.000	0.308	0.204
65	0.200	0.290	0.000	0.308	0.485
66	0.200	0.330	0.000	0.306	0.714
67	0.200	0.373	0.000	0.308	0.401
68	0.200	0.415	0.000	0.308	0.116
69	0.200	0.500	0.000	0.308	0.010
70	0.250	0.290	0.000	0.307	0.583
71	0.250	0.330	0.000	0.307	0.715
72	0.250	0.373	0.000	0.307	0.516
73	0.300	0.085	0.000	0.307	0.001
74	0.300	0.170	0.000	0.307	0.079
75	0.300	0.250	0.000	0.307	0.437
76	0.300	0.330	0.000	0.307	0.678
77	0.300	0.415	0.000	0.307	0.267
78	0.350	0.128	0.000	0.307	0.041
79	0.350	0.170	0.000	0.307	0.123
80	0.350	0.210	0.000	0.307	0.295
81	0.400	0.000	0.000	0.306	0.030
82	0.400	0.085	0.000	0.306	0.025
83	0.400	0.128	0.000	0.306	0.020
84	0.400	0.170	0.000	0.307	0.173
85	0.400	0.210	0.000	0.306	0.374
86	0.400	0.250	0.000	0.306	0.474
87	0.400	0.330	0.000	0.306	0.558
88	0.400	0.500	0.000	0.306	0.116
89	0.450	0.128	0.000	0.306	0.042
90	0.450	0.170	0.000	0.306	0.221
91	0.450	0.210	0.000	0.306	0.418
92	0.500	0.085	0.000	0.305	-0.001
93	0.500	0.170	0.000	0.305	0.247
94	0.500	0.250	0.000	0.305	0.545
95	0.600	0.000	0.000	0.304	-0.061
96	0.600	0.170	0.000	0.304	0.279
97	0.600	0.330	0.000	0.304	0.549

NODE	X-COORD	Y-COORD	U-COORD	POTENTIAL HEAD	CONCENTRATION
98	0.600	0.500	0.000	0.304	0.225
99	0.800	0.000	0.000	0.302	-0.012
100	0.800	0.170	0.000	0.302	0.194
101	0.800	0.330	0.000	0.302	0.382
102	0.800	0.500	0.000	0.302	0.227

