DISSERTATION

A FOURIER FINITE ELEMENT MODEL

FOR

UNSATURATED FLOW IN POROUS MEDIA

Submitted by

Mohamed Ali Dhbeel AL-Dhamari

Department of Civil Engineering

In partial fulfillment of the requirements

for the Degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Summer 2002

UMI Number: 3063970

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

COLORADO STATE UNIVERSITY

April 18th 2002

WE HEREBY RECOMMEND THAT THE DISSERTATION PREPARED UNDER OUR SUPERVISION BY MOHAMED BIN ALI SALEH DHBEEL AL-DHAMARI ENTITLED A FOURIER FINITE ELEMENT MODEL FOR UNSATURATED FLOW BE ACCEPTED AS FULFILLING IN PART REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY.

Committee on Graduate Work

Adviser

Department Head

بسم الله الرحمن الرحيم

((وأنزلنا من السماء ماءً بقدر فأسكناه في الأرض و إنا على ذهابٍ به لقادرون))

صدق الله العظيم "سورة المؤمنون ١٨"

((And we send down water from the sky according to (Due) measure, and we cause it to soak in the earth. And we certainly can able to drain it off (with ease))) "23-18"

ABSTRACT

A FOURIER FINITE ELEMENT MODEL FOR UNSATURATED FLOW IN POROUS MEDIA

A new numerical methodology for solving the partial differential equation governing the unsaturated flow of water in porous media is developed and applied to approximate the hydraulic head and to solve for the flux functions that obey the one-dimensional unsaturated flow equation. The algorithms are based on a Fourier finite element type of formulation, which combines the efficiency of the Fourier series expansion as a refinement tool and the finite element method, where the spatial domain is divided into discrete elements and nodal points. A truncated Fourier series expansion was used as the approximating function over orthogonal discrete elements. The Fourier cosine series was selected for the expansion series and the same denomination is used for the numerical integration techniques using polynomials of a high order as approximating functions. The solution to the partial differential equation is obtained by marching through time in discrete steps, beginning with a known solution of Fourier series coefficients at time zero. A Fourier finite element computer program based on the new methodology was written and applied to modeling a one-dimensional flow of water into porous media. The nonlinear solution behavior was well simulated by the high order approximating functions. The accuracy of the numerical solution used in this model was

evaluated by analyzing a test problem for which the solution is available, and a good agreement was obtained.

Mohamed Ali Dhbeel Al-Dhamari
Department of Civil Engineering
Colorado State University
Fort Collins, CO 80523
Summer, 2002

ACKNOWLEDGEMENTS

The completion of my entire graduate studies, including this research work, would have been impossible without the help and gracious guidance of Almighty Allah. I thank him first and last for His generosity and mercy. I would like to extend my deep and sincere thanks and love to my beloved parents, my father Ali S. Dhbeel and my mother Fatema A. Sina'ah, for their unconditional love and unlimited support, which words cannot express. My thanks are also due to the rest of my family for their abundant love and encouragement.

I would like to thank Prof. James W. Warner, my advisor, for his continuous advice and generous assistance throughout this study. I wish to thank other members of my graduate committee; Prof. Darrell G. Fontaine, Dr. William E. Sanford, Dr. Sameh M. Affifi. Thanks also to Dr. Ramiz Namvar-Golin and Dr. Thomas C. Sale for their friendship and time.

My greatest debt is to my wife, Dhekra, who endured many hardships and made many personal sacrifices during the past several years; this work was possible only with her unending love, support and infinite patience. I thank my children, Majed, Ahmed, and Manar, for tolerating my absence from home and for the joy they provided on difficult days. Again, I would like to thank my beloved parents who have been a great source of inspiration and to whom I dedicate this work.

DEDICATION

To my beloved parents

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Description of the Problem

Unsaturated flow is a process of simultaneous flow of two immiscible fluids, air and water. Most of the time water flows in soil under unsaturated conditions. Flow in the unsaturated zone is becoming extremely important in groundwater studies to simulate the artificial recharge and/or natural recharge as a result of vertical infiltration process.

Moreover, unsaturated zone plays a significant role in a large number of groundwater engineering problems; for example, the movement of contaminant and toxic leachates to groundwater aquifers takes place largely via a vertical flow process under unsaturated conditions.

Application of the governing Richards' equation, which is nonlinear by nature, usually simulates the dynamic of such process. Whereby, the exact analytical solutions for such an equation, for most cases, are almost impossible. Numerical solutions of the governing unsaturated flow equation by classical finite elements and finite difference methods are easily available. However, the non-linearity nature of such an equation, have made the numerical solutions via the finite-element method more attractive technique due to its flexibility in terms of selecting the appropriate shape functions when dealing with various types of complicated geometries.

Numerical solutions of Richards' equation via conventional finite element approach have sacrificed either accuracy or computational efficiency when modeling vertical infiltration into relatively coarser and dry porous media due to the great numerical difficulties caused by the rapid advancement of the sharp wetting front on a local domain of interest. In addition, the non-linearity nature of the Richards' equation have made the conventional mass distributed finite element method suffers from numerical oscillations at the wetting front, especially under heterogeneous conditions for vertical infiltration into initially dry soils.

To overcome such difficulties, and to obtain accurate and proper solutions, the application of some sort of resolution enhancement techniques are thought to be extremely important steps for both solutions accuracy and computational efficiency.

Although, significant progress has been made in the field of the adaptive grid methods for the solution of partial differential equations, the computational time required for most of these methods is typically high. For example, the use of multigrid methods for local refinement, it is often necessary to calculate the solution in the sharp wetting front zone more accurately than in other areas of the problem domain (i.e boundary layers). Therefore, a non-uniform grid with more grid points in these regions is used. These local grid refinements increase the computational time because the discretized differential equations become much more complicated. The moving-grid method has limitations regarding applicability to layered systems and use under time-varying boundary conditions. The disadvantage associated with the application of multi-level adaptive techniques is that to work not only with a single grid but also with a sequence of grids of increasing fineness at the sharp wetting front zone.

Each grid may be introduced and changed several times during the solution process and according to the advancement of the sharp wetting front; these would result in enormous waste of efforts.

Although, such regions often represent small fraction of the physical domain, the application of a finer grid would certainly increase the number of equations to be solved, maximize storage requirements and minimize computational efficiency.

To describe unsaturated flow, three available standard forms of the governing Richards' equation may be used: 1) Pressure-head based form, 2) Volumetric water content-based form and 3) The mixed-form of the governing unsaturated Richards' equation. Although, when solved numerically, the pressure head-based form of the governing Richards' unsaturated flow equation has been shown to lead to significant mass-conservation error in the solution when compared to the alternative mixed-form and volumetric water content-based formulations. The pressure head-based form has been selected for the model developed here in order to overcome such difficulties.

On the other hand, a literature survey has revealed that numerical solution based on volumetric water content-based formulation of Richards' unsaturated flow equation generally yields poor results, characterized by significant discontinuity in the volumetric water content at the interface of the layering soil materials where θ create a discrete jump at layer interfaces. Conversely, for infiltration into layered soil materials, the standard pressure-head based form will produce continuous hydraulic head and pressure head functions from ground surface to water table.

Moreover, for infiltration into dry soils, the conventional finite element approximations produce oscillatory solutions even while conserving mass. However, the

mass-conservative formulation is not adequate to guarantee good approximations. In some situations, the spatial approximation used is also very significant.

Therefore, the need for a new local solution refinement technique becomes of great practical important to enhance resolution for the domain of the local interest. However, there is no currently available general model of the Richards' unsaturated flow equation that incorporates recent advances in the computational effort and accuracy of the solution of this equation system via the application of a Fourier finite element method.

1.2 Objectives of Research

The broad objective of the present study is to develop an alternative numerical technique called: the "Fourier Finite Element Method for Unsaturated Flow". This method deals with numerical solutions to the modified Richards' equation to simulate one-dimensional vertical flow processes in the unsaturated zone of layered soil profiles.

This approach combines the efficiency of the Fourier series expansion as a refinement tool and a finite element method that provides a mechanism to split the overall unsaturated flow domain into a collection of nodal points and elements within which local resolution refinement can be specified based on the non-linearity of the problem and the local behavior of the sharp wetting front in the unsaturated flow domain.

The specific objectives of this dissertation are to:

1. Develop a numerical solution procedure for local solution refinement of the modified pressure-head based form of the Richards' equation to simulate one-dimensional flow process in the unsaturated zone of layered soil profiles.

2. Develop a computer code to simulate a vertical one-dimensional flow process in the unsaturated zone of homogenous and layered soil profiles.
3. Verify the proposed numerical solution procedure by comparing its results for example problems with other existing solutions.

The solution procedures as developed for the modified Richards' unsaturated flow equation are readily applicable to other problems governed by parabolic diffusion partial differential equations.

1.3 Basic Concepts

A finite element method using a Fourier series expansion technique is designed to model a one-dimensional vertical flow process in the unsaturated zone of layered soil profiles. The problem subdomain, which exhibit a local phenomena of highly nonlinear behavior in the solution and/or that require a higher accuracy of the solution, are discretized into collection of elements and nodal points.

A finite Fourier cosine series expansion of variable order is applied as the approximating function to approximate the head solution over each element. More accurate solutions can be achieved by using a higher order approximating function where required (higher order of harmonics or modes). To insure continuity of the solution, the number of harmonics applied for the direction perpendicular to the interface between two neighboring elements, must be the same.

For the application to the unsaturated flow problem presented in this study, the hydraulic head and pressure head are required to be continuos over the element interfaces in order to provide a better accuracy of the solution. The main purpose for applying a

Fourier finite element technique is to obtain a highly accurate approximation while minimizing storage requirements.

For local solution refinement, this new technique could be used in connection with other methods or linked with simpler numerical models to simulate rapid change of the solution in a particular direction.

CHAPTER 2

LITERATURE REVIEW

2.1 Unsaturated Flow

Unsaturated flow is a process of simultaneous flow of two immiscible fluids, air and water. Most of the time water flows in soil under unsaturated conditions. Moreover, it is assumed that air phase is inter-connected and continuous. Consequently, air can easily escape ahead of the wetting front and allow the soil moisture to proceed downward by gravity and vertical hydraulic gradients. Thereby, three important physical processes are related and commonly encountered by groundwater hydrologists are, infiltration of surface water into the soil to become soil moisture, subsurface flow or unsaturated flow through the soil, and ultimately, groundwater flow or saturated flow through the porous media.

Historically, the study of unsaturated flow has been a field of interest of soil physicists and agronomists but, recently it also become an important domain of groundwater hydrologists for the practical prospective in studying contaminant movements from the unsaturated to saturated zone. [Sposito, 1986], attributes the beginning of the study of unsaturated flow as a subdiscipline of physics to the fundamental work of [Buckingham, 1907], who proved experimentally that soil water matric potential is a function of water content at constant temperature and applied pressure.

Consequently, Buckingham was able to modify the differential form of Darcy's law to describe the unsaturated flow in porous media. He introduced the concept of the unsaturated hydraulic conductivity, K (θ), soil water matric potential, $\psi(\theta)$, and diffusivity, D (θ), as all being functions of the volumetric soil water content. Yet, his experimental works on soil water diffussivity was limited. Buckinghams' modified equation still employed today in modeling unsaturated flow and it is known as Darcy-Buckingham equation.

[Scheidegger, 1974], declared that the extensions of Darcy's law to multiple phase flow is only theoretical thought. He points to the phenomenon of hysteresis in wetability as indicative of some of the limitation of the approach.

