Analytical Solutions Describing Transient, Two-Dimensional Solute Transport in a Two Layer Aquifer System

by

Kent W. Lindberg

Independent Study
Submitted in Partial Fulfillment of the
Requirement for the Degree of Master of Science
in Hydrology

New Mexico Institute of Mining and Technology
Socorro, New Mexico

July 28, 1989

ABSTRACT

The mission of aquifer remediation expands with the growing dependence on groundwater. Analytical solutions are important tools used to characterize the extent of aquifer contamination. Three mathematical models (Models I, II, and III) are developed to study two-dimensional solute transport in a two layer aquifer system. Model I uses an "exact" approach in which two existing layers are coupled at their interface with two boundary conditions, a continuity of concentration and a continuity of solute mass flux. Models II and III employ a "decoupled" approximation approach to find the solutions for solute concentration distributions in the system. This approach assumes that the transport phenomena in layer *1* are not affected by the presence of layer *2*; the thickness of the first layer in the direction of groundwater flow is mathematically set to infinity. The second layer is then imposed upon the flow domain of the first layer at some finite distance from the solute source and the solution to solute transport in the first layer is used as the inlet concentration for the second layer. At the interface of layers *1* and *2*, Model II assumes the continuity condition of solute mass flux while Model III of concentration. Models II and III were conceived for their mathematical simplicity and are approximations to the results of Model I. The Laplace transform and the Fourier cosine transform are used to determine the solutions for the three models. Analysis is carried out at the laboratory scale and at the field scale. The three models yield virtually the same results if the dispersion coefficients of layers *1* and *2* are the same order of magnitude. If the dispersion coefficients of layers *1* and *2* are significantly different (e.g. a few orders of magnitude difference), the results of Models I and II are almost identical in both layers except within a small region in layer *1* near the interface. Model III, however, fails to give accurate results in layer *2* when the dispersion contrast between the two layers is large. Due to its less complicated mathematics and overall accuracy (except in a small region in layer *1* near the interface), Model II is suggested for use in lieu of Model I when dealing with the problem of interest.

ACKNOWLEDGEMENTS

Funds for this research were provided by the Hantush Fellowship in the hydrology program of New Mexico Tech, the National Science Foundation grant ECE-8696079 to Dr. Chen, and the U.S. Environmental Protection Agency through the Robert S. Kerr Environmental Research Laboratory as part of a research project (CR-813529). This study has not been subjected to the Agency's required peer and policy review and therefore does not necessarily reflect the views of the Agency and no official endorsement should be inferred.

The following individuals contributed to this research: Annette Schafer-Perini and Dr. Bill Stone.

ACKNOWLEDGEMENTS

The author wishes to thank Dr. Chia-Shyun Chen for his assistance, support, and guidance. Financial support for this work was provided by the Hantush Fellowship in the hydrology program of New Mexico Tech, the National Science Foundation grant ECE-8696079 to Dr. Chen, and the U.S. Environmental Protection Agency through the Robert S. Kerr Environmental Research Laboratory as part of a research project (CR-813529). This study has not been subjected to the Agency's required peer and policy review and therefore does not necessarily reflect the views of the Agency and no official endorsement should be inferred. The assistance of Annette Schafer-Perini of the Hydrology Department and Dr. Bill Stone of the Mathematics Department is also greatly appreciated. I wish to thank my friends at New Mexico Tech and my family for their encouragement.

Table of Contents

List of Figures

List of Tables

Nomenclature

B, thickness of flow domain (m)

D_{L_i}, longitudinal dispersion coefficient for the *i*th layer (m^2/d)

D_{T_i}, transverse dispersion coefficient for the *i*th layer (m^2/d)

L, distance from solute source to interface between the first and second layers (m)

M, amount of input solute mass (kg)

n_i, porosity in the *i*th layer

t, time since injection of solute (d)

v_i, seepage velocity in the *i*th layer (m/d)

x, distance from solute source perpendicular to layering (m)

y, distance from solute source parallel to layering (m)

α_{L_i}, longitudinal dispersivity in the *i*th layer (m)

α_{T_i}, transverse dispersivity in the *i*th layer (m)

The subscripts, $i = 1,2$, refer to the first and second layers, respectively

1: INTRODUCTION

Groundwater contamination is an ever-increasing concern as our society's dependence on groundwater grows. Most groundwater is stored in aquifers, saturated permeable geologic units transmitting significant quantities of water. As the world's population increases so does the need for reliable sources of clean, uncontaminated fresh water. It is imperative that we prevent further contamination of our groundwater supplies and remediate existing contamination. Once a contaminant, miscible with water, moves through the soil and reaches the water table it diffuses through the aquifer, traveling with the groundwater and dispersing among the grains of the porous medium. At this stage removal of existing contamination requires not only elimination of the contaminant source, but extraction of the chemicals present in the groundwater. Before the physical removal of the contaminated groundwater can be initiated, however, the extent and severity of the contaminant plume must be estimated. Mathematical models of solute transport in porous media are useful tools in approximating the size and location of a contaminant plume.

Both analytical and numerical models have been used to characterize groundwater contamination. Analytical solutions can be very useful for calibrating numerical models. Numerical models are excellent tools for incorporating sophisticated boundary conditions and intricate flow heterogeneities. Though it is possible to incorporate these detailed aspects of fluid flow and the movement of solutes into numerical models, many times, unfortunately, not enough field data is available to feed sophisticated numerical models. Depending on the sophistication of these models, they can be very costly to execute. As opposed to numerical models, analytical techniques can provide adequate results with limited input data requirement for many aquifer remediation problems and are generally much less expensive to implement.

Many analytical solutions dealing with simple flow conditions such as uniform groundwater flow through a homogeneous, isotropic porous media are available. One-dimensional solutions have been previously derived by Ogata and Banks [1961], Gershon and Nir [1969], Bear [1972], and Kreft and Zuber [1978]. Wilson and Miller [1978], Lenda and Zuber [1970], and Bear [1972] discuss analytical solutions in multiple dimensions for uniform flow fields. Many analytical or approximate solutions are also available for solute transport under radial flow conditions as could be caused by pumping or injection problems. Since these solutions deal with nonuniform groundwater flow fields, which are different from the flow conditions encountered in the current study, they are not reviewed here. Useful information on these radial dispersion solutions can be found in Chen [1986, 1989].

Of greater interest to this study is the previous research on layered aquifer systems. Al-Niami and Rushton [1979] gave a solution to one-dimensional flow across

three layers. Selim et al [1977] compared the solution of an equation describing one-dimensional flow across three layers to experimental results. They included linear and nonlinear adsorption in their mathematical model. Barry and Parker [1987] gave a one-dimensional Laplace domain solution to radioactive solute transport with flow transverse to layering. One-dimensional flow across multiple layers was also investigated by Gureghian and Jansen [1985]. Their solution to solute transport included the effects of radioactive decay and adsorption on a contaminant plume resulting from both continuous injection and band release sources. Shamir and Harleman [1967] presented a solution to one-dimensional flow perpendicular to layering. To find this solution, they initially extended the first layer to infinity. This first layer solution is used to give the inlet boundary concentration for the succeeding layer.

Often a one-dimensional model is not sufficient to analyze the movement of a solute. The ability to model solute transport in multiple dimensions is important since contamination rarely moves in only one direction. Although the flow domain may be in one direction, the solute can spread perpendicular to flow by the so-called transverse dispersion mechanism as it propogates through the medium. Multi-dimensional solutions are more widely applicable to transport problems encountered in the field because of the fact that a solute will disperse in any direction that a concentration gradient is present. Domenico and Robbins [1984] studied the importance of modeling multi-dimensional transport with multi-dimensional solutions. They observed a scaling effect of the dispersion coefficients when an $(n\text{-}1)$-dimensional model is used to describe an n-dimensional system. If, for example, a two-dimensional contaminant plume were modeled using a one-dimensional model the dispersion coefficient of the one-dimensional model would have to be increased by a scale factor in order for the breakthrough curve predicted by the model to fit the concentration curve generated from the field data. The dispersion coefficient used in the model will necessarily vary with time and space even though the dispersive nature of the actual contaminant plume may be constant in time and space. With this analysis Domenico and Robbins [1984] have shown a need for two- and three-dimensional analytical solutions to help broaden our understanding of solute transport.

A few papers are available concerning multi-dimensional solute transport in stratified porous media. Verruijt [1971] gave a solution for the dispersion of a solute across an interface between two aquifer layers where the velocity of the groundwater is parallel to the layers. Van Duijn and van der Zee [1986] approximated an analytical solution for two-dimensional flow parallel to two infinitely thick layers. Valocchi [1988] studied the effects of adsorption on a vertically averaged contaminant plume flowing through and parallel to stratified aquifers that resulted from an instantaneous mass input. Al-Niami and Rushton [1979] provided a solution to two-dimensional flow parallel to two layers; Shamir and Harleman [1967] also presented a solution with the

same flow domain. Some analytical solutions for solute transport through heterogeneous media were given by (e.g. Güven et al [1984]). The soil heterogeneity was incorporated into those solutions by using different types of hydraulic conductivity variations. It is noted that many analytical solutions are available in literature for one-dimensional solute transport through homogeneous or layered soil systems, or multi-dimensional solute transport in single layer, homogeneous soil systems. Little attention has been paid to multi-dimensional solute transport in layered soil systems.

In this study, analytical or semi-analytical solutions for two-dimensional solute transport in a two layered soil system are determined. Those solutions have practical application for many field problems. For example, many pipes or conduits carrying contaminated fluids exist in this country. Leakage from those pipes/conduits can contaminate the soils below them. This contamination problem can be conceptualized as the problem depicted in Figure *1*, where the pipe/conduit is simulated as a line source. A vertical flow field is considered, which, for example, may arise from seepage due to rainfall. A two layered soil sytem is employed to represent the heterogeneous conditions in the simplest form.

The rigorous mathematical approach in modelling the problem of interest treats layer *1* as a semi-infinite domain attached to layer *2*, which for practical purposes can be considered to have a semi-infinite domain as well. The outlet of layer *1* acts as the inlet of layer *2*. Thus, two boundary conditions are required at the interface between layers *1* and *2*; one for the outlet boundary condition for layer *1* and the other for the inlet boundary condition for layer *2*. Normally, these two boundary conditions can be appropriately prescribed by invoking the continuity principles of solute mass flux and concentration. To do so, determination of the analytical solutions becomes difficult and the solutions obtained may have little practical usefulness. Therefore, in this study two different approximate solutions are determined, which are of simpler mathematics. In these approximate solutions, layer *1* is mathematically extended to infinity such that its outlet boundary condition is not important. Consequently, only one boundary condition for the inlet to layer *2* is needed at the interface. This inlet boundary condition can be either the continuity principle of solute mass flux or concentration (since only one of the two required principles is satisfied at the interface, solutions so determined are termed approximate solutions here). Although the approximate solutions are easier to determine and use, it is very important to know which one yields better accuracy. Therefore, the major purpose of this study is to find the conditions under which the approximate solutions can be used to study the problem of interest in lieu of the exact solution.

2: PURPOSE AND SCOPE

The specific purposes of this study are (1) to find the exact and two approximate solutions for the problem illustrated in Figure *1*, and (2) to determine the conditions under which the approximate solutions can be reasonably used in lieu of the exact solution which is mathematically much more complicated and difficult to use. Three models (Models I, II, and III) are developed to achieve the specific goals. Model I relates to the exact approach; that is, layer *1* is considered to be semi-infinite and both continuity of solute mass flux and concentration principles are imposed at the interface between layers *1* and *2*. Model II pertains to the approximate solution where only the continuity of solute mass flux is prescribed at the interface. Model III relates to the approximate solution where only the continuity of concentrations is used at the interface. In the three models, each layer is considered to be homogeneous and isotropic. The longitudinal and transverse dispersivities are assumed to be constant in each layer, and a uniform groundwater flow field exists in both layers.

3: METHODOLOGY

3.1: Model Development

As shown in Figure *1*, the problem of interest is related to the study of a contaminant plume emanating from an instantaneous line source moving with the groundwater flow into a two layered system. Solutes in this plume can be transmitted by advection as caused by the groundwater flow and by mechanical dispersion; molecular diffusion is neglected. In both layers, longitudinal (the direction of groundwater flow) and transverse (the direction normal to groundwater flow) dispersion are taken into consideration. A solute miscible with water is carried through the layering via advection and dispersion and parallel to the layers by dispersion only. The layers are infinite in extent in the direction normal to flow. Both layers are considered to be semi-infinite but the concentration source is a finite distance away from the interface between the first and second layer. Each layer is homogeneous and isotropic throughout.

A partial differential equation, and boundary and initial conditions can now be created to mathematically preserve the details of this model. We begin with the partial differential equation governing the space, time distribution of a conservative, nonradioactive tracer in a homogeneous, isotropic aquifer

$$D_{L_i}\frac{\partial^2 C_i}{\partial x^2} + D_{T_i}\frac{\partial^2 C_i}{\partial y^2} - v_i\frac{\partial C_i}{\partial x} = \frac{\partial C_i}{\partial t} \qquad i = 1,2 \tag{1}$$

where the subscript $i = 1$ indicates the first layer and $i = 2$ indicates the second. Equation (1) is used in all three models to develop the solutions to solute transport for the two layers separately. The three models differ by the use of different boundary conditions at the layer interface. The parameters used in this equation are defined in the nomenclature section. The x-axis is in the direction of flow and normal to stratification, the y-axis is parallel to layering, and t is the time variable. The seepage velocity and dispersion coefficients are assumed to be constant.

In this study the mechanical dispersion is assumed to be the dominant term and molecular diffusion is neglected so the dispersion parameter is equal to

$$D_{L,T} = \alpha_{L,T}\, v \tag{2}$$

This assumption is valid provided the mechanical dispersion is appreciably larger than the molecular diffusion term; this will be the case for most flow regimes where $|v_i| > 0$.

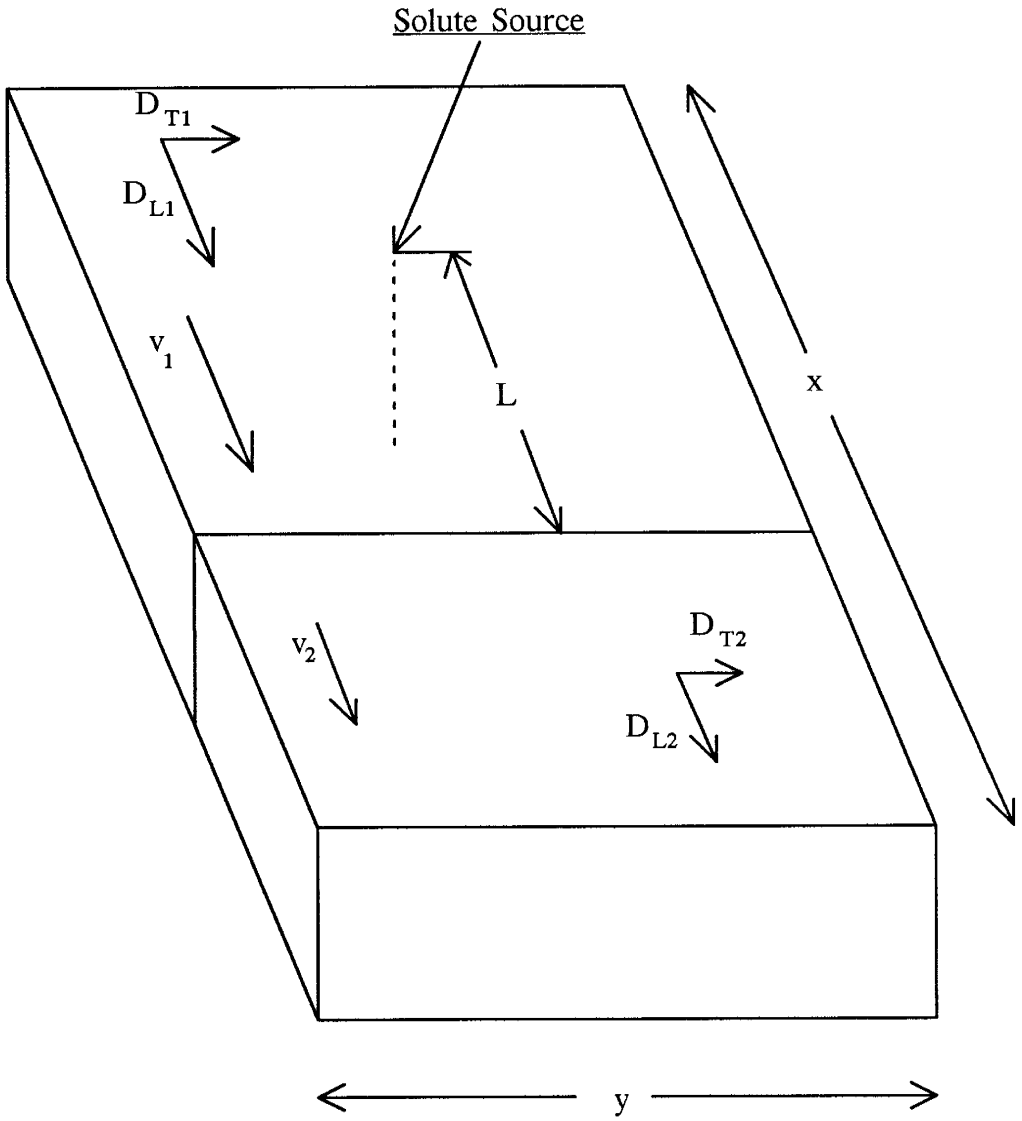

Figure 1. Conceptual model of solute transport through a two layer aquifer system.

Three different mathematical formulations used to characterize the same conceptual model are defined in this section and its subsections by using the partial differential equation given as (1) and different boundary conditions at the interface between the layered media. Each model is comprised of two parts; each part of the mathematical model governs solute transport in one layer or the other. For the first model (Model I) each layer is considered to be semi-infinite and each layer becomes coupled with the adjoining layer via the use of two boundary conditions: a continuity of concentration condition and a continuity of solute mass flux condition. The continuity of concentration boundary condition is the most widely used due to its simplicity. However, this condition does not force a mass balance of solute across the interface. The continuity of solute mass flux boundary condition is more difficult to implement since it is a third-type (Cauchy) boundary condition. This third-type boundary condition does preserve solute mass moving across the interface but does not guarantee that the concentration of solute in the two different layers will be continuous over the boundary between the layers.

A technique used by Shamir and Harleman [1967] to model solute transport perpendicular to layering has been applied in the last two of three models (Models II and III). Their technique initially considers the first layer to be infinite. The solution for this infinite layer is used to calculate the space, time-dependent input concentration at the inlet boundary for the second layer, which is superimposed upon the domain of the first layer, thus requiring only one boundary condition at the interface. Model II imposes the continuity of solute mass flux boundary condition at the interface while Model III imposes the continuity of concentration condition. In all models advantage is taken of the fact that the concentration of the solute in the system is symmetric about an imaginary line passing through the point of mass input and normal to the layer stratification. With this in mind a boundary condition imposing a zero concentration gradient in the y-direction at this imaginary line can be applied. As a result of this only one half of the infinite extent of the layers in the y-direction is considered in the mathematical characterization.

Most of the boundary conditions used by the three models are identical; the models differ only by the use of different boundary conditions between the two layers. The initial and boundary conditions that are common among the three models are given here as

Initial Condition: (first layer)

$$C_1(x,y,0) = \frac{M}{n_1 B}\,\delta(x)\,\delta(y) \tag{3a}$$

Boundary Conditions: (first layer)

$$C_1(x, y, t) = 0 \qquad x \rightarrow -\infty \tag{3b}$$

$$\frac{\partial C_1}{\partial y} = 0 \qquad at\ y = 0 \tag{3c}$$

$$C_1(x, y, t) = 0 \qquad y \rightarrow \infty \tag{3d}$$

Initial Condition: (second layer)

$$C_2(x, y, 0) = 0 \tag{4a}$$

Boundary Conditions: (second layer)

$$C_2(x, y, t) = 0 \qquad x \rightarrow \infty \tag{4b}$$

$$\frac{\partial C_2}{\partial y} = 0 \qquad at\ y = 0 \tag{4c}$$

$$C_2(x, y, t) = 0 \qquad y \rightarrow \infty \tag{4d}$$

where $\delta(x)$ and $\delta(y)$ are Dirac delta functions. The initial condition, (3a), models a mass of solute located along the line, $x=y=0$, at a time, $t=0$. It will be stated here without proof that the solution to (1) with boundary conditions (3) and (4) provides the same solution as equation (1) with boundary conditions (3c) and (4c) written as

$$C_{1,2}(x,y, t) = 0 \qquad y \rightarrow -\infty \tag{5}$$

3.1.1: Model I

In this model two semi-infinite layers are present with the interface between them residing a distance, L, from the contaminant source. A situation exists in this model where two solutions governing transport in separate layers are solved

simultaneously and coupled together through two boundary conditions imposed at the interface. The continuity of concentration and mass flux boundary conditions are both imposed at the layer interface. This model applies the following boundary conditions (including the common initial and boundary conditions given in section 3.1):

Boundary Condition: (first layer)

$$C_1(L, y, t) = C_2(L, y, t) \tag{6a}$$

Boundary Condition: (second layer)

$$n_1\left[v_1 C_1 - D_{L_1}\frac{\partial C_1}{\partial x}\right] = n_2\left[v_2 C_2 - D_{L_2}\frac{\partial C_2}{\partial x}\right] \qquad at\ x = L \tag{6b}$$

In (6b), the terms $n_1 v_1 C_1$ and $n_2 v_2 C_2$ represent the mass flux due to advection in layers *1* and *2*, respectively. The rest of the terms represent the mass flux due to dispersion in the different layers.

3.1.2: Model II

For Models II and III the simplification used by Shamir and Harleman [1967] is applied. Initially, a solution is determined for an infinite, homogeneous first layer; this requires that no boundary condition be needed at the forthcoming layer interface. A second layer is then superimposed upon the domain a finite distance from the solute source. The space, time-dependent solution to concentration in the first layer is used as the inlet solute concentration for the second layer. Model II uses the continuity condition of mass flux alone at the layer interface. By the principle of mass balance, the mass concentration flux out of the first layer has to be equal to the mass concentration flux into the second layer. The following boundary conditions (along with the common initial and boundary conditions given in section 3.1) are used for Model II:

Boundary Condition: (first layer)

$$C_1(x, y, t) = 0 \qquad x \to \infty \tag{7a}$$

Boundary Condition: (second layer)

$$n_1\left[v_1 C_1 - D_{L_1}\frac{\partial C_1}{\partial x}\right] = n_2\left[v_2 C_2 - D_{L_2}\frac{\partial C_2}{\partial x}\right] \qquad at\ x = L \tag{7b}$$

Use of the prescribed mass flux (Cauchy) boundary condition in Models I and II requires that the solute concentration being modeled be considered to be a resident or volume-averaged concentration.

3.1.3: Model III

This model uses the same simplification described in section 3.1.2. The boundary condition at the interface between the two layers for this model is a continuity of concentration condition. Besides the common initial and boundary conditions given in section 3.1 this model is characterized by the following boundary conditions:

Boundary Condition: (first layer)

$$C_1(x, y, t) = 0 \qquad x \rightarrow \infty \tag{8a}$$

Boundary Condition: (second layer)

$$C_1(L, y, t) = C_2(L, y, t) \tag{8b}$$

3.2: Transformation Techniques

Integral transforms are applied to the partial differential equation given in (1) to convert it to an ordinary differential equation, an equation for which a solution can be found. An integral transform takes a function of some variable, t, and multiplies it by some function of t and some other parameter, p. The definite integral of the new function of t and p is taken with respect to t to leave a new function of p alone. The two transforms used were chosen to fit this particular equation and its unique set of boundary conditions. The application of the transforms to the partial differential equation reduces the dimensionality of the equation producing an ordinary differential equation that can be solved directly. Both the Laplace transform and a type of Fourier transform were used to solve equation (1).

3.2.1: Laplace transform and inversion

The Laplace transform changes a function, $f(t)$, into a function, $F(p)$, with the integral transform

$$F(p) = \int_0^\infty f(t) e^{-pt}\, dt \tag{9}$$

The Laplace transform is used in this analysis to transform the time variable, t, to a parameter, p. It is desirable when applying the Laplace transform to the time variable that the concentration, C_i , be continuous over time and equal to zero if time were negative; negative time corresponds to any time before the solute source was introduced to the system. The assumption is made that there is no concentration of solute inherent in the system prior to the point injection or, if there is, that it is constant throughout the aquifer and that this amount of background concentration can be subtracted from all measurements. The Laplace domain concentration will be defined as

$$\overline{C_i}(x, y, p) = \int_0^\infty C_i(x,y,t)\, e^{-pt}\, dt \qquad t \rightarrow p, \qquad i = 1,2 \tag{10}$$

Since the Laplace transform with respect to time has no dependence on the spatial coordinates, x and y, the partial and second partial derivatives of C_i with respect to these variables are not affected by the transformation from C_i to $\overline{C_i}$. The Laplace transform of the partial derivative of C_i with respect to time is needed to effect the transformation of the partial differential equation into the Laplace domain. It is shown in literature (e.g. Boas [1983]) that by using the definition of the Laplace transform (10) and integration by parts the Laplace transform of the partial derivative with respect to time results in

$$\int_0^\infty \frac{\partial C_i(x, y, t)}{\partial t} e^{-pt}\, dt = p\, \overline{C_i}(x,y,p) - C_i(x, y, 0) \tag{11}$$

where

$$C_1(x, y, 0) = \frac{M}{n_1 B}\, \delta(x)\, \delta(y)$$

$$C_2(x, y, 0) = 0$$

The shifting property of Laplace transforms will be needed and it can be stated as (Spiegel [1965])

$$F(p + a) = \int_0^\infty e^{-at} f(t) e^{-pt}\, dt \tag{12}$$

where a is a constant. A Laplace transform identity which will also be necessary for the solution determination is given as (Oberhettinger and Badii [1973])

$$\int_0^\infty t^{-1/2} \exp\left[-at - \frac{b}{t} - pt \right] dt = \left(\frac{\pi}{p+a} \right)^{1/2} \exp\left[-2\sqrt{b\,(p+a)} \right] \tag{13}$$

where a and b are two constants.

Once the solutions for the transformed model in the Laplace domain are known, they need to be inverted back to the real time domain so that the "true" solutions for the original model can be acquired. The Laplace inversion of $F(p)$ is formally defined, by the Mellin inversion theory (Spiegel [1965]) as

$$f(t) = L^{-1}[F(p)] = \frac{1}{2\pi i} \int_{c-i\infty}^{c+i\infty} F(p) e^{pt}\, dp \tag{14}$$

where i denotes the complex number of $\sqrt{-1}$, and c is a positive real number so chosen that all the singularities of the integrand lie left of the vertical line defined by $c - i\infty$ and $c + i\infty$ on the complex plane. The calculation of (14) usually makes use of the Gaussian theorem for complex integrations. However, many useful inversion formulae have been tabulated for practical use (e.g. Oberhettinger and Badii [1973], Bateman [1954]). The following inversion formulae are given in those references and are useful for the current study.

$$L^{-1}\left[\frac{\exp\left[-a\sqrt{p} \right]}{\sqrt{p}} \right] = \frac{\exp\left[-\frac{a^2}{4t} \right]}{\sqrt{\pi t}} \tag{15a}$$

$$L^{-1}\left[\exp\left[-a\sqrt{p} \right] \right] = \frac{a}{2\sqrt{\pi}\, t^{3/2}} \exp\left[-\frac{a^2}{4t} \right] \tag{15b}$$

$$L^{-1}\left[\frac{\exp\left[-a\sqrt{p}\right]}{b+\sqrt{p}}\right] =$$

$$\frac{1}{\sqrt{\pi t}}\exp\left[-\frac{a^2}{4t}\right] - b\exp\left[ab + b^2 t\right] Erfc\left(\frac{a}{2\sqrt{t}} + b\sqrt{t}\right) \qquad (15c)$$

where $Erfc(x)$ is the complimentary error function of the argument, x, and a and b are constants.

The shifting and convolution properties of Laplace inversion are also useful. The shifting property for Laplace inversion states (Spiegel [1965])

$$L^{-1}\left[F(p+a)\right] = e^{-at} f(t) \qquad (16a)$$

where a is constant. The convolution property indicates that

$$L^{-1}[F_1(p)F_2(p)] \equiv L^{-1}[F_1] * L^{-1}[F_2] = \int_0^t f_1(u) f_2(t-u)\, du \qquad (16b)$$

where the symbol, *, denotes the convolution operation, and f_1 and f_2 are the known inversions of F_1 and F_2, respectively. These formulae will be put to use in section 3.3 for determining the solutions to solute transport for the three models.

Since the functions involved in the solutions (a separate solution for each layer) to Model I present difficulties in the Laplace inversion procedure if the Mellin inversion theory or the tabulated inversion formulae are used, inversion of the Laplace domain solutions to Model I was achieved using two different numerical inversion techniques separately, the Stehfest and Crump methods. The Stehfest [1970] method treats the Laplace domain variable for time, p, as a real number in computation and the Laplace inversion is approximated as

$$f(t) \approx p \sum_{n=1}^{N} W_n F(np); \qquad p = \frac{\ln(2)}{t} \qquad (17)$$

where $f(t)$ is the desired Laplace inversion of $F(p)$. In (17) the weighting factors, W_n, can be determined by the following formula given by Stehfest [1970]:

$$W_n = (-1)^{n+N/2} \sum_{k=\frac{(n+1)}{2}}^{\min(n,N/2)} \frac{k^{N/2}\,(2k)!}{(N/2-k)!\,k!\,(k-1)!\,(n-k)!\,(2k-n)!} \tag{18}$$

where the number of weighting factors, N, must be even. Twenty-two weighting factors were used for the Stehfest inversion. The Stehfest method was implemented by applying the routine given by Stehfest [1970].