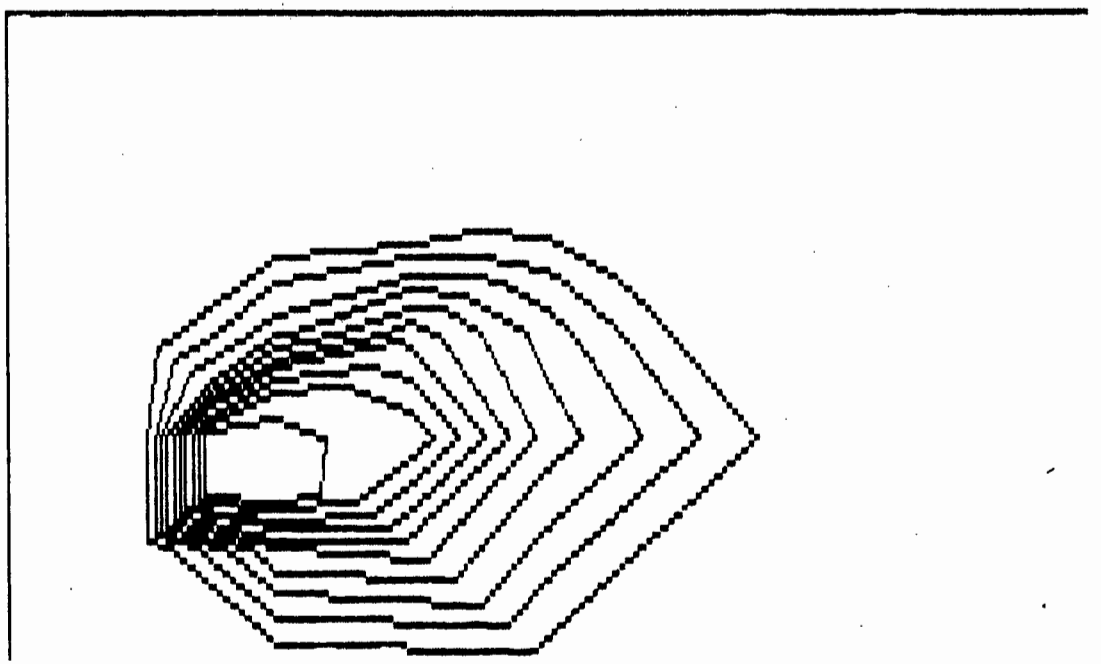
ELEMENT	X-VELOCITY	Y-VELOCITY	U-VELOCITY
1	0.0000265	-0.0000012	-0.0000004
2	0.0000245	-0.0000058	-0.0000034
3	0.0000208	-0.0000012	-0.0000023
4	0.0000194	-0.0000003	-0.0000024
5	0.0000215	0.0000022	-0.0000042
6	0.0000219	-0.0000126	-0.0000098
7	0.0000164	-0.0000033	-0.0000080
8	0.0000160	0.0000004	-0.0000082
9	0.0000213	0.0000083	-0.0000098
10	0.0000014	-0.0000264	-0.0000227
11	0.0000016	0.0000218	-0.0000225
12	0.0000529	-0.0000268	-0.0000226
13	0.0000520	0.0000214	-0.0000225
14	0.0000335	-0.0000134	-0.0000094
15	0.0000383	-0.0000050	-0.0000076
16	0.0000376	-0.0000009	-0.0000082
17	0.0000316	0.0000077	-0.0000099
18	0.0000302	0.0000001	0.0000022
19	0.0000314	-0.0000024	0.0000005
20	0.0000324	-0.0000072	-0.0000026
21	0.0000340	-0.0000059	-0.0000010
22	0.0000332	-0.0000015	-0.0000028
23	0.0000306	0.0000014	-0.0000046
24	0.0000318	0.0000035	0.0000052
25	0.0000354	-0.0000023	0.0000037
26	0.0000365	-0.0000054	0.0000025
27	0.0000342	-0.0000100	0.0000038
28	0.0000451	0.0000119	0.0000128
29	0.0000458	-0.0000180	0.0000125
30	0.0000141	0.0000124	0.0000132
31	0.0000143	-0.0000174	0.0000131
32	0.0000262	0.0000043	0.0000055
33	0.0000232	-0.0000006	0.0000046
34	0.0000237	-0.0000030	0.0000044
35	0.0000271	-0.0000089	0.0000051
36	0.0000267	0.0000010	0.0000024
37	0.0000257	-0.0000003	0.0000017
38	0.0000266	-0.0000019	0.0000015
39	0.0000287	-0.0000046	0.0000014
40	0.0000297	-0.0000013	-0.0000003
41	0.0000266	-0.0000002	0.0000003
42	0.0000272	-0.0000006	0.0000002
43	0.0000281	-0.0000004	0.0000000

1.00
0.90
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10



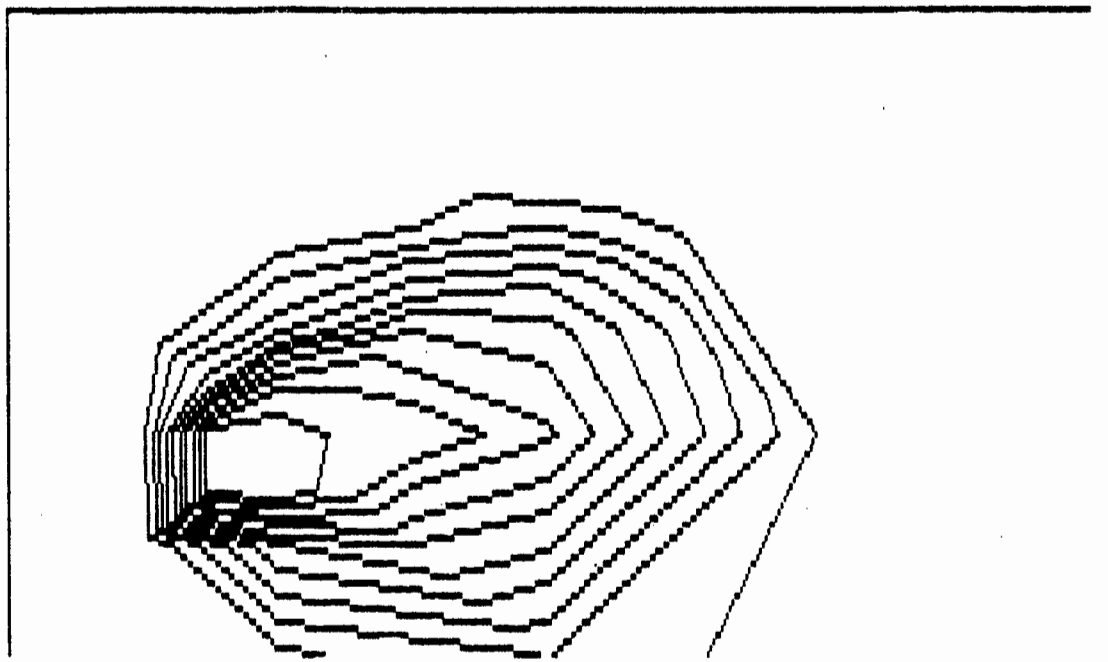
STEP NUMBER 8 PRESS ANY KEY TO CONTINUE ('C' FOR MULTI DISPLAY)