[Marino and Luthin, 1982], stated that water flow through porous media follows Darcy's law and concur with Hillel's, 1980, statement that perhaps the most important differences between unsaturated and saturated flows is the hydraulic conductivity.

2.1.1 Richards' Equation

The next forward step was taken by [Richard, 1931], a soil physicist, who was able to combine Darcy's law with the law of conservation of mass to derive a non-linear partial differential equation known as the governing Richards' equation for unsaturated flow, the equation assumes water is incompressible, soil matrix is non-deformable and the effect of air is negligible.

Substantially, all studies related to unsaturated zone, the fluid motion is assumed to obey the conventional form of Richards' equation. Three standard forms of the governing Richards equation may be identified:

1) Pressure-head based form, 2) Volumetric water content-based form, and 3) The mixed form, these equations are described below.

2.1.1.1 Pressure-head based form

This equation is the classical form of Richards' equation, it was written in terms of soil water pressure head, ψ, as dependent variable. This particular form of equation was developed by combining the Darcy's equation with the principle of mass balance, replacing the saturated hydraulic conductivity Ksat by K (ψ) and considering flow continuity through two opposite side of a nondeformable differential cubic element in which a balance of mass flow entering, leaving, and changing within the element are maintained. This process can be described mathematically by a partial differential equation as the net mass efflux on one side of the equation is equated to the other side of the equation as a rate of change in the mass flow.

Applying expansion to the time derivative term, the resultant partial differential equation contains the implicit assumption that water is homogeneous and incompressible which allow discarding the variable of water density, ρ, from both sides of the equation. Upon substitution for the hydraulic head in terms of unsaturated flow as '$h = \psi \pm z$', depending on z direction, we can obtain the pressure head-based form of Richards' equation. This particular form can be used for describing water flow in combined saturated-unsaturated porous media.

2.1.1.2 Water content based form.

This form of Richards, equation was written in terms of moisture water content, θ, as dependent variable, known as the water content based form of Richards' equation. It

is simply derived by replacing K (ψ) with K (θ) in the pressure-head based form and, assuming that θ and ψ are uniquely related, as smooth continuous functions, and multiply both sides of the equation by $\partial\theta/\partial\theta$. Interchanging $\partial\theta$ with $\partial\psi$ in the numerator and substituting the diffusivity term as $D(\theta) = K(\theta)\,\partial\psi/\partial\theta$, yields the water content based form.

2.1.1.3 The Mixed form

The last form of Richards' equation was obtained by applying the time derivative to the water content and the space derivative to the pressure head, the resulted equation of the two unknowns, ψ and θ, is so-called the mixed form of Richards' equation. Each of the above-described forms of equations has its own disadvantages and difficulties in modeling infiltration process as will be described hereafter.

2.1.2 Infiltration Modeling

2.1.2.1 General Principles

Infiltration is the process whereby water penetrating into the soil from surficial sources, mainly, rainfalls, irrigation, flooding and snowmelt etc. It is usually considered as a process that occurs predominately in the vertical direction when water penetrates from the ground surface into the soil. Thus, infiltration rate is the volume rate at which water flowing into a unit area of soil surface. Several factors influence the infiltration processes such as the rate of water application, antecedent moisture in the soil, soil hydraulic properties, soil heterogeneity and others.

Most of the theoretical and experimental developments concerning unsaturated flow in the literature classify the distribution of soil moisture, within the soil profile,

during infiltration process, into four zones: starting from the soil surface, a saturated zone, near the surface, a transmission zone, in which the unsaturated flow occur, a wetting zone, in which moisture decrease with depth and, a wetting front zone, which is located at the mid-point of the wetting zone and regarded as the most critical zone in modeling the unsaturated flow due to the appearance of a sharp discontinuity between the wet soil above and the dry soil below.

2.1.1.2 Theoretical and Experimental Investigations

Several investigators have proposed quantitative and operational models to describe the vertical infiltration process. One of the early model was proposed by [Green, W. H. and G.A. Ampt, 1911], known widely as Green-Ampt method, this method considered as an early solution to Richards' equation.

The model assumed water to move vertically into a dry soil as a sharp wetting front boundary located between the upper and lower layers of the investigated soil profile. Such a boundary separates soil of saturated water content, θ_s, above from that of initial water content, θ_i below. The model neglects the depth of ponding at the soil surface.

[Horton, 1933], showed empirically that water infiltrates from soil surface at a rate exponentially decreases with time. He defined the curve of infiltration capacity versus time as a limited curve to determine the maximum possible rate of infiltration for a given soil. [Eagleson, 1970] and [Raudkivi, 1979], have shown that Horton's equation can be derived form Richards' equation assuming that diffussivity and the unsaturated hydraulic conductivity are constant parameters independent of soil moisture content.

[Kostiakov, A. N., 1932], proposed an empirical infiltration model in which the infiltration rate and the time be exponentially related via constant variables depend on the soil and its initial conditions. The model has limitations regarding applicability to other soils and conditions different from which its constant variables were determined.

[Philip, 1957, 1969], provided a series of solutions for vertical infiltration into a semi-infinite homogeneous soil with initial water content and a constant pressure head maintained at soil surface. Virtually, Philip solved Richards' equation under less constrained conditions as he employed the Boltzman transformation to convert the differential water content-based form of Richards' equation into an ordinary differential equation, assuming that the unsaturated hydraulic conductivity and diffusivity can vary with water content.

[Parlange J.-Y and D.E. Hill, 1979], presented an analysis of air and water movement in a sand column, taking air compressibility into account. In the analysis, Parlange found that the influence of air on water intake was assessed when water content was imposed at the soil surface. He showed that air compressibility was negligible if the air moved freely a head of the wetting front. However, the air movement increases water intake by a small percentage. On the other hand, when air escaped through the sand surface, air compressibility increased water intake by small percentage as well. Parlange concluded that the overall effect of air movement through the sandy soil column was to reduce the water flow.

[Parlange, J-Y. 1980], analyzed the effect of gravity on water flow instability and water intake by layered soil during ponded infiltration. Several layers of sandy loam and coarse sand soils of variable depths of 20, 40, 60, 120 and 180 cm were placed in a

column of 1.8-m diameter. The effects were analyzed when the flow was one dimensional in the z-direction, measured positively downwards. Parlange found that the water intake by soil was increased by the presence of gravity.

A second effect of gravity was analyzed when the vertical flow became unstable and moved along preferred paths or fingers. Observation showed that water moved through out the whole soil in the top 60 cm but only in a few zones, i.e. fingers below that depth.

Parlange indicated that the flow was faster through the fingers as the total cross section of the fingers being small. He concluded that the instability of the flow was responsible for the formation of the fingers originates at the interface of the fine and coarse layers.

2.1.3 Analytical Solutions

Analytical solution to the governing Richards equation for unsaturated flow exists for certain special cases i.e homogeneous soils, although such solutions are difficult to obtain under various boundary and initial conditions due to the non-linearity in soil hydraulic properties. The analytical solution becomes extremely difficult especially under heterogeneous conditions.

During the past few decades, many analytical and quasi-analytical solutions to the unsaturated flow equation have been developed. Recently, many analytical solutions for water infiltration into homogeneous soil profiles have been developed by [Broadbridge et al.1988], [Broadbridge and White, 1988], [Warrick et al.1990 and 1991], [Sanders et al.1991], these solutions are based on transformation methods developed by [Rogers et al.1983], which allows exact analytical solutions to be obtained for infiltration problems

for particular nonlinear diffussivity and conductivity functions by using Backlund transformation but, this method is applicable only for the special case of a constant fluxes at the surface boundary.

[Barry, D. A. and G. C. Sander, 1991], attempted an analytical solution to Richards, equation using the Backlund transformation to linearized the equation and to derive a solution that is valid for arbitrary surface fluxes and arbitrary initial specification of water content. In their paper, they were able to obtain a general solution for an arbitrary temporal variation in the surface flux. But, their solution involves an unknown boundary integral function specified by a Volterra integral equation and must be evaluated numerically.

The analytical solutions of the governing Richards' equation in one-dimensional vertical flow almost always require simplified boundary conditions; however, these approaches are useful for simplified engineering models for hydrologic prediction and sensitivity analysis.

[Salvucci, G. D.1996], derived a series solution to Richards' equation for the case of uniform initial condition and concentration boundary condition, his solution was similar to that of [Philip, 1957], with the exception of using a transformed time variable never exceed unity. The resulted series yields a general analytical solution that is not restricted to time less than that at which the effect of gravity is as large as capillarity, that time was defined as the gravity time. Salvucci showed that the resulted solution is applicable for infiltration without the use of approximate joining techniques.

[Serrano, S. E., 1998], proposed an analytical decomposition series for the solution of the nonlinear unsaturated flow equation in which a horizontal and vertical

infiltration equations subjected to constant boundary conditions and continuous wetting were approximated. For the verification of the solution to the vertical infiltration equation, he indicated that the decomposition solution reasonably predicts the evaluation of water content versus depth profile only for the earlier infiltration times. However, for large times, the decomposition solution underestimates the position of the wetting front and the need for a third term in the series increases, indicating that the accuracy of the solution will improve as more terms are calculated.

Serrano concluded that at near saturation values of the water content, a linearized differential equation with a limiting high diffusivity values is a reasonable model while in any other region, the effect of the non-linearity becomes strong.

A comparison' study between the analytical solution described by [Hauntush, 1967] and a numerical approach based on finite difference computer code VS2D [Lappala et al.1987] was recently conducted by [Sumner, et al.1999]. The aim of the study was to estimate the effect of unsaturated zone on the magnitude of groundwater mounding beneath an infiltration basin.

They found the error introduced by the analytical solution that ignores the effect of the unsaturated zone on groundwater mounding was increased as the basin-loading period is shortened. In another words, the error increased as depth to the water table increases with increasing subsurface anisotropy and with the inclusion of fine-texture strata.

Although, the analytical solutions are essential tools for investigating the effect of heterogeneity on flow and transport in the unsaturated zone, however, the non-linearity of the governing unsaturated flow equation, the degree of non-linearity and the spatial

variability in the hydraulic parameters often restricted the use of analytical solutions to some special cases.

For more general and large-scale problems, groundwater hydrologists often rely on numerical modeling, such as finite difference and finite elements.

2.1.4 Numerical Methods

Flow in the unsaturated zone is becoming extremely important in groundwater studies to simulate the artificial recharge and/or natural recharge as a result of vertical infiltration process. Application of the governing Richards' equation, which is nonlinear by nature, usually simulates the dynamic of such process. Whereby, the exact analytical solutions for such an equation, for most cases, are almost impossible.

Numerical solution techniques of the Richards' equation via finite elements and finite difference methods have been widely developed and used as a powerful tactic to solve the governing unsaturated flow equation. Such tactics offer a great flexibility in terms of geometry of the modeled domain, representation of heterogeneity, and accurate description of the initial and boundary conditions.

Several investigators have examined the accuracy of various numerical solution techniques. [Haverkamp et al.1977], [Van Genuchten, 1982], indicated that a highly accurate solution to the Richards' equation can be achieved by using implicit finite difference schemes with implicit or explicit evaluation of hydraulic properties.

[Van Genuchten, 1982], compared several numerical solutions and concluded that Hermitian finite elements formulation with a four- or five-point Lobatto integration scheme was the most accurate.