The Crump [1976] inversion method treats the Laplace domain variable, p, as a complex number and the transformed function is expressed in a Fourier series which approximates the inversion of $F(p)$ as

$$f(t) \approx \frac{e^{at}}{T}\left[\frac{F(a)}{2} + \sum_{k=1}^{\infty}\{\,\mathbf{Re}\left[F\left(a + \frac{ik\pi}{T}\right)\right]\cos\left(\frac{k\pi t}{T}\right) - \mathbf{Im}\left[F\left(a + \frac{ik\pi}{T}\right)\right]\sin\left(\frac{k\pi t}{T}\right)\}\right] \tag{19}$$

where **Re** and **Im** denote the use of the real or imaginary partss of the complex value of the function, F, and $2T$ is the period of the approximation to the function, $f(t)$. The Crump method was executed by calling the DINLAP subroutine of the IMSL computer library [1987].

The accuracy of the Crump and Stehfest methods can be analyzed by applying them to a Laplace domain solution for which the solution is known in the real time domain. Models II and III will be used to test the Crump and Stehfest method's accuracy because their solutions in real time exist (they are derived in section 3.3) and because their solutions are similar to the solution to Model I for which the numerical inversion methods will be used. The transport parameters corresponding to Case *3* given in Table *4* are arbitrarily chosen to portray the accuracy of the numerical inversion methods. Figure *2* shows the actual results of the solution to Model II given as (60) and the results of applying the Crump and Stehfest methods to (53) (the second layer, Laplace domain solution for Model II). Figure *3* shows the actual results of the solution to Model III given as (68) and the results of applying the Crump and Stehfest methods to (62) (the second layer, Laplace domain solution to Model III). It is noted from Figures *2* and *3* that the Stehfest method yields more accurate results, and thus it will be used in the following work.

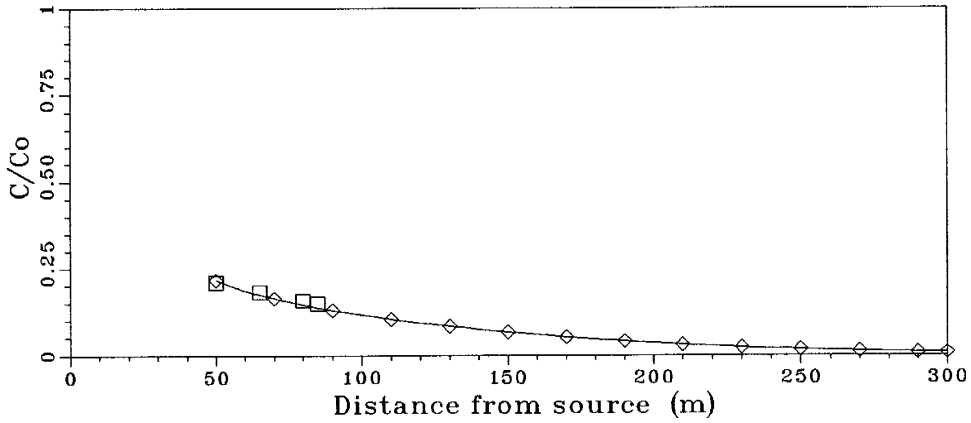

Figure *2.* Comparison of Numerical Laplace inversion methods with the actual results from Model II using Case *3* parameters from Table *4* (section 4.2). The solid line represents the actual solution to Model II given as (60). The diamonds represent the Stehfest method approximation to the Laplace domain solution (52) and the squares, the Crump method approximation.

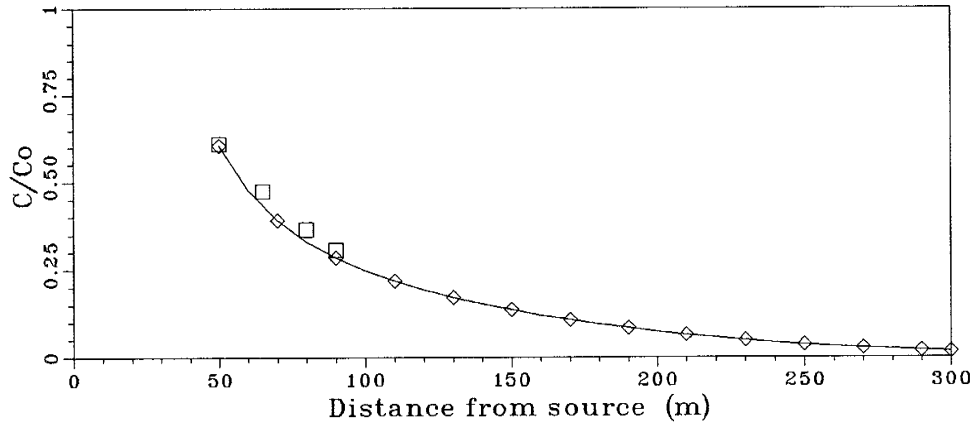

Figure *3.* Comparison of Numerical Laplace inversion methods with the actual results from Model III using Case *3* parameters from Table *4* (section 4.2). The solid line represents the actual solution to Model II given as (68). The diamonds represent the Stehfest method approximation to the Laplace domain solution (61) and the squares, the Crump method approximation.

3.2.2: Fourier cosine transform and inversion

The Fourier integral transform is another method which performs the same operation as the Laplace transform only that a different kernel, the term containing

both variables that is multiplied by the function, is used. The Fourier transform is defined as

$$F(s) = \frac{1}{2\pi} \int_{-\infty}^{\infty} f(y) e^{-isy}\, dy \quad y \rightarrow s \tag{20}$$

Since the function of solute concentration of this particular study is an even, symmetrical function about the x-axis $(C(x,y,t) = C(x, -y,t))$, a simplified transform called the Fourier cosine transform can be applied. It is defined as

$$F(s) = F_c[f(y)] = \left(\frac{2}{\pi}\right)^{1/2} \int_{0}^{\infty} f(y) \cos(sy)\, dy \quad y \rightarrow s \tag{21}$$

The Fourier cosine transform of $\overline{C_i}$ will be given the symbol, G_i. The Fourier cosine integral transform is used to further reduce the dimensionality by decoupling the governing equation from any dependence on y. Two Fourier cosine transform formulae are useful for this current study. First, Sneddon [1951] shows the second partial derivative of $\overline{C_i}$ with respect to y transforms to

$$\left(\frac{2}{\pi}\right)^{1/2} \int_{0}^{\infty} D_{T_i} \frac{\partial^2 \overline{C_i}}{\partial y^2} \cos(sy)\, dy = -s^2 D_{T_i} G_i - D_{T_i} \left(\frac{2}{\pi}\right)^{1/2} \frac{\partial \overline{C_i}}{\partial y}_{y=0} \quad i = 1,2 \tag{22}$$

Second is the transform of the Dirac delta function.

$$\left(\frac{2}{\pi}\right)^{1/2} \int_{0}^{\infty} \delta(y) \cos(sy)\, dy = \frac{1}{\sqrt{2\pi}} \tag{23}$$

The second formula assumes that

$$\int_{0}^{\infty} \delta(y) \cos(sy)\, dy = \frac{1}{2} \tag{24}$$

This appears to be valid considering the Dirac delta and cosine functions are even

functions about the x-axis and that the following equation is an identity of the delta function:

$$\int_{-\infty}^{\infty} \delta(y)\cos(sy)\,dy = 1 \tag{25}$$

The Fourier cosine inversion is much simpler than the Laplace inversion due to its symmetry. The inverse Fourier cosine inversion is defined as

$$f(y) = F_c^{-1}[F(s)] = \left(\frac{2}{\pi}\right)^{1/2} \int_0^{\infty} F(s)\cos(sy)\,ds \tag{26}$$

With this particular integral transform, the kernel is the same for inversion as it was for transformation. No inversion formula, per se, need be defined for the Fourier cosine inversion.

3.3: Solution Determination and Evaluation

The primary task is to "decouple" the governing equation (1) from its dependence on t and y so that a solution to it can be recognized. Equation (1) is first modified using the Laplace transform which removes the dependence on time from the governing equation. Making use of inversion formula given as (11) and applying the initial conditions given as (3a) and (4a) the equation can be rewritten as (27a) for the first layers of the three models and as (27b) for the second layers.

$$D_{L_1}\frac{\partial^2 \overline{C}_1}{\partial x^2} + D_{T_1}\frac{\partial^2 \overline{C}_1}{\partial y^2} - v_1\frac{\partial \overline{C}_1}{\partial x} = p\,\overline{C}_1 - \frac{M}{n_1 b}\delta(x)\,\delta(y) \tag{27a}$$

$$D_{L_2}\frac{\partial^2 \overline{C}_2}{\partial x^2} + D_{T_2}\frac{\partial^2 \overline{C}_2}{\partial y^2} - v_2\frac{\partial \overline{C}_2}{\partial x} = p\,\overline{C}_2 \tag{27b}$$

The initial conditions have now been incorporated. Along with the governing equation, the remaining boundary conditions must be transferred to the Laplace domain as well. The boundary conditions used by all the models become

Model I: (first layer)

$$C_1(x,y,t) = 0 = \overline{C}_1(x,y,p) \qquad x \rightarrow -\infty \tag{28a}$$

$$C_1(L,y,t) = C_2(L,y,t) \longrightarrow \overline{C}_1(L,y,p) = \overline{C}_2(L,y,p) \tag{28b}$$

$$\frac{\partial C_1}{\partial y} = 0 = \frac{\partial \overline{C}_1}{\partial y} \qquad at\ y = 0 \tag{28c}$$

$$C_1(x,y,t) = 0 = \overline{C}_1(x,y,p) \qquad y \rightarrow \infty \tag{28d}$$

Models II and III: (first layer)

$$C_1(x,y,t) = 0 = \overline{C}_1(x,y,p) \qquad x \rightarrow -\infty \tag{29a}$$

$$C_1(x,y,t) = 0 = \overline{C}_1(x,y,p) \qquad x \rightarrow \infty \tag{29b}$$

$$\frac{\partial C_1}{\partial y} = 0 = \frac{\partial \overline{C}_1}{\partial y} \qquad at\ y = 0 \tag{29c}$$

$$C_1(x,y,t) = 0 = \overline{C}_1(x,y,p) \qquad y \rightarrow \infty \tag{29d}$$

Models I and II: (second layer)

$$n_1\left(v_1 C_1 - D_{L_1}\frac{\partial C_1}{\partial x}\right) = n_2\left(v_2 C_2 - D_{L_2}\frac{\partial C_2}{\partial x}\right) \longrightarrow$$

$$n_1\left(v_1 \overline{C}_1 - D_{L_1}\frac{\partial \overline{C}_1}{\partial x}\right) = n_2\left(v_2 \overline{C}_2 - D_{L_2}\frac{\partial \overline{C}_2}{\partial x}\right) \qquad at\ x = L \tag{30a}$$

$$C_2(x,y,t) = 0 = \overline{C}_2(x,y,p) \qquad x \to \infty \tag{30b}$$

$$\frac{\partial C_2}{\partial y} = 0 = \frac{\partial \overline{C}_2}{\partial y} \qquad at\ y = 0 \tag{30c}$$

$$C_2(x,y,t) = 0 = \overline{C}_2(x,y,p) \qquad y \to \infty \tag{30d}$$

Model III: (second layer)

$$C_1(L,y,t) = C_2(L,y,t) \longrightarrow \overline{C}_1(L,y,p) = \overline{C}_2(L,y,p) \tag{31a}$$

$$C_2(x,y,t) = 0 = \overline{C}_2(x,y,p) \qquad x \to \infty \tag{31b}$$

$$\frac{\partial C_2}{\partial y} = 0 = \frac{\partial \overline{C}_2}{\partial y} \qquad at\ y = 0 \tag{31c}$$

$$C_2(x,y,t) = 0 = \overline{C}_2(x,y,p) \qquad y \to \infty \tag{31d}$$

The Fourier cosine transform is now applied to (27a) and (27b) using (22) and (23) to eliminate the governing equation's y dependence. The last term on the right hand side of (22) is dropped due to the existence of boundary conditions (28c), (29c), (30c), and (31c) which are used in all three models. By applying (22) and (23), (27a) can be rewritten as

$$D_{L_1} \frac{\partial^2 G_1}{\partial x^2} - v_1 \frac{\partial G_1}{\partial x} - (p + D_{T_1} s^2) G_1 = - \frac{M}{\sqrt{2\pi}\, n_1 B} \delta(x) \tag{32a}$$

for the first layers of Models I, II, and III, and the governing equation for the second layers (27b) can be rewritten as

$$D_{L_2} \frac{\partial^2 G_2}{\partial x^2} - v_2 \frac{\partial G_2}{\partial x} - (p + D_{T_2} s^2) G_2 = 0 \tag{32b}$$

The remaining boundary conditions must also be transformed to the Fourier cosine domain. The boundary conditions used by the models are transformed to the following relationships:

Model I: (first layer)

$$\overline{C}_1(x,y,p) = 0 = G_1(x,s,p) \qquad x \rightarrow -\infty \tag{33a}$$

$$\overline{C}_1(L,y,p) = \overline{C}_2(L,y,p) \quad \longrightarrow \quad G_1(L,s,p) = G_2(L,s,p) \tag{33b}$$

Models II and III: (first layer)

$$\overline{C}_1(x,y,p) = 0 = G_1(x,s,p) \qquad x \rightarrow -\infty \tag{34a}$$

$$\overline{C}_1(x,y,p) = 0 = G_1(x,s,p) \qquad x \rightarrow \infty \tag{34b}$$

Models I and II: (second layer)

$$n_1\left[v_1\overline{C}_1 - D_{L_1}\frac{\partial\overline{C}_1}{\partial x}\right] = n_2\left[v_2\overline{C}_2 - D_{L_2}\frac{\partial\overline{C}_2}{\partial x}\right] \quad \longrightarrow$$

$$n_1\left[v_1G_1 - D_{L_1}\frac{\partial G_1}{\partial x}\right] = n_2\left[v_2G_2 - D_{L_2}\frac{\partial G_2}{\partial x}\right] \quad at\ x = L \tag{35a}$$

$$\overline{C}_2(x,y,p) = 0 = G_2(x,s,p) \qquad x \rightarrow \infty \tag{35b}$$

Model III: (second layer)

$$\overline{C}_1(L,y,p) = \overline{C}_2(L,y,p) \quad \longrightarrow \quad G_1(L,s,p) = G_2(L,s,p) \tag{36a}$$

$$\overline{C}_2(x,y,p) = 0 = G_2(x,s,p) \qquad x \rightarrow \infty \tag{36b}$$

The governing equation given as (1) has now been reduced from a partial differential equation to ordinary differential equations, one equation regulating solute transport in each layer. The remaining boundary conditions have also been transformed to the Laplace and Fourier domains. The solution to each model are now derived separately using the above information.

3.3.1: Model I

First Layer

The equations governing solute transport in the first and second layers of the first model are coupled at the layer interface; this requires that both solutions be solved simultaneously. The equation to flow in the first layer of this model (32a) is an inhomogeneous equation with constant coefficients which is solved utilizing the method of variation of parameters. Equation (32b), the equation to flow in the second layer, is recognized as a homogeneous equation with constant coefficients. A general solution to this equation is found by creating the characteristic equation of (32b) and solving for its roots. The process of obtaining solutions for Model I are detailed in Appendix A. The general solution to the first layer of Model I is given as

Model I: (first layer)

$$G_1 = A_1 \exp\left[\lambda_1 x + \lambda_3 (x - 2L)\right] + A_2\left\{ \exp\left[(\lambda_1 + \lambda_3)x\right] - \exp\left[\lambda_1 x + \lambda_3 (x - 2L)\right]\right\}$$

$$+ A_2\left\{ H(x) \exp\left[(\lambda_1 - \lambda_3)x\right] - H(x) \exp\left[(\lambda_1 + \lambda_3)x\right]\right\} \qquad (37)$$

where

$$A_1 = \frac{M}{\sqrt{2\pi}\, B}\left[\frac{1}{n_1 D_{L_1}(\lambda_1 + \lambda_3) - n_2 D_{L_2}(\lambda_2 - \lambda_4)}\right]$$

$$A_2 = \frac{M}{2\sqrt{2\pi} n_1 B D_{L_1} \lambda_3}$$

$$\lambda_1 = \frac{v_1}{2D_{L_1}}$$

$$\lambda_3 = \frac{\left[v_1{}^2 + 4D_{L_1}(p + D_{T_1}s^2)\right]^{1/2}}{2D_{L_1}}$$

$$H(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x > 0 \end{cases}$$

where $H(x)$ is usually termed the Heaviside function. The solution now resides in the Laplace and Fourier cosine domain. To be useful, it must be inverted back to real time and space. Eliminating the Heaviside functions from the solution changes the appearance of (37) to

$$G_1(x, s, p) = A_1 \exp\left[\lambda_1 x + \lambda_3(x - 2L)\right] + A_2 \exp\left[(\lambda_1 \pm \lambda_3)x\right]$$

$$- A_2 \exp\left[\lambda_1 x + \lambda_3(x - 2L)\right] \qquad (38)$$

$$\lambda_1 \pm \lambda_3 = \begin{cases} \lambda_1 + \lambda_3 & \text{if } x < 0 \\ \lambda_1 - \lambda_3 & \text{if } x > 0 \end{cases}$$

It will become evident later in the derivation that the sign on this particular exponential power does not affect the outcome of the solution; there will not be two separate solutions, one for the $x < 0$ domain and one for the $x > 0$ domain. To invert the first term of equation (38) was not possible because of the complexity of the denominator of A_1 . The Laplace inversion of this solution begins by rewriting the solution in a form conducive to this process.

$$G_1 = \frac{a_4}{a_6\left\{a_7 + \sqrt{p + a_1}\right\} - \left\{a_3 - \sqrt{p + a_2}\right\}} \exp\left[-a_8\sqrt{p + a_1}\right]$$

$$+ \frac{a_5}{\sqrt{p + a_1}} \exp\left[\pm a_9\sqrt{p + a_1}\right]$$

$$- \frac{a_5}{\sqrt{p + a_1}} \exp\left[-a_8\sqrt{p + a_1}\right] \qquad (39)$$

where

$$a_1 = \frac{v_1{}^2}{4D_{L_1}} + D_{T_1}s^2$$

$$a_2 = \frac{v_2{}^2}{4D_{L_2}} + D_{T_2}s^2$$

$$a_3 = \frac{v_2}{2(D_{L_2})^{1/2}}$$

$$a_4 = \frac{M}{n_2 B\left(2\pi D_{L_2}\right)^{1/2}} \exp\left[\frac{v_1 x}{2D_{L_1}}\right]$$

$$a_5 = \frac{M}{2n_1 B\left(2\pi D_{L_1}\right)^{1/2}} \exp\left[\frac{v_1 x}{2D_{L_1}}\right]$$

$$a_6 = \frac{n_1}{n_2}\left(\frac{D_{L_1}}{D_{L_2}}\right)^{1/2}$$

$$a_7 = \frac{v_1}{2(D_{L_1})^{1/2}}$$

$$a_8 = \frac{(2L - x)}{(D_{L_1})^{1/2}}$$

$$a_9 = \begin{cases} -\dfrac{+x}{(D_{L_1})^{1/2}} & \text{for } x > 0 \\ +\dfrac{-x}{(D_{L_1})^{1/2}} & \text{for } x < 0 \end{cases}$$

It is obvious from the character of a_9 that the exponential power of the second term of equation (39) is always negative and thus only one solution is obtained regardless of the sign on the variable, x. The last two terms of (39) are similar in appearance to the inversion formula given as (15a); these terms may be inverted directly with the use of the shifting property. For convenience a new state of the solutions is defined as

$$\overline{G_i}(x, s, t) = L^{-1}\left[G_i(x, s, p)\right] \qquad p \rightarrow t, \qquad i = 1,2 \qquad (40)$$

$\overline{G_i}$ is a representation of the solution inverted out of the Laplace domain but remaining still in the Fourier cosine domain. The Laplace inversion can be carried out and is given by

$$\overline{G}_1 = L^{-1}[G_1] =$$

$$L^{-1}\left[\frac{a_4}{a_6\left\{a_7 + \sqrt{p + a_1}\right\} - \left\{a_3 - \sqrt{p + a_2}\right\}} \exp\left[-a_8\sqrt{p + a_1}\right]\right]$$

$$+ L^{-1}\left[\frac{a_5}{\sqrt{p + a_1}} \exp\left[\pm a_9\sqrt{p + a_1}\right]\right]$$

$$- L^{-1}\left[\frac{a_5}{\sqrt{p + a_1}} \exp\left[-a_8\sqrt{p + a_1}\right]\right] \tag{41}$$

The results of the inversion of the last two terms in (41) are given in (15a) with the aid of the shifting property given in (16a) and thus (41) becomes

$$\overline{G}_1 = L^{-1}\left[\frac{a_4}{a_6\left\{a_7 + \sqrt{p + a_1}\right\} - \left\{a_3 - \sqrt{p + a_2}\right\}} \exp\left[-a_8\sqrt{p + a_1}\right]\right] +$$

$$\frac{a_5}{\sqrt{\pi t}} \exp\left[-a_1 t - \frac{a_9{}^2}{4t}\right] - \frac{a_5}{\sqrt{\pi t}} \exp\left[-a_1 t - \frac{a_8{}^2}{4t}\right] \tag{42}$$

The Laplace inversion of the first term is not known and as a result it will be carried through these derivations as it now exists.

The Fourier cosine inversion is much less difficult to implement than the Laplace inversion. The entire inversion process is a matter of multiplying the above equation by the kernel of the Fourier cosine integral transform given in (26) and integrating the new function with respect to s from zero to infinity. Equation (42) is rewritten below with newly defined coefficients to simplify the inversion out of the Fourier cosine

domain.

$$\overline{G}_1 = L^{-1}\left[\frac{a_4}{a_6\left\{a_7 + \sqrt{p + a_1}\right\} - \left\{a_3 - \sqrt{p + a_2}\right\}} \exp\left[-a_8\sqrt{p + a_1}\right]\right] +$$

$$\frac{a_5'}{\sqrt{\pi t}}\left\{\exp\left[-\frac{a_9^2}{4t}\right] - \exp\left[-\frac{a_8^2}{4t}\right]\right\}\exp\left[-a_1's^2\right] \tag{43}$$

where

$$a_1' = D_{T_1} t$$

$$a_5' = a_5 \exp\left[-\frac{v_1^2 t}{4D_{L_1}}\right]$$

The Fourier cosine inversion defined in (26) can now be applied to (43) with the following result:

$$C_1(x, y, t) = F_c^{-1}[\overline{G}_1] =$$

$$\left(\frac{2}{\pi}\right)^{1/2}\int_0^\infty L^{-1}\left[\frac{a_4}{a_6\left\{a_7 + \sqrt{p + a_1}\right\} - \left\{a_3 - \sqrt{p + a_2}\right\}} \exp\left[-a_8\sqrt{p + a_1}\right]\right]\cos(sy)\ ds +$$

$$\frac{a_5'}{\sqrt{\pi t}}\left\{\exp\left[-\frac{a_9^2}{4t}\right] - \exp\left[-\frac{a_8^2}{4t}\right]\right\}\left(\frac{2}{\pi}\right)^{1/2}\int_0^\infty \exp\left[-a_1's^2\right]\cos(sy)\ ds \tag{44}$$

The integral with respect to s can be found in Gradshteyn and Ryzhik [1980] to be

$$\int_0^\infty \exp\left[-a_1'(u)s^2\right]\cos(sy)\,ds = \frac{1}{2}\left[\frac{\pi}{a_1'(u)}\right]^{1/2}\exp\left[-\frac{y^2}{4a_1'(u)}\right] \tag{45}$$

Using the above equation the solution to the first layer of Model I can be written as

$$C_1(x,y,t) =$$

$$\left(\frac{2}{\pi}\right)^{1/2}\int_0^\infty L^{-1}\left[\frac{a_4}{a_6\left\{a_7+\sqrt{p+a_1}\right\}-\left\{a_3-\sqrt{p+a_2}\right\}}\exp\left[-a_8\sqrt{p+a_1}\right]\right]\cos(sy)\,ds +$$

$$\frac{a_5'}{\sqrt{2a_1'\pi t}}\exp\left[-\frac{y^2}{4a_1'}\right]\left\{\exp\left[-\frac{a_9^2}{4t}\right]-\exp\left[-\frac{a_8^2}{4t}\right]\right\} \tag{46}$$

Second Layer

The solution to the second layer is very similar to the first term of the first layer solution for which the Laplace inversion was not known. The derivation of the second layer solution is carried out in detail in Appendix A and the solution is given below.

Model I: (second layer)

$$G_2 = A_1\exp\left[(\lambda_1-\lambda_3)L+(\lambda_2-\lambda_4)(x-L)\right] \tag{47}$$

where

$$\lambda_2 = \frac{v_2}{2D_{L_2}}$$

$$\lambda_4 = \frac{\left[v_2^2+4D_{L_2}(p+D_{T_2}s^2)\right]^{1/2}}{2D_{L_2}}$$

The parameters, A_1, λ_1, and λ_3 used in the second layer solution are previously

defined in this section. For this second layer, the Laplace inversion of the entire solution is not known. This function can be written in terms of different coefficients to help simplify the Laplace inversion process.

$$G_2(x,s,p) = \frac{b_3}{a_6\left\{a_7 + \sqrt{p + a_1}\right\} - \left\{a_3 - \sqrt{p + a_2}\right\}} \exp\left[-b_1\sqrt{p + a_1} - b_2\sqrt{p + a_2}\right] \qquad (48)$$

where

$$b_1 = \frac{L}{\left(D_{L_1}\right)^{1/2}}$$

$$b_2 = \frac{(x - L)}{\left(D_{L_2}\right)^{1/2}}$$

$$b_3 = \frac{M}{n_2 B\,(2\pi D_{L_2})^{1/2}} \exp\left[\frac{v_1 L}{2D_{L_1}} + \frac{v_2(x - L)}{2D_{L_2}}\right]$$

The coefficients, a_1, a_2, a_3, a_6, and a_7, are defined previously. The denominator of the second layer solution is identical to that of the first term of the first layer. As was the case in the first layer, the Laplace inversion of this term is not carried out. The Laplace inversion of the second layer solution can be represented with the aid of (40) as

$$\overline{G}_2(x,s,t) = L^{-1}[G_2] =$$

$$L^{-1}\left[\frac{b_3}{a_6\left\{a_7 + \sqrt{p + a_1}\right\} - \left\{a_3 - \sqrt{p + a_2}\right\}} \exp\left[-b_1\sqrt{p + a_1} - b_2\sqrt{p + a52}\right]\right] \qquad (49)$$

The Fourier cosine inversion of the solution is written simply and directly with the aid of (26). The following equation presents the final form of the solution to solute

transport in the second layer of Model I:

$$C_2(x, y, t) = F_c^{-1}[\overline{G}_2] = \left(\frac{2}{\pi}\right)^{1/2} \int_0^{\infty} \cos(sy) \bullet$$

$$L^{-1}\left[\frac{b_3}{a_6\left\{a_7 + \sqrt{p + a_1}\right\} - \left\{a_3 - \sqrt{p + a_2}\right\}} \exp\left[-b_1\sqrt{p + a_1} - b_2\sqrt{p + a_2}\right]\right] ds \qquad (50)$$

Both the solution for the first layer and the one for the second are written in the real space, time domain as functions of the transform spaces; they are not closed form solutions. However, numerical inversion techniques can be applied to the solutions in their given final forms to receive output from these solutions in the real space, time domain. These numerical techniques are discussed in section 3.2. The solutions to these equations are given in dimensional units and any set of parameters with consistent units is acceptable.