1.00
0.90
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10



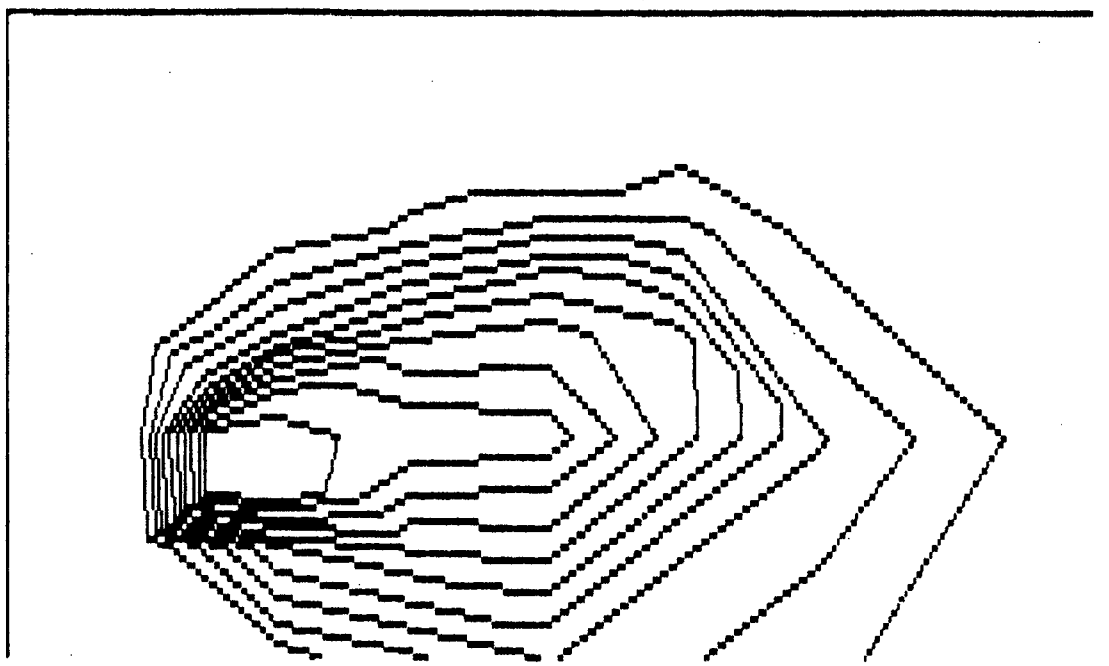
STEP NUMBER 16 PRESS ANY KEY TO CONTINUE ('C' FOR MULTI DISPLAY)

1.00
0.90
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10



STEP NUMBER 24 PRESS ANY KEY TO CONTINUE ('C' FOR MULTI DISPLAY)

1.00
0.90
0.80
0.70
0.60
0.50
0.40
0.30
0.20
0.10



STEP NUMBER 32 PRESS ANY KEY TO CONTINUE ('C' FOR MULTI DISPLAY)