[Milly, 1985], presented a mass-conserving solution that modified the specific moisture capacity term to force a global mass balance. In another paper [Milly, 1988], indicated mass balance problem associated with using the pressure-head based form of Richards' equation. Thereby, he suggested an appropriate assumption for the solution involves the air phase essentially remains at a constant pressure equal to atmospheric and, solution was considered for the water phase only.

[Celia et al.1990], thought that selection of the appropriate form of the governing partial differential equation is an important consideration in the development of numerical approximation, indicated that the pressure-head based form of the Richards' equation is inherently non-mass conserving and, concluded that the mixed form should be preferred to both water content and pressure-head formulations.

[Ross, 1990], investigated the computational efforts of two finite difference methods for solving the mixed form of the Richards' equation in one-dimensional infiltration process through a range of soils of different initial conditions. The two techniques are so-called the fixed grid and advancing front methods, in the latter method, the spatial grid points were added as the infiltration/ redistribution front advanced. He concluded that the latter method was highly simulating the infiltration process even in heterogeneous soils under non-uniform conditions.

[Kirkland et al.1992], state that the pressure head-based form of the Richards' equation is inaccurate for infiltration into a very dry soil. [Celia et al. 1992], concurs with their statement and concluded that the resulting unsaturated equation can be numerically difficult to solve. Moreover, [Celia et al. 1992], presented a numerical solution for two-phase flow in porous media using the mixed form of the Richards' equation coupled with

a finite-element approximation lumped in space and a modified Picard linearization as a dynamic time step control. They concluded that the obtained solutions were mass conservative and oscillation free even in the presence of steep infiltration front.

During the current decade, more sophisticated numerical techniques to the unsaturated flow equations have been developed to improve the computational efficiency, which lead to convergence in the iterative solution process.

[Harter and Yeh, 1993], used an analytical perturbation solution as an initial guess to a finite element model to simulate a vertical infiltration process in random porous media. The model was able to solve the governing nonlinear Richards equations reducing CPU time by one to two orders of magnitude when a perturbation solution was applied.

[Pan et al.1995], proposed a nonlinear transformed pressure to solve the pressure head-based form of the Richards' equation for variably saturated heterogeneous media, their approach was compared to and contrasted with φ-based transformation method by [Kirkland et al.1992] and h-based modified Picard method, [Celia et al.1990]. In comparison with those methods, they indicated that their approach offers an excellent CPU efficiency and did not require difficult numerical coding.

[Romano et al.1998], developed a finite difference algorithm based upon flux conservation and pressure continuity at layers interfaces to estimate the hydraulic conductivity values between two neighboring nodes of layered soil profiles, refer to as interlayer conductivity. The result was then implemented in Crank-Nicholson method to solve the pressure head-based form of the Richards' equation to simulate numerically one-dimensional unsaturated flow processes. By comparison with a numerical model based on a geometric-averaged hydraulic conductivity values, they concluded that their

results show significant reduction in relative mass balance errors and a very high computational efficiency.

[Babel et al.1997], developed a numerical model based on the pressure head-based form of Richards' equation to simulate stable and unstable flow in unsaturated coarse sands for one-dimensional experiment column conducted in the laboratory, the unstable flow due to wetting front instability was modeled using the proposed steady-state theory by [Hillel and Baker.1988].

They concluded that the antecedent wetness on fingering behaviors was to reduce the pore water velocity drastically from air dry to field capacity. Moreover, they indicated that the use of water content-based form of Richards' equation and potential variables averaged over total cross-sectional area was not valid for water flow in instability-prone boundary conditions of infiltration of non-ponding rainfall or irrigated sands and unsaturated sandy soil.

[Mirabzadeh et al.1997], developed a numerical model for the water flow in two dimensions in the unsaturated zone using the pressure head-form of Richards' equation. The model based on finite difference scheme and the solution to the flow equation was obtained via a dynamic programming method. They concluded that such a method was able to reduce the computational time, storage requirements and the computing error due to inversion of large matrices.

[Miller et al.1998], introduced computational approaches involve interblock permeability, using non-linear algebraic system approximation methods and two time integration approaches to investigate the numerical convergence difficulties associated with using Richards' equation for non-uniform porous media. They indicated that a

Hermit spline interpolation method is more accurate than the standard linear interpolation methods.

[Totoev et al.1998], described an infiltration numerical model which has been developed to simulate time dependence of the vector of sources and sinks for the finite element solution of the Richards' equation. As a result of the meteorological input data, the aforementioned model was able to predict the seasonal suction changes in the soil profile and to reflect the major meteorological changes in a small area (i.e. several hundred square meters). In addition, they indicated that such a model could be used with the Richards' equation to simulate the soil moisture flow in a small urban area.

[Wildenschild et al.1999], investigated the performance of a deterministic model applied to a heterogeneous soil system and the applicability of estimated effective parameters were tested by comparing numerical simulation results to the measurements of transient flow events obtained in two dimensional laboratory experiment.

The aforementioned model was able to solve the Richards' equation in two dimensions in which the Galerkin finite element scheme, based on triangular elements and linear interpolation functions was used to solve the time-independent part of the equation for each time step, while the time derivatives was approximated by finite difference using a fully implicit approximation. Moreover, the Picard's iterative scheme was applied to overcome the nonlinearity troublesome arising from the specific water capacity and hydraulic conductivity parameters.

They concluded that the model was able to simulate reasonably the values of the capillary suction measurements only under the geometric statistical averaging approach, indicating that the arithmetic approach did not produce accurate results. In addition, the

numerical simulation failed to reproduce the pressure drops due to the change in flow rates.

2.1.5 Numerical Difficulties in Modeling Unsaturated flow

Numerical solutions of the governing unsaturated flow equation by classical finite elements and finite difference methods are easily available. But, the nonlinearirty nature of such an equation, have made the numerical solutions via the finite-element method more attractive technique due to its flexibility in terms of selecting the appropriate shape functions when dealing with various types of complicated geometries.

[Gottardi et al.1992 and 1994], reported that the numerical integration of the Richards' equation by classical fixed-grid finite element and finite difference methods often requires a fine discretization on space and time to obtain accurate solutions. This would lead to an intensive computational task and the efficiency of the relevant computer simulation could be severely diminished.

2.1.6 Alternative Methods

Unfortunately, numerical difficulties caused by wetting front instability can also arise sometimes, even with numerical methods by finite element solutions, especially under heterogeneous conditions for infiltration into initially dry soils, or for coarser materials that are characterized by sharp wetting fronts.

For this reason, there are strong efforts to overcome such difficulties and to obtain substantial computational efficiencies. If some type of iterative solution technique is used, the Fourier finite element method can provide robust, parallel and scalable preconditioned iterative methods for the large-scale linear systems arising from

discretization of the problem domain by finite elements, finite differences or Fourier finite element methods.

2.2 Local Resolution Enhancement Methods

Instability of the wetting front is a major numerical difficulty associated with the solutions of the governing Richards' equations. The wetting front zone is regarded in the literatures as the most critical zone in modeling the infiltration process. To overcome such difficulties, and to obtain accurate and proper solutions, the application of a resolution enhancement techniques are have to be extremely important steps for solutions accuracy.

Since these difficulties are limited to small portion of the problem domain, the use of the finer global grid over the entire problem domain is costly and computationally inefficient. Therefore, the need for local solution refinement technique becomes of great practical important to enhance resolution for the domain of the local interest.

Even though, several application methods have been used in the literature to obtain local resolution enhancement, namely Fixed Grid with variable cell spacing, Fast Adaptive Composite Grids (FACG), Moving-grids, Multi-grids, Domain Decomposition method.

On the other hand, a new method called 'Fourier finite element method', which combines the efficiency of the Fourier series as a refinement tool with the finite element method, will be used in this work for the solution of the governing Richards' equation to obtain local resolution enhancement in the wetting front zone associated with unsaturated flow modeling.

2.2.1 The Adaptive Methods

Adaptive method is a promising technique in the numerical solution of differential equations. Adaptivity is particularly advantageous when the region requiring high resolution consists of only a small fraction of the problem domain. An adaptive mesh generation procedure adjusts the location of mesh points or adds and subtracts points using feedback from a previous numerical solution of a problem.

Three main types of adaptive techniques for the finite element methods were found in the literature, namely (1) h-method, which refines and coarsens the mesh locally according to certain error indicators. (2) p-method, which selects the polynomial degree used in the finite element approximation in each element according to the smoothness of the solutions and (3) r-method, which relocates the grid points to concentrate them in the desired regions.

Moreover, An adaptive algorithm may be considered to be a computational procedure for constructing a finite element discretization for a given problem. Such as, the error associated with an approximate solution in a given model is located within a given tolerance such that the number of degree of freedom is insignificant. An advantage of using adaptive finite element method may be expected to be the substantial saving in computational effort for a given accuracy.

During the last and current decades, several investigators have developed many adaptive grid methods for the solution of partial differential equations. As stated in the literatures, significant progress has been made in finite element adaptive mesh methods. However, progress has been slower in finite difference application due to the difficulties associated with discretization on nonuniform mesh.

Several researchers have studied the use of the adaptive techniques via the application of the finite element methods [Babuska and Rheinboldt, 1978], [Bank, 1981], [Miller and Miller, 1981], [Miller, 1981], [Davis and Flaherty, 1982].

[Babaska et al. 1983], have given a review of adaptive finite element grid methods, they developed a set of data entry finite element methods for a one-dimensional boundary value problem. Moreover, they presented the most effective use of their mathematical results with respect to several performance measures. In addition, similar application and results were illustrated and discussed for a two-dimensional problem for plane elasticity.

[Flaherty et al. 1983], introduced an adaptive finite element grid that was applied to solve a one-dimensional initial-boundary value problems for vector systems of partial differential equations. They indicated that the method was capable to adjust the computational grid in a self-regulating manner as the solution progress over time. Moreover, the method was also able to roughly minimize the local discretization error. In comparison with the classical uniform grid method, the former method was able to produce highly accurate solutions with small number of elements.

[Devloo et al. 1988], used an h-p hybrid adaptive version of finite element method to numerically simulate the flow of a compressible fluid. The refinement technique was applied locally in a region of large solution variation. In addition, high-order polynomials were also used for the same purpose.

They concluded that the degree of the polynomial of the elements was noticeably increased as a result of the local refinement application. However, the employment of the

high-order polynomials produced highly accurate approximations to the equations governing viscous compressible flow.

[Altas, I. and J. Stephenson, 1989], developed an automatic two-dimensional adaptive mesh generation method. The method was designed so that a small portion of the mesh can be modified without disturbing a large number of adjacent mesh points. Moreover, the method could be used with or without boundary-fitted coordinate generation procedure.

They concluded that the method generates a well-suited mesh for problems whose solutions have large variations like large gradients, boundary layers and sharp peaks. In addition, the method could handle problems having mild singularities in their solutions.

[Berger, M.1989], presented computations that use local adaptive mesh refinement technique to solve multidimensional, time dependent shock hydrodynamics problems. In this approach, the refined regions consist of a small number of rectangular grid patches with finer mesh spacing than the underlying global coarse grid. These rectangular subgrids contain points where the error in the coarser grid solution is too high and other points as well. The use of the rectangular subgrids was to allow integration methods of known convergence properties. He indicated that utilizing adaptive techniques were essential for such computations in order to resolve features in the solution within computer storage limitations.

[Eriksson K., and C. Johnson, 1991], presented and analyzed an adaptive algorithm for choosing the space and time discretization in a finite element method for a linear parabolic problem. The finite element method uses a space discretization with

mesh-size variable in space and time and a third-order accurate time discretization with time-steps variable in time.