3.3.2: Model II

The solution to solute transport in the first layer of Model II need not be derived since it is well known. Bear (1972) found the solution to (1) with boundary conditions (3) and (4b) to be

Model II: (first layer)

$$C_1(x, y, t) = \frac{M}{4\pi n_1 B t\left(D_{L_1} D_{T_1}\right)^{1/2}} \exp\left[-\frac{(x - v_1 t)^2}{4D_{L_1} t} - \frac{y^2}{4D_{T_1} t}\right] \qquad (51)$$

Since the governing equation and boundary conditions for the second layer of Models II and III are not coupled to any other system, their solutions may be computed directly. The approach to solving the transport equation (32b) for the second layer of Model II will be the same as the approach for Model III. Different solutions for the two models emerge from this equation due to the third-type boundary condition at the layer interface (35a) used in Model II and the first-type boundary condition (36a) used in Model III. Equation (32b) is recognized as a homogeneous equation with constant coefficients. A general solution to this equation is found by creating the

characteristic equation of (32b) and solving for its roots. The derivation of the characteristic equation, the process of finding its roots, the incorporation of those roots into the general solution, and the use of the boundary conditions to solve for the coefficients of the general solution for the second layer of Model II can be found in Appendix B. For Model II the general solution to equation (32b) with boundary conditions (35)is given here as

Model II: (second layer)

$$G_2 = A_3 \exp\left[\gamma_1 L + \beta_1 (x - L)\right] \tag{52}$$

where

$$A_3 = \frac{M}{\sqrt{2\pi}\, n_2 B}\left[\frac{1}{v_2 + \left[v_2^2 + 4D_{L_2}(p + D_{T_2} s^2)\right]^{1/2}}\right] \cdot$$

$$\left[\frac{v_1}{\left[v_1^2 + 4D_{L_1}(p + D_{T_1} s^2)\right]^{1/2}} + 1\right]$$

$$\gamma_1 = \frac{v_1 - \left[v_1^2 + 4D_{L_1}(p + D_{T_1} s^2)\right]^{1/2}}{2D_{L_1}}$$

$$\beta_1 = \frac{v_2 - \left[v_2^2 + 4D_{L_2}(p + D_{T_2} s^2)\right]^{1/2}}{2D_{L_2}}$$

Equation (52) must now be inverted back to the real time, space domain. Rewriting (52) to simplify the Laplace inversion yields

$$G_2(x, s, p) = c_1 \frac{\exp\left[-b_1\sqrt{p + a_1}\right]}{\sqrt{p + a_1}} \frac{\exp\left[-b_2\sqrt{p + a_2}\right]}{a_3 + \sqrt{p + a_2}} +$$

$$c_2 \exp\left[-b_1\sqrt{p + a_1}\right] \frac{\exp\left[-b_2\sqrt{p + a_2}\right]}{a_3 + \sqrt{p + a_2}} \qquad (53)$$

where

$$c_1 = \frac{M v_1}{4 n_2 B (2\pi D_{L_1} D_{L_2})^{1/2}} \exp\left[\frac{v_1 L}{2D_{L_1}} + \frac{v_2 (x - L)}{2D_{L_2}}\right]$$

$$c_2 = \frac{2(D_{L_1})^{1/2} b_1}{v_1}$$

The constants, $a_1, a_2, a_3, b_1,$ and b_2 are previously defined in the solution derivation of Model I. The Laplace inversion of (53) can be carried out by applying the convolution theorem (16b) in the following manner:

$$\overline{G}_2 = L^{-1}[G_2] =$$

$$L^{-1}\left[c_1 \frac{\exp\left[-b_1\sqrt{p + a_1}\right]}{\sqrt{p + a_1}}\right] * L^{-1}\left[\frac{\exp\left[-b_2\sqrt{p + a_2}\right]}{a_3 + \sqrt{p + a_2}}\right] +$$

$$L^{-1}\left[c_2 \exp\left[-b_1\sqrt{p + a_1}\right]\right] * L^{-1}\left[\frac{\exp\left[-b_2\sqrt{p + a_2}\right]}{a_3 + \sqrt{p + a_2}}\right] \qquad (54)$$

The results of the inversions shown in (54) were given by (15a), (15b), and (15c) with the aid of the shifting property in (16a) and (54) becomes

$$\overline{G}_2(x, s, t) = \frac{1}{\sqrt{\pi}} \int_0^t \left\{ c_1 + \frac{c_2 b_1}{2(t-u)} \right\} \left\{ \exp \frac{\left[-a_2 u - a_1 (t-u) - \frac{{b_1}^2}{4(t-u)} \right]}{\sqrt{t-u}} \right\} \bullet$$

$$\left\{ (\pi u)^{-1/2} \exp\left[-\frac{{b_2}^2}{4u} \right] - a_3 \exp\left[b_2 a_3 + {a_3}^2 u \right] Erfc \left(\frac{b_2}{2\sqrt{u}} + a_3 \sqrt{u} \right) \right\} du \qquad (55)$$

The Fourier cosine inversion of (55) may be immediately written from this form.

$$C_2(x, y, t) = {F_c}^{-1}[\overline{G}_2] = \frac{\sqrt{2}}{\pi} \int_0^\infty \int_0^t \left\{ c_1 + \frac{c_2 b_1}{2(t-u)} \right\} \left\{ \exp \frac{\left[-a_2 u - a_1 (t-u) - \frac{{b_1}^2}{4(t-u)} \right]}{\sqrt{t-u}} \right\} \bullet$$

$$\left\{ (\pi u)^{-1/2} \exp\left[-\frac{{b_2}^2}{4u} \right] - a_3 \exp\left[b_2 a_3 + {a_3}^2 u \right] Erfc \left(\frac{b_2}{2\sqrt{u}} + a_3 \sqrt{u} \right) \right\} \cos(sy)\, du\, ds \qquad (56)$$

As with the solution inversion of Model I, a_1 and a_2 are not constant with respect to s. Upon changing the order of integration the solution adopts the following form:

$$C_2 = \frac{\sqrt{2}}{\pi} \int_0^t \left\{ (\pi u)^{-1/2} \exp\left[-\frac{{b_2}^2}{4u} \right] - a_3 \exp\left[b_2 a_3 + {a_3}^2 u \right] Erfc \left(\frac{b_2}{2\sqrt{u}} + a_3 \sqrt{u} \right) \right\} \bullet$$

$$\left\{ c_1 + \frac{c_2 b_1}{2(t-u)} \right\} \left\{ \frac{\exp\left[-\frac{{b_1}^2}{4(t-u)} - {a_1}'(u) \right]}{\sqrt{t-u}} \right\} \int_0^\infty \exp\left[-{a_2}'(u) s^2 \right] \cos(sy)\, ds\, du \qquad (57)$$

where

$${a_1}'(u) = \left[\frac{{v_1}^2 (t-u)}{4 D_{L_1}} + \frac{{v_2}^2 u}{4 D_{L_2}} \right]$$

$$a_2'(u) = \left[D_{T_1}(t-u) + D_{T_2}u\right]$$

The integral with respect to s in (57) is similar to the known integral of (45). Substituting the known value for this integral into (57) yields the solution to the second layer of Model II.

$$C_2 = \frac{1}{\sqrt{2\pi}} \int_0^t \left\{ (\pi u)^{-1/2} \exp\left[-\frac{b_2^2}{4u}\right] - a_3 \exp\left[b_2 a_3 + a_3^2 u\right] Erfc\left(\frac{b_2}{2\sqrt{u}} + a_3\sqrt{u}\right) \right\} \bullet$$

$$\left\{c_1 + \frac{c_2 b_1}{2(t-u)}\right\} \left\{ \frac{\exp\left[-a_1'(u) - \frac{b_1^2}{4(t-u)} - \frac{y^2}{4a_2'(u)}\right]}{\sqrt{a_2'(u)}\sqrt{t-u}} \right\} du \qquad (58)$$

Within this single integral form of the solution there exists two separate, distinct sets of integrals, one containing a complimentary error function and the other containing a unique exponential function. The solution is displayed in two sections as

$$C_2 = \frac{1}{\sqrt{2\pi}} \int_0^t \left\{c_1 + \frac{c_2 b_1}{2(t-u)}\right\} \left\{ \frac{\exp\left[-a_1'(u) - \frac{b_1^2}{4(t-u)} - \frac{b_2^2}{4u} - \frac{y^2}{4a_2'(u)}\right]}{\sqrt{t-u}\sqrt{u\,a_2'(u)}} \right\} du$$

$$- \frac{a_3 e^{b_2 a_3}}{\sqrt{2\pi}} \int_0^t \left\{c_1 + \frac{c_2 b_1}{2(t-u)}\right\} \left\{ \frac{\exp\left[-a_1'(u) - \frac{b_1^2}{4(t-u)} + a_3^2 u - \frac{y^2}{4a_2'(u)}\right]}{\sqrt{t-u}\sqrt{a_2'(u)}} \right\} \bullet$$

$$Erfc\left(\frac{b_2}{2\sqrt{u}} + a_3\sqrt{u}\right) du \qquad (59)$$

At this point a change of variables is executed to create a simpler, more elegant form of the solution to the second layer of Model II. The relationship between τ and u is given by

$$\tau^2 = \frac{L^2}{4D_{L_1}(t-u)}$$

The limits of integration correspondingly transform to the following:

$$u = 0,\ \tau = \frac{L}{2(D_{L_1}t)^{1/2}} = k$$

$$u = t,\ \tau \rightarrow \infty$$

Evoking the change of variables transforms the solution to

$$C_2(x,y,t) = \frac{c_3}{\sqrt{\pi}}\exp^{-c_5^2 t}\int_k^\infty \frac{\left[\frac{c_6}{L\alpha_{T_1}} + \tau^2\right]}{\left(c_6 t(\tau^2 - k^2) + c_7 t(\tau^2 - k^2)^2\right)^{1/2}} \bullet$$

$$\exp\left[-\tau^2 - \frac{c_4^2 k^2 t}{\tau^2} + \frac{c_5^2 k^2 t}{\tau^2} - \frac{c_8^2\tau^2}{\tau^2 - k^2} - \frac{y^2\tau^2}{4(c_6 + c_7(\tau^2 - k^2))}\right] d\tau$$

$$- \frac{c_3 v_2}{2(D_{L_2})^{1/2}}\int_k^\infty \frac{\left[\frac{c_6}{L\alpha_{T_1}\tau} + \tau\right]}{\left(c_6 + c_7(\tau^2 - k^2)\right)^{1/2}} \exp\left[-\tau^2 - \frac{c_4^2 k^2 t}{\tau^2} - \frac{y^2\tau^2}{4(c_6 + c_7(\tau^2 - k^2))}\right] \bullet$$

$$Erfc\left[\frac{c_8\tau}{\left((\tau^2 - k^2)\right)^{1/2}} + \frac{c_5\left(t(\tau^2 - k^2)\right)^{1/2}}{\tau}\right] d\tau \qquad (60)$$

where

$$c_3 = \frac{M}{2\pi n_2 B (D_{L_2})^{1/2}} \exp\left[\frac{v_1 L}{2D_{L_1}} + \frac{v_2(x-L)}{2D_{L_2}}\right]$$

$$c_4 = \frac{v_1}{2(D_{L_1})^{1/2}}$$

$$c_5 = \frac{v_2}{2(D_{L_2})^{1/2}}$$

$$c_6 = D_{T_1} k^2 t$$

$$c_7 = D_{T_2} t$$

$$c_8 = \frac{(x - L)}{2(D_{L_2} t)^{1/2}}$$

3.3.3: Model III

The solution to solute transport in the first layer of Model III is identical to Model II's first layer solution and is given as (51). The approach used to obtain a solution to the second layer of Models I and II can be applied to the second layer of Model III. The roots of the characteristic equation and the incorporation of these roots and the appropriate boundary conditions into the general solution for the second layer of Model III can be found in Appendix C. For this model the general solution to equation (32b) with boundary conditions (36) is given here as

Model III: (second layer)

$$G_2 = A_4 \exp\left[\gamma_1 L + \beta_1 (x - L)\right] \tag{61}$$

where

$$A_4 = \frac{M}{\sqrt{2\pi}\, n_1 B\left[{v_1}^2 + 4D_{L_1}(p + D_{T_1} s^2)\right]^{1/2}}$$

The values for γ_1 and β_1 are the same for Model III as they were for Model II. The coefficient, A_4, is less complicated than A_3 due to the simplicity of the boundary

condition used at the interface in Model III. The format followed to invert the solution to the second layer of Model III is identical to that followed for Model II. To invert the solution for the second layer it is helpful to rewrite (61) as

$$G_2 = \frac{d_1}{\sqrt{p + a_1}} \exp\left[-b_1\sqrt{p + a_1}\right] \exp\left[-b_2\sqrt{p + a_2}\right] \tag{62}$$

where

$$d_1 = \frac{M}{2n_1 B \left(2\pi D_{L_1}\right)^{1/2}} \exp\left[\frac{v_1 L}{2D_{L_1}} + \frac{v_2 (x - L)}{2D_{L_2}}\right]$$

The constants, $a_1, a_2, b_1,$ and b_2 are previously defined in section 3.3.1. The Laplace inversion of (62) can be carried out by making use of the convolution theorem (16b) as

$$\overline{G}_2 = L^{-1}[G_2] = L^{-1}\left[\frac{d_1}{\sqrt{p + a_1}} \exp\left[-b_1\sqrt{p + a_1}\right]\right] * L^{-1}\left[\exp\left[-b_2\sqrt{p + a_2}\right]\right] \tag{63}$$

The results of the two Laplace inversions shown in (63) were given by (15a) and (15b) with the aid of the shifting property given in (16a) and thus (63) becomes

$$\overline{G}_2(x, s, t) = \frac{d_1 b_2}{2\pi} \int_0^t \frac{\exp\left[-a_1 u - \frac{b_1^2}{4u}\right]}{\sqrt{u}} \frac{\exp\left[-a_2 (t - u) - \frac{b_2^2}{4(t - u)}\right]}{(t - u)^{3/2}} \, du \tag{64}$$

To arrive at the solution for the second layer in the real space, time domain, the Fourier cosine inversion (26) must be applied at this time. The solution to solute transport in the second layer of Model III can be written in double integral form as

$$C_2(x,y,t) = F_c^{-1}[\overline{G}_2] =$$

$$\frac{d_1 b_2}{\sqrt{2}\pi^{3/2}} \int_0^{\infty} \int_0^{t} \frac{\exp\left[-a_1 u - \frac{b_1^2}{4u}\right]}{\sqrt{u}} \frac{\exp\left[-a_2(t-u) - \frac{b_2^2}{4(t-u)}\right]}{(t-u)^{3/2}} \cos(sy)\, du\, ds \qquad (65)$$

By changing the order of integration, the equation above can be rewritten and the integral with respect to s can be evaluated. Both a_1 and a_2 were constants with respect to the Laplace domain variable, p, but they are functions of the Fourier cosine domain variable, s, so they must be included in the integration over this variable. After the change in the order of integration the solution becomes

$$C_2(x,y,t) = \frac{d_1 b_2}{\sqrt{2}\,\pi^{3/2}} \int_0^{t} \frac{\exp\left[-\frac{b_1^2}{4u} - \frac{b_2^2}{4(t-u)} - a_1'(u)\right]}{\sqrt{u}\,(t-u)^{3/2}} \bullet$$

$$\int_0^{\infty} \exp\left[-a_2'(u)\, s^2\right] \cos(sy)\, ds\, du \qquad (66)$$

where

$$a_1'(u) = \left[\frac{v_1^2 u}{4D_{L_1}} + \frac{v_2^2 (t-u)}{4D_{L_2}}\right]$$

$$a_2'(u) = \left[D_{T_1} u + D_{T_2}(t-u)\right]$$

After the expansion and redefinition of a_1 and a_2 into $a_1'(u)$ and $a_2'(u)$, the integration with respect to s is given by (45) and the solution to solute transport in the second layer of Model III is rewritten in single integral form as

$$C_2 = \frac{d_1 b_2}{2\sqrt{2}\,\pi} \int_0^{t} \frac{\exp\left[-a_1'(u) - \frac{b_1^2}{4u} - \frac{b_2^2}{4(t-u)} - \frac{y^2}{4a_2'(u)}\right]}{\sqrt{u\,a_2'(u)}\,(t-u)^{3/2}}\, du \qquad (67)$$

This solution can be written in a simpler form by making the following change of variables from u to τ :

$$\tau^2 = \frac{(x-L)^2}{4D_{L_2}(t-u)}$$

The limits of integration then become

$$u = 0, \quad \tau = \frac{(x-L)}{2(D_{L_2}t)^{1/2}} = k$$

$$u = t, \quad \tau \rightarrow \infty$$

After this change of variables equation (67) is transformed to

$$C_2(x,y,t) = d_2 \int_k^\infty \frac{\tau^2}{\left(d_3 t(\tau^2 - k^2)^2 + d_4 t(\tau^2 - k^2)\right)^{1/2}} \cdot$$

$$\exp\left[-\tau^2 + \frac{d_5}{\tau^2} - \frac{d_6}{\tau^2} - \frac{d_7\tau^2}{(\tau^2 - k^2)} - \frac{y^2\tau^2}{4(d_3(\tau^2 - k^2) + d_4)}\right] d\tau \qquad (68)$$

where

$$d_2 = \frac{M}{2\pi^{3/2} n_1 B (D_{L_1})^{1/2}} \exp\left[\frac{v_1 L}{2D_{L_1}} + \frac{v_2(x-L)}{2D_{L_2}} - \frac{{v_1}^2 t}{4D_{L_1}}\right]$$

$$d_3 = D_{T_1} t$$

$$d_4 = D_{T_2} k^2 t$$

$$d_5 = {v_1}^2 k^2 t$$

$$d_6 = v_2^2 k^2 t$$

$$d_7 = \frac{L^2}{4 D_{L_1} t}$$

3.3.4: Preliminary Test of Solution Accuracy

The following example is given as an illustration that the solutions provide accurate results. The solution for two-dimensional solute transport in one layer is known and given as equation (51); from this a true or exact breakthrough curve along any given transect can be generated. The solutions derived in this work can be used to simulate this curve if the same time to injection is used, the layer interface is positioned somewhere along the breakthrough curve, and the parameters in both layers are equal. At first, a breakthrough curve is generated perpendicular to layering using (51). Second, breakthrough curves using the solutions for the three models (given in (46), (50), (51), (60), and (68)) are generated using the same parameters with all the respective parameters in both layers equal to each other; the layer interface is positioned close to the solute source so that the solute front has propogated into the second layer. This test is performed on both a laboratory and a field scale. The parameters used to produce the four curves are given in Table *1*. When comparing the output from two simulations either from the laboratory or from the field where the layer interface has been shifted, the output from one simulation must be multiplied by a ratio between the distance from source to interface lengths. This is a result of defining C_0 as

$$C_0 = \frac{M}{2\pi B L \alpha_{L_1} n_2} \tag{69}$$

To compare the two simulations directly, the value for C_0 must be the same, but since it depends on the source to interface length, *L*, the output from one of the simulations must be multiplied by a scaling factor. After scaling one of the simulations the results on the laboratory scale from both simulations are identical as depicted in Figures *4a* and *4b*. The results on the field scale from both simulations are also identical as depicted in Figures *5a* and *5b*. This analysis provides a degree of confidence that the numerical integration scheme and numerical inversion techniques are correctly evaluating the function.

Parameter Values for Test of Solution Accuracy										
	$t\,(d)$	$L\,(m)$	$y\,(m)$	$v_1\,(m/d)$	$\alpha_{L_1}\,(m)$	$\alpha_{T_1}\,(m)$	$v_2\,(m/d)$	$\alpha_{L_2}\,(m)$	$\alpha_{T_2}\,(m)$	$n2/n1$
Figure 4a	6	2.0	0.0	0.1	0.05	0.005				
Figure 4b	0.5	0.0	0.1	0.05	0.005	0.1	0.05	0.005	1.0	
Figure 5a	600	200.0	0.0	0.1	10.0	1.0				
Figure 5b	600	200.0	0.0	0.1	10.0	1.0	0.1	10.0	1.0	1.0

Table *1*

To compare the results for varying parameters, the value of C_0 must remain constant. For this reason, all the tests performed on both the laboratory and field scale in the succeeding section are performed with the source to interface length, L, and the longitudinal dispersivity in the first layer, α_{L_1}, held constant. If C_0 were defined as the actual concentration of solute input to the system, then values for C/C_0 must always be less than one. Since C_0 is defined as (69), its value is arbitrarily determined by the values for *M, B, L,* α_{L_1} , and n_2 and thus values for C/C_0 that are greater than one are not surprising.

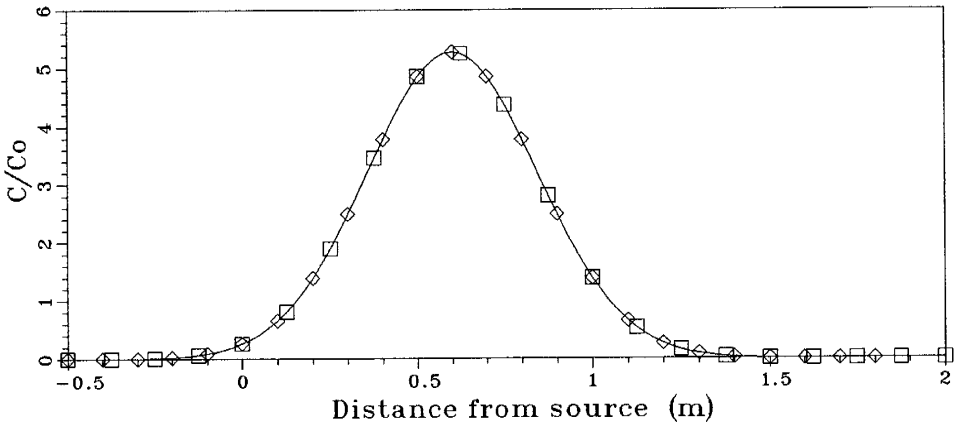

Figure 4a. Concentration distribution (using transport parameters given in Table *1*) in a homogeneous aquifer, laboratory scale. The solid line represents Model I, the squares, Model II, and the diamonds, Model III.

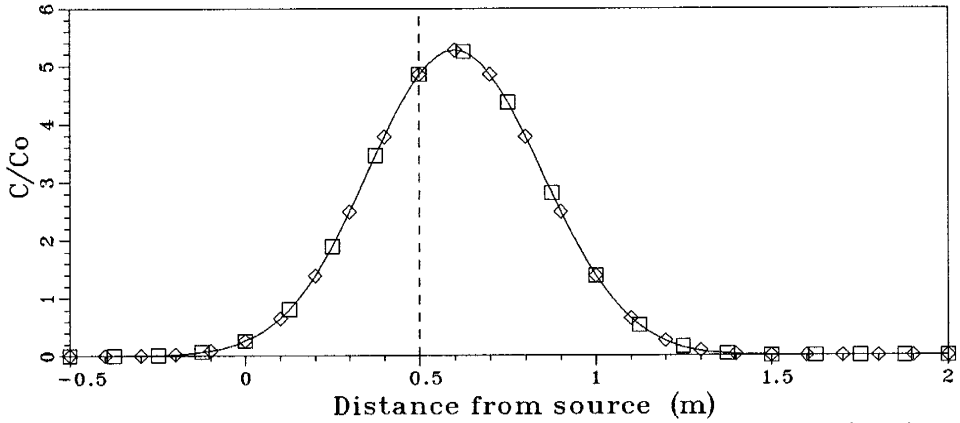

Figure 4b. Concentration distribution (using transport parameters given in Table *1*) in a two-layer aquifer with identical transport properties in each layer. The solid line represents Model I, the squares, Model II, and the diamonds, Model III.

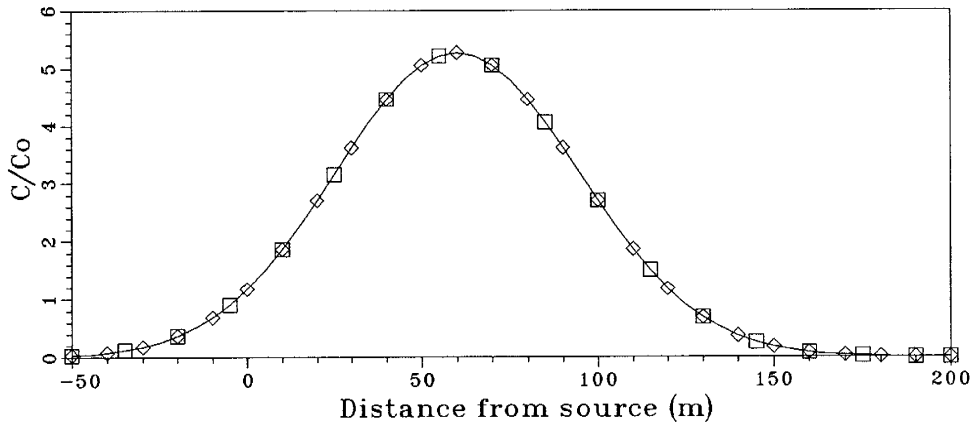

Figure 5a. Concentration distribution (using transport parameters given in Table *1*) in a homogeneous aquifer, field scale. The solid line represents Model I, the squares, Model II, and the diamonds, Model III.

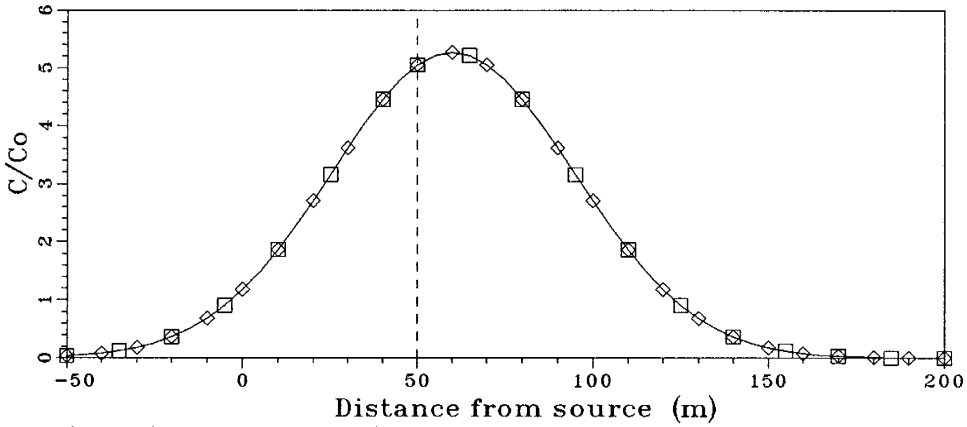

Figure 5b. Concentration distribution (using transport parameters given in Table *1*) in a two-layer aquifer with identical transport properties in each layer. The solid line represents Model I, the squares, Model II, and the diamonds, Model III.

4: RESULTS AND DISCUSSION

The three solutions presented in the previous section are analyzed and compared to one another in this section to try to determine under what conditions one may safely use one of the two approximate solutions to calculate concentrations of the solute instead of using the more complex Model I solution. It was not possible to invert a portion of the first layer Model I solution or all of the second layer Model I solution out of the Laplace domain which makes this solution difficult to implement and more susceptible to error due to multiple numerical approximations and round-off error and therefore the Model I solution is less desirable for application.

Two methods for numerical integration were analyzed and applied. The first method used was a 32-point fixed Gaussian quadrature. To improve on the accuracy of the quadrature, it was applied to sections of the function domain and these sections were then added together,. The second method implemented was an adaptive rhomberg extrapolation given by de Boor (1971). This second method did not need to be applied to sections of the function domain to increase accuracy. Both methods gave very similar results everywhere except for the area in close proximity to the layer interface; this problem was eliminated by applying the quadrature to sections of the function domain. The Gaussian quadrature was used to evaluate the integrand of the three models and the adaptive rhomberg extrapolation was used to check the accuracy of the quadrature.

Two different domain sizes will be used to examine the behavior of the solutions to the three models. The first type compares the three solutions on a laboratory scale with shorter travel distances and shorter dispersivity lengths. The second type compares the three solutions on a field scale with longer travel distance to the second layer and not only longer dispersivity lengths but also larger variations in dispersivities between the two layers. The scenario where the dispersivities are small will be examined on the laboratory scale and it will be shown that the three models predict similar breakthrough curves. The solutions behave differently on the field scale than they do on the laboratory scale due to the larger variation in the dispersivities on the field scale and so this makes each domain size worth examining. For each domain size, four cases representing different combinations of transport parameters will be used to compare the three models.

4.1: Laboratory Scale Analysis

The first type to be analyzed is solute transport over short distances likely encountered in a laboratory setting. It is under these conditions that the solutions exhibit the most similarity to one another since in all cases the dispersion coefficients are small and are of the same order of magnitude. A finite domain model may be

characterized with these solutions provided that end boundary effects of the model are minimized or that measurements are not retrieved close to the end boundary of the model. In this analysis the calculations cease at two meters from the solute source. The width of the model is considered to be infinite.

The first case considered is one in which the seepage velocity and the rate of dispersion in the first layer are higher than in the second layer, i.e. $v_1 > v_2$, $D_{L_1} > D_{L_2}$, and $D_{T_1} > D_{T_2}$. This may represent the existence of a "clogging layer". The slower movement of fluid in the second layer tends to retard the movement of the solute. The breakthrough curve is observed at three different times for each case. The values used in this first case for the transport parameters can be found in Table 2.

Parameter Values for Four Cases of Layer Stratification, Laboratory Scale

	t (d)	L (m)	y (m)	v_1 (m/d)	α_{L_1} (m)	α_{T_1} (m)	v_2 (m/d)	α_{L_2} (m)	α_{T_2} (m)	$n2/n1$
Case 1	4.0 6.0 20.0	0.5	0.0	0.1	0.05	0.005	0.05	0.01	0.001	2
Case 2	4.0 6.0 12.0	0.5	0.0	0.1	0.05	0.005	0.1	0.01	0.001	1
Case 3	4.0 6.0 10.0	0.5	0.0	0.1	0.05	0.005	0.2	0.1	0.01	0.5
Case 4	4.0 6.0 14.0	0.5	0.0	0.1	0.05	0.005	0.1	0.1	0.01	1

Table 2

A relationship that must be taken into account at all times is the equality of mass flux of water across the layer interface.

$$n_1 v_1 = n_2 v_2 \quad \text{or} \quad v_1 = \frac{n_2}{n_1} v_2 \tag{70}$$

Therefore, the seepage velocity in the first layer is related to the seepage velocity in the second layer multiplied by a ratio of the porosities.

Breakthrough curves generated using the three solutions and these transport parameters are shown in Figures *6a*, *6b*, and *6c*. The solid line represents the output from the solution employing both boundary conditions at the layer interface (Model I). The diamonds in the figures depict the output from the approximate solution that was derived using the continuity of solute mass flux boundary condition at the layer interface (Model II). The dotted line depicts the output from the approximate solution that was derived using the continuity of concentration boundary condition (Model III). The dashed line represents the layer interface. The plume location is shown at three different times since injection.