They resolved that the algorithm was reliable in the sense that the posteriori error was guaranteed to be within a given tolerance for all time-steps. Moreover, the algorithm was also efficient in the sense that the approximation error was not essentially below the given tolerance for most time-steps.

[Adjerid et al.1992], investigated a higher-order adaptive solution of a parabolic partial differential systems in one and two space dimensions by finite element procedures that automatically refine and coarsen computational meshes, vary the degree of the piecewise polynomial basis and move the computational mesh in one-dimension. In addition, two-dimensional meshes of triangular, quadrilateral or a mixture of triangular and quadrilateral elements were generated using a finite quadtree procedure.

Moreover, a posteriori estimates used to control adaptive enrichment were generated from the hierarchical polynomial basis. Temporal integration, within a method-of-lines framework, uses either backward difference methods or a variant of the singly implicit Rung-Kutta (SIRK) methods. A high-level user interface facilitates use of the adaptive software.

They resolved that the high-order and variable-order methods combined with mesh refinement offer several advantages relative to the usual low-order adaptive methods. When used with a hierarchical basis, they provide an efficient means of obtaining high accuracy. P-based adaptive strategies are somewhat simpler to implement than h-refinement techniques.

Adjerid et al., provide a natural framework for obtaining a posteriori error estimates at reasonable cost. When used with a variant of the singly implicit Rung-Kutta methods, the higher-order methods may also provide efficient local refinement techniques for time- dependent problems.

[Cao et al.1999], presented an r-adaptive finite element method for solving time-dependent partial differential equation. A moving mesh partial differential equation MMPDE, [Huang and Russell, 1999], was employed to move the (unstructured) mesh in time. The advantage of using MMPED for unstructured mesh movement was to initially define the computational domain Ω_c and to compute the computational mesh Ω^h_c. Moreover, the mesh movement for finite element method was done using MMPDEs in order to facilitate smooth mesh evolution in time.

They indicated that the resulted initial adaptive mesh $\Omega^h(0)$, was insensitive to the choice of the computational domain Ω_c. Furthermore, the initial adaptive mesh showed to have a tendency to concentrate the points in the same regions as the initial mesh. The method showed several limitations, the most significant one was the inability to do error estimation nor to add or remove grid points.

2.2.1.1 Multi-Level Adaptive Techniques (MLAT)

The multi-level adaptive technique employed a succession of grid levels of increasing fineness rather than using a single grid. For highest overall accuracy, the solution of a problem is obtained by alternately solving the problem on several level of coarse to fine grids.

The coarser grids are correction grids that accelerate the convergence of the finest grid by efficiently removing low frequency error components. Conversely, the fine grids

are correction grids that improve the accuracy of the coarser grids by introducing fine grid transfer functions to the coarse grid discrete system.

The multi-level adaptive technique was first introduced by [Achi Brandt, 1977], to solve boundary value problems. Later on, more improvement was suggested by [Bai and Brandt, 1987].

In the literature, [McCormick, 1984] and [McCormick and Thomas, 1986], classified the Multi-level adaptive technique to be generalized by a method called Fast Adaptive Composite Grid. (FAC). Such generalization allows any grid solver to be used to operate each level, whereby making MLAT an adaptive well-organized technique.

[Liu and Joe, 1996], introduced a local refinement algorithm based mainly on an 8-subtetrahedron subdivision, and have shown that the algorithm produces guaranteed-quality meshes.

Experimental results showed that the number of tetra-hedra actually refined in order to keep a conforming mesh is limited by small number times the number of tetra-hedra chosen for refinement, which implies that the algorithm is truly local. They indicated that the aforementioned algorithm is preferable if the mesh in a refined region is relatively coarse; otherwise, the local refinement algorithm based on a bisection procedure, [Liu and Joe1995], may be better.

[Hoppe and Wohlmuth, 1997] developed a fully adaptive algorithm for mixed discretization based on the lowest-order Raviart-Thomas elements featuring a multilevel iterative solver and a posteriori estimator as indicator for local refinement. The adaptive multilevel algorithm was applied to two selected second-order elliptic boundary problems in which the refinement process started from an initial coarse to final refined

triangulations. The local refinement of the triangulation was based on a posteriori error estimator that can be easily derived from super-convergence results.

[Heuer et al.1998], studied a multilevel adaptive Schwarz method for the h-p version of the Galerkin boundary element method with geometrically graded meshes. The method was applied to both hyper-singular and weakly singular integral equations.

Heuer et al., indicated that the use of h-p version with geometrically graded meshes resulted in an exponentially fast convergence in the energy norm. However, such a fast convergence was at the expense of an exponentially fast increasing condition number of the Galerkin system. They recommended a pre-conditioner design in order to solve the systems efficiently.

[Axelsson and Kaporin, 1998], Analyzed a minimum residual adaptive multilevel finite element procedure for the solution of a general nonlinear differential problem with respect to linearization and adaptive finite element approximation errors.

In comparison with a nonadaptive multilevel finite element method, they concluded that the adaptive method was more efficient. However, the error norm was considerably smaller than the residual norm in both cases.

2.2.1.2 Fast Adaptive Composite Grid (FAC)

The Fast Adaptive Composite Grid Method is an adaptive systematic technique often applied for solving differential boundary-value problems. This technique is very similar to MALT.

[McCormick, 1989], described the fast adaptive composite grid methods to be characterized by their use of a composite grid, which is nominally the union of various uniform grids. Furthermore, this method is capable of producing a composite grid with

tailored resolution, and a corresponding solution with commensurate accuracy, at a cost proportional to the number of composite grid points.

Basically, the method develops discretizations based on several regular uniform grids that are used as basis for fast solution. The distinguishing features of the method are that the fine grids cover only local areas. Moreover, discretization is developed directly and not as a result of solution procedure.

The use of fast adaptive composite technique can greatly improve the accuracy of finite element computations, and has been among the most important advances in the field over the past decade. Generally, the method employed both global and local uniform grids to define the composite grid problem and to alternate for relatively fast and accurate solution. However, the accuracy of the method increases if a finer grid is used, especially near the critical regions of large and rapid variation in the solution as the sharp wetting front and steep gradient.

Although, such regions are often represent small fraction of the physical domain, the application of a finer grid would certainly increase the number of equations to be solved. Consequently, maximize storage requirements and minimize computational efficiency. To overcome such an acute situation, substantial amounts of researches have been conducted to develop procedures that enhance the accuracy of the solution for the partial differential equations.

Detailed studies on mathematical progress and prospective of FAC can be found in [McCormick, 1984], [McCormick and Thomas, 1986], [Hart et al.1986]. Considerable amount of studies have been done to maximize computational efficiency and minimize storage requirements.

A fast adaptive composite grid method for time dependent problems (TDFAC), which is an extension of the fast composite grid method for steady state problems, was presented by [Heroux, M.1989].

The method allows local refinement in both time and space by utilizing independent rectangular patches or overlapping groups of similar patches. In his solution procedure, Heroux used smaller time step in the local areas that exhibit rapid transient behavior, while a large time step was used over the entire problem domain.

[Hamzeh, 1990] investigated the fast adaptive composite grid method as a resolution enhancement technique applied to a finite element method. He indicated that such an application was noticeably improved the results of the finite element solution especially on the area of significant local effect.

[McKay, S.1990], employed an adaptive technique along with the fast adaptive composite grid method to improve the computational efficiency and minimize storage requirements. He developed a numerical model in which a composite grid of several patches was adapted to locally refine regions where approximation errors are unacceptable. Similar adaptation technique was applied to a time dependent fast adaptive composite grid method (TDFAC). He concluded that the adaptive mesh refinement technique provided a highly accurate solution from FAC.

Adaptive mesh refinement algorithms for three-dimensional time dependent groundwater flow problems have been developed by [Al-Arawi, M. 1991]. In addition to the use of fast adaptive composite grid method, the model also employed two finite difference methods; the implicit method and alternating direction implicit method represented by the Douglas Rachford method. In his model, Al-Arawi used several levels

of coarse to fine grids in which the latter grids were concentrated in regions of rapid solution variations.

Conservative approximations of divergence-type second-order elliptic boundary value problems on grids with local refinements have been widely used in the petroleum engineering literatures.

In consideration of the problem of local flow modeling around wells (singularity points) in the petroleum reservoir, [Bramble, J. et al.1988] developed a preconditioning method for elliptic problems, which allow for dynamic local grid refinement. The aim of their work is to develop a preconditioner for the discrete problem that results from finite element approximation with composite grids and to reduce overhead requirements.

They concluded that the radial-like refinement technique is the most advantageous from the accuracy as well as the programming point of view indicating that the radial-like refinement technique converges somewhat faster than the regular refinement scheme.

[Ewing, R. et al. 1991], adopted the strategy of a regular local grid refinement by subdividing each grid cell in a certain local subdomain into a number of rectangular cells. Their objective was to develop a general approach for deriving finite difference approximations on a composite grid and to study the problems arising from the local refinement using a piecewise constant interpolation.

A discretization technique based on the finite volume method was employed by the approximation of the balance equation over each grid cell. The fine grid is introduced by subdividing the coarse grid cells of a sub-region into a certain number of fine grid cells in which the centers of the new finer cells was introduced as grid points.

Ewing, R. et al., indicated that in the case of local refinement, the natural approximation near the composite grid interfaces produce a nonsymmetric scheme. Moreover, they resolved that the approximation properties of the finite difference scheme were resulted in low convergence rates due to the employment of the piecewise constant interpolation.

To acquired higher accuracy at the points of singularity, [Abdel-Meguid, 1994] applied a fast adaptive composite grid method for a refinement of a two-dimensional finite element groundwater flow problems. In his thesis, abdel-Meguid defined the fine grids within a coarse domain where the approximated solution on the coarse grids was interpolated to serve as boundaries for the fine grids; the fine grids are then solved.

The residuals between coarse and fine grids were calculated and transferred to the coarse grid via an iterative scheme between the coarse and fine grids until a convergence occurred and the final solution of the composite grid was obtained. He concluded that the approximated results showed a good agreement with the analytical solutions.

[Lee and Greengard, 1997], presented a fast adaptive automatic algorithm code for the solution of two-points boundary value problems, with mesh selection based on a sequence of computed solutions. The objective of the paper is to determine which subintervals require further refinement before any resolution can be achieved. Hence an integral equation reformulation of the original two-point boundary value problem was employed.

Lee and Greengard, showed that the final mesh constructed was fine in regions which require it and coarse in regions which do not even without a priori information about the location of complicated features. In addition, they found that the method

requires about twice as much work as a non-adaptive code in which the adaptive algorithm was required nine level of mesh refinement.

2.2.2 Multi-grid Methods

The Multi-grid method is an iterative technique that approximates the solution to the partial differential equation at various grid resolutions in the solution process; thus the equation is not only solved on one grid, but also on coarser grids with less grid points. Thereby, the coarser grid solutions improve the finer grid solutions and increase the rate of convergence. Usually, a set of grids with decreasing number of grid points from one grid to the next coarser by a factor of two in each dimension is used.

Such a method is applicable to both linear and non-linear partial differential equations to overcome the computational difficulties associated with problems having a large numbers of unknowns.