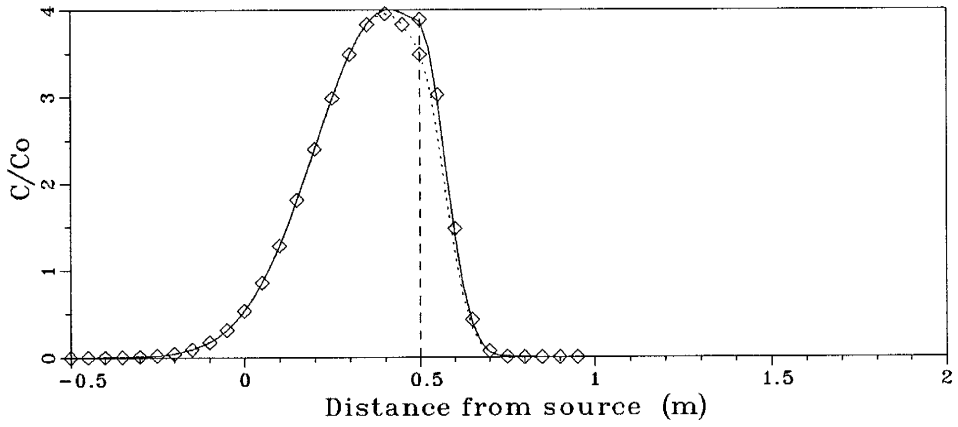

Figure *6a*.

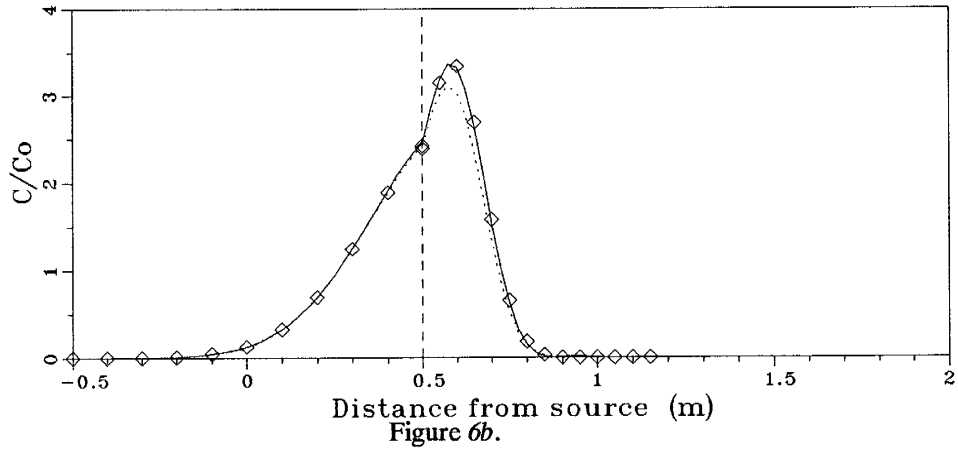

Figure *6b*.

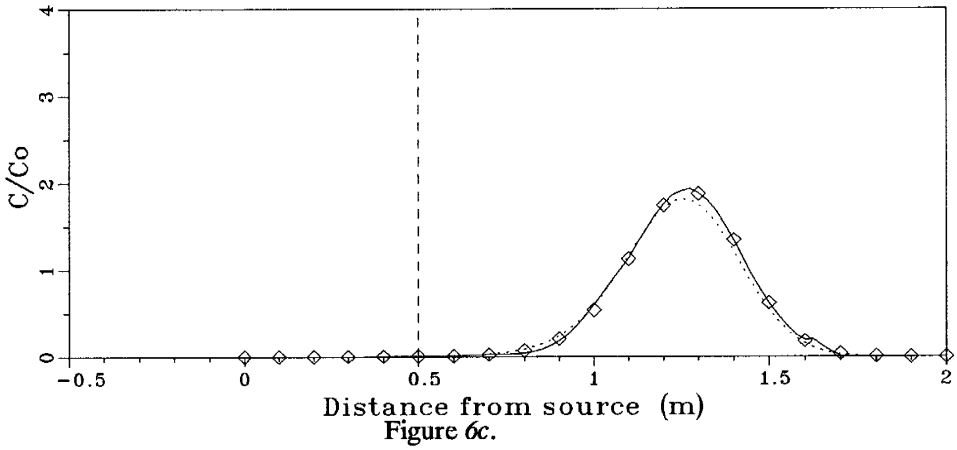

Figure *6c*.

Figure *6*. Concentration distribution (using Case *1* transport parameters given in Table *2*) parameters at t = *4, 6,* and *20*. The solid line represents the solution to Model I, the diamonds, Model II, and the dotted line, Model III.

The output from the three solutions at the three different times are quite similar except for the region right at the interface where a small discrepancy exists. Models II and III have in common the same equation for solute transport in the first layer and so in all cases the diamonds and the dotted line represent the same curve in this region. The solutions to Models I and III are indeed continuous across the interface; this is evident in all the graphs. In each of the three graphs of Figure *6*, Model II yields the same results in the second layer as Model I. In most cases this produces a jump discontinuity in the breakthrough curve for Model II. Model III, however, produces a peak slightly lower than the other two solutions even after the breakthrough curve has moved completely into the second layer as shown in figure *6c*.

Unusual behavior is observed in Figures *6a* and *6b*; the tail of the breakthrough curve extends behind the solute source implying that concentrations are greater than zero to some distance behind the point of injection. This is hardly plausible when considering that there are no driving forces accounted for in the mathematical model that could advance the solute behind the source. Chen and Woodside (1988) encountered a similar phenomenon and they attributed this adverse (backward) dispersion to the infinitely large concentration gradient existing at the plume boundary. It seems this adverse dispersion is a mathematical artifact originating from any initial condition which has an inherently large concentration gradient. This occurrence of adverse dispersion will also be evident in subsequent cases.

The second case examined is one in which the rate of dispersion in the first layer is higher than in the second layer but the seepage velocities are the same, i.e. $v_1 = v_2$, $D_{L_1} > D_{L_2}$, and $D_{T_1} > D_{T_2}$. Figures *7a*, *7b*, and *7c* depict the breakthrough curves calculated from the three solutions using the set of transport

parameters labeled Case *2* in Table *2*.

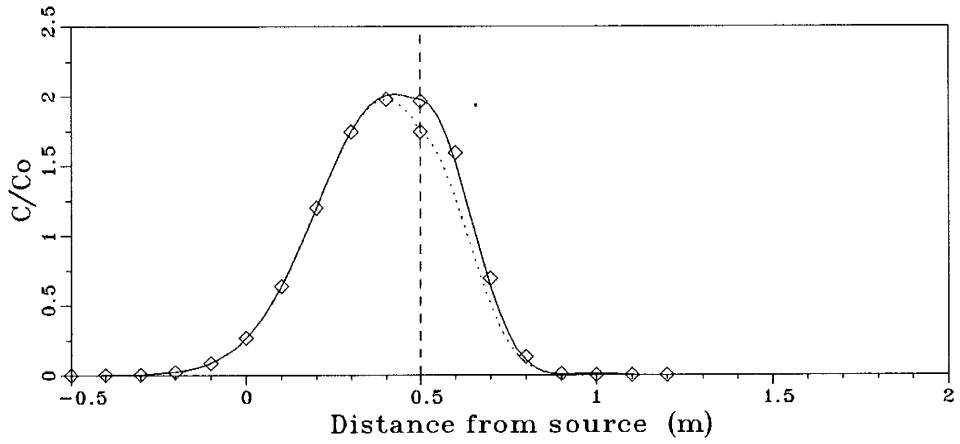

Figure *7a*.

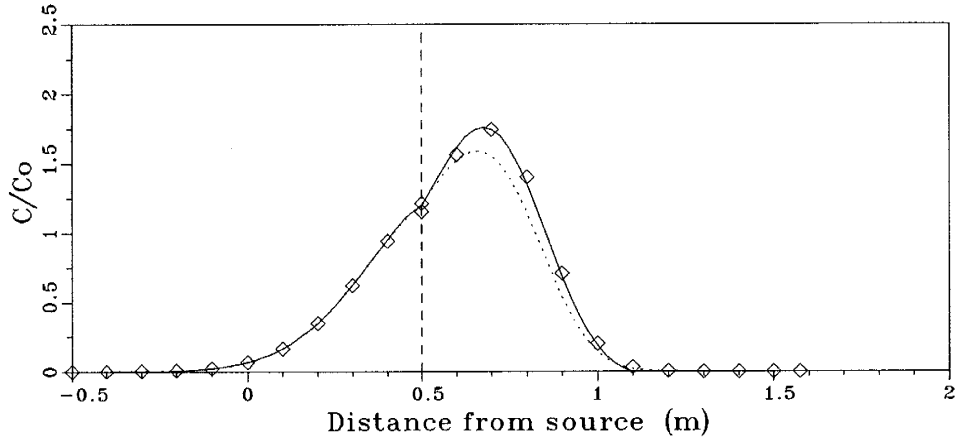

Figure *7b*.

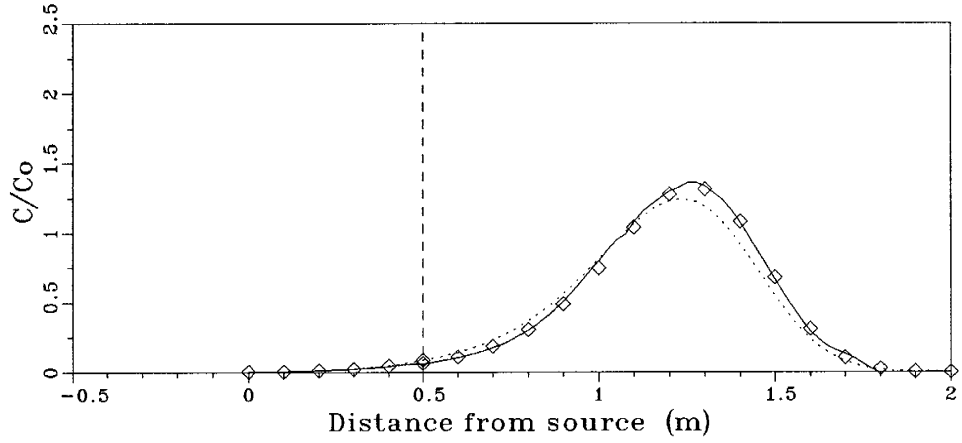

Figure *7c*.

Figure *7*. Concentration distribution (using Case *2* transport parameters given in Table *2*) at t = *4, 6,* and *12*. The solid line represents the solution to Model I, the diamonds, Model II, and the dotted line, Model III.

The general behavior of this set of breakthrough curves is similar to the behavior of the curves generated using the Case *1* parameters. The increase in the second layer's velocity between the Case *1* and Case *2* layering scenarios serves both to propogate the center of mass of the solute plume more quickly and also reduces the relative difference between the three solutions. The relative difference between the solutions decreases from Figure *6* to Figure *7* because although the dispersivity values remain constant from Case *1* to Case *2*, the magnitude of the difference in dispersion coefficients between the two layers decreases from Case *1* to Case *2*. The curves in Figure *7* are smoother than the curves in figure *6* due to the increase in the dispersion coefficients, D_{L_2} and D_{T_2}, caused by the increase in the seepage velocity of the second layer. As before, all three solutions produce very similar results. Again, Model III tends to underestimate solute concentrations slightly when compared to the other two.

The third stratification case represents solute transport from a low velocity layer to a high velocity layer. The solute upon reaching the second layer quickly moves forward due to the increased seepage velocity and disperses more thoroughly due to the increase in the dispersion coefficients. The transport parameter values used to model this type of stratification can be found in Table *2*. The breakthrough curves depicting movement through this configuration of porous media are shown in figures *8a*, *8b*, and *8c*.

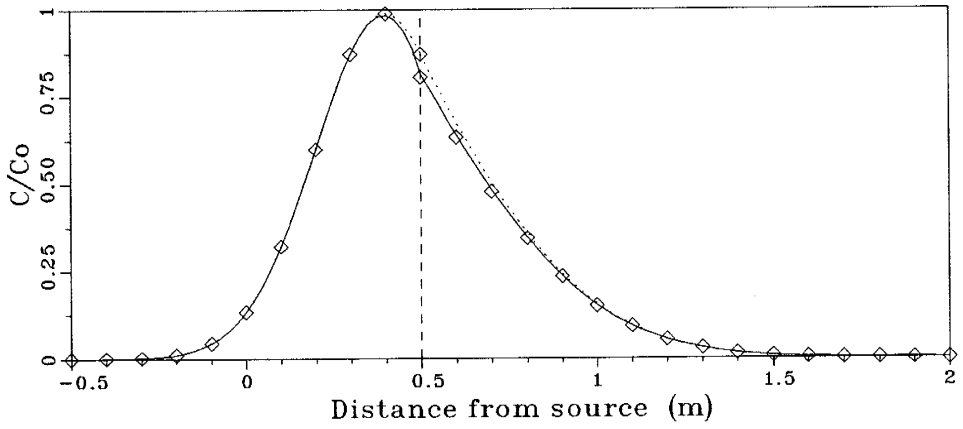

Figure *8a*.

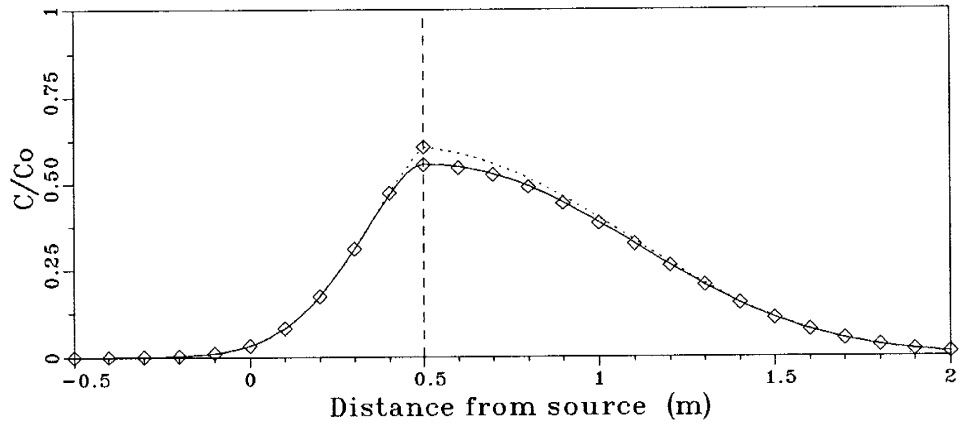

Figure *8b*.

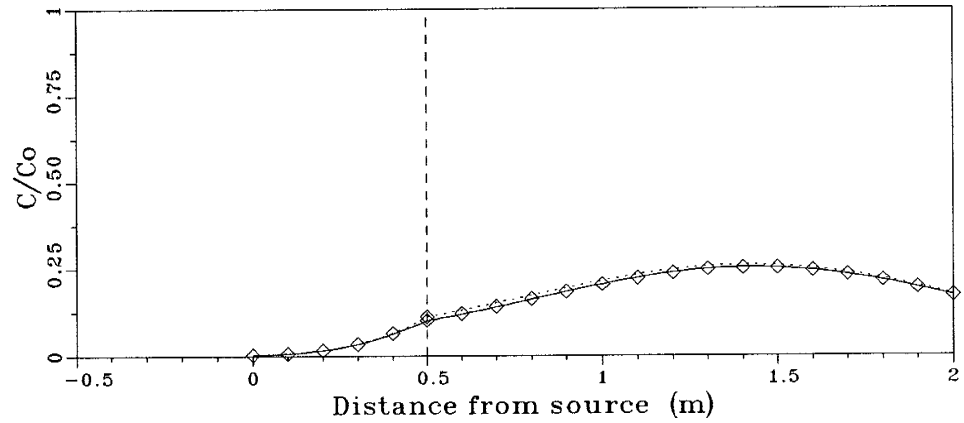

Figure *8c*.

Figure *8*. Concentration distribution (using Case *3* transport parameters given in Table *2*) at t = *4, 6,* and *10*. The solid line represents the solution to Model I, the diamonds, Model II, and the dotted line, Model III.

The results from this layering strategy are opposite from the results obtained using the Case *1* or Case *2* strategies. The Model I and Model II solutions still produce identical results in the second layer but the Model III solution now tends to overestimate the solute concentration. In contrast to the previous two cases, the three solutions produce almost the same results for large times.

The fourth case portrays the situation where $v_1 = v_2$, $D_{L_1} < D_{L_2}$, and $D_{T_1} < D_{T_2}$. This layering sequence is the same as the third case only that the seepage velocities are kept constant between the two layers. This scenario is the same as the second case except with the relationship between the dispersivities of the two layers reversed. The transport parameters used in this example can be found in Table *2*. The breakthrough curves generated using these parameters are shown in Figures *9a*, *9b*, and *9c*. The Model III solution again overestimates the solute concentration in the second layer as compared to the other two solutions. The difference between the

solutions, however, diminishes at large times.

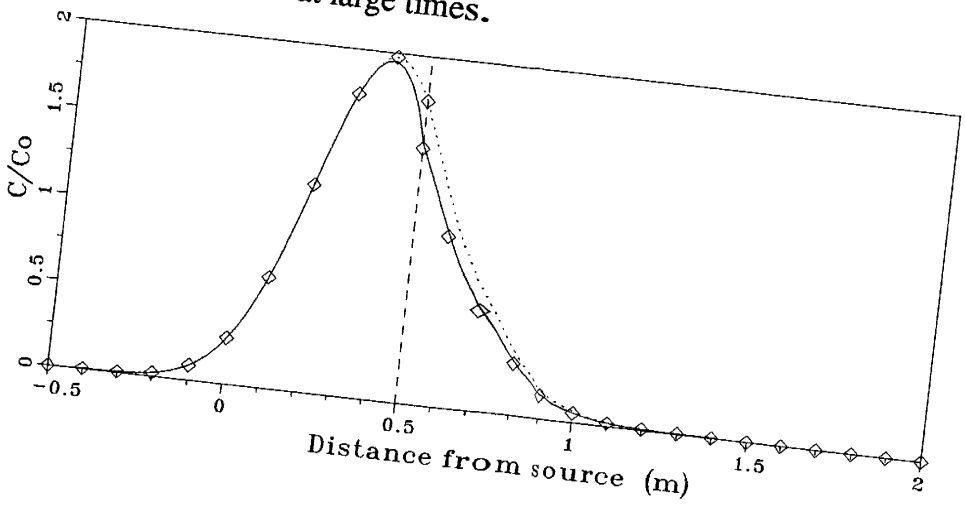

Figure *9a*.

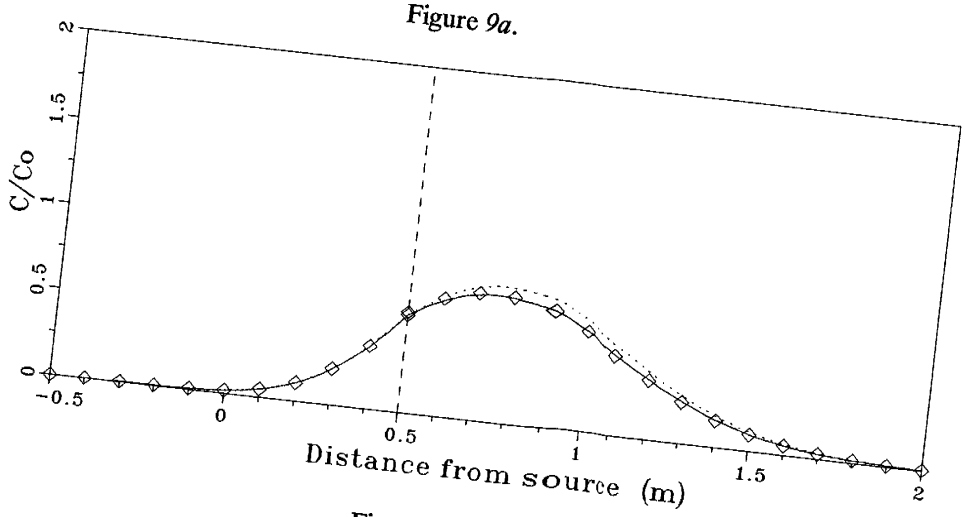

Figure *9b*.

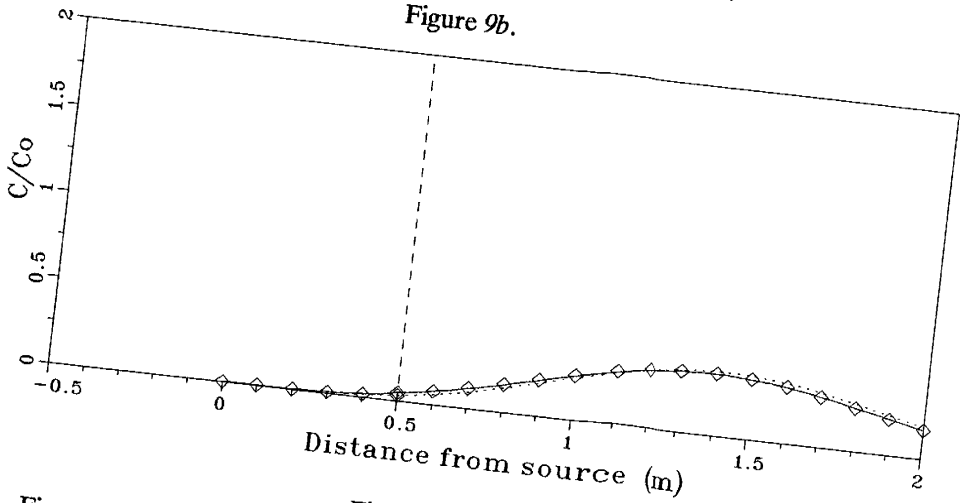

Figure *9c*.

Figure *9*. Concentration distribution (using Case *4* transport parameters given in Table *2*) at t = *4, 6,* and *14*. The solid line represents the solution to Model I, the diamonds, Model II, and the dotted line, Model III.

The three solutions model solute transport in two dimensions although as yet in this paper their two-dimensional behavior has not been portrayed. The one-dimensional breakthrough curves provided in this section have all been oriented normal to the layer interface. Using this type of analysis, the differences between the solutions are the most visible. It is difficult to recognize differences in the solutions using contour plots. Figure *10* is a plot of C/C_0 concentration contours using the solutions from Models II and III and the transport parameters from Case *1* with a time since injection of six days.

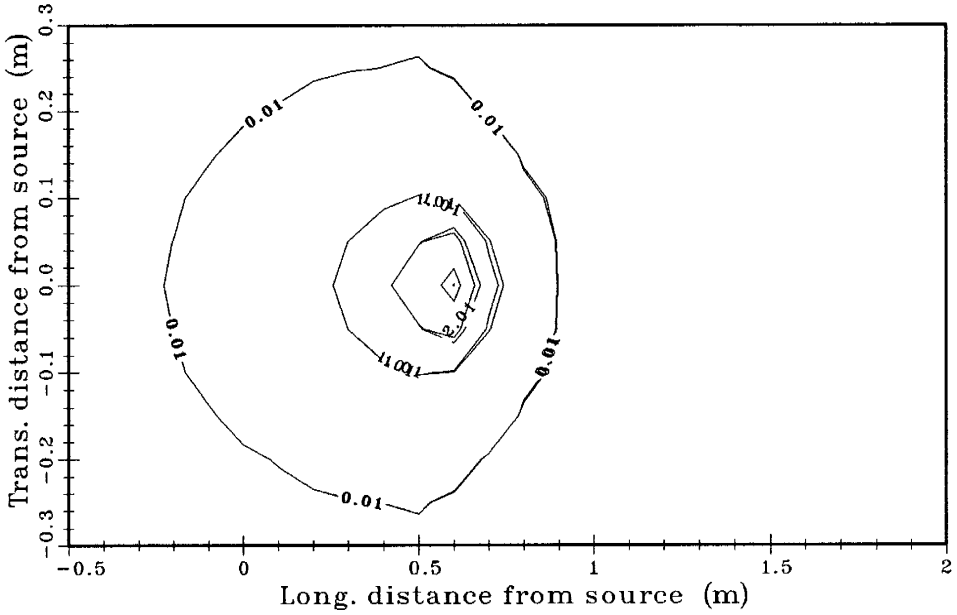

Figure *10*. Concentration contours generated using solutions from Models II and III (and Case *1* transport parameters given in Table *2*) at $t=6$. The interface between layers is located at $x=0.5$.

Note that the vertical axis is exaggerated when compared to the scale of the horizontal axis. From this plot it is possible to see that differences in the solutions exist in the lateral direction as well as the longitudinal direction. Model III underestimates the travel distance of the concentration when the dispersivity in the second layer is less than in the first, but to a negligible degree. Figure *11* is a plot of concentration contours using the parameters from Case *3* with a time since injection of six days. The solute spreads out laterally to a greater degree in figure *11* because the transverse dispersion coefficient is greater in this case. In this scenario the travel distance of the solute according to Model III is overestimated only slightly when comparing it to the Model II solution. This similarity will disappear when transport parameters likely encountered on a field scale are incorporated.

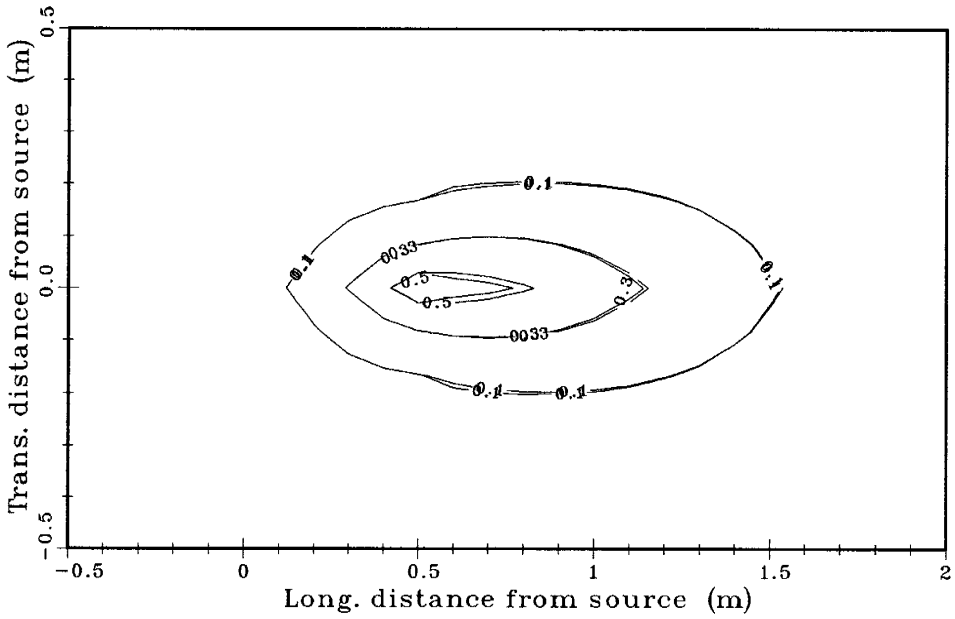

Figure *11.* Concentration contours generated using solutions from Models II and III (and Case *3* transport parameters given in Table *2*) at $t=6$. The interface between layers is located at $x=0.5$.

In summary, in every scenario the three solutions are all very similar except at the interface boundary where Model III sometimes underestimates or overestimates the solute concentration, as compared to the other two models, depending on if the dispersion coefficients decrease or increase in the second layer, respectively. In all cases the location of the peak of concentration of all three solutions coincide. The breakthrough curves generated under circumstances where the seepage velocities in both layers are equal, look very similar to the curves generated with seepage velocities varying between the two layers. The conclusion can be drawn that the difference in dispersion between the two layers is more the cause of the difference in the three solutions than the difference in the velocities. Any of the three solutions will give accurate results provided that the difference in dispersivity values between the layers is not great. With this in mind Model III would be the best solution to use if the layer parameters are very similar since it is the fastest and easiest of the three to implement. Model III does not provide accurate results when the character of each of the layers are very different. Examples of when this occurs will be examined at the field scale.

4.2: Field Scale Analysis

The second model type involves the comparison of the three solutions with transport parameters evident in field scale problems. Instead of observing solute transport

over short domains and small durations, the length of the domain and the dispersion are both increased to several meters in length and the time duration is increased to several hundred days. Field scale values of longitudinal dispersivity appear in literature on the order of a fraction of a meter to tens of meters. Murty and Scott (1977) modeled contaminant transport using longitudinal dispersivities from *20* to *60* meters. Lenda and Zuber (1970) fitted a model to experimental data using dispersivities from *0.26* to *6.0* meters. Pinder (1973) used a value for dispersivity of *21.3* meters for his finite-element model of contamination of Long Island, New York. Bredehoeft and Pinder (1973) modeled a contaminant plume in Brunswick, Georgia using longitudinal dispersivities of *100* to *400* feet (*30.5* to *122* meters). The range of longitudinal dispersivities used in this section for solution comparison range from *0.1* to *100.0* meters. As in the laboratory scale analysis the ratio of $\alpha_T/\alpha_L = 0.1$ is a fixed constant.

Similar to the laboratory scale analysis, parameters depicting four cases of layer stratification will be used to point out differences in the breakthrough curves produced by the three different solutions. In the field scale analysis of the transport parameters, the distance from solute source to the interface between the layers is arbitrarily fixed to a length of *50.0* meters.