Historically, the first paper on multi-grid method was published by [Fedorenko, 1964], to investigate the convergence of one iterative process using finite difference formulations. [Bakhvalove1966], took the next forward step, where he analyzed Fedorenko's method. [Brandt, 1973] supported the previous method as being a powerful technique for solving problems associated with elliptic equations. The work of [Brandt, 1977 and 1986] was considered as valuable contributions toward the popularity of the multi-grid method.

An optimal order process for solving finite element equations was first introduced by [Bank and Dupont, 1981], in which they have presented the discretization techniques. Whereby, they assumed nested spaces and proved convergence of the W-cycle using many smoothing steps and assuming regularity in the continuous problem. The proof that

the V-cycle convergence uniformly with respect to the number of levels was first made by [Brass and Hackbusch, 1983]. Although, their proof required a full elliptic regularity, but, investigators considered it as a first step led to the proof that W-cycle could give a uniform reduction under reduced regularity assumptions.

[Brandt, 1979], [Nicolaides, 1979], [Bank and Dupont, 1981], investigated the convergence rate of the multi-grid algorithm applied to the solution of linear differential equations and concluded that the multi-grid algorithm converges at a rate independent of the grid size. Other investigators including [Deconinck and Hirsch, 1982], [Caughey 1985] and [Vanka, 1985], showed empirically that the multi-grid method applied to non-linear differential equation also converges at a rate that still independent of grid size.

[Mckeon et al.1987], developed a multi-grid solution algorithm to study flow in partially saturated porous media with variation in the hydraulic conductivity values. He indicated that the method was capable of solving the highly nonlinear partially saturated flow equation efficiently whereby he was able to use a very fine grid resolution necessitated by certain difficult partially saturated flow problems. In comparison with other iterative methods, line successive over relaxation techniques, the multi-grid method was found to be 25 times faster (in terms of CPU).

Mckeon et al.,concluded that the computational time varies linearly with the number of unknowns in the domain.

[McKinney and Tsai, 1996], applied a multi-grid method in GIS-grid-cell-based modeling to examine the efficiency of GIS to solve steady groundwater flow problems in heterogeneous aquifer where the grid-cell-based portion of the GIS was used to represent the spatial data as grid-cells and to implement the multi-grid solution. They showed that

the application of the multi-grid method was an efficient factor to handle the boundary conditions and to accelerate the iteration process.

[Stals, L. 1997], described the implementation of a multi-grid on parallel computer systems using adaptive finite element methods. The procedure was based upon the use of a node-edge data structure in which the geometrical information stored in a node table while the topological information stored in an edge table. The grids were refined via a node bisection method by [Mitchell, 1988], in which bisection and compatibly divisible triangles were used.

However, Mitchill's method was able to refine only one triangle at a time, an extension has been made so that several triangles could be refine at once. She indicated that the overall efficiency of the system decreased as the grids spread across the processors. On the other hand, the efficiency markedly increased as the number of nods in the processors increased.

[Oosterlee and Washio, 1997], investigated singularly perturbed problems and in which a multi-grid method was evaluated as a solver and as a pre-conditioner using a generalized minimal residual algorithm for solving non-symmetric linear systems (GMRES), developed by [Saad et al.1986].

Oosterlee and Washio, indicated that the multi-grid method was much robust when it was used as a pre-conditioner in which the convergence was not found to be sensitive to parameter changes.

As a result of the tested problems, they found that many of the eigenvalues of a multi-grid iteration matrix were clustered around the origin.

2.2.3 Moving-grid Methods

Moving-grid methods are solution-adaptive methods for time-dependent partial differential equations specifically applied to problems whose solutions involve sharp transition layers and front's propagation in time.

With these methods, the movement and readjustment of grid points are dynamically occurred in order to allow many nods to be concentrated in the critical regions of rapid variation of the solution, as the sharp wetting front or steep gradient, and to move simultaneously with them. To limit truncation errors and to insure high degree of smoothness, the movement of any consecutive points in the grid must be intrinsically coupled.

[Miller, K. and R. N. Miller, 1981], investigated the solutions of the moving finite element method applied mainly to Burgers' equation with a large Reynolds' number and to several other problems whose solutions characterized by shocks' growth. They indicated that the convergence of the iteration required extremely small time step Δt when some of the nodes become closely spaced. However, the nodes were able to concentrate sharply within the shock region and their movements were concomitantly with the shocks.

[Miller, K.1981], introduced an improved moving finite element method and presented the results of later computations related to his previous method, the new method employed a small inter-nodal spring forces to keep the nodes separated. In addition, the new method has been tested on several hyperbolic and parabolic partial differential equations.

He concluded that the new method was able to work Newton's method for implicit stiff ordinary differential equation solver. The method was efficient and powerful in terms of shocks' movements and flexibility of time-stepping Δt. Moreover, the method was able to achieve an excellent fine resolution in the critical regions with only small numbers of nodes.

[Adjerid and Flaherty, 1986], presented a moving finite element method in which the spatial component of the discretization error was equidistributed during the mesh movements. In addition, a computational procedure using quadratic hierarchic finite elements was applied for estimating the disctretization error; the resulted error was employed to develop a procedure for combining the mesh movement and refinement.

Adjerid and Flaherty concluded that the method was particularly well suited to reaction-diffusion problems in which it was able to resolve the rapidly evolving phenomena produced by the nonlinear reaction terms.

[Gottardi, and Venutelli, 1992], introduced a moving finite element model for one-dimensional infiltration problem in unsaturated soil by which a numerical integration of Richards' equation was carried out. The method allowed the use of a coarse mesh in which grid points are moved along the wetting front during computation, thereby allowing a small numbers of nodes and offer improved CPU efficiency for longer domain problem without sacrificing numerical accuracy.

The method has limitations regarding applicability to layered systems and use under time-varying boundary conditions.

[Huang et al.1994], developed a continuos moving mesh equation that so-called moving mesh partial differential equations (MMPDEs), in one-dimension. The method

was based on the equidistribution principle with respect to a monitor function M., which preserved the scaling invariance. A finite difference method in space was employed for the discretization process.

They concluded that for structured grids, the finite difference discretization of MMPDE produces quite satisfactory meshes, which moves smoothly in time and are concentrated in regions where the solution changes rapidly. In addition, the discrete solutions were generally inherited the theoretical properties of the continuous solutions to the moving mesh partial differential equations.

[Gottardi, and Venutelli, 1994], applied a moving finite element method to solve the advection dispersion equation for one-dimensional transport problems. They indicated that the method provided very accurate and efficient solutions for the advection-dispersion problems in both homogeneous and heterogeneous soils and under different initial and boundary conditions. In comparison to the classical fixed-grid finite difference method, the method was able to solve those advection dominated transport problems without suffering numerical smearing or oscillations.

[Shengati and Petzold, 1996], investigated moving mesh methods with upwinding schemes applied to reaction-convection-diffusion partial differential equations in which the method of line was applied to discretize the time-dependent partial differential equations. They indicated that during the discretization, using the method of line, the convection term was the most difficult term to work with.

Shengati and Petzold concluded that the combination technique of the moving mesh with up winding schemes provided stability to the solutions and highly improved the results of the moving mesh methods. In addition, the aforementioned combination

was able to track the wave front more accurately than the commonly used central difference methods.

[Huang and Russell, 1999], developed a moving mesh method for solving partial differential equations in two dimensions as an extension to the previous one-dimensional method developed in 1994. The method has been derived from a gradient flow equation method and explicitly included a mesh speed within a continuous moving mesh equation that was applied to direct the mesh points toward the critical regions of the steep physical solution. They concluded that the method concentrate the mesh points around the areas of large variation in the physical solution and was able to control the mesh skewness.

2.2.4 Domain Decomposition Methods

Domain decomposition methods for solving the partial differential equations received significant attention in scientific and engineering fields, not only as a computing technique suitable to high performance computing systems, but also as a powerful tool to handle highly nonlinear problems that required fine discretizations in time and space. Such technique is increasingly becoming a major area of contemporary research in numerical analysis of partial differential equations.

Historically, the first domain decomposition method was introduced by [Schwarz H. A.1890], as a solution technique to elliptic partial differential equations, the method was based on the use of Dirichlet boundary conditions as interface conditions, such a method is widely known in the literature as Schwarz method, where convergence can be achieved only with overlapping sub-domains. Consequently, the convergence becomes very slow when the overlap is small.

The advantage of using the domain decomposition method is the possibility to divide the problem domain into smaller sub-domains that can be solved independently. These sub-domains are then connected back together to yield the solution of the original problem. The connection between the sub-domains, however, is obtained by means of iterative technique.

In order to speed up the convergence and to handle non-overlapping sub-domain, [Lions, 1989], proposed the use of Fourier or more complex boundary conditions in place of the Dirichlet boundary conditions. But, the main concern was about the best choice of interface conditions that can satisfy the convergence rate and simplify the use and implementations. [Despres, 1990] and [Charion, et al., 1991], [Carlenzoli, et al.1993], [Tan, et al.1994], [Nataf, et al.1995] and [Japhet, 1996], proposed the use of a Fourier analysis for a two-domain decomposition as an important step to quantify the effect of the boundary conditions on the convergence rate before the choice of the interface conditions can be made.

The first step of numerical simulation of unsaturated flow problems is to generate a grid on the domain where the physical variables are defined. Finite-element and finite-difference methods that used fixed grids could waste computing efforts; thus it is better to adapt the grid to the features of the physical variables being simulated.

For example, to simulate a transient unsaturated flow problem with a moving wetting front, more nodes should be concentrated about the moving wetting front than elsewhere.

Therefore, multi-grid, moving-grid and local grid refinement are the main adaptation strategies.

2.2.5 Fourier Series and Finite Element Method

2.2.5.1 Fourier series

Fourier series named in the honor of the French mathematician Joseph Fourier who submitted a paper in 1807 to the Academy of Science of Paris. The paper was a mathematical description of problems involving heat conduction, and was at first rejected for lack of mathematical rigorous. However, it contained ideas that have developed into an important area of mathematics. Since then, the name Fourier became attached to all kinds of analysis using trigonometric series and integrals and their generalizations.

The Fourier method is particularly efficient for the solution of differential equations with constant coefficients since the differential operators are represented in the Fourier basis by diagonal matrices, which can be easily inverted.

[Spiegal, M. 1974], wrote an entire book on the fundamental concepts and applications of Fourier series, Fourier integrals and orthogonal functions. [Haberman, 1987], presented two-chapter in his book with considerable details on Fourier series and orthogonal functions. Likewise, [Pinkus and Zafrany, 1997], wrote a whole book on Fourier series and integral transformations.

The computation and study of Fourier series is known as Harmonic Analysis. The Fourier series are widely used in the study of error estimation, and stability and perturbation analyses of numerical methods.

2.2.5.2 Fourier Finite Element Method

[Burie and Marion, 1999] presented an adaptive multilevel method in space and time for periodic problems in which a Fourier type spatial discretization was considered. Moreover, the utilized schemes were based on multilevel spatial splitting and the use of

different time steps for the various spatial components. Using a Fourier type spatial discretization allows the approximate solution to split naturally into the sum of a low and high frequency components. In the solution procedure, the heat equation and a nonlinear problem were investigated and numerical experiments for both problems using the one-level and the multilevel algorithms were conducted. They concluded that the multilevel method was up to 70% faster than the one-level method.

[Costa and Dettori, 1998] introduced a pseudo-spectral Fourier collocation splitting for two-dimensional partial differential equations in which the splitting was originally motivated by the nonlinear Galerkin Method. In addition, the numerical features as spatial precision and computational cost were analyzed by means of the Burgers and Reaction-Diffusion equations.

A refinement technique was used in which the fine grid was split into four coarse grids each one having one fourth of the points of the fine grid. The high modes component was decomposed into three components. Each component can be represented by any one of the coarse grids, allowing most of the computations to be performed using only one fourth of the number of points of the original fine grids.

Costa and Dettori indicated that the further refinement of the grid yields a super-convergence order when the time step Δt becomes small enough so that the spatial errors start to dominate. Moreover, they concluded that the splitting motivated by the nonlinear Galerkin method, in high and low modes, is a very attractive technique in solving partial differential equations via a pseudo-spectral discretization.

The Fourier-finite-element method FFEM, which combines the approximating Fourier and finite-element method, was applied by [Heinrich, 1996] to the Dirichlet problem of the Poisson equation in axi-symmetric domain with reentrant edge. The edge singularity function was presented by a non-tensor product representation and treated numerically by mesh grading in a two-dimensional meridian of the domain with linear finite elements. The primary aim of the work was the error analysis of the FFEM for elliptic problems of second order with less regularity due to edge singularities.

In his paper, Heinrich concluded that the local mesh grading in the meridian domain is suited for getting more accuracy and a higher approximation order of the FFEM applied to problem with edge singularities.

[Quarteroni, 1987] examined the spectral numerical approximations to nonlinear periodic initial value problems based on both Fourier-Galerkin and Fourier-collocation methods. Moreover, convergence with spectral accuracy and the stability of the discretization in time by explicit one-step technique were presented and analyzed for both methods. His error analysis for the Fourier-Galerkin method was very similar to that of [Arnold et al.1981] for the finite element-Galerkin method.

Quarteroni indicated that a very high accuracy could be achieved if the function is smooth and the number of Fourier harmonic used in the approximation is large. Also, the computation with the pseudo-spectral method was shown to be faster than the one with the Galerkin method. Regarding the time discretization of the problem he showed that using Euler method leads to an unconditionally stable scheme.

A Fourier cosine series was employed by [Chehata, 1988] to approximate the head distribution for one and two-dimensional equations of groundwater flow problems in a confined aquifer.

In his model, Chehata developed a spectral domain decomposition method at which the study region was decomposed into rectangular discrete domains. A truncated Fourier cosine series expansion along with a rectangular domain discretization was used to approximate the head distribution in the discrete domain. The method of weighted residuals was applied on a discrete domain basis in which the Fourier cosine series was used as the weighting function. He concluded that the higher order approximating functions was able to effectively simulate the nonlinear solution behavior.

[Choi, 1989] developed a finite difference and spectral decomposition method for local solution refinement of two-dimensional groundwater flow equation in a confined aquifer. Two different discretization schemes were utilized in his model, a block centered finite difference scheme on a global grid and a spectral domain decomposition scheme on a local grid. Meanwhile, a finite Fourier cosine series expansion of variable order was applied over each local discrete domain in the region of interest.

The basic idea in the solution process is first, to approximate solutions on a global grid and then on a local grid, using the former approximations as boundary values for the local grid. Choi concluded that the results of the solution on a local grid were closely approximated the analytical solutions except at the singularity point. Furthermore, the accuracy enhanced by increasing the harmonics where needed.

CHAPTER 3

GENERAL THERORY OF UNSATURATED FLOW IN POROUS MEDIA

3.1 BASIC EQUATIONS

The governing equation that describes the flow of water in unsaturated porous media can be derived mathematically by combining the continuity equation for an element control volume with Darcy's law (for the flux terms in the continuity equation)

3.1.1 Darcy's Law

Darcy's law that conceived for flow in saturated porous media in three-dimension is written as:

$$q = -K\nabla h \tag{3.1}$$

Where:

q = Darcy's flux [$L^3/T/L^2$]

K = saturated hydraulic conductivity [L/T]

∇h = gradient of the hydraulic head in three-dimensional space [L]

An alternative expression of equation (3.1), Darcy's flux equation may expressed as the gradient of scalar force potential and be written as:

$$q = -\left[\frac{k_w \rho_w g}{\mu_w}\right]\nabla\left(\frac{P_w}{\rho_w g} + z\right) \tag{3.2}$$

The quantity in brackets represents the saturated hydraulic conductivity for isotropic conditions, while the quantity in parentheses is the piezometric head.

Where:

k = intrinsic permeability of the porous medium[L^2]

ρ_w = density [M/L^3]

g = gravity [L/T^2]

μ_w = dynamic viscosity [M/LT^2]

P_w = water pressure [M/LT^2]

z = gravitational head [L]

3.1.2 Mass Balance

The mass conservation law expressed in the equation of continuity for one-dimensional vertical flow with q_z being the flux in z direction stated that the rate of increase of q_z with z must equal the time rate of decrease of volumetric water content θ with time t.

$$\frac{\partial(\rho \vec{q}_z)}{\partial z_j} = -\frac{\partial(\rho\theta)}{\partial t} \tag{3.3}$$

or

$$\frac{\partial(\rho \vec{q}_z)}{\partial z_j} = -\frac{\partial}{\partial t}(\rho\phi S)$$

Where:

q_z = Darcy's flux in the vertical direction z [$L^3/T/L^2$]

S = saturation [dimensionless]

ϕ = porosity of the porous media [dimensionless]

θ = volumetric water content [L^3/L^3] =$\phi.S$

z_j = spatial coordinate in the vertical z-direction

t = elapsed time [T]

The combination of continuity equation and Darcy's flux equation may be written in different alternative formulations depending on their use and the point of interest.

3.1.3 Basic Assumptions

The derivation of the equation governing the flow in unsaturated soils requires some essential assumption upon which we will setup our formulation:

1. Flow occurs only in a one-dimensional vertical system.
2. Physical properties are constant with space and time.
3. Incompressible fluid and non-consolidating porous media.
4. Hysteresis of soil-water characteristics and airflow effects is negligible.
5. No sources or sinks of moisture within the unsaturated flow system
6. The relationships between pressure head and the relative quantity of water at any point in the unsaturated flow system at any time is described by the linear, Brooks-Corey and Van Genuchten relationships.

3.2 GOVERNING EQUATIONS

Combining the continuity equation with Darcy's law can derive the governing equation that describes the flow of water in the unsaturated soil system.

3.2.1 Unsaturated Flow Equation

Darcy's law in equation (3.1) was extended to unsaturated flow in soil, with the provision that the saturated hydraulic conductivity K becomes a function of the pressure head Ψ (Richards, 1931). Therefore, equation (3.1) may be written as:

$$q = -K(\psi)\nabla h \tag{3.4}$$

The gradient of the total hydraulic head ∇H is the sum of the pressure head Ψ and the gravitational head z. Moreover, K (Ψ) may be decomposed as the product of a relative permeability ($0< k_{rw}<1$), which is pressure head dependent and a saturated hydraulic conductivity tensor K_s. Thus, K (Ψ) can be expressed as

$$K(\psi) = K_s k_{rw}(\psi) \tag{3.5}$$

Where k_{rw} (Ψ) is the relative permeability [dimensionless], ranges from 0 to 1.

Thus, equation (3.4) becomes

$$q = -K_s k_{rw}(\psi)\nabla(\psi + z) \tag{3.6}$$

Flow occurs in the vertical direction z through a one-dimensional unsaturated soil system. Thus, equation (3.6) becomes:

$$q_z = -K_s k_{rw}(\psi)\left[\frac{\partial \psi}{\partial z} + \frac{\partial z}{\partial z}\right] \tag{3.7a}$$

or

$$q_z = -K_s k_{rw}(\psi)\left[\frac{\partial \psi}{\partial z} + 1\right] \tag{3.7b}$$

3.2.2 Mass Balance Equation

In this research, the study of the unsaturated soil system is restricted to homogenous incompressible fluid flow. Thus, ρ_w in equation (3.3) is constant and therefore equation (3.3) will be reduced to,

$$\frac{\partial(q_z)}{\partial z_j} = -\frac{\partial(\theta)}{\partial t} \tag{3.8a}$$

And further

$$\frac{\partial(q_z)}{\partial z_j} = -\frac{\partial(\phi S)}{\partial t} \tag{3.8b}$$

3.3 CONSTITUTIVE RELATIONSHIPS

Solution of the partial differential equations governing the flow of water into unsaturated soil system requires two auxiliary equations. These auxiliary equations describe the relationship between pressure head and volumetric water content (saturation) and the relationship between pressure head and relative permeability (hydraulic conductivity). Common constitutive relationships are the linears, Brooks-Corey and Van-Genuchten relationships.

3.3.1 Effective Saturation

The constitutive relationships that describe the relative quantity of water at certain time in the vicinity of a point in the unsaturated flow system may be expressed in terms of effective saturation and effective volumetric water content. Effective saturation, Se is defined as:

$$S_e = \frac{S - S_r}{1 - S_r}, \quad \text{and} \tag{3.9a}$$

$$S_e = \frac{\theta - \theta_r}{\phi - \theta_r} \tag{3.9b}$$

Where:

S = water saturation [dimensionless]

S_e = effective saturation

S_r = residual effective saturation

θ = volumetric water content [L^3/L^3]

θ_r = residual or irreducible volumetric water content

ϕ = porosity = volumetric water content at saturation

additionally, effective volumetric water content can be defined as

$$\theta_e = \frac{\theta - \theta_r}{1 - S_r} \tag{3.10}$$

Where:

θ_e = effective volumetric water content [L^3/L^3]

The volumetric water content and saturation are related to each other by

$$\theta = \phi S \tag{3.11a}$$

or

$$\theta_e = \phi_e S_e \tag{3.11b}$$

3.3.2 Linear Relationship

The Linear relationship is the simplest semi-empirical expression that can be used to describe the dependence of saturation and relative permeability on pressure head as:

$$S_e(\psi) = 1 - \left(\frac{\psi}{\psi_r}\right) \quad \text{for } \psi_r \le \psi \le 0 \tag{3.12}$$

and $S_e = 1$ for $\psi \ge 0$

or

$$S(\psi) = 1 - [1 - S_r]\left(\frac{\psi}{\psi_r}\right) \quad \text{for } \psi_r \le \psi \le 0 \tag{3.13}$$

and $S = 1$ for $\psi \ge 0$

For the linear relationship, the dependence of relative permeability on effective saturation and pressure head may also expressed respectively as

$$k_{rw}(S_e) = S_e \qquad \text{for } S_e \le 1$$

And

$$k_{rw} = 1 \qquad \text{for } S_e = 1 \tag{3.14}$$

And

$$k_{rw}(\psi) = 1 - \left(\frac{\psi}{\psi_r}\right) \qquad \text{for } \psi_r \le \psi \le 0$$

$$k_{rw} = 1 \qquad \text{for } \psi \ge 0 \tag{3.15}$$

3.3.3 Brooks-Corey Relationship

Brooks and Corey (1966) developed empirical formulations to relate effective saturation and volumetric water content of a soil to pressure head. The dependence of effective saturation on pressure head is expressed as

$$S_e(\psi) = \left(\frac{\psi_d}{\psi}\right)^{\lambda} \qquad \text{for } \psi \le \psi_d$$

and

$$S_e = 1 \qquad \text{for } \psi \ge \psi_d \tag{3.16}$$

or

$$S(\psi) = S_r + [1 - S_r]\left(\frac{\psi_d}{\psi}\right)^{\lambda} \qquad \text{for } \psi \le \psi_d$$

$$S = 1 \qquad \text{for } \psi \ge \psi_d \tag{3.17}$$

The dependence of volumetric water content on pressure head is also expressed as:

$$\theta(\psi) = \theta_r + [\phi - \theta_r]\left(\frac{\psi_d}{\psi}\right)^{\lambda} \qquad \text{for } \psi \le \psi_d \tag{3.18}$$

and $\theta = \phi$ for $\psi \ge \psi_d$

Where:

Ψ_d = displacement pressure head [L]

λ = pore-size distribution index [dimensionless]. Other parameters as defined earlier. The dependence of volumetric water content on pressure head is also expressed as:

$$k_{rw} = \left(\frac{\psi_d}{\psi}\right)^{(2+3\lambda)} \qquad \text{for } \psi \le \psi_d \tag{3.19}$$

and $k_{rw} = 1$ for $\psi \ge \psi_d$

3.3.4 Van Genuchten Relationship

Van Genuchten (1980) also derived an empirical relationship. The dependence of effective saturation on pressure head is proposed as:

$$S_e(\psi) = \left[1 + |\alpha\psi|^n\right]^{-m} \qquad \text{for } \psi \le 0 \tag{3.20}$$

and $S_e = 1$ for $\psi \ge 0$

$$S(\psi) = S_r + \frac{1 - Sr}{\left[1 + |\alpha\psi|^n\right]^m} \qquad \text{for } \psi \le 0 \tag{3.21}$$

and

$$S = 1 \qquad \text{for } \psi \ge 0$$

The dependence of volumetric water content on pressure head is expressed as:

$$\theta(\psi) = \theta_r + \frac{\phi - \theta_r}{\left[1 + |\alpha\psi|^n\right]^m} \qquad \text{for } \psi \le 0 \tag{3.22}$$

and

$$\theta = \phi \qquad \text{for } \psi \ge 0$$

The dependence of the relative permeability on pressure head is also expressed as:

$$k_{rw}(\psi) = S_e^{1/2}\left[1 - \left(1 - Se^{1/m}\right)^m\right]^2 \qquad \text{for } \psi \le 0 \tag{3.23}$$

and

$$k_{rw} = 1 \qquad \text{for } \psi \ge 0$$

Wherein: α is a positive parameter equal to the inverse of the air entry pore water pressure, $-\Psi_a$. m, n are empirically best fitted parameters.

The relationships between m and n may be expressed as:

$$m = 1 - \frac{1}{n} \tag{3.24}$$

and

$$n = \frac{1}{1 - m}$$

3.4 DERIVATION OF THE GOVERNING EQUATIONS

3.4.1 Pressure Head-Based Form of Richards Equation

Combining Darcy's flux equation (3.7b) with the continuity equation (3.8a) yield the governing equation for the flow through the unsaturated soil system in a vertical z-direction

$$\frac{\partial}{\partial z_j}\left[K_z k_{rw}(\psi)\left(\frac{\partial \psi}{\partial z_j}+1\right)\right]=\frac{\partial \theta}{\partial t} \tag{3.25}$$

The resulting equation expressed in term of the pressure head Ψ as the unknown variable is referred to as the pressure-head based form of Richards's equation (Richards, 1931).

Where:

K_z = saturated hydraulic conductivity [L/T]

k_{rw} = effective relative permeability

Ψ = pressure head [L]

θ = volumetric water content [L^3/L^3]

z_j = spatial coordinate in the vertical z-direction

To model the unsaturated flow system, equation (3.25) will be mathematically modified and expressed in terms of total hydraulic head and effective saturation rather than its classical form of pressure head and volumetric water content.

Noting that $\theta = \theta_e$, the right hand-side of equation (3.25) can be expressed in terms of the effective volumetric water content.

$$\frac{\partial}{\partial z_j}\left[K_z k_{rw}(\psi)\left(\frac{\partial \psi}{\partial z_j}+1\right)\right]=\frac{\partial \theta_e}{\partial t} \tag{3.26}$$

3.4.2 Derivation of the modified Unsaturated Flow Equation

Generally, the total hydraulic head is, the sum of the pressure head and the gravitational head thus, the simplest relationship between Ψ and h is:

$$h=\psi+z \tag{3.27a}$$

$$\psi=h-z \tag{3.27b}$$

Differentiating Ψ with respect to z in (3.27b) yield.

$$\frac{\partial \psi}{\partial z}=\frac{\partial h}{\partial z}-\frac{\partial z}{\partial z} \tag{3.28a}$$

and further

$$\frac{\partial \psi}{\partial z}=\frac{\partial h}{\partial z}-1 \tag{3.28b}$$

Substituting for the pressure head gradient, equation (3.28b) into equation (3.26) yield:

$$\frac{\partial}{\partial z_j}\left[K_{zj} k_{rw}(\psi)\left(\frac{\partial h}{\partial z_j}-1+1\right)\right]=\frac{\partial \theta_e}{\partial t} \tag{3.29a}$$

Thus

$$\frac{\partial}{\partial z_j}\left[K_{zj} k_{rw}(\psi)\left(\frac{\partial h}{\partial z_j}\right)\right]=\frac{\partial \theta_e}{\partial t} \tag{3.29b}$$

The relationships between the effective volumetric water content θe, residual volumetric water content θr, and saturated volumetric water content θs that is close to porosity ϕ, under no airflow effect, are expressed in terms of effective saturation as:

$$S_e = \frac{\theta - \theta_r}{\phi - \theta_r} = \frac{\theta_e}{\phi_e} \tag{3.30a}$$

$$\theta_e = \phi_e S_e \tag{3.30b}$$

Differentiating equation (3.30b) with respect to time t yield

$$\frac{\partial \theta_e}{\partial t} = \phi_e \frac{\partial S_e}{\partial t} + S_e \frac{\partial \phi_e}{\partial t} \tag{3.31}$$

The change of the effective volumetric water content with respect to time may also expressed in terms of hydraulic head by means of the chain rule to obtain

$$\frac{\partial \theta_e}{\partial t} = \frac{\partial \theta_e}{\partial h} \frac{\partial h}{\partial t} \tag{3.32}$$

Likewise, differentiating equation (3.30b) with respect to hydraulic head yield

$$\frac{\partial \theta_e}{\partial h} = \phi_e \frac{\partial S_e}{\partial h} + S_e \frac{\partial \phi_e}{\partial h} \tag{3.33}$$

Based on our third assumption, the second term on the right hand-side of equations (3.31) and (3.33) will vanish thus, both equations may also be written as:

$$\frac{\partial \theta_e}{\partial t} = \phi_e \frac{\partial S_e}{\partial t} \tag{3.34a}$$

and

$$\frac{\partial \theta_e}{\partial h} = \phi_e \frac{\partial S_e}{\partial h} \tag{3.34b}$$

Substituting equation (3.34b) into equation (3.32), the right hand side form of the modified unsaturated flow equation may expressed as

$$\frac{\partial \theta_e}{\partial t} = \phi_e \frac{\partial S_e}{\partial h} \frac{\partial h}{\partial t} \tag{3.35}$$

By substituting equation (3.35) into the right hand side term of equation (3.29b), the final form of the modeling equation may be written as:

$$\frac{\partial}{\partial z_j}\left[K_z k_{rw}(\psi)\left(\frac{\partial h}{\partial z_j}\right)\right] = \phi_e \frac{\partial S_e}{\partial h} \frac{\partial h}{\partial t} \tag{3.36}$$

or

$$\frac{\partial}{\partial z_j}\left[K_z k_{rw}(\psi)\left(\frac{\partial h}{\partial z_j}\right)\right] = \xi \frac{\partial h}{\partial t} \tag{3.37a}$$

Where:

K_z = saturated hydraulic conductivity in the vertical z-direction [L/T]

$k_{rw}(\Psi)$ = relative permeability[dimensionless]

h = hydraulic head [L]

θ_e = effective volumetric water content [L^3/L^3]

ϕ_e = effective porosity [dimensionless]

S_e = effective saturation [dimensionless]

z_j = spatial coordinate in the vertical z-direction

t = time [T], and

$$\xi = \phi_e \frac{\partial Se}{\partial h} \tag{3.37b}$$

The second term on the right hand-side of equation (3.37b) is the change in the effective saturation with respect to the hydraulic head, which may also be expressed in terms of pressure head Ψ by means of the chain rule to obtain

$$\frac{\partial Se}{\partial h} = \frac{\partial Se}{\partial \psi}\frac{\partial \psi}{\partial h} \tag{3.37c}$$

or

$$\frac{\partial Se}{\partial h} = \frac{\partial Se}{\partial \psi}\left(\frac{\partial (h-z)}{\partial h}\right) \tag{3.37d}$$

and further

$$\frac{\partial Se}{\partial h} = \frac{\partial Se}{\partial \psi}\left(1-\frac{\partial z}{\partial h}\right) \tag{3.38}$$

Substituting equation (3.38) into equation (3.37b) yield

$$\xi = \phi_e \frac{\partial Se}{\partial \psi}\left(1-\frac{1}{\frac{\partial h}{\partial z}}\right) \tag{3.39}$$

The change in the effective saturation with respect to the pressure head may be solved analytically by differentiating the commonly used Brooks-Corey and/or Van-Genuchten relationships for the dependence of effective saturation on pressure head, equations (3.16) and/or (3.20) respectively. Differentiating equation (3.16) with respect to ψ yield

$$\frac{\partial Se}{\partial \psi} = \frac{-\lambda}{\psi} Se \tag{3.40}$$

Likewise, differentiating equation (3.20) yield

$$\frac{\partial Se}{\partial \psi} = \frac{mn|\alpha\psi|^n}{|\psi|} Se^{\frac{m+1}{m}} \tag{3.41}$$

The hydraulic gradient in equation (3.39) may be solved numerically, as will be presented in the following chapter.

3.5 INITIAL AND BOUNDARY CONDITIONS

The solutions of equation (3.36), that describe the flow of water in the vertical direction through the unsaturated soil system, require the specification of an appropriate initial and boundary conditions.

3.5.1 Initial Conditions

The initial conditions to be considered in this study include the specification of the initial hydrostatic head distributions, constant initial head distributions or user defined initial head distributions. The general form of the initial condition is described as:

$$h(z_j,0) = h_o(z_j) \tag{3.42}$$

Wherein h_o is a known function of z_j at every point along the entire flow system and z_j is the spatial coordinate in the vertical z-direction.

3.5.2 Boundary Conditions

Vertical flow of water from ponded ground surface to the water table is the problem to be considered in this study.

3.5.2.1 Ground Surface

The ponded ground surface is the upper boundary of the unsaturated flow problem thus; two types of boundary conditions may be considered at this location, the Dirichlet or the first type boundary condition (specified head) and the Neuman or second type of boundary condition (specified normal flux of water)

1. Specified head

$$h(z_j,t) = h(z_j,t) \qquad \text{on } S_g \tag{3.43}$$

Where S_g is a known function represent the prescribed head boundary.

2. Specified normal flux of water

$$-K_z k_{rw}(\psi)\left(\frac{\partial h}{\partial z_j}\right)n_j = q_n(z_j,t) \qquad \text{on } S_f \tag{3.44}$$

Where n is the unit vector normal to the boundary surface and S_f represent the applied flux boundary.

3.5.2.2 Water table

The water table is the lower boundary of the unsaturated flow problem at which the pressure head $\Psi = 0$. However, small amount of flow will be specified at this boundary thus, the total hydraulic head gradient $\neq 0$.