The first case examines the effect of a clogging layer on the solute front; both the seepage velocity and the dispersion coefficients are greater in the first layer than they are in the second. Table *3* contains the transport parameters for the four cases of field scale layering. Breakthrough curves using the first layering strategy are shown for three different times in Figures *12a*, *12b*, and *12c*. The solid line displays the Model I solution, the diamonds represent the Model II solution, the dotted line is the output from the Model III solution, and the dashed line indicates where the layer interface is located. As with the laboratory scale analysis, the two approximate solutions for transport in the first layer are the same and so it is expected that the diamonds depicting the Model II solution and the dotted line depicting the Model III solution coincide in the first layer. The breakthrough curves generated in figure *12* are very similar in shape to the curves of figure *6*. As in the laboratory scale model, the Model III solution underestimates the solute concentration when compared to the other two solutions. The degree of underestimation is greater in the field scale analysis, however. Since the velocities used in the field scale analysis are the same as those used for the laboratory scale, the greater degree of underestimation is attributed to the fact that larger differences in the dispersion coefficients between the layers exist in the field scale model when compared to the laboratory scale model. This reinforces the observation made in section *4.1* that the difference in dispersive properties between the two layers, not the difference in velocities, is the cause of the different predictions of solute concentrations by the three models.

Parameter Values for Four Cases of Layer Stratification, Field Scale										
	$t\,(d)$	$L\,(m)$	$y\,(m)$	$v_1\,(m/d)$	$\alpha_{L_1}\,(m)$	$\alpha_{T_1}\,(m)$	$v_2\,(m/d)$	$\alpha_{L_2}\,(m)$	$\alpha_{T_2}\,(m)$	$n2/n1$
Case 1	400 600 1200	50.0	0.0	0.1	5.0	0.5	0.05	0.1	0.01	2
Case 2	400 600 1200	50.0	0.0	0.1	5.0	0.5	0.1	0.1	0.01	1
Case 3	400 600 1250	50.0	0.0	0.1	5.0	0.5	0.2	100.0	10.0	0.5
Case 4	400 600 1500	50.0	0.0	0.1	5.0	0.5	0.1	100.0	10.0	1

Table *3*

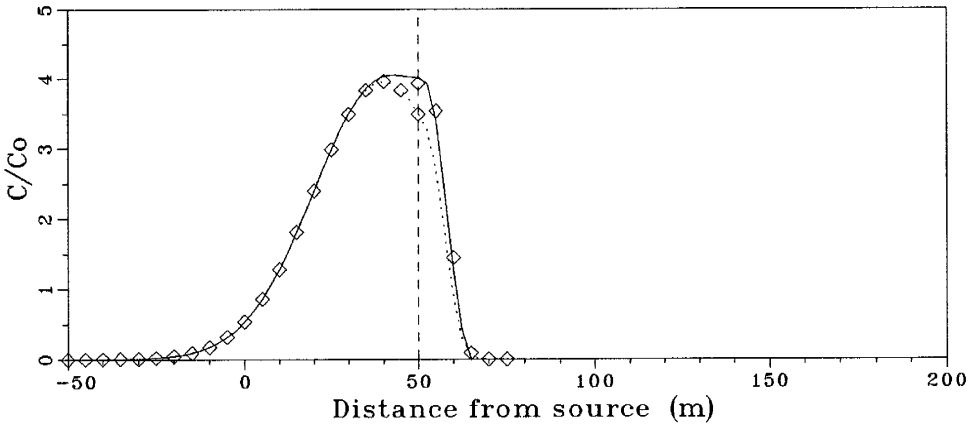

Figure *12a.*

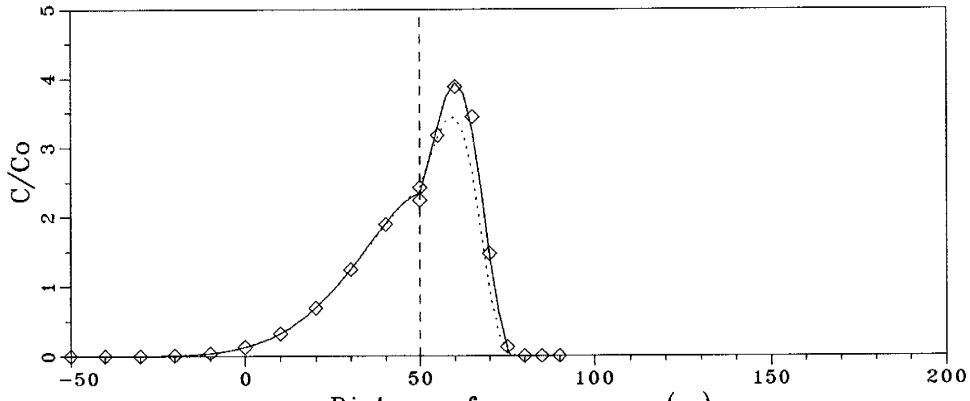

Figure *12b*.

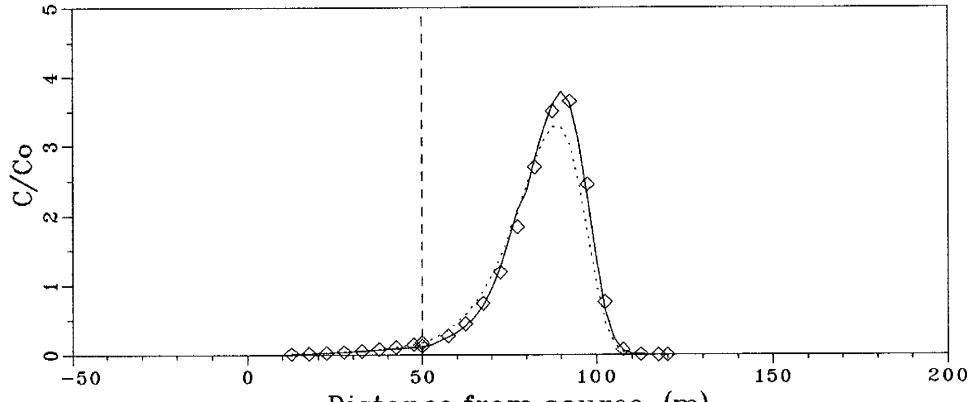

Figure *12c*.

Figure *12*. Concentration distribution (using Case *1* transport parameters given in Table *3*) at t = *400, 600,* and *1200*. The solid line represents the solution to Model I, the diamonds, Model II, and the dotted line, Model III.

The second case to be analyzed is the same as the first case only with the seepage velocities in both layers equal. The transport parameters can be found in Table *3*. Figures *13a*, *13b*, and *13c* show the breakthrough curves for the three solutions at the fixed times given in the table. The breakthrough curves of Figure *13* appear very similar to the curves of Figure *12* except that they are smoothed out due to the increased dispersion in the second layer. The curves presented in Figure *13* also look very much like the curves of Figure *7* except that as in the previous field case the Model III solution more severely underestimates the solute concentration as compared to the laboratory scale analysis.

The third case to consider is a layering sequence where the seepage velocity and dispersion coefficients in the second layer are greater than in the first. Table *3* contains the transport parameters for Case *3*. The breakthrough curves for the third case are shown in figures *14a*, *14b*, and *14c*.

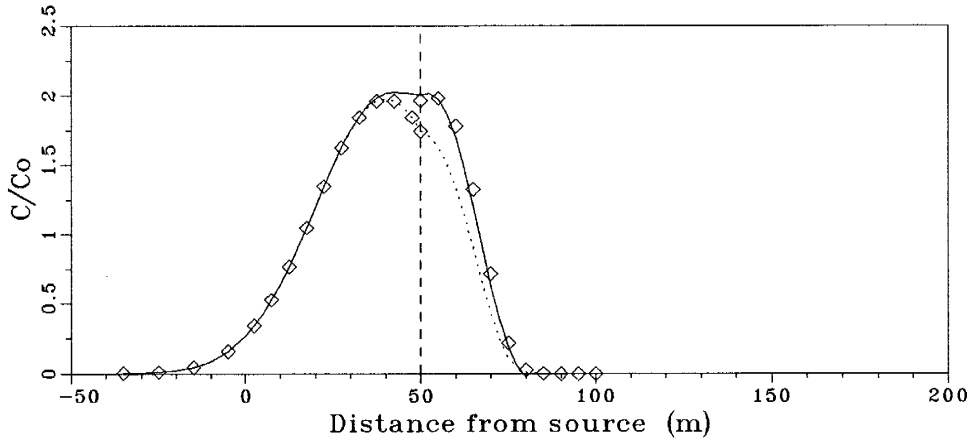

Figure *13a.*

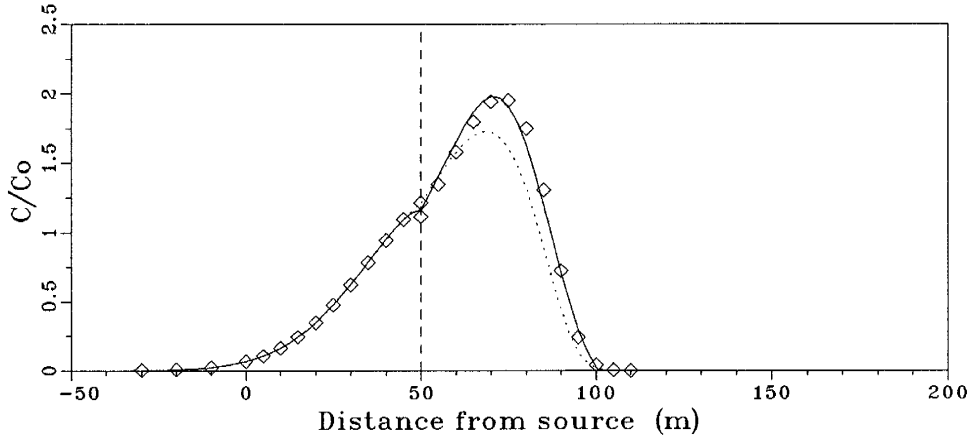

Figure *13b.*

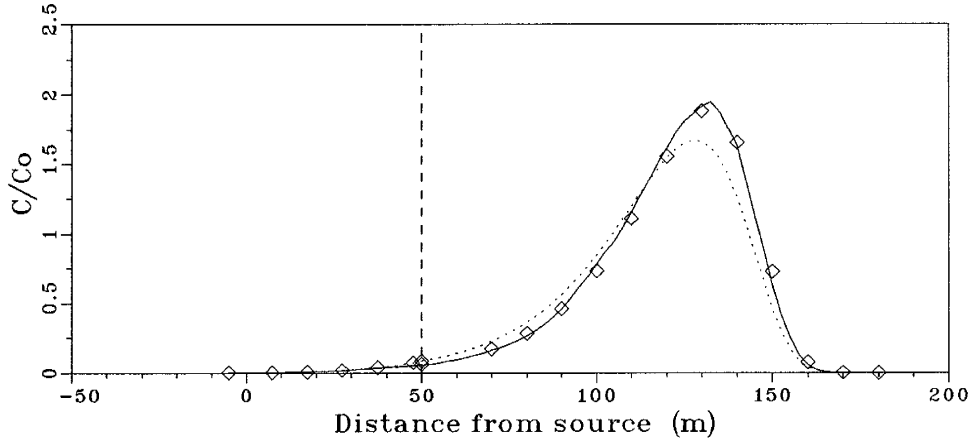

Figure *13c.*

Figure *13.* Concentration distribution (using Case *2* transport parameters given in Table *3*) at t = *400, 600,* and *1200.* The solid line represents the solution to Model I, the diamonds, Model II, and the dotted line, Model III.

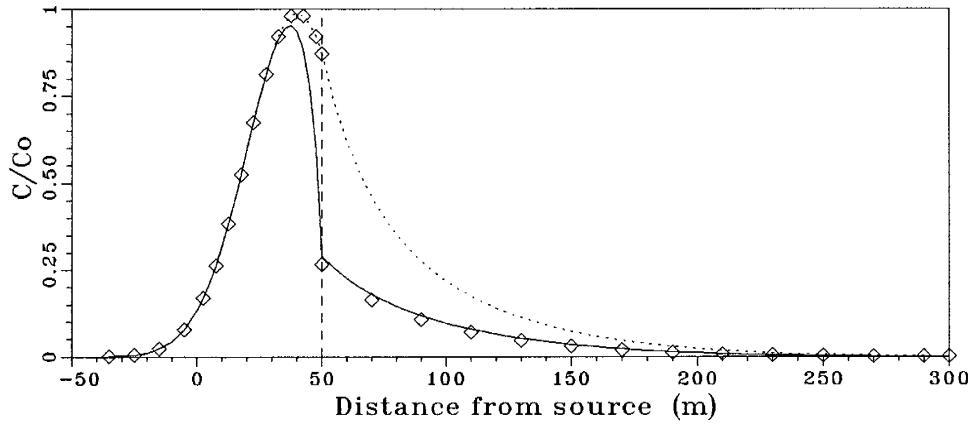

Figure *14a*.

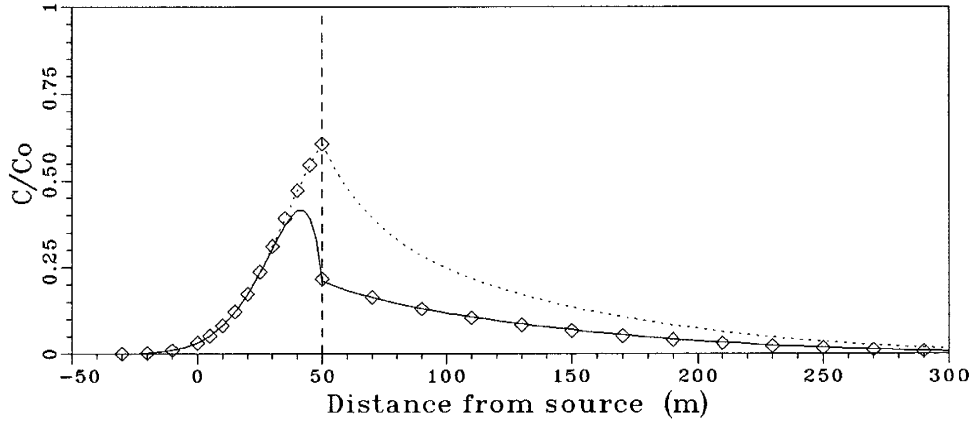

Figure *14b*.

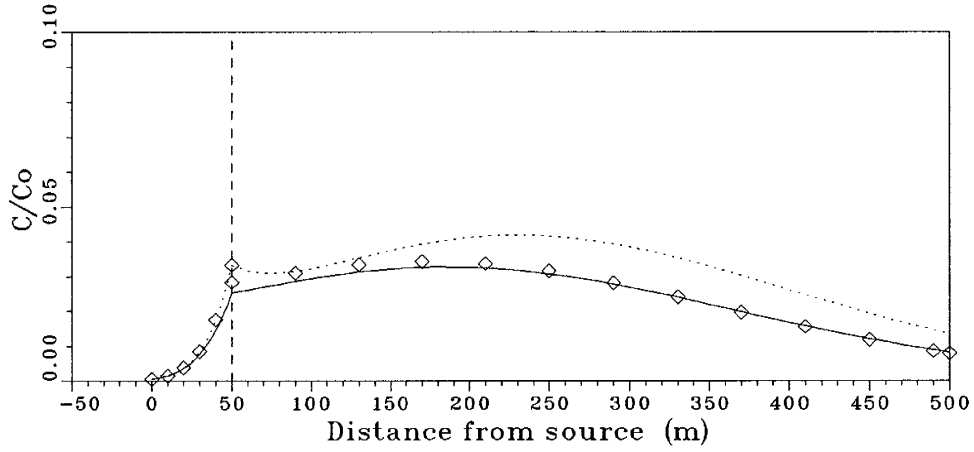

Figure *14c*.

Figure *14*. Concentration distribution (using Case *3* transport parameters given in Table *3*) at t = *400, 600,* and *1250*. The solid line represents the solution to Model I, the diamonds, Model II, and the dotted line, Model III.

Figure *14c* is vertically exaggerated by a factor of ten in order to allow the differences

between the solutions to be seen. The *x*-axis is also extended in this figure to display the extent of the dispersion effect. Although the absolute differences between the curves in Figure *14* are small, the relative error is quite high. Unlike Cases *1* and *2* of the field scale analysis, Model III grossly overestimates the solute concentration when using the parameters of the third case. This is the same behavior (except to a more excessive degree) that was observed on the laboratory scale.

The fourth case examines the effects of an increase in the dispersion coefficients in the second layer with the seepage velocities remaining the same over the domain. The transport parameters for this case can be found in table *3*. The breakthrough curves generated from these parameters are shown in Figures *15a*, *15b*, and *15c*.

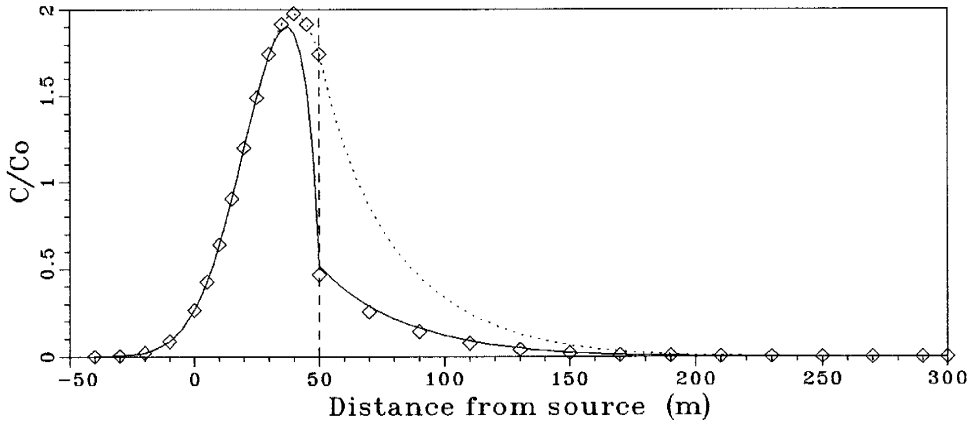

Figure *15a*.

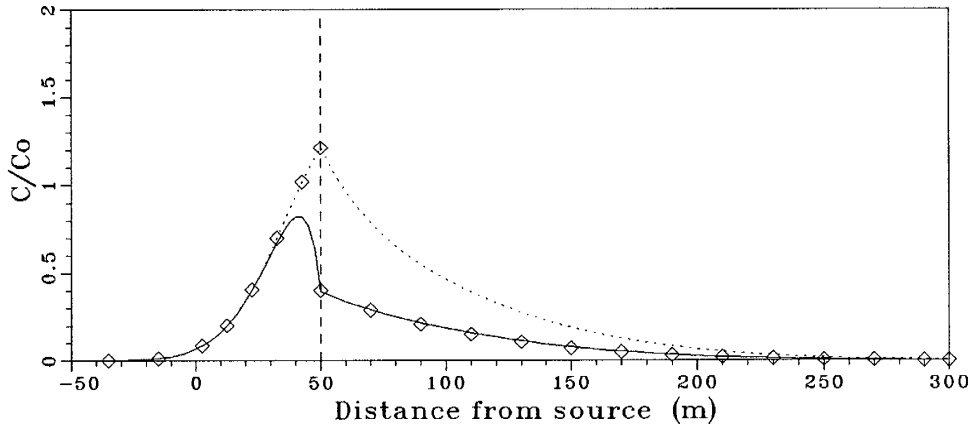

Figure *15b*.

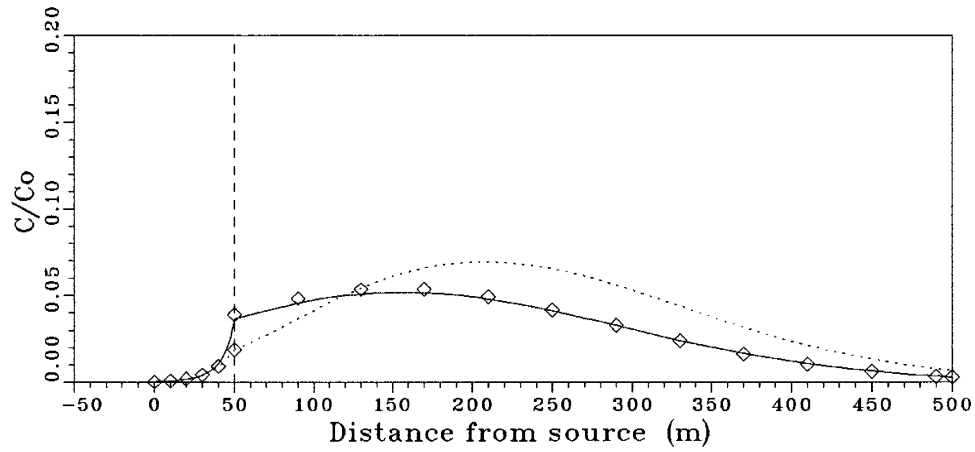

Figure *15c*.

Figure *15*. Concentration distribution (using Case *4* transport parameters given in Table *3*) at t = *400, 600,* and *1500*. The solid line represents the solution to Model I, the diamonds, Model II, and the dotted line, Model III.

Figure *15c*'s vertical axis is exaggerated by a power of ten to make examination of the differences between the solutions easier. The solutions express the same behavior with this set of parameters as they did with the parameters used in the third case. Again, Model III overestimates the solute concentration as it has in all tests where the dispersion coefficients were larger in the second layer than in the first.

The two-dimensional nature of the solutions is shown in figures *16* and *17*. Figure *16* shows the output from the solutions to Models II and III using the tranport parameters of Case *1*, Table *3*.

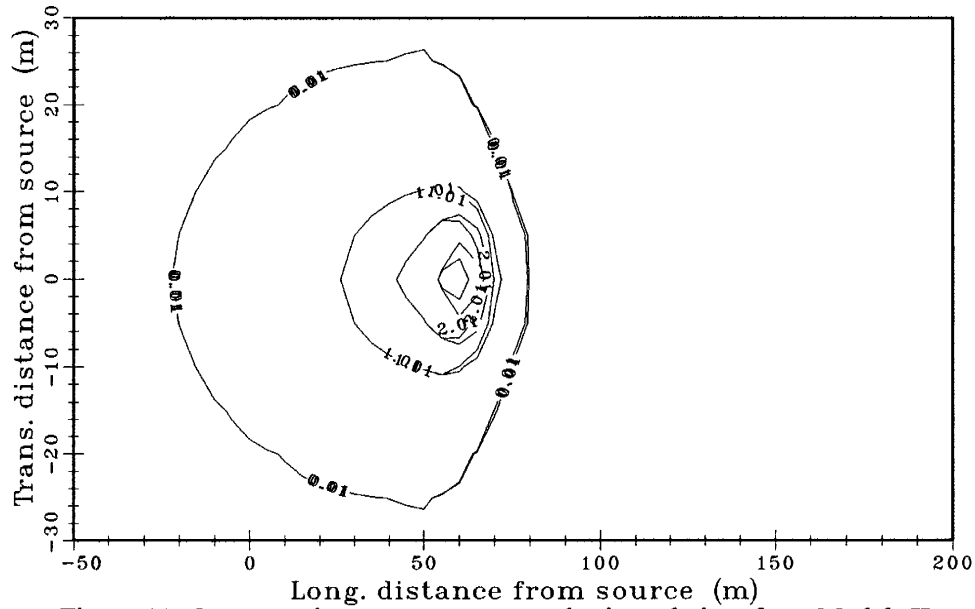

Figure *16*. Concentration contours generated using solutions from Models II and III (and Case *1* transport parameters given in Table *3*) at t=*600*. The interface between layers is located at x=*50*.

When compared to Figure *10* the contour plot shown above depicts a larger difference between the concentrations predicted by the two solutions. This is likely due to the greater variation in dispersion coefficients employed in Figure *16*. Model III underestimates the travel distance of the solute when compared to Model II as it also did in Figure *10*. Figure *17* shows the same two solutions using the parameters from Case *3*.

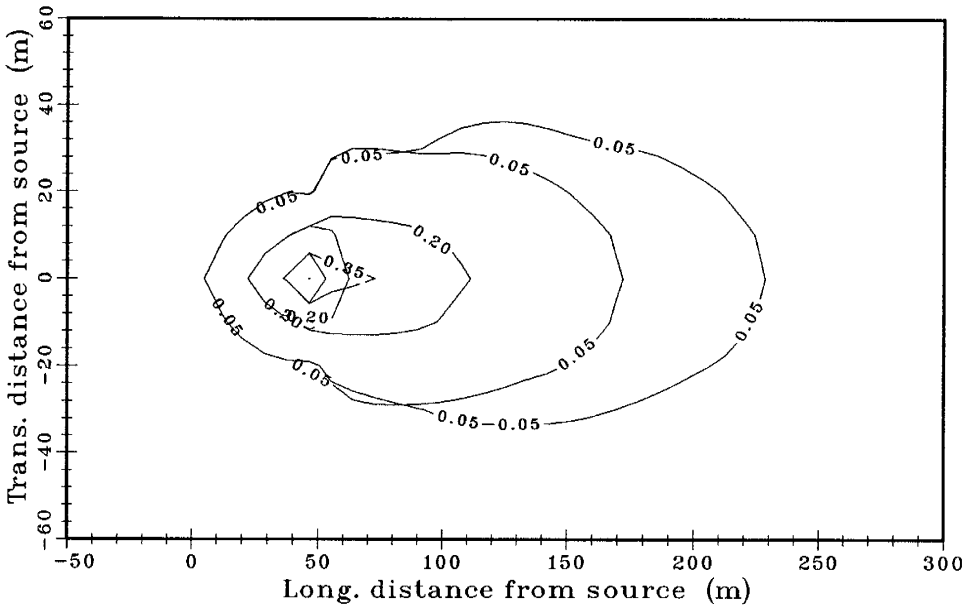

Figure *17*. Concentration contours genereated using solutions from Models II and III (and Case *3* transport parameters given in Table *3*) at $t=600$. The interface between layers is located at $x=50$.

The differences between the solutions are difficult to examine but the transverse spreading of the plume is evident. The larger transverse dispersion coefficients of Case *3* produce a wider spreading of the plume in Figure *17* than in Figure *16*. Comparing the concentrations generated by the two solutions in figure 17 portrays a grossly overestimated travel distance by Model III. As in the laboratory scale analysis, similar differences between the solutions are observed in the transverse direction as they are in the longitudinal direction; when the dispersion coefficients are greater in the first layer than in the second, Model III underestimates the solute concentration in both the longitudinal and transverse directions. When the dispersion coefficients inherent to the second layer are greater, Model III overestimates the solute concentration in both spreading directions.

In summary of the field scale analysis, all three solutions give predictable results for all parameter sets. The Model III solution underestimated the solute concentration when compared to Models I and II in cases where the dispersion coefficients were decreased in the second layer. Model III overestimated the solute concentration when the dispersion coefficients were increased in the second layer. The difference and the

relative error between the Model III solution and the other two solutions are greater on the field scale. Large differences in dispersion coefficients between the layers have a major effect on the differences between the predicted concentrations of the three solutions. Models I and II continue to give the same results in the second layer while Model III deviates from the other two under the field conditions.

A simple criterion involving the dispersion coefficients can be established to determine whether Model III may be applied in lieu of Models I or II. Table *4* lists the ratios of the longitudinal dispersion coefficients for the eight laboratory scale and field scale cases examined in this section.

Comparison of the Ratios of Longitudinal Dispersion Coefficients for the Eight Laboratory and Field Cases Examined	
Set of Transport Parameters	D_{L_1}/D_{L_2}
Case 1, Table 2	10.0
Case 2, Table 2	5.0
Case 3, Table 2	0.25
Case 4, Table 2	0.5
Case 1, Table 3	100.0
Case 2, Table 3	50.0
Case 3, Table 3	0.025
Case 4, Table 3	0.05

Table *4*

If the ratio of longitudinal dispersion coefficients is approximately greater than *10.0* or less than *0.1* (less than an order of magnitude difference between the coefficients), then Model I or Model II should be utilized for determining solute concentration. The analysis of these three solutions lead to some basic conclusions about using analytical solutions for solute transport through heterogeneous media.

5: CONCLUSIONS AND RECOMMENDATIONS

When concerned with laboratory scale transport of contaminants miscible with water, the three pairs of solutions produce very similar results. Their similarity is due to the small scale variation of the dispersion coefficients. In modeling solute transport on a laboratory scale any of the three solutions should correspond well to measured data values.

Results from Model III differ greatly from the results of the other two models when transport parameters inherent in field scale measurements are incorporated. Model III does not yield results similar to the other two models when a large variation in the dispersion coefficients between the two layers exists; it is therfore not recommended for use. In all cases, Model I and Model II bear identical results in the second layer and generate different results in the first layer only when in close proximity to the layer interface. It is recommended here that Model II be used to model solute transport. Model I is not recommended because it creates problems due to its mathematical complexity. Model I requires numerical inversion and numerical integration methods to be applied to the solutions to provide values of concentration; this is difficult and time consuming. Only in cases where the concentration must be known very accurately in a location of close proximity to the interface while still in the first layer would the recommendation be made that the solutions to Model I be used.

This paper is not addressing the issue of scale-dependent dispersivity. The cases mentioned above are examined to show the three models give reasonable results when using dispersivity values over a wide range. In both the laboratory and field scales the concern that the solute has travelled a sufficient distance for the dispersivity to become constant is taken into account. The breakthrough curves were generated after sufficient time had passed for the center of mass of the solute plume to have moved approximately ten dispersivity lengths from the solute source, the assumption of constant dispersivity being more valid after this has occurred.

The techniques employed in this research to create solutions to two-dimensional solute transport normal to layering may also be used to determine solutions in three dimensions; this work is recommended as a further study. The Fourier cosine transform may be applied to both of the spatial coordinates parallel to the layer interface and the Laplace transform to the time variable in order to reduce the governing equation to an ordinary differential equation for which a unique solution may be derived.

6: REFERENCES

Al-Niami, A. N. S., and K. R. Rushton, Dispersion in stratified porous media: analytical solutions, *Water Resources Research,* 15(5), 1044-1048, 1979.