CHAPTER 4

MODEL DEVELOPMENT

4.1 Introduction

Solution of the partial differential equations governing the flow of water into unsaturated soil system have been widely developed and used in the field of groundwater studies. The most commonly used methods are analytical solutions and numerical simulations. Each solution method, however, has its advantages and disadvantages and work better for certain classes of problems than others.

For instance, analytical solutions of the partial differential equations are greatly affected by soil heterogeneity, spatial variability of the hydraulic properties and complicated boundary conditions. Although, numerical solutions by the classical finite elements and finite difference methods have been widely applied to the solution of the partial differential equations. Such methods were unsuccessful to handle the highly nonlinear flow problems into initially dry soil system under heterogeneous conditions.

Moreover, such methods failed to simulate the solution under the sharp wetting front instability associated with the flow of water into unsaturated soil system. Alternatively, a numerical solution method called the "Fourier Finite element Method" is to be developed in this study to numerically simulate the solution of the partial differential equation governing the flow of water into unsaturated soil system.

The Fourier finite element method, which combines the approximating Fourier series with the finite element method, is applied to the partial differential equation governing the flow of water into unsaturated soil system.

The approximate numerical solution by the new method is able to predict the head and the flux functions that obey the one-dimensional vertical flow of water represented by Equation (3.36).

Furthermore, with the application of the Fourier harmonics at each element, the new method is readily useable for local solution refinement by increasing the number of Fourier harmonics at specific regions where highly non-linear behavior of the solution occurs.

Moreover, the numerical solution of the partial differential equation by the Fourier finite element method provides a great deal of flexibility in selecting the Fourier trigonometric shape functions in one space direction. The true dependent hydraulic head function h (z,t) is replaced by an approximating function where z represents the vector of the space coordinates and t is the time.

Additionally, the numerical solution of the partial differential equation governing the flow of water into unsaturated soil system by means of the Fourier finite element method is applicable to the problem of wetting front instability associated with the flow of water into unsaturated soil system.

This chapter presents the mathematical formulations and the numerical development of a Fourier finite element method for solving the partial differential equations governing the one-dimensional unsaturated flow of water in porous media.

4.2 THE GOVERNING UNSATURATED FLOW EAQUATION, AND INITIAL AND BOUNDARY CONDITIONS

The equation governing the vertical one-dimensional unsaturated flow of water in heterogeneous soil system expressed in terms of total hydraulic head and effective saturation is the modified pressure-head based form of Richards' equation.

$$\frac{\partial}{\partial z_j}\left[K_z\, k_{rw}\left(\frac{\partial h}{\partial z_j}\right)\right] = \phi_e \frac{\partial S_e}{\partial h}\frac{\partial h}{\partial t} \tag{4.1}$$

or

$$\frac{\partial}{\partial z_j}\left[K_{zz}\left(\frac{\partial h}{\partial z_j}\right)\right] = \xi \frac{\partial h}{\partial t} \qquad \text{on } 0 \le z \le L \tag{4.2a}$$

Where:

$$\xi = \phi_e \frac{\partial S_e}{\partial h} \tag{4.2b}$$

K_z = saturated hydraulic conductivity in the vertical z-direction [L/T]

k_{rw} = relative permeability [dimensionless]

$K_{zz} = K_z\, k_{rw}$

h = hydraulic head [L]

ϕ_e = effective porosity [dimensionless]

S_e = effective saturation [dimensionless]

$S_e = S_e(\Psi)$

L = total length of the one-dimensional region, [L]

z_j = spatial coordinate in the vertical z-direction [L], and

t = time [T]

4.2.1 Initial Conditions

$$h(z_j,0) = h_o(z_j) \tag{4.3}$$

Wherein h_o is a known function of z_j

4.2.2 Boundary Conditions

(1) On ground surface: Constant infiltration rate I, applied at the ground surface

$$q_I = q_o \tag{4.4}$$

or

$$q_o = -K_z k_{rw}\left(\frac{\partial h}{\partial z}\right)$$

$$q_o = -K_{zz}\left(\frac{\partial h}{\partial z}\right) \tag{4.5}$$

$$\frac{\partial h}{\partial z} = \frac{q}{K(\psi)}$$

(2) On water table: $\Psi = 0$

$$\frac{\partial h}{\partial z} = \frac{q_{small}}{K_s} \tag{4.6}$$

4.3 FOURIER FINITE ELEMENT APPROXIMATION

The Fourier finite element approximation can be performed through the following main steps:

1. Discretize the vertical one-dimensional problem domain into a collection of nodal points and elements.
2. Select a Fourier cosine shape functions that satisfy continuity conditions within and between elements.
3. Define a truncated Fourier cosine series approximating function for each discrete element j as

$$\hat{h}_j(z,t) = \sum_{m=0}^{M_j} C_{mj}(t)\, N_{mj}(z) \tag{4.7}$$

Where M is the number of Fourier harmonics in the approximating function, $C_{mj}(t)$ is the Fourier cosine series coefficients for discrete element j and $N_{mj}(z)$ is the Fourier cosine shape functions for discrete element j, defined by

$$N_{mj}(z) = \left\{ \cos \frac{m\pi(z - z_i)}{L_j} \right\} \qquad \text{for } z_i \le z \le z_{i+1} \tag{4.8}$$

And

$$N_{mj}(z) = 0 \qquad \text{elsewhere}$$

Wherein L_j is the element length defined as $L_j = z_{i+1} - z_i$ and z_i is the global z-location of node i, [L].

4. Select the number of Fourier harmonics to be applied at each discrete element. Number of Fourier harmonics at each element may be increased or decreased depending on the shape and the location of the element under consideration.

5. Apply the weighted residuals method, substitute for Fourier cosine series approximating function into the partial differential equations and integrate by parts.
6. Formulate the individual element matrices by differentiating inside the integrals and carry out integration in a piecewise manner over each element.
7. Select the appropriate number of quadrature points and perform the numerical integration over each element
8. Formulate the global matrix by grouping the individual element matrices.
9. Introduce the boundary and continuity conditions into the system of equations in the global matrix.
10. Solve the system of equations.

4.4 APPLICATION OF THE WEIGHTED RESIDUALS METHOD

The weighted residuals method is applied over the problem domain [0,L], using the Fourier cosine shape functions N_{mj} (z) defined in equation (4.8) as the weighting functions. Define a linear differential operator L (h)=0. Thus, the linear differential operators corresponding to equation (4.2a) are defined as

$$L(h) = \frac{\partial}{\partial z_j}\left[K_{zz}\left(\frac{\partial h}{\partial z_j}\right)\right] - \xi\frac{\partial h}{\partial t} = 0 \qquad (4.9)$$

Wherein $K_{zz} = K_z\, k_{rw}$, and other variables as defined earlier.

If the trial solution in equation (4.7) is assumed to be a true solution, thus:

$$\hat{h}(z,t) = \text{true solution}$$

Then

$$L(\hat{h}) \equiv 0 \tag{4.10}$$

Else

$$L(\hat{h}) = \text{Residual} = R(z,t)$$

Substitution of a trial solution into the linear differential operator L(h), yields the residual

$$R(z,t) = L(\hat{h}) = L\left(\sum_{m=0}^{Mj} C_{mj}(t)\, N_{mj}(z)\right) \tag{4.11}$$

Also, substitution of the trial solution into the linear differential operator L(h), defined in equation (4.9) yields

$$L(\hat{h}) = \frac{\partial}{\partial z_j}\left[K_{zz}\left(\frac{\partial \hat{h}}{\partial z_j}\right)\right] - \xi\frac{\partial \hat{h}}{\partial t} = R(z,t) \tag{4.12}$$

An approximating integral function over the entire problem domain [0,L] can be formed as

$$\int_0^L R(z,t)\, N_{ij}(z)\, dz = 0 \qquad \begin{matrix} i = 1,2\cdots\cdots, n_d \\ j = 1,2\cdots\cdots, n_e \end{matrix} \tag{4.13}$$

Wherein N_{ij} (z) are weighting functions used to force the residual R (z, t) to be zero on an average sense over the entire problem domain [0,L].

Substituting equation (4.13) for the residual into equation (4.12) yields the following integral form

$$\int_0^L \left\{ \frac{\partial}{\partial z_j}\left[K_{zz} \frac{\partial \hat{h}}{\partial z} \right] - \xi \frac{\partial \hat{h}}{\partial t} \right\} N_{ij}(z)\, dz = 0 \tag{4.14}$$

The approximating function and its derivative in equation (4.7) is defined as

$$\hat{h}(z,t) = \sum_{m=0}^{M_j} C_{mj}(t)\, N_{mj}(z) \tag{4.15a}$$

$$\frac{\partial \hat{h}_j(z,t)}{\partial z_j} = \sum_{m=0}^{Mj} C_{mj}(t) \frac{\partial N_{mj}(z)}{\partial z_j} \tag{4.15b}$$

$$\frac{\partial N_{mj}(z)}{\partial z_j} = \frac{\partial}{\partial z_j}\left(\cos \frac{m\pi(z - z_i)}{L_j} \right) \qquad \text{for } z_i \le z \le z_{i+1} \tag{4.15c}$$

$$N_{mj}(z) = 0 \qquad \text{elsewhere}$$

Substitution of the trial solution, equation (4.15a), into the integral function formed in equation (4.14) yields

$$\int_0^L \left\{ \frac{\partial}{\partial z_j}\left[K_{zz} \frac{\partial}{\partial z_j}\left(\sum_{m=0}^{Mj} C_{mj}(t)\, N_{mj}(z) \right) \right] N_{ij}(z)\, dz \right\} \tag{4.16}$$

$$- \int_0^L \left\{ \xi \frac{\partial}{\partial t}\left[\sum_{m=0}^{Mj} C_{mj}(t)\, N_{mj}(z) \right] N_{ij}(z)\, dz \right\} = 0$$

$$i = 1,2\cdots\cdots, n_d$$
$$j = 1,2\cdots\cdots, n_e$$
$$m = 1,2\cdots\cdots, M_j$$

Integration by parts

$$\int u\,dv = uv\,|\, - \int v\,du \tag{4.17}$$

Let:

$$u = K_{zz} \quad du = \frac{\partial K_{zz}}{\partial z} dz$$

$$v = N_{ij} \quad dv = \frac{\partial N_{ij}}{\partial z} dz \tag{4.18}$$

The first term on the right hand-side of (4.17) represents the boundary conditions at ground surface. Applying the integral function in (4.17) and the differential terms in (4.18) to equation (4.16) and differentiate inside the summations, equation (4.16) after expansion can be written as:

$$\int_0^L \frac{\partial}{\partial z_j}\left[K_{zz}\sum_{m=0}^{Mj} C_{mj}(t)\frac{\partial N_{mj}}{\partial z}\right] N_{ij}(z)\,dz$$

$$= \left[K_{zz}\sum_{m=0}^{Mj} C_{mj}(t)\frac{\partial N_{mj}(z)}{\partial (z)} N_{ij}(z)\right]_0^L \tag{4.19}$$

$$-\int_0^L \left[\xi \sum_{m=0}^{Mj} \frac{\partial C_{mj}(t)}{\partial t} N_{mj}(z) N_{ij}(z)\right]$$

Multiplying (4.19) by –1 and rearranging terms. The entire system of equations may be written as:

$$\int_0^L \frac{\partial}{\partial z_j}\left[K_{zz}\sum_{m=0}^{Mj} C_{mj}(t)\frac{\partial N