Aris, R., On the dispersion of a solute in a fluid flowing in a tube, *Proc. Royal Soc. London Ser. A,* 235, 67-77, 1956.

Aris, R., On the dispersion of kinematic waves, *Proc. Royal Soc. London,* A245, 268-277, 1958.

Bachmat, Y., and J. Bear, The general equations of hydrodynamic dispersion in homogeneous, isotropic porous medium, *Journal of Geophysical Research,* 69, 2561-2567, 1964.

Barry, D. A., and J. C. Parker, Approximations for solute transport through porous media with flow transverse to layering, *Transport in Porous Media,* 2, 65-82, 1987.

Bateman, H., *Tables of integral transforms,* compiled by the staff of the Bateman manuscript project, vol. 1, 125-303, McGraw-Hill Book Co., Inc., New York, 1954.

Bear, J., Some experiments in dispersion, *Journal of Geophysical Research,* 66(8), 2455-2468, 1961.

Bear, J., *Dynamics of fluids in porous media,* 579-664, American Elsevier, New York, 1972.

Blackwell, R. J., Laboratory studies of microscopic dispersion phenomena, *Journal of Society of Petroleum Engineering,* 2(1), 1-8, 1962.

Boas, M. L., *Mathematical methods in the physical sciences,* 635-685, John Wiley and Sons, New York, 1983.

Boyce, W. E., and R. C. DiPrima, *Elementary differential equations and boundary value problems,* 132-162, John Wiley and Sons, New York, 1986.

Bredehoeft, J. D., and G. F. Pinder, Mass transport in flowing groundwater, *Water Resources Research,* 9(1), 194-210, 1973.

Bresler, E., and G. Dagan, Unsaturated flow in spatially variable fields, 3, Solute transport models and their application to two fields, *Water Resources Research,*

19(2), 429-435, 1983.

Chen, C. S., Solutions for radionuclide transport from an injection well into a single fracture in a porous formation, *Water Resources Research,* 22(4), 508-518, 1986.

Chen, C. S., and G. D. Woodside, Analytical solution for decontamination by pumping, *Water Resources Research,* 24(8), 1329-1338, 1988.

Chen, C. S., Solutions approximating solute transport in a leaky aquifer receiving wastewater injection, *Water Resources Research,* 25(1), 61-72, 1989.

Crump, K. S., Numerical inversion of Laplace transforms using a Fourier series approximation, *Journal of the Association for Computing Machinery,* 23, 89-96, 1976.

de Boor, C., *ACM Monograph Series, Mathematical Software,* Edited by Rice, Academic Press, New York, 1971.

de Josselin de Jong, G., Longitudinal and transverse diffusion in granular deposits, *EOS Trans., AGU,* 39(1), 67-74, 1958.

Domenico, P. A., and G. A. Robbins, A dispersion scale effect in model calibrations and field tracer experiments, *Journal of Hydrology,* 70, 123-132, 1984.

Fried, J. J., *Groundwater pollution,* 47-48, Elsevier, Amsterdam, 1975.

Gershon, N. D., and A. Nir, Effect of boundary conditions of models on tracer distribution in flow through porous mediums, *Water Resources Research,* 5(4), 830-840, 1969.

Gradshteyn, I. S., and I. M. Ryzhik, *Table of integrals, series, and products; corrected and enlarged edition,* Academic Press, Orlando, 1980.

Gureghian, A. B. and G. Jansen, One-dimensional analytical solutions for the migration of a three-member radionuclide decay chain in a multilayered geologic medium, *Water Resources Research,* 21, 733-742, 1985.

Güven, O., F. J. Molz, and J. G. Melville, An analysis of dispersion in a stratified aquifer, *Water Resources Research,* 20(10), 1337-1354, 1984.

Hantush, M. S., Analysis of data from pumping tests in leaky aquifers, *Transactions,* American Geophysical Union, 37(6), 702-714, 1956.

Harleman, D. R., and R. R. Rumer, Longitudinal and lateral dispersion in an isotropic porous medium, *Journal of Fluid Mechanics,* 16(3), 385-394, 1963.

Hibsch, G., and A. Kreft, Determinationn of aquifer transport parameters, *J. Hyd. Div., Proc. Am. Soc. Civ. Eng.,* 105(HY9), 1137-1151, 1979.

International Mathematics and Statistics Library, Inc., *IMSL User's Manual*, vol. 2, Houston, TX 1987.

Klotz, D., K. P. Seiler, H. Moser, and F. Neumaier, Dispersivity and velocity relationship from laboratory and field experiments, *Journal of Hydrology,* 45, 169-184, 1980.

Kreft, A., and A. Zuber, On the physical meaning of the dispersion equation and its solution for different initial and boundary conditions, *Chemical Engineering Science,* 33, 1471-1480, 1978.

Lenda, A., and A. Zuber, Tracer dispersion in groundwater experiments, *Isotope Hydrology, IAEA, Vienna,* 619-641, 1970.

Murty, V. V. N., and V. H. Scott, Determination of transport model parameters in groundwater aquifers, *Water Resources Research,* 13(6), 941-947, 1977.

Oberhettinger, F., and L. Badii, *Tables of Laplace Transforms,* Springer-Verlag, 1973.

Ogata, A. and R. B. Banks, Solution of the differential equation of longitudinal dispersion in porous media, *U.S. Geological Survey, Prof. Paper 411-A,* 7pp., 1961.

Parker, J. C., Analysis of solute transport in column tracer studies, *Soil Science Society of America,* 48, 719-724, 1984.

Parker, J. C., and M. Th. van Genuchten, Flux-averaged and volume-averaged concentrations in continuum approaches to solute transport, *Water Resources Research,* 20(7), 866-872, 1984.

Pinder, G. F., A Galerkin-finite element simulation of groundwater contamination on Long Island, New York, *Water Resources Research,* 9(6), 1657-1669, 1973.

Selim, H. M., J. M. Davidson, and P. S. C. Rao, Transport of reactive solutes through multilayered soils, *Soil Science Society of America, Division S-1 (Soil Physics)*, 41, 3-10, 1977.

Shamir, U. Y., and D. R. Harleman, Dispersion in layered porous media, *J. Hyd. Div., Proc. Am. Soc. Civ. Eng.,* 93, 236-260, 1967.

Sneddon, I. N., *Fourier transforms,* International series in pure and applied mathematics, Mc Graw-Hill, New York, 1-29, 1951.

Spiegel, M. R., *Theory and problems of Laplace transforms,* Schaum's outline series in mathematics, Mc Graw-Hill, New York, 1965.

Stehfest, H., Algorithm 368: Numerical inversion of Laplace transforms, *Comm. ACM,* 13, 47-48, 1970.

Valocchi, A. J., Theoretical analysis of deviations from local equilibrium during sorbing solute transport through idealized stratified aquifers, *J. Contam. Hydrol.,* 2, 191-207, 1988.

van Duijn, C. J., and S. E. A. T. M. van der Zee, Solute transport parallel to an interface separating two different porous materials, *Water Resources Research,* 22(13), 1779-1789, 1986.

van Genuchten, M. Th., and W. J. Alves, Analytical solutions of the one-dimensional convection-dispersion equation, *U.S. Department of Agriculture, Technical Bulletin No. 1661,* 151 p., 1982.

Verriujt, A., Steady dispersion across an interface in a porous medium, *Journal of Hydrology,* 14, 337-347, 1971.

Wilson, J. L., and P. J. Miller, Two-dimensional plume in uniform ground-water flow, *J. Hyd. Div., Proc. Am. Soc. Civ. Eng.,* 104(HY4), 503-514, 1978.

APPENDIX A: SOLUTION DETERMINATION FOR MODEL I:

The derivation of the general solution for Model I is slightly more difficult than the derivations of the other two models. The general solution to the first layer will be developed to the point where the solutions governing transport in the two layers must be coupled. Following this derivation the general solution for the second layer will be carried out as completely as possible. The remaining unknown constants within the two solutions will then be solved by simultaneously applying the boundary conditions at the interface. The ordinary differential equation and the boundary conditions defining transport in the first layer were given as (32a) and (33) and are restated here.

$$D_{L_1} \frac{\partial^2 G_1}{\partial x^2} - v_1 \frac{\partial G_1}{\partial x} - (p + D_{T_1} s^2) G_1 = h(x) \tag{A1}$$

where

$$h(x) = -\frac{M}{\sqrt{2\pi}\, n_1 B} \delta(x)$$

Boundary Conditions:

$$G_1(x, s, p) = 0 \qquad x \to -\infty \tag{A2a}$$

$$G_1(L, s, p) = G_2(L, s, p) \tag{A2b}$$

Equation (A1) is an inhomogeneous equation with constant coefficients. The method of variation of parameters was employed to derive a solution. It involves determining a particular solution and a complimentary solution to create a general solution from the two. The complimentary solution, a solution to an equation similar to (A1) with the right side equal to zero, is found using the method of characteristic equations and roots. First, a solution is sought to the differential equation having the general form

$$e^{\beta x}$$

Substituting this form for, G_1, the differential equation becomes

$$D_{L_1} \frac{\partial^2 e^{\beta x}}{\partial x^2} - v_1 \frac{\partial^2 e^{\beta x}}{\partial x} - (p + D_{T_1} s^2) e^{\beta x} = 0 \tag{A3}$$

Taking the derivatives of the exponential function results in the characteristic equation

$$e^{\beta x}\left[D_{L_1}\beta^2 - v_1\beta - (p + D_{T_1}s^2)\right] = 0 \tag{A4}$$

Since $e^{\beta x}$ is not zero, the above equation can be rewritten as

$$\left[D_{L_1}\beta^2 - v_1\beta - (p + D_{T_1}s^2)\right] = 0 \tag{A5}$$

If β is a root of this characteristic equation then $e^{\beta x}$ is a solution of equation (A3). Recognizing that (A5) is a simple quadratic equation, the roots of the characteristic equation are given by

$$\beta_1 = \frac{v_1 - \left[v_1{}^2 + 4D_{L_1}(p + D_{T_1}s^2)\right]^{1/2}}{2D_{L_1}} \tag{A6a}$$

and

$$\beta_2 = \frac{v_1 + \left[v_1{}^2 + 4D_{L_1}(p + D_{T_1}s^2)\right]^{1/2}}{2D_{L_1}} \tag{A6b}$$

Both β_1 and β_2 are roots of the characteristic equation. The two linearly independent solutions created with the roots, β_1 and β_2, are added together to produce the complimentary solution to (A1) which is given by

$$G_{complimentary} = k_1 e^{(\lambda_1 + \lambda_3)x} + k_2 e^{(\lambda_1 - \lambda_3)x} \tag{A7}$$

where

$$\lambda_1 = \frac{v_1}{2D_{L_1}}$$

$$\lambda_3 = \frac{\left[v_1{}^2 + 4D_{L_1}(p + D_{T_1}s^2)\right]}{2D_{L_1}}$$

The values, k_1 and k_2, are temporary unknown constants. Two additional functions,

g_1 and g_2, are defined below for convenience.

$$g_1(x) = e^{(\lambda_1 + \lambda_3)x} \tag{A8a}$$

$$g_2(x) = e^{(\lambda_1 - \lambda_3)x} \tag{A8b}$$

The functions, g_1 and g_2, are two linearly independent solutions to the homogeneous equation. To derive the particular solution, the constants, k_1 and k_2, are replaced in equation (A7) by two functions, $\mu_1(x)$ and $\mu_2(x)$. The particular solution has the form

$$G_{particular} = \mu_1(x)\, g_1(x) + \mu_2(x)\, g_2(x) \tag{A9}$$

The derivatives of the functions, $\mu_1(x)$ and $\mu_2(x)$ are shown in literature (e.g. Boyce and DiPrima [1986]) to be equal to

$$\mu_1'(x) = -\frac{g_2(x)h(x)}{W(g_1, g_2)} \tag{A10a}$$

$$\mu_2'(x) = \frac{g_1(x)h(x)}{W(g_1, g_2)} \tag{A10b}$$

where

$$W(g_1, g_2) = g_1 g_2' - g_1' g_2$$

The function, $W(g_1, g_2)$, is termed the Wronskian and the primed functions denote the derivatives with respect to x of those functions. After some manipulation the functions, μ_1' and μ_2' become

$$\mu_1'(x) = \frac{M}{4 n_1 B D_{L_1} \lambda_3}\, \delta(x)\, e^{(-\lambda_1 - \lambda_3)x} \tag{A11a}$$

$$\mu_2'(x) = \frac{M}{4 n_1 B D_{L_1} \lambda_3}\, \delta(x)\, e^{(\lambda_1 - \lambda_3)x} \tag{A11b}$$

The final form of these two functions upon integration with respect to x is

$$\mu_1(x) = k_3 - \frac{M}{4n_1 B D_{L_1} \lambda_3} H(x) \tag{A12a}$$

$$\mu_2(x) = k_4 + \frac{M}{4n_1 B D_{L_1} \lambda_3} H(x) \tag{A12b}$$

where

$$H(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x > 0 \end{cases}$$

The function, $H(x)$, is termed the heaviside function. The values of k_3 and k_4 are unknown and will be evaluated by imposing the boundary conditions on the solution. The general solution is obtained by adding together the complimentary and particular solutions which results in the following solution:

$$G_1 = \left\{k_1 - \frac{M}{4n_1 B D_{L_1} \lambda_3} H(x)\right\} e^{(\lambda_1 + \lambda_3)x} + \left\{k_2 + \frac{M}{4n_1 B D_{L_1} \lambda_3} H(x)\right\} e^{(\lambda_1 - \lambda_3)x} \tag{A13}$$

The unknown constants, $k_1, k_2, k_3,$ and k_4, have been added together and combined into two constants, k_1 and k_2. Boundary condition (A2a) can now be applied to eliminate one of the unknown constants. For boundary condition (A2a) to be satisfied, the unknown constant contained in the term

$$\left\{k_2 + \frac{M}{4n_1 B D_{L_1} \lambda_3} H(x)\right\} e^{(\lambda_1 - \lambda_3)x}$$

must be identically zero since the exponential portion of this term is diverging from zero for negative values of x. The general solution in its temporary form is given as

$$G_1 = \left\{k_1 - \frac{M}{4n_1 B D_{L_1} \lambda_3} H(x)\right\} e^{(\lambda_1 + \lambda_3)x} \tag{A14}$$

The last unknown constant, k_1 cannot be evaluated until the solution to solute transport in the second layer is known.

The solution procedure for the second layer is very similar to the procedure used in finding the complimentary solution for the first layer. The governing ordinary differential equation (32b) and boundary conditions (35) are restated here for

convenience.

$$D_{L_2}\frac{\partial^2 G_2}{\partial x^2} - v_2\frac{\partial G_2}{\partial x} - (p + D_{T_2}s^2)G_2 = 0 \qquad \text{(A15)}$$

Boundary Conditions:

$$n_1\left[v_1 G_1 - D_{L_1}\frac{\partial G_1}{\partial x}\right] = n_2\left[v_2 G_2 - D_{L_2}\frac{\partial G_2}{\partial x}\right] \qquad at\ x = L \qquad \text{(A16a)}$$

$$G_2(x, s, p) = 0 \qquad x \to \infty \qquad \text{(A16b)}$$

Using the method of characteristic equations explained in the beginning of this appendix, the solution displays the following familiar form:

$$G_2 = k_5 e^{(\lambda_2 + \lambda_4)x} + k_6 e^{(\lambda_2 - \lambda_4)x} \qquad \text{(A17)}$$

where

$$\lambda_2 = \frac{v_2}{2D_{L_2}}$$

$$\lambda_4 = \frac{\left[v_2^2 + 4D_{L_2}(p + D_{T_2}s^2)\right]}{2D_{L_2}}$$

The values, k_5 and k_6, are temporary unknown constants. Boundary condtion (A16b) can be immediately applied to show that k_5 must be identically zero to restrain the solution from tending towards infinity as x becomes very large. The general solution to solute transport in the second layer is now given by

$$G_2 = k_6 e^{(\lambda_2 - \lambda_4)x} \qquad \text{(A18)}$$

The two remaining boundary conditions at the layer interface must be applied and the two general solutions solved simultaneously as a system of equations. Boundary condition (A16a) can be simplified in this case. Since boundary condition (A2b) has also been imposed at the interface and since there must exist a water mass flux

across the boundary, the following identity is true:

$$n_1 v_1 G_1 = n_2 v_2 G_2$$

With this in mind, boundary condition (A16a) can be simplified to

$$n_1 D_{L_1} \frac{\partial G_1}{\partial x} = n_2 D_{L_2} \frac{\partial G_2}{\partial x} \tag{A19}$$

By substituting the equations (A13) and (A18) into boundary conditions (A2b) and (A19), the unknown constants, k_1 and k_6, can be determined. The final form of the solution governing solute transport in the first layer of Model I is given by

$$G_1 = A_3 e^{\left[\lambda_1 x + \lambda_3 (x - 2L)\right]} + A_4 \left\{ e^{\left[(\lambda_1 + \lambda_3)x\right]} - e^{\left[\lambda_1 x + \lambda_3 (x - 2L)\right]} \right\}$$

$$+ A_4 \left\{ H(x)\, e^{\left[(\lambda_1 - \lambda_3)x\right]} - H(x)\, e^{\left[(\lambda_1 + \lambda_3)x\right]} \right\} \tag{A20}$$

where

$$A_3 = \frac{M}{\sqrt{2\pi}\, B} \left[\frac{1}{n_1 D_{L_1} (\lambda_1 + \lambda_3) - n_2 D_{L_2} (\lambda_2 - \lambda_4)} \right]$$

$$A_4 = \frac{M}{2\sqrt{2\pi} n_1 B D_{L_1} \lambda_3}$$

The final form of the solution for the second layer of Model I has the form

$$G_2 = A_3 e^{\left[(\lambda_1 - \lambda_3) L + (\lambda_2 - \lambda_4)(x - L)\right]} \tag{A21}$$

APPENDIX B: SOLUTION DETERMINATION FOR MODEL II:

The method of finding the general solution for this model is identical to the method presented in Appendix A for finding the solution for the second layer of Model I. Only the details of employing the boundary conditions will be outlined here. The governing differential equation is given by (A15) and the remaining boundary conditions for the second layer of model two are restated here as

Boundary Conditions:

$$n_1\left[v_1 G_1 - D_{L_1}\frac{\partial G_1}{\partial x}\right] = n_2\left[v_2 G_2 - D_{L_2}\frac{\partial G_2}{\partial x}\right] \qquad at\ x = L \qquad \text{(B1a)}$$

$$G_2(x, s, p) = 0 \qquad x \to \infty \qquad \text{(B1b)}$$

Equation (A15) is a homogeneous equation with constant coefficients. As in Appendix A, the characteristic equation is developed from (A15) and is of a similar form as (A5). The roots of this equation are given as (A6a) and (A6b). Since the characteristic roots are presently known, the general solution may be written in the following form:

$$G_2 = h_1 e^{\beta_1 x} + h_2 e^{\beta_2 x} \qquad \text{(B2)}$$

where h_1 and h_2 are two unknown constants. These two constants will be evaluated by applying the two boundary conditions. At this point the results begin to differ from those in Appendix A. First, boundary condition (B1b) is input to the solution to give

$$G_2(x, s, p) = h_1 e^{\beta_1 x} + h_2 e^{\beta_2 x} = 0 \qquad x \to \infty \qquad \text{(B3)}$$

The value of β_1 is always negative while the value of β_2 is always positive. For equation (B3) to approach zero as x approaches infinity, the constant, h_2, must be zero. This reduces the general solution to

$$G_2(x, s, p) = h_1 e^{\beta_1 x} \qquad \text{(B4)}$$

To apply boundary condition (B1a) in the pursuit of determining h_1, the value for G_1 must be known. The value of C_1 is known and given as (51). This function for C_1 must be transformed using the Laplace and Fourier cosine integral transforms. By making use of (12) and (13) and by applying the Fourier cosine transform the value of G_1 is known to be

$$G_1(L, s, p) = A_1 e^{\gamma_1 L} \tag{B5}$$

where

$$A_1 = \frac{M}{\sqrt{2\pi}\, n_1 B \left[v_1{}^2 + 4D_{L_1}(p + D_{T_1} s^2)\right]^{1/2}}$$

$$\gamma_1 = \frac{v_1 - \left[v_1{}^2 + 4D_{L_1}(p + D_{T_1} s^2)\right]^{1/2}}{2D_{L_1}}$$

The value of h_1 can be found by substituting (B4) and (B5) into boundary condition (B1a) and evaluating at $x = L$. This substitution provides the following set of equations:

$$n_1 \left(v_1 A_1 e^{\gamma_1 L} - D_{L_1} A_1 \gamma_1 e^{\gamma_1 L}\right) = n_2 \left(v_2 h_1 e^{\beta_1 L} - D_{L_2} h_1 \beta_1 e^{\beta_1 L}\right) \tag{B6a}$$

$$h_1 = \frac{n_1 v_1 - n_1 D_{L_1} \gamma_1}{n_2 v_2 - n_2 D_{L_2} \beta_1} A_1 e^{(\gamma_1 - \beta_1)L} \tag{B6b}$$

where (B6b) originates from solving (B6a) for h_1. After further manipulation to the constant, h_1, the general solution for the second layer of model two is given by

$$G_2 = A_2 e^{\left[\gamma_1 L + \beta_1 (x - L)\right]} \tag{B7}$$

where

$$A_2 = \frac{M}{\sqrt{2\pi}\, n_2 B} \left[\frac{1}{v_2 + \left[v_2{}^2 + 4D_{L_2}(p + D_{T_2} s^2)\right]^{1/2}}\right] \cdot$$

$$\left[\frac{v_1}{\left[v_1{}^2 + 4D_{L_1}(p + D_{T_1} s^2)\right]^{1/2}} + 1\right]$$

APPENDIX C: SOLUTION DETERMINATION FOR MODEL III:

This appendix details the derivation of the characteristic roots from the characteristic equation and the production of the general solution from these roots for Model III. The governing differential equation is given by (A15) and the remaining boundary conditions for the second layer of Model III are restated here as

Boundary Conditions:

$$G_1(L, s, p) = G_2(L, s, p) \tag{C1a}$$

$$G_2(x, s, p) = 0 \qquad x \to \infty \tag{C1b}$$

Equation (A15) is recognized as a homogeneous equation with constant coefficients. To solve this ordinary differential equation the method of creating a characteristic equation is employed and this method is detailed in the derivation of the solution to the second layer of Model I in Appendix A. As in Appendix A, the characteristic equation is developed from (A15) and is of a similar form as (A5). The roots of this equation are given as (A6a) and (A6b). Since the characteristic roots are presently known, the general solution may be written in the following form:

$$G_2 = h_1 e^{\beta_1 x} + h_2 e^{\beta_2 x} \tag{C2}$$

where h_1 and h_2 are two unknown constants. These two constants will be evaluated by applying the two boundary conditions. First, boundary condition (C1b) is input to the solution to give

$$G_2(x, s, p) = h_1 e^{\beta_1 x} + h_2 e^{\beta_2 x} = 0 \qquad x \to \infty \tag{C3}$$

The value of β_1 is always negative while the value of β_2 is always positive. To keep equation (C3) from diverging from zero as x approaches infinity, the constant, h_2 must be zero. This reduces the general solution to

$$G_2(x, s, p) = h_1 e^{\beta_1 x} \tag{C4}$$

Boundary condition (C1a) is now applied to determine the value h_1 .

$$G_1(L, s, p) = G_2(L, s, p) = h_1 e^{\beta_1 x} \tag{C5}$$

Solving the above equation for h_1 yields

$$h_1 = G_1 e^{-\beta_1 L} \tag{C6}$$

The value for G_1 is derived in Appendix B and is given as (B5). This identity can be substituted for G_1 in (C6) and the value of h_1 becomes

$$h_1 = A_1 e^{\left[\gamma_1 L - \beta_1 L\right]} \tag{C7}$$

The general solution for G_2 can now be written in final form as

$$G_2 = A_1 e^{\left[\gamma_1 L - \beta_1 (x - L)\right]} \tag{C8}$$

APPENDIX D: COMPUTER PROGRAM FOR EVALUATING SOLUTIONS

```
C*****************************************************************
C       This program combines the three solutions to produce
C     output using all three. The first analytical solution is the
C     solution to resident
C     concentrations and uses the boundary condition
C                        C1=C2              at the interface.
C     This particular version of the program only looks at calculations
C     along one y-direction.
C     The second solution is the solution to
C     resident concentrations and uses the boundary condition
C           N1(V1C1-DL1*dC1/dX)=N2(V2C2-dC2/dX) at the interface.
C     This particular version of the program only looks at calculations
C     along one y-direction.
C       The third solution is the solution to resident
C     concentrations and uses two boundary conditions,
C          1. N1(V1C1-DL1*dC1/dX)=N2(V2C2-dC2/dX)
C          2. C1=C2                    at the interface.
C     This solution has a truly finite boundary at the interface.
C     This particular version of the program only looks at calculations
C     along one y-direction.
C     The solution to the second layer is written entirely in
C     exponential form to help cure underflow and overflow problems.
C       We have an integral that has a lower limit of some finite
C     number and an upper limit of infinity. A subroutine is included
C     in this program to allow the user to look at the behavior of the
C     integral to gain an understanding of an appropriate value for
C     infinity.
C       The Stehfast inversion with 32-point Gaussian quadrature
C     is used to evaluate the third solution. The adaptive rhomberg
C     extrapolation routine is used to evaluate the first and second
C     solutions.
C       The Crump inversion routine was executed using a different
C     program. The format was exactly the same but instead of using
C     the subroutines, COEFF and STEHINV, to do the Stehfest inversion
C     an subroutine call to IMSL was executed.
C*****************************************************************
C
C*****************************************************************
C  PARAMETER   TYPE     DESCRIPTION
C-----------------------------------------------------------------
C  X          [L]      direction normal to layers, X=0 at
C                        concentration source
C  FACE       [L]      point where boundary between layers exists
```

```
C  v1       [L/T]     velocity in the first layer
C  Dl1      [L**2/T]  longitudinal dispersion in first layer
C  Dt1      [L**2/T]  transverse dispersion in first layer
C  Al1      [L]       longitudinal dispersivity in first layer
C  At1      [L]       transverse dispersivity in first layer
C  v2       [L/T]     velocity in second layer
C  Dl2      [L**2/T]  longitudinal dispersion in second layer
C  Dt2      [L**2/T]  transverse dispersion in second layer
C  Al2      [L]       longitudinal dispersivity in second layer
C  At2      [L]       transverse dispersivity in second layer
C  T        [T]       time since injection of tracer
C  U        [T]       dummy variable of integration for time
C  Y        [L]       distance parallel to layers away from
C                      source, y=0 at source
C  P        [?]       The Laplace inversion variable for time
C                      used for the third solution
C  S        [L?]      The Fourier cosine inversion variable for y
C                      used for the third solution.
C  por21    [-]       A ratio of porosities, n2/n1
C  neq      [-]       The number of the solution presently being
C                      evaluated
C  sola     [function] The function governing flow of solute in
C                      the first layer for both approximate
C                      solutions.
C  solc     [function] The function governing flow of solute in the
C                      second layer for approximate solution using
C                      concentration continuity B.C.
C  solm1    [function] The first of two functions governing flow
C                      of solute in the second layer for appr.
C                      solution using mass flux B.C.
C  solm2    [function] The second of two functions governing
C                      flow of solute in the second layer for
C                      appr. solution using mass flux B.C.
C  V        [real array] An array containing the coefficients for
C                      Stehfast inversion.
C  sole1    [function] The first of three functions governing
C                      flow of solute in the first layer, "exact"
C                      solution.
C  sole2    [function] The second of three functions governing
C                      flow of solute in the first layer, "exact"
C                      solution.
C  sole3    [function] The third of three functions governing
C                      flow of solute in the first layer, "exact"
C                      solution.
C  sole4    [function] The function governing flow
C                      of solute in the second layer, "exact"
C                      solution.
```

```
C  GETDATA     [subroutine] Inputs and outputs the input data.
C  INT_BHVR    [subroutine] Looks at the behavior of the integrand
C                  to aid in determining a suitable value
C                  for infinity.
C  solcprint   [subroutine] Calculates and outputs the concentration
C                      at various values of X for approximate
C                      solution using concentration continuity B.C.
C  solmprint   [subroutine] Calculates and outputs the concentration
C                  at various values of X for approximate
C                  solution using mass flux B.C.
C  soleprint   [subroutine] Calculates and outputs the concentration
C                      at various values of X for exact solution
C                      using mass flux and continuity B.C.'s.
C  COEFF       [subroutine] Calculates the values of the stehfast
C                  inversion coefficients for any number
C                  of coefficients.
C  STEHINV     [function]  Numerically inverts the function out of
C                  the laplace domain prior to being inverted
C                  out of the fourier cosine domain.
C  DQG32       [subroutine] 32-point Gaussian quadrature for
C                  evaluating integrals in "exact" solution.
C  CADRE       [subroutine] Numerically integrates the functions of the
C                      first and second solution using
C                  the cautious adaptive rhomberg extrapolation.
C*****************************************************************

C.....MAIN BODY OF PROGRAM .....
C.....

            PROGRAM MAIN

C.....
C
C   The main body is a skeleton which delegates authority to
C   various subroutines. The parameters, functions, and subroutines
C   are defined above and any other different parameters used in the
C   functions or subroutines will be defined where needed.
C.....

    implicit double precision (A-H,O-Z)
    dimension V(30)
    common/param/POR21,FACE,V1,AL1,DL1,DT1,V2,AL2,DL2,DT2,PI,DK2,DK,
   &          RL2,RL,xend,xbeg,scale
    CHARACTER*15 FINAME1,FINAME2,FINAME3

C   enter the input and output file names
    write(6,*)'ENTER THE INPUT FILENAME:'
```

```
read(5,'(A)')FINAME1
OPEN (UNIT=10,STATUS='OLD',FILE=FINAME1)

call GETDATA(T,STEP1,STEP2,Y,AERR,RERR,LEVEL,N)
call COEFF(V,N)
write(6,*)'Do you want to look at the integrand behavior for'
write(6,*)'  Solm? (1=yes)'
read(5,*)option1
neq=1
do while (option1 .eq. 1)
 call INT_BHVR(T,Y,neq,N,V)
 write(6,*)
 write(6,*)'Do you want to check the behavior again? (1=yes)'
 read(5,*)option1
end do
call solmprint(T,STEP1,STEP2,Y,AERR,RERR,LEVEL)

write(6,*)'Do you want to look at the integrand behavior for'
write(6,*)'  Solc? (1=yes)'
read(5,*)option1
neq=2
do while (option1 .eq. 1)
 call INT_BHVR(T,Y,neq,N,V)
 write(6,*)
 write(6,*)'Do you want to check the behavior again? (1=yes)'
 read(5,*)option1
end do
call solcprint(T,STEP1,STEP2,Y,AERR,RERR,LEVEL)

write(6,*)'Do you want to look at the integrand behavior for'
write(6,*)'  Sole? (1=yes)'
read(5,*)option1
neq=3
do while (option1 .eq. 1)
 call INT_BHVR(T,Y,neq,N,V)
 write(6,*)
 write(6,*)'Do you want to check the behavior again? (1=yes)'
 read(5,*)option1
end do
call soleprint(V,N,STEP1,STEP2,Y,T)

STOP
END
```

```
C*****************************************************************
C    Subroutines and functions follow
C*****************************************************************

C|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
C    Subroutine GETDATA
C     This subroutine obtains all the input from an input file and
C    also from the user and outputs these parameters to a file.
C|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
C  PARAMETER   TYPE    DESCRIPTION
C-----------------------------------------------------------------
C  DK        [-]     lower limit of integration=sqrt(face*face/
C                   (4*Dl1*T))
C  DK2       [-]     DK**2
C  pi                the value of pi
C  step1              The x-direction step size for calculating
C                       concentration in the first layer
C  step2              The x-direction step size for calculating
C                       concentration in the second layer
C  xbeg               The beginning point of calculations
C  xend               The ending point of calcuations in the x-dir.
C  scale              Used for comparing solutions against
C                       homogeneous data for test of accuracy so that
C                       the peak values will be the same after moving
C                       the interface.  An artifact of the definition
C                       of Co.  For most analyses, this is set to 1.
C  AERR               Absolute error for Cadre solver
C  RERR               Relative error for Cadre solver
C  Level              Level of error output for Cadre solver,
C                       (=1-5)
C  N                  Number of Stehfast coefficients
C*****************************************************************

      subroutine GETDATA(T,STEP1,STEP2,Y,AERR,RERR,LEVEL,N)

      implicit double precision (A-H,O-Z)
      CHARACTER*15 FINAME2,FINAME3
      common/param/POR21,FACE,V1,AL1,DL1,DT1,V2,AL2,DL2,DT2,PI,DK2,DK,
     &       RL2,RL,xend,xbeg,scale

      PI=DACOS(-1.D0)

C     Obtain the output file names
      read(10,'(A)')FINAME2
      read(10,'(A)')FINAME3

C     unit 20 outputs the input parameters and the data
```

```
C    unit 30 outputs only the data in a form ready for plotting by
C    DISSPLA

     OPEN (UNIT=20,STATUS='NEW',FILE=FINAME2,CARRIAGE CONTROL
    &     ='LIST')
     OPEN (UNIT=30,STATUS='NEW',FILE=FINAME3,CARRIAGE CONTROL
    &     ='LIST')

C    THE SIZE OF THE STEP IN THE FIRST LAYER IS DEFINED
     write(6,*)'ENTER THE VALUE OF THE DISTANCE STEP'
     write(6,*)'IN THE FIRST LAYER.'
     read(5,*)STEP1

C    THE SIZE OF THE STEP IN THE SECOND LAYER IS DEFINED
     write(6,*)'ENTER THE VALUE OF THE DISTANCE STEP'
     write(6,*)'IN THE SECOND LAYER.'
     read(5,*)STEP2

C    A scale size is needed if one wants to compare the homogeneous data so
C     that the peak values will be the same when the layer interface moves.
     write(6,*)'Enter the scaling factor (usually=1)'
     read(5,*)scale

C    READ THE DATA AND SET A FEW PARAMETERS
     read(10,*)T
     read(10,*)FACE
     read(10,*)V1
     read(10,*)AL1,AT1
     read(10,*)V2
     read(10,*)AL2,AT2
     read(10,*)POR21
     read(10,*)Y
     read(10,*)AERR
     read(10,*)RERR
     read(10,*)LEVEL
     read(10,*)xbeg
     read(10,*)xend
     read(10,*)N

     DL1=V1*AL1
     DT1=V1*AT1
     DL2=V2*AL2
     DT2=V2*AT2
     DK2=FACE*FACE/4.D0/DL1/T
     DK=DSQRT(DK2)
C    Draw the interface line
     write(30,*)FACE,0.d0
```

```
      write(30,*)FACE,100.d0
      write(30,*)

C     WRITE OUT THE INPUT DATA TO THE SCREEN AND THE OUTPUT FILE
      write(6,*)'OUTPUT FOR SOLUTION NUMBER 2.'
      write(6,*)'------------------------------'
      write(6,*)
      write(20,*)'OUTPUT FOR SOLUTION NUMBER 2.'
      write(20,*)'------------------------------'
      write(20,*)
      write(6,*)'TIME SINCE INJECTION=                    ',T
      write(20,*)'TIME SINCE INJECTION=                   ',T
      write(6,*)'DISTANCE FROM SURFACE TO TOP OF SECOND AQUIFER=',FACE
      write(20,*)'DISTANCE FROM SURFACE TO TOP OF SECOND AQUIFER=',FACE
      write(6,*)'VERTICAL VELOCITY IN FIRST LAYER=            ',V1
      write(20,*)'VERTICAL VELOCITY IN FIRST LAYER=           ',V1
      write(20,*)'LONGITUDINAL DISPERSION IN FIRST LAYER=       ',DL1
      write(6,*)'LONGITUDINAL DISPERSION IN FIRST LAYER=      ',DL1
      write(20,*)'TRANSVERSE DISPERSION IN FIRST LAYER=       ',DT1
      write(6,*)'TRANSVERSE DISPERSION IN FIRST LAYER=        ',DT1
      write(20,*)'VERTICAL VELOCITY IN SECOND LAYER=          ',V2
      write(6,*)'VERTICAL VELOCITY IN SECOND LAYER=           ',V2
      write(20,*)'LONGITUDINAL DISPERSION IN SECOND LAYER=      ',DL2
      write(6,*)'LONGITUDINAL DISPERSION IN SECOND LAYER=     ',DL2
      write(20,*)'TRANSVERSE DISPERSION IN SECOND LAYER=      ',DT2
      write(6,*)'TRANSVERSE DISPERSION IN SECOND LAYER=       ',DT2
      write(6,*)'RATIO OF POROSITIES (N2/N1)=                 ',POR21
      write(20,*)'RATIO OF POROSITIES (N2/N1)=                ',POR21
      write(6,*)'TRANSVERSE LOCATION OF EVALUATION, Y=        ',Y
      write(20,*)'TRANSVERSE LOCATION OF EVALUATION, Y=       ',Y
      write(6,*)'The absolute error for cadre solver=         ',AERR
      write(20,*)'The absolute error for cadre solver=        ',AERR
      write(6,*)'The relative error for cadre solver=         ',RERR
      write(20,*)'The relative error for cadre solver=        ',RERR
      write(6,*)'The LEVEL for cadre solver=                  ',LEVEL
      write(20,*)'The LEVEL for cadre solver=                 ',LEVEL
      write(6,*)'The start point of calc. in x-direction=     ',xbeg
      write(20,*)'The start point of calc. in x-direction=    ',xbeg
      write(6,*)'The last point of calculation in x-direction= ',xend
      write(20,*)'The last point of calculation in x-direction= ',xend
      write(6,*)'The number of stehfast coefficients for SOLE= ',N
      write(20,*)'The number of stehfast coefficients for SOLE= ',N

      return
      end
```

```
C||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
C    Subroutine INT_BHVR
C      This subroutine checks the behavior of the integrand to aid in
C    the approximation of the upper limit of integration, i.e. an
C    approximation for infinity.
C||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
C  PARAMETER   TYPE     DESCRIPTION
C-------------------------------------------------------------------
C  RL              RL=(X-FACE)/2.D0/DSQRT(DL2*T)
C  RL2                RL2 = RL * RL
C  I                  Number of points of the function to observe
C                     between the lower limit and the upper limit
C                     of integration
C  k                  The number of the function one wishes to
C                     observe
C  upperlim           The estimation of infinity
C*******************************************************************

      subroutine INT_BHVR(T,Y,neq,N,V)

      implicit double precision (A-H,O-Z)
      dimension V(30)
      external sole1,sole4
      common/param/POR21,FACE,V1,AL1,DL1,DT1,V2,AL2,DL2,DT2,PI,DK2,DK,
     &          RL2,RL,xend,xbeg,scale

C     Look at behavior for first solution if neq= 1
      if (neq .eq. 1) then
       write(6,*)'Which function do you wish to look at?'
       write(6,*)'  1. First function of the second layer'
       write(6,*)'  2. Second function of the second layer'
       read(5,*)k
       write(6,*)'Enter the number of steps for the time variable:'
       read(5,*)I
       write(6,*)'Enter the x-direction location for evaluation:'
       write(6,*)'  the interface is at x=',face
       read(5,*)X
       write(6,*)'Enter the approximation for infinity:'
       write(6,*)'  the lower limit =',DK
       read(5,*)upperlim

       step=(upperlim-DK)/I
       XL=DK+step
       XU=UPPERLIM
       write(6,*)'U=                FCT='

C     PRINT OUT THE RESULTS FOR VARYING VALUES OF U
```

```
      IF (K .EQ. 1) THEN
       DO 100 U=XL,XU,step
        Q=solm1(T,U,X,Y)
        write(6,*)U,Q
100     CONTINUE

      ELSE IF (K .EQ. 2) THEN
       DO 110 U=XL,XU,step
        Q=solm2(T,U,X,Y)
        write(6,*)U,Q
110     CONTINUE
      END IF

     else if (neq .eq. 2) then
      write(6,*)'Enter the number of steps for the time variable:'
      read(5,*)I
      write(6,*)'Enter the x-direction location for evaluation:'
      write(6,*)'  the interface is at x=',face
      read(5,*)X
      RL=(X-FACE)/2.D0/DSQRT(DL2*T)
      write(6,*)'Enter the approximation for infinity:'
      write(6,*)'  the lower limit =',RL
      read(5,*)upperlim
      step=(upperlim-RL)/I
      XU=UPPERLIM
      RL2=RL*RL
      XL=RL
      write(6,*)'U=                    FCT='

C     PRINT OUT THE RESULTS FOR VARYING VALUES OF U

      DO 120 U=XL,XU,step
       IF (U .EQ. XL) THEN
        Q=0.D0
       ELSE
        Q=solc(T,U,X,Y)
       END IF
       write(6,*)U,Q
120    CONTINUE

     else
      write(6,*)'Which function do you wish to look at?'
      write(6,*)'  1. Solution for function, layer 1, Stehfast'
      write(6,*)'     inversion'
      write(6,*)'  2. Solution for function, layer 2, Stehfast'
      write(6,*)'     inversion'
      read(5,*)k
```

```
      write(6,*)'Enter the number of steps for the time variable:'
      read(5,*)I
      write(6,*)'Enter the x-direction location for evaluation:'
      write(6,*)'  the interface is at x=',face
      read(5,*)X
      write(6,*)'Enter the approximation for infinity:'
      write(6,*)'  the lower limit = 0'
      read(5,*)upperlim

      step=upperlim/I
      XL=0.D0
      XU=UPPERLIM
      write(6,*)'S=                    FCT='

C     PRINT OUT THE RESULTS FOR VARYING VALUES OF S

      IF (K .EQ. 1) THEN
       DO 130 S=XL,XU,step
        Q=STEHINV(S,X,Y,sole1,V,N,T)
        WRITE(5,*)S,Q
130     CONTINUE
      ELSE IF (K .EQ. 2) THEN
       DO 140 S=XL,XU,step
        Q=STEHINV(S,X,Y,sole4,V,N,T)
        WRITE(5,*)S,Q
140     CONTINUE
      END IF
     end if

     return
     end
```

```
C||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
C    Subroutine solmprint
C      This subroutine calculates the actual values of concentration
C    and prints them to an output file for the first approximate
C    equation which uses the mass flux B.C.
C      For accurate results the integral evaluated
C    is divided into "bit" number of pieces for which the cadre
C    subroutine evaluates each piece and the pieces are added
C    together to increase the accuracy of the quadrature.  The
C    derivative of the integrand is very large near the interface
C    boundary.  The time since injection is held constant.
C||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
C  PARAMETER   TYPE     DESCRIPTION
C-------------------------------------------------------------------
C  IFLAG              Flag for error output for Cadre solver
C  RESULT2            The final value for concentration
C  PRTRES2            A portion of the result added together to
C                     obtain RESULT2
C  RES1,RES2                Portions of the result added together to
C                     obtain PRTRES2
C  bit                The number of sections to divide the
C                     funciton domain into.  Usually equal to 1
C                     but sometimes set to 5 or 10 to check the
C                     accuracy of the numerical integration
C                     subroutines
C*******************************************************************

      subroutine solmprint(T,STEP1,STEP2,Y,AERR,RERR,LEVEL)

      implicit double precision (A-H,O-Z)
      external solm1,solm2
      common/param/POR21,FACE,V1,AL1,DL1,DT1,V2,AL2,DL2,DT2,PI,DK2,DK,
     &          RL2,RL,xend,xbeg,scale


C     WE NEED A NUMBER TO APPROXIMATE INFINITY
      write(6,*)'ENTER THE NUMBER YOU WISH TO USE TO APPROXIMATE'
      write(6,*)'INFINITY:  (the lower limit = ',DK,')'
      read(5,*)UPPERLIM
      write(20,*)'The number approximating infinity=        ',upperlim

C     THE INITIAL OUTPUT DATA FORMAT IS WRITTEN
      write(6,*)
      write(20,*)
      write(6,*)'OUTPUT RESULTS FOR SOLM'
      write(20,*)'OUTPUT RESULTS FOR SOLM'
      write(6,*)'C/Co VALUES FOR VARYING DISTANCES AND TIME= ',T
```

```
      write(20,*)'C/Co VALUES FOR VARYING DISTANCES AND TIME= ',T
      write(6,*)'X                C/Co'
      write(20,*)'X                C/Co'

C     THE INITIAL VERTICAL DISTANCE IS DEFINED

      X=xbeg

      DO WHILE (X .LE. FACE+0.0001D0)
       Q=sola(T,X,Y)
       write(6,*)X,Q
       write(20,*)X,Q
       write(30,*)X,Q
       X=X+STEP1
      END DO
      write(6,*)
      write(20,*)
      write(30,*)


C     This while loop is set up to propogate through the second aquifer
C      perpendicular to the layers at the specified Y-value until a
C      C/Co concentration of 0 (or close to it) is obtained.

      X=FACE
C     THE FIRST UPPER AND LOWER LIMIT OF INTEGRATION ARE SET

      write(6,*)'Enter # bits you wish to break the curve into:'
      read(5,*)bit
      step=(upperlim-DK)/bit
      DO WHILE (X .LE. xend)
       xl=DK+1.d-10
       xu=xl+step
       result2=0.d0
       do 200 i=1,bit
C       CALCULATE C/Co.
        CALL CADR(solm1,xl,xu,AERR,RERR,LEVEL,ERROR,IFLAG,Q,T,X,Y)
        RES1=Q
        if (IFLAG .GE. 4) then
         write(6,*)'IFLAG= ',IFLAG,' for fct1 at x=',X
         write(20,*)'IFLAG= ',IFLAG,' for fct1 at x=',X
        end if

        CALL CADR(solm2,xl,xu,AERR,RERR,LEVEL,ERROR,IFLAG,Q,T,X,Y)
        RES2=Q
        if (IFLAG .GE. 4) then
         write(6,*)'IFLAG= ',IFLAG,' for fct2 at x=',X
```

```
        write(20,*)'IFLAG= ',IFLAG,' for fct2 at x=',X
       end if

C      Here the two parts of the solution, after being evaluated,
C        are added and subtracted accordingly from each other to
C        produce a part of the solution, called PRTRES1.  All the
C        PRTRES1'S are added together to produce the whole solution
C        at one particular location via the inner while loop.
       PRTRES2=RES1-RES2
       RESULT2=RESULT2+PRTRES2

       XL=XU
       XU=XU+step
200    continue
C  We now print out the results
       write(6,*)X,RESULT2
       write(20,*)X,RESULT2
       write(30,*)X,RESULT2
       X=X+STEP2
      END DO
      write(30,*)

      return
      end
```

```
C||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
C    Subroutine solcprint
C      This subroutine calculates the actual values of concentration
C    and prints them to an output file for the approximate solution
C    using the concentration continuity B.C.
C      For accurate results the integral is divided into "bit" number
C    of pieces for which the cadre subroutine evaluates each piece
C    and the pieces are added together to increase the accuracy.
C      The derivative of the integrand is very large near the
C    interface boundary. The time since injection is held constant.
C||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
C  PARAMETER   TYPE     DESCRIPTION
C---------------------------------------------------------------------
C  all parameters previously defined
C**********************************************************************

      subroutine solcprint(T,STEP1,STEP2,Y,AERR,RERR,LEVEL)

      implicit double precision (A-H,O-Z)
      external solc
      common/param/POR21,FACE,V1,AL1,DL1,DT1,V2,AL2,DL2,DT2,PI,DK2,DK,
     &          RL2,RL,xend,xbeg,scale

C     WE NEED A NUMBER TO APPROXIMATE INFINITY
      write(6,*)'ENTER THE NUMBER YOU WISH TO USE TO APPROXIMATE'
      write(6,*)'INFINITY:'
      read(5,*)UPPERLIM
      write(20,*)'The number approximating infinity=          ',upperlim

C     THE INITIAL OUTPUT DATA FORMAT IS WRITTEN
      write(6,*)
      write(20,*)
      write(6,*)'OUTPUT RESULTS FOR SOLC'
      write(20,*)'OUTPUT RESULTS FOR SOLC'
      write(6,*)'C/Co VALUES FOR VARYING DISTANCES AND TIME= ',T
      write(20,*)'C/Co VALUES FOR VARYING DISTANCES AND TIME= ',T
      write(6,*)'X                  C/Co'
      write(20,*)'X                  C/Co'


      X=xbeg

      DO WHILE (X .LE. FACE+0.0001D0)
       Q=sola(T,X,Y)
       write(6,*)X,Q
       write(20,*)X,Q
       write(30,*)X,Q
```

```
      X=X+STEP1
     END DO
     write(6,*)
     write(20,*)
     write(30,*)

C    This while loop is set up to propogate through the second aquifer
C      perpendicular to the layers at the specified Y-value until a
C      C/Co concentration of 0 (or close to it) is obtained.

     X=FACE
     write(6,*)'Enter # bits to divide the curve into:'
     read(5,*)bit
     DO WHILE (X .LE. xend)
      RL=(X-FACE)/2.D0/DSQRT(DL2*T)
      step=(upperlim-RL)/bit
      RL2=RL*RL
      xl=RL+1.d-10
      xu=xl+step
      result2=0.d0
      do 300 i=1,bit
C       CALCULATE C/Co.
       CALL CADR(solc,xl,xu,AERR,RERR,LEVEL,ERROR,IFLAG,Q,T,X,Y)
       RES1=Q

C       All the RESULT2'S are added together to produce the
C         whole solution at one particular location via the inner
C         while loop.
       RESULT2=RESULT2+RES1

       XL=XU
       XU=XU+step
300    continue
C    We now print out the results
      write(6,*)X,RESULT2
      write(20,*)X,RESULT2
      write(30,*)X,RESULT2
      X=X+STEP2
     END DO
     write(30,*)

     return
     end
```

```
C||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
C    Subroutine soleprint
C      This subroutine calculates the actual values of concentration
C    and prints them to an output file for our "exact" solution.
C||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
C  PARAMETER   TYPE     DESCRIPTION
C---------------------------------------------------------------------
C  all parameters previously defined
C**********************************************************************

      subroutine soleprint(V,N,STEP1,STEP2,Y,T)

      implicit double precision (A-H,O-Z)
      external sole1,sole4
      dimension V(30)
      common/param/POR21,FACE,V1,AL1,DL1,DT1,V2,AL2,DL2,DT2,PI,DK2,DK,
     &          RL2,RL,xend,xbeg,scale

C     WE NEED A NUMBER TO APPROXIMATE INFINITY
      write(6,*)'ENTER THE NUMBER YOU WISH TO USE TO APPROXIMATE'
      write(6,*)'INFINITY:  (the lower limit = 0)'
      read(5,*)UPPERLIM
      write(20,*)'The number approximating infinity=         ',upperlim

C     THE INITIAL OUTPUT DATA FORMAT IS WRITTEN
      write(6,*)
      write(20,*)
      write(6,*)'OUTPUT RESULTS FOR SOLC'
      write(20,*)'OUTPUT RESULTS FOR SOLC'
      write(6,*)'C/Co VALUES FOR VARYING DISTANCES AND TIME= ',T
      write(20,*)'C/Co VALUES FOR VARYING DISTANCES AND TIME= ',T
      write(6,*)'X                  C/Co'
      write(20,*)'X                   C/Co'

C     THE INITIAL VERTICAL DISTANCE IS DEFINED

      X=xbeg
      xl=0.d0
      xu=upperlim

      DO WHILE (X .LE. FACE+0.0001d0)
       RESULT1=0.D0
       CALL DQG32(T,XL,XU,X,Y,sole1,Q,V,N)
       RES1=Q

       RES2=sole2(T,X,Y)
```

```
      RES3=sole3(T,X,Y)
C     HERE THE THREE PARTS OF THE SOLUTION, AFTER BEING EVALUATED,
C       ARE ADDED AND SUBTRACTED ACCORDINGLY FROM EACH OTHER TO
C       PRODUCE THE SOLUTION, CALLED RESULT1.

      RESULT1=RES1-RES2+RES3

      write(6,*)X,RESULT1
      write(20,*)X,RESULT1
      write(30,*)X,RESULT1
      X=X+STEP1
     END DO

     write(6,*)
     write(20,*)
     write(30,*)

     X=FACE
     xl=0.d0
     xu=upperlim

C    THIS WHILE LOOP IS SET UP TO PROPOGATE THROUGH THE SECOND AQUIFER
C     IN A VERTICAL DIRECTION AT THE SPECIFIED Y-VALUE UNTIL A C/Co
C     CONCENTRATION OF 0 (OR CLOSE TO IT) IS OBTAINED.
C     THE TIME SINCE INJECITON IS HELD CONSTANT.

     DO WHILE (X .LE. xend)
      RESULT2=0.D0
      CALL DQG32(T,XL,XU,X,Y,sole4,Q,V,N)
      RESULT2=Q
      write(6,*)X,RESULT2
      write(20,*)X,RESULT2
      write(30,*)X,RESULT2
      X=X+STEP2
     END DO
     write(30,*)

     return
     end
```

```
C||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
C    Function sola
C      This function calculates the value of concentration for the
C      first layer for the two approximate solutions.
C||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
C  PARAMETER   TYPE     DESCRIPTION
C---------------------------------------------------------------------
C  w1,w2       [-]      dummy variables
C  WT                   [-]      total dummy variable
C*********************************************************************

      double precision function sola(T,X,Y)

      implicit double precision (A-H,O-Z)
      common/param/POR21,FACE,V1,AL1,DL1,DT1,V2,AL2,DL2,DT2,PI,DK2,DK,
     &           RL2,RL,xend,xbeg,scale

      W1=DEXP(-((X-(V1*T))*(X-(V1*T))/4.D0/DL1/T)-(Y*Y/4.D0/DT1/T))
      W2=FACE*AL1*POR21/2.D0/DSQRT(DL1*DT1)/T
      WT=W1*W2

      sola=WT*scale

      return
      end
```

```
C||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
C    Function solm1
C      This is the first of two functions used to find the value of
C    concentration for the approximate solution using the mass flux
C    B.C.
C||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
C  PARAMETER   TYPE     DESCRIPTION
C-------------------------------------------------------------------------
C  w1-w7       [-]      dummy variables
C  WT                   [-]      total dummy variable
C*************************************************************************

      double precision function solm1(T,U,X,Y)

      implicit double precision (A-H,O-Z)
      common/param/POR21,FACE,V1,AL1,DL1,DT1,V2,AL2,DL2,DT2,PI,DK2,DK,
     &           RL2,RL,xend,xbeg,scale

      W1=T-(DK2*T/U/U)
      W2=(DT2*W1)+(DT1*DK2*T/U/U)
      W3=DLOG(FACE*FACE/4.D0/U/U/DSQRT(PI*DL2)+FACE*AL1/DSQRT(PI*DL2))
      W4=-0.5D0*DLOG(W1*W2)
      W5=(V1*FACE/2.D0/DL1)+((X-FACE)*V2/2.D0/DL2)-(V2*V2*T/4.D0/DL2)
      W6=(V2*V2*DK2*T/4.D0/DL2/U/U)-(V1*V1*DK2*T/4.D0/DL1/U/U)
      W7=-((X-FACE)*(X-FACE)/4.D0/DL2/W1)-(U*U)-(Y*Y/4.D0/W2)

      WT=W3+W4+W5+W6+W7

      solm1=DEXP(WT)*scale

      return
      end
```

```
C||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
C    Function solm2
C      This is the second of two functions used to find the value of
C    concentration for the approximate solution using the mass flux
C    B.C. In this function, an
C    approximation is used for the DERFC function if the argument of
C    it is greater than 9.
C||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
C  PARAMETER   TYPE    DESCRIPTION
C-------------------------------------------------------------------
C  w1-w6,w      [-]     dummy variables
C  WT                   [-]     total dummy variable
C*******************************************************************

      double precision function solm2(T,U,X,Y)

      implicit double precision (A-H,O-Z)
      common/param/POR21,FACE,V1,AL1,DL1,DT1,V2,AL2,DL2,DT2,PI,DK2,DK,
     &          RL2,RL,xend,xbeg,scale

      W1=T-(DK2*T/U/U)
      W2=(DT2*W1)+(DT1*DK2*T/U/U)
      W3=DLOG(FACE*FACE/8.D0/AL2/U/U+FACE*AL1/2.D0/AL2)
      W4=-0.5D0*DLOG(W2)
      W5=(V1*FACE/2.D0/DL1)+(V2*(X-FACE)/DL2)-(V1*V1*DK2*T/4.D0/
     &   DL1/U/U)
      W6=-(U*U)-(Y*Y/4.D0/W2)
      Z=((X-FACE)/2.D0/DSQRT(DL2*W1))+(V2*DSQRT(W1/DL2)/2.D0)

      IF (Z .LT. 9.D0) THEN
       WT=W3+W4+W5+W6
       W=DERFC(Z)
       solm2=DEXP(WT)*W*scale
      ELSE
       W=(1.D0/Z/DSQRT(PI))-(1.D0/2.D0/Z/Z/Z/DSQRT(PI))
       WT=W3+W4+W5+W6-Z*Z
       solm2=DEXP(WT)*W*scale
      END IF

      return
      end
```

```
C||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
C    Function solc
C      This function is used to find the concentration in the second
C    layer for our approximate solution using the concentration
C    continuity B.C.
C||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
C  PARAMETER   TYPE     DESCRIPTION
C-------------------------------------------------------------------
C  w1-w7       [-]      dummy variables
C  WT                   [-]      total dummy variable
C*******************************************************************

      double precision function solc(T,U,X,Y)

      implicit double precision (A-H,O-Z)
      common/param/POR21,FACE,V1,AL1,DL1,DT1,V2,AL2,DL2,DT2,PI,DK2,DK,
     &          RL2,RL,xend,xbeg,scale

      W1=DT2*RL2*T/U/U+DT1*(T-RL2*T/U/U)
      W2=T-RL2*T/U/U
      W3=-0.5D0*DLOG(W2*W1)
      STUFF=DSQRT(PI*DL1)
      W4=DLOG(FACE*AL1*POR21/STUFF)
      W5=-V1*V1*T/4.D0/DL1
      W6=-FACE*FACE/4.D0/DL1/W2
      W7=FACE*V1/2.D0/DL1+(X-FACE)*V2/2.D0/DL2+V1*V1*RL2*T/4.D0/DL1/U
     &  /U-V2*V2*RL2*T/4.D0/DL2/U/U
      W8=-Y*Y/4.D0/W1

      WT=W4+W3+W7+W5+W8+W6-U*U

      solc=DEXP(WT)*scale

      return
      end
```

```
C||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
C    Function sole1
C      This function is the first of three for finding the
C    concentration in the first layer of our "exact" solution.
C||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
C  PARAMETER   TYPE     DESCRIPTION
C---------------------------------------------------------------------
C  w1-w4       [-]      dummy variables
C  WT                   [-]      total dummy variable
C*********************************************************************

      double precision function sole1(P,S,X,Y)
      implicit double precision (A-H,O-Z)
      common/param/POR21,FACE,V1,AL1,DL1,DT1,V2,AL2,DL2,DT2,PI,DK2,DK,
     &           RL2,RL,xend,xbeg,scale

      W1=DSQRT(V1*V1+4.D0*DL1*P+4.D0*DL1*DT1*S*S)
      W2=DSQRT(V2*V2+4.D0*DL2*P+4.D0*DL2*DT2*S*S)
      W3=DLOG(4.D0*FACE*Al1*POR21*DCOS(S*Y)/(V1+W1-POR21*V2+POR21*W2))
      W4=V1*X/2.D0/DL1+(X/2.D0-FACE)*W1/DL1

      WT=W3+W4

      sole1=DEXP(WT)*scale

      return
      end
```

```
C||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
C    Function sole2
C      This is the second of three functions used to find the value of
C    concentration for our "exact" solution.
C||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
C  PARAMETER   TYPE     DESCRIPTION
C---------------------------------------------------------------------
C  w1,w2       [-]      dummy variables
C*********************************************************************

      double precision function sole2(T,X,Y)
      implicit double precision (A-H,O-Z)
      common/param/POR21,FACE,V1,AL1,DL1,DT1,V2,AL2,DL2,DT2,PI,DK2,DK,
     &           RL2,RL,xend,xbeg,scale

      W1=DLOG(FACE*AL1*POR21/2.D0/T/DSQRT(DL1*DT1))
      W2=V1*X/2.D0/DL1-V1*V1*T/4.D0/DL1-(2.D0*FACE-X)*(2.D0*FACE-X)/
     &  4.D0/DL1/T-Y*Y/4.D0/DT1/T

      sole2=DEXP(W1+W2)*scale

      RETURN
      END
```

```
C||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
C    Function sole3
C      This is the third of three functions used to find the value of
C    concentration for our "exact" solution.
C||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
C  PARAMETER   TYPE     DESCRIPTION
C---------------------------------------------------------------------
C  w1,w2       [-]      dummy variables
C*********************************************************************

      double precision function sole3(T,X,Y)
      implicit double precision (A-H,O-Z)
      common/param/POR21,FACE,V1,AL1,DL1,DT1,V2,AL2,DL2,DT2,PI,DK2,DK,
     &           RL2,RL,xend,xbeg,scale

      W1=DLOG(FACE*AL1*POR21/2.D0/T/DSQRT(DT1*DL1))
      W2=V1*X/2.D0/DL1-V1*V1*T/4.D0/DL1-X*X/4.D0/DL1/T-Y*Y/4.D0/DT1/T

      sole3=DEXP(W1+W2)*scale

      RETURN
      END
```

```
C|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
C    Function sole4
C      This is the only function needed to find the value of
C    concentration in the second layer for our "exact" solution.
C|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
C  PARAMETER   TYPE     DESCRIPTION
C-------------------------------------------------------------------
C  w1-w5       [-]      dummy variables
C  WT                   [-]      total dummy variable
C*******************************************************************

      double precision function sole4(P,S,X,Y)
      implicit double precision (A-H,O-Z)
      common/param/POR21,FACE,V1,AL1,DL1,DT1,V2,AL2,DL2,DT2,PI,DK2,DK,
     &           RL2,RL,xend,xbeg,scale

      W1=DSQRT(V1*V1+4.D0*DL1*P+4.D0*DL1*DT1*S*S)
      W2=DSQRT(V2*V2+4.D0*DL2*P+4.D0*DL2*DT2*S*S)
      W3=DLOG(4.D0*FACE*AL1*POR21*DCOS(S*Y)/(V1+W1-POR21*V2+POR21*W2))
      W4=V1*FACE/2.D0/DL1+V2*(X-FACE)/2.D0/DL2
      W5=-FACE*W1/2.D0/DL1-(X-FACE)*W2/2.D0/DL2

      WT=W3+W4+W5

      sole4=DEXP(WT)*scale

      RETURN
      END
```

```
C||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
C     subroutine COEFF
C       This subroutine calculated the coefficients necessary for
C       doing the stehfast inversion.  This is handy since it allows
C       me to see what number of coefficients is optimal for the
C       functions I am inverting.  Perhaps a certain number of
C       coefficients gives a more accurate inversion than a different
C       number.
C||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
C  PARAMETER   TYPE     DESCRIPTION
C---------------------------------------------------------------------
C  V          [real array] The output stehfast coefficients
C  N          [-]      The number or stehfast coefficients
C  All other variables and arrays are local dummy values
C*********************************************************************

      Subroutine coeff(v,n)
      implicit double precision (a-h,o-z)
      dimension v(30),g(100),h(50)
      common/param/POR21,FACE,V1,AL1,DL1,DT1,V2,AL2,DL2,DT2,PI,DK2,DK,
     &          RL2,RL,xend,xbeg,scale

       g(1)=1.d0

       nh=n/2
       do 10 i=2,n
         g(i)=g(i-1)*i
10     continue
       h(1)=2.d0/g(nh-1)
       do 30 i=2,nh
         fi=i
         if(i.ne.nh)then
           h(i)=fi**nh*g(2*i)/(g(nh-i)*g(i)*g(i-1))
         else
           h(i)=fi**nh*g(2*i)/(g(i)*g(i-1))
         endif
30     continue
       sn=2*(nh-nh/2*2)-1
       do 40 i=1,n
         v(i)=0.d0
         k1=(i+1)/2
         k2=i
         if(k2.gt.nh)k2=nh
         do 50 k=k1,k2
           if(2*k-i.eq.0)then
             v(i)=v(i)+h(k)/(g(i-k))
           elseif(i.eq.k)then
```

```
            v(i)=v(i)+h(k)/(g(2*k-i))
          else
            v(i)=v(i)+h(k)/(g(i-k)*g(2*k-i))
          endif
50       continue
        v(i)=sn*v(i)
        sn=-sn
40    continue
    return
    end
```

```
C||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
C    function STEHINV
C      This function numerically inverts a function out of the
C      laplace domain.  This is done using the stehfast method.
C      The solution is numerically inverted at each point of
C      the gaussian quadrature.
C||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
C  PARAMETER   TYPE     DESCRIPTION
C---------------------------------------------------------------------
C  W          [real array] The output stehfast coefficients
C*********************************************************************

      double precision function STEHINV(S,X,Y,FCT,W,N,T)
      implicit double precision (A-H,O-Z)
      common/param/POR21,FACE,V1,AL1,DL1,DT1,V2,AL2,DL2,DT2,PI,DK2,DK,
     &          RL2,RL,xend,xbeg,scale

      DIMENSION W(30)

      SUM=0.D0
      P2=DLOG(2.D0)/T

C Loop 6 calculates the inversion for a given time
      DO 6 I=1,N
       P1=DLOG(2.D0)*I/T
       SUMn=W(I)*FCT(P1,S,X,Y)
       SUM=SUM+SUMn
6     CONTINUE
      STEHINV=P2*SUM

      RETURN
      END
```

```
C||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
C    Subroutine DQG32
C      This subroutine numerically integrates the functions sole1 and
C    sole4 using 32 point Gaussian quadrature.
C||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
C  PARAMETER   TYPE     DESCRIPTION
C-------------------------------------------------------------------------
C  XL          [-]      lower limit of integration
C  XU          [-]      upper limit of integration
C  FCT         [-]      function to be integrated
C  Q           [-]      result
C  A,B,C       [-]      dummy variable
C  V           [real array] The stehfast coefficients
C  N           [-]      The number or stehfast coefficients
C*************************************************************************

      SUBROUTINE DQG32(T,XL,XU,X,Y,FCT,Q,V,N)
      implicit double precision (A-H,O-Z)
      external FCT
      dimension V(30)
      common/param/POR21,FACE,V1,AL1,DL1,DT1,V2,AL2,DL2,DT2,PI,DK2,DK,
     &          RL2,RL,xend,xbeg,scale

      A=.5D0*(XU+XL)
      B=XU-XL
      C=.49863193092474078D0*B
       Q1=STEHINV(A+C,X,Y,FCT,V,N,T)
       Q2=STEHINV(A-C,X,Y,FCT,V,N,T)
      Q=.35093050047350483D-2*(Q1+Q2)
      C=.49280575577263417D0*B
       Q1=STEHINV(A+C,X,Y,FCT,V,N,T)
       Q2=STEHINV(A-C,X,Y,FCT,V,N,T)
      Q=Q+.8137197365452835D-2*(Q1+Q2)
      C=.48238112779375322D0*B
       Q1=STEHINV(A+C,X,Y,FCT,V,N,T)
       Q2=STEHINV(A-C,X,Y,FCT,V,N,T)
      Q=Q+.12696032654631030D-1*(Q1+Q2)
      C=.46745303796886984D0*B
       Q1=STEHINV(A+C,X,Y,FCT,V,N,T)
       Q2=STEHINV(A-C,X,Y,FCT,V,N,T)
      Q=Q+.17136931456510717D-1*(Q1+Q2)
      C=.44816057788302606D0*B
       Q1=STEHINV(A+C,X,Y,FCT,V,N,T)
       Q2=STEHINV(A-C,X,Y,FCT,V,N,T)
      Q=Q+.21417949011113340D-1*(Q1+Q2)
      C=.42468380686628499D0*B
       Q1=STEHINV(A+C,X,Y,FCT,V,N,T)
```

```
  Q2=STEHINV(A-C,X,Y,FCT,V,N,T)
Q=Q+.25499029631188088D-1*(Q1+Q2)
C=.39724189798397120D0*B
  Q1=STEHINV(A+C,X,Y,FCT,V,N,T)
  Q2=STEHINV(A-C,X,Y,FCT,V,N,T)
Q=Q+.29342046739267774D-1*(Q1+Q2)
C=.36609105937014484D0*B
  Q1=STEHINV(A+C,X,Y,FCT,V,N,T)
  Q2=STEHINV(A-C,X,Y,FCT,V,N,T)
Q=Q+.32911111388180923D-1*(Q1+Q2)
C=.33152213346510760D0*B
  Q1=STEHINV(A+C,X,Y,FCT,V,N,T)
  Q2=STEHINV(A-C,X,Y,FCT,V,N,T)
Q=Q+.36172897054424253D-1*(Q1+Q2)
C=.29385787862038116D0*B
  Q1=STEHINV(A+C,X,Y,FCT,V,N,T)
  Q2=STEHINV(A-C,X,Y,FCT,V,N,T)
Q=Q+.39096947893535153D-1*(Q1+Q2)
C=.25344995446611470D0*B
  Q1=STEHINV(A+C,X,Y,FCT,V,N,T)
  Q2=STEHINV(A-C,X,Y,FCT,V,N,T)
Q=Q+.41655962113473378D-1*(Q1+Q2)
C=.21067563806531767D0*B
  Q1=STEHINV(A+C,X,Y,FCT,V,N,T)
  Q2=STEHINV(A-C,X,Y,FCT,V,N,T)
Q=Q+.43826046502201906D-1*(Q1+Q2)
C=.16593430114106382D0*B
  Q1=STEHINV(A+C,X,Y,FCT,V,N,T)
  Q2=STEHINV(A-C,X,Y,FCT,V,N,T)
Q=Q+.45586939347881942D-1*(Q1+Q2)
C=.11964368112606854D0*B
  Q1=STEHINV(A+C,X,Y,FCT,V,N,T)
  Q2=STEHINV(A-C,X,Y,FCT,V,N,T)
Q=Q+.46922199540402283D-1*(Q1+Q2)
C=.7223598079139825D-1*B
  Q1=STEHINV(A+C,X,Y,FCT,V,N,T)
  Q2=STEHINV(A-C,X,Y,FCT,V,N,T)
Q=Q+.47819360039637430D-1*(Q1+Q2)
C=.24153832843869158D-1*B
  Q1=STEHINV(A+C,X,Y,FCT,V,N,T)
  Q2=STEHINV(A-C,X,Y,FCT,V,N,T)
Q=B*(Q+.48270044257363900D-1*(Q1+Q2))
RETURN
END
```

```
C||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
C    Subroutine CADR
C This subroutine uses Cautious Adaptive Rhomberg Extrapolation to
C  Compute the integral of function F, between the limits a and b.
C  This routine is not modular, is written in bad fortran, but gives
C  great results. For a more complete write up see De Boor, ACM
C  Monograph Series: Mathematical Software edited by Rice
C||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
C  PARAMETER   TYPE     DESCRIPTION
C-----------------------------------------------------------------------
C  AERR                 Absolute error for Cadre solver
C                       Usually, a value of 1.D-10 was used
C  RERR                 Relative error for Cadre solver
C                       Usually, a value of 1.D-10 was used
C  Level                Level of error output for Cadre solver,
C                        (=1-5). Usually the value of 1 was used.
C  IFLAG               Flag for error output for Cadre solver
C     For a more complete description of these variables, see
C                     de Boor.
C***********************************************************************

    Subroutine CADR(F,A,B,AERR,RERR,LEVEL,ERROR,IFLAG,cadre,time,X,Y)
    implicit double precision (A-H,O-Z)
    DOUBLE PRECISION LENGTH,JUMPTL
    common/param/POR21,FACE,V1,AL1,DL1,DT1,V2,AL2,DL2,DT2,PI,DK2,DK,
   &          RL2,RL,xend,xbeg,scale

    DIMENSION T(10,10),R(10),AIT(10),DIF(10),RN(4),
   1      TS(2049),IBEGS(30),BEGIN(30),FINIS(30),EST(30)
    LOGICAL H2CONV,AITKEN,RIGHT,REGLAR,REGLSV(30)
    DATA TOLMCH,AITLOW,H2TOL,AITTOL,JUMPTL,MAXTS,MAXTBL,MXSTGE
   1 / 1.E-7, 1.1 , .15 ,  .1  , .01  , 2049,  10  ,  30/
    DATA RN/.71420053,.34662815,.843751,.12633046/
    DATA ALG402/ .3010299956639795/
    DATA PT1/.1/
    data nout/5/
    CADRE = 0.
    ERROR = 0.
    IFLAG = 1
    LENGTH = DABS(B-A)
    IF(LENGTH .EQ. 0.0)RETURN
    ERRR = DMIN1(PT1,DMAX1(DABS(RERR),10.*TOLMCH))
    ERRA = DABS(AERR)
    STEPMN = DMAX1(LENGTH/dble(2**MXSTGE),
   1      DMAX1(LENGTH,DABS(A),DABS(B))*TOLMCH)
    STAGE = .5
    ISTAGE = 1
```

```
      CUREST = 0.
      FNSIZE = 0.
      PREVER = 0.
      REGLAR = .FALSE.
Comment: The given interval of integration is the first interval considered
      BEG = A
      FBEG = F(time,BEG,x,y)/2.
      TS(1) = FBEG
      IBEG = 1
      END = B
      FEND = F(time,END,x,y)/2.
      TS(2) =  FEND
      IEND = 2
5     RIGHT = .FALSE.
6     STEP = END - BEG
      ASTEP = DABS(STEP)
      IF(ASTEP .LT. STEPMN) GOTO 950
      IF(LEVEL .GE. 3) WRITE(NOUT,609)BEG,STEP,ISTAGE
609   FORMAT(10H BEG,STEP ,2E16.8,I5)
      T(1,1) = FBEG + FEND
      TABS = DABS(FBEG) + DABS(FEND)
      L = 1
      N = 1
      H2CONV = .FALSE.
      AITKEN = .FALSE.
      GOTO 10
9     IF(LEVEL .GE. 4) WRITE(NOUT,692)L,T(1,LM1)
10    LM1 = L
      L = L+1
      N2 = N*2
      FN = N2
      ISTEP = (IEND - IBEG)/N
      IF(ISTEP .GT. 1) GOTO 12
      II = IEND
      IEND = IEND +N
      IF(IEND .GT. MAXTS) GOTO 900
      HOVN = STEP/FN
      III = IEND
      DO 11 I = 1,N2,2
       TS(III) = TS(II)
       TS(III-1) = F(time,END - dble(I)*HOVN,x,y)
       III = III -2
       II = II -1
11    CONTINUE
      ISTEP = 2
12    ISTEP2 = IBEG +ISTEP/2
      SUM = 0.
```

```
      SUMABS = 0.
      DO 13 I = ISTEP2,IEND,ISTEP
       SUM = SUM + TS(I)
       SUMABS = SUMABS + DABS(TS(I))
   13  CONTINUE
      T(L,1) = T(L-1,1)/2. + SUM/FN
      TABS = TABS/2. + SUMABS/FN
      ABSI = ASTEP*TABS
      N = N2
      IT = 1
      VINT = STEP*T(L,1)
      TABTLM = TABS*TOLMCH
      FNSIZE = DMAX1(FNSIZE,DABS(T(L,1)))
      ERGOAL = DMAX1(ASTEP*TOLMCH*FNSIZE,
     1       STAGE*DMAX1(ERRA,ERRR*DABS(CUREST+VINT)))
      FEXTRP = 1.
      DO 14 I = 1,LM1
       FEXTRP = FEXTRP*4.
       T(I,L) = T(L,I)-T(L-1,I)
       T(L,I+1) = T(L,I) + T(I,L)/(FEXTRP -1.)
   14  CONTINUE
      ERRER = ASTEP*DABS(T(1,L))
      IF(L .GT. 2)GOTO 15
      IF(DABS(T(1,2)).LE.  TABTLM) GOTO 60
      GOTO 10
   15  DO 16 I = 2,LM1
       DIFF = 0.
       IF(DABS(T(I-1,L)) .GT. TABTLM) DIFF = T(I-1,LM1)/T(I-1,L)
       T(I-1,LM1) = DIFF
   16  CONTINUE
      IF(DABS(4. - T(1,LM1)) .LE. H2TOL) GOTO 20
      IF(T(1,LM1) .EQ. 0.) GOTO 18
      IF(DABS(2. - DABS(T(1,LM1))) .LT. JUMPTL) GOTO 50
      IF(L .EQ. 3) GOTO 9
      H2CONV = .FALSE.
      IF(DABS((T(1,LM1)-T(1,L-2))/T(1,LM1)) .LE. AITTOL)GOTO 30
   17  IF(REGLAR)GOTO 18
      IF(L.EQ.4) GOTO 9
   18  IF(ERRER .LE. ERGOAL) GOTO 70
      IF(LEVEL .GE. 4) WRITE(NOUT,692)L,T(1,LM1)
      GOTO 91
   20  IF(LEVEL .GE. 4)WRITE(NOUT,619)L,T(1,LM1)
  619  FORMAT(I5,E16.8,5X,6HH2CONV)
      IF(H2CONV)GOTO 21
      AITKEN = .FALSE.
      H2CONV = .TRUE.
      IF(LEVEL .GE. 3) WRITE(NOUT,620)L
```

```
620  FORMAT(22H H2 CONVERGENCE AT ROW,I5)
21   FEXTRP = 4.
22   IT= IT +1
     VINT = STEP*T(L,IT)
     ERRER = DABS(STEP/(FEXTRP - 1.)*T(IT-1,L))
     IF(ERRER .LE. ERGOAL )GOTO 80
     IF(IT .EQ.LM1)GOTO 40
     IF(T(IT,LM1).EQ. 0.) GOTO 22
     IF(T(IT,LM1) .LE. FEXTRP) GOTO 40
     IF(DABS(T(IT,LM1)/4.-FEXTRP)/FEXTRP .LT. AITTOL)FEXTRP=FEXTRP*4.
     GOTO 22
30   IF(LEVEL .GE. 4)WRITE(NOUT,629)L,T(1,LM1)
629  FORMAT(I5,E16.8,5X,6HAITKEN)
     IF(T(1,LM1) .LT. AITLOW) GOTO 91
     IF(AITKEN)GOTO 31
     H2CONV = .FALSE.
     AITKEN = .TRUE.
     IF(LEVEL.GE.3)WRITE(NOUT,630)L
630  FORMAT(14H AITKEN AT ROW,I5)
31   FEXTRP = T(L-2,LM1)
     IF(FEXTRP .GT. 4.5)GOTO 21
     IF(FEXTRP .LT. AITLOW)GOTO 91
     IF(DABS(FEXTRP-T(L-3,LM1))/T(1,LM1).GT.H2TOL)GOTO 91
     IF(LEVEL.GE.3)WRITE(NOUT,631)FEXTRP
631  FORMAT(6H RATIO,F12.8)
     SING = FEXTRP
     FEXTM1 = FEXTRP -1.
     DO 32 I = 2,L
      AIT(I) = T(I,1) +(T(I,1)-T(I-1,1))/FEXTM1
      R(I) = T(1,I-1)
      DIF(I) = AIT(I) -  AIT(I-1)
32   CONTINUE
     IT =2
33   VINT = STEP*AIT(L)
     IF(LEVEL.LT.5)GOTO 333
     WRITE(NOUT,632)(R(I+1),I=IT,LM1)
     WRITE(NOUT,632)(AIT(I),I=IT,L)
     WRITE(NOUT,632)(DIF(I+1),I=IT,LM1)
632  FORMAT(1X,8E15.8)
333  ERRER = ERRER/FEXTM1
     IF(ERRER .GT. ERGOAL) GOTO 34
     ALPHA = DLOG10(SING)/ALG402 -1.
     IF(LEVEL.GE.2)WRITE(NOUT,633)ALPHA,BEG,END
633  FORMAT(11X,42HINTEGRAND SHOWS SINGULAR BEHAVIOR OF TYPE ,
    1  4HX**( ,F4.2,9H) BETWEEN ,E15.8,4H AND ,E15.8)
     IFLAG = MAX0(IFLAG,2)
     GOTO 80
```

```
34   IT = IT+1
     IF(IT.EQ.LM1)GOTO 40
     IF(IT.GT.3)GOTO 35
     H2NEXT = 4.
     SINGNX = 2.*SING
35   IF(H2NEXT .LT. SINGNX) GOTO 36
     FEXTRP = SINGNX
     SINGNX = 2.*SINGNX
     GOTO 37
36   FEXTRP = H2NEXT
     H2NEXT = 4.*H2NEXT
37   DO 38 I = IT,LM1
      R(I+1)=0.
      IF(DABS(DIF(I+1)) .GT. TABTLM)R(I+1)=DIF(I)/DIF(I+1)
38   CONTINUE
     IF(LEVEL .GE. 4)WRITE(NOUT,638)FEXTRP,R(L-1),R(L)
638  FORMAT(16H FEXTRP + RATIOS,3E15.8)
     H2TFEX = -H2TOL*FEXTRP
     IF(R(L) - FEXTRP .LT. H2TFEX)GOTO 40
     IF(R(L-1) - FEXTRP .LT. H2TFEX)GOTO 40
     ERRER = ASTEP*DABS(DIF(L))
     FEXTM1 = FEXTRP -1.
     DO 39 I = IT,L
      AIT(I) = AIT(I) + DIF(I)/FEXTM1
      DIF(I) = AIT(I)-AIT(I-1)
39   CONTINUE
     GOTO 33
40   FEXTRP = DMAX1(PREVER/ERRER,AITLOW)
     PREVER = ERRER
     IF(L.LT.5)GOTO10
     IF(LEVEL.GE.3)WRITE(NOUT,641)ERRER,ERGOAL,FEXTRP,IT
641  FORMAT(23H ERRER,ERGOAL,FEXTRP,IT,2E15.8,E14.5,I3)
     IF(L-IT .GT. 2 .AND. ISTAGE .LT. MXSTGE) GOTO 90
     IF(ERRER/FEXTRP**(MAXTBL-L) .LT. ERGOAL)GOTO 10
     GOTO 90
50   IF(LEVEL .GE. 4)WRITE(NOUT,649)L,T(1,LM1)
649  FORMAT(I5,E16.8,5X,4HJUMP)
     IF(ERRER .GT. ERGOAL) GOTO 90
     DIFF = DABS(T(1,L))*2.*FN
     IF(LEVEL .GE. 2)WRITE(NOUT,650)DIFF,BEG,END
650  FORMAT(13X,36HINTEGRAND SEEMS TO HAVE JUMP OF SIZE ,E13.6,
    1 8H BETWEEN ,E15.8,4H AND ,E15.8)
     GOTO 80
60   IF(LEVEL .GE. 4)WRITE(NOUT,660)L
660  FORMAT(I5,21X,13HSTRAIGHT LINE)
     SLOPE = (FEND-FBEG)*2.
     FBEG2 = FBEG*2.
```

```
      DO 61 I = 1,4
       DIFF = DABS(F(time,BEG+RN(I)*STEP,x,y)-FBEG2-RN(I)*SLOPE)
       IF(DIFF .GT. TABTLM) GOTO 72
61    CONTINUE
      IF(LEVEL.GE.3)WRITE(NOUT,667)BEG,END
667   FORMAT(27X,43HINTEGRAND SEEMS TO BE STRAIGHT LINE BETWEEN ,
     1E15.8,4H AND ,E15.8)
      GOTO 80

70    IF(LEVEL .GE. 4) WRITE(NOUT,670)L,T(1,LM1)
670   FORMAT(I5,E16.8,5X,5HNOISE)
      SLOPE = (FEND-FBEG)*2.
      FBEG2 = FBEG*2.
      I=1
71    DIFF = DABS(F(time,BEG+RN(I)*STEP,x,y) - FBEG2-RN(I)*SLOPE)
72    ERRER = DMAX1(ERRER,ASTEP*DIFF)
      IF(ERRER.GT.ERGOAL)GOTO 91
      I = I+1
      IF(I.LE.4) GOTO 71
      IF(LEVEL .GE. 3)WRITE(NOUT,671)BEG,END
671   FORMAT(15H NOISE BETWEEN ,E15.8,4H AND,E15.8)
      IFLAG = 3
80    CADRE = CADRE + VINT
      ERROR = ERROR + ERRER
      IF(LEVEL .LT. 3) GOTO 83
      IF(LEVEL.LT.5) GOTO 82
      DO 81 I = 1,L
       WRITE(NOUT,692)I,(T(I,J),j=1,L)
81    CONTINUE
82    WRITE(NOUT,682)VINT,ERRER,L,IT
682   FORMAT(12H INTEGRAL IS,E16.8,7H, ERROR,E15.8,9H  FROM T(,
     1 I1,1H,,I1,1H))
83    IF(RIGHT) GOTO 85
      ISTAGE = ISTAGE-1
      IF(ISTAGE .EQ. 0) RETURN
      REGLAR = REGLSV(ISTAGE)
      BEG = BEGIN(ISTAGE)
      END = FINIS(ISTAGE)
      CUREST = CUREST-EST(ISTAGE+1) + VINT
      IEND = IBEG-1
      FEND = TS(IEND)
      IBEG = IBEGS(ISTAGE)
      GOTO 94
85    CUREST = CUREST + VINT
      STAGE = STAGE*2.
      IEND = IBEG
      IBEG = IBEGS(ISTAGE)
```

```
      END = BEG
      BEG = BEGIN(ISTAGE)
      FEND = FBEG
      FBEG = TS(IBEG)
      GOTO 5
90    REGLAR = .TRUE.
91    IF(ISTAGE .EQ. MXSTGE) GOTO 950
      IF(LEVEL.LT.5) GOTO 93
      DO 92 I = 1,L
       WRITE(NOUT,692)I,(T(I,J),J=1,L)
92    CONTINUE
692   FORMAT(I5,7E16.8/8E16.8)
93    IF(RIGHT)      GOTO 95
      REGLSV(ISTAGE +1) = REGLAR
      BEGIN(ISTAGE) = BEG
      IBEGS(ISTAGE) = IBEG
      STAGE = STAGE/2.
94    RIGHT = .TRUE.
      BEG = (BEG + END)/2.
      IBEG = (IBEG+IEND)/2
      TS(IBEG) = TS(IBEG)/2.
      FBEG = TS(IBEG)
      GOTO 6
95    NNLEFT = IBEG - IBEGS(ISTAGE)
      IF(IEND+NNLEFT .GE. MAXTS) GOTO 900
      III = IBEGS(ISTAGE)
      II = IEND
      DO 96 I = III,IBEG
       II = II +1
       TS(II) = TS(I)
96    CONTINUE
      DO 97 I = IBEG,II
       TS(III) = TS(I)
       III = III+1
97    CONTINUE
      IEND = IEND +1
      IBEG = IEND - NNLEFT
      FEND = FBEG
      FBEG = TS(IBEG)
      FINIS(ISTAGE) = END
      END = BEG
      BEG = BEGIN(ISTAGE)
      BEGIN(ISTAGE) = END
      REGLSV(ISTAGE) = REGLAR
      ISTAGE = ISTAGE +1
      REGLAR = REGLSV(ISTAGE)
      EST(ISTAGE) = VINT
```

```
      CUREST = CUREST + EST(ISTAGE)
      GOTO 5

  900 IF(LEVEL .GE. 2)WRITE(NOUT,6900)BEG,END
 6900 FORMAT(37H TOO MANY FUNCTION EVALUATIONS AROUND/ ,
     1 10X,E15.8,4H AND,E15.8)
      IFLAG = 4
      GOTO 999
  950 IFLAG = 5
      IF(LEVEL .LT. 2) GOTO 999
      IF(LEVEL.LT.5)GOTO 959
      DO 958 I = 1,L
       WRITE(NOUT,692)I,(T(I,J),J=1,L)
  958 CONTINUE
  959 WRITE(NOUT,6959)BEG,END
 6959 FORMAT(12X,38HINTEGRAND SHOWS SINGULAR BEHAVIOR OF ,
     1 20HUNKNOWN TYPE BETWEEN ,E15.8,4H AND ,E15.8)
  999 CADRE = CUREST + VINT

      RETURN
      END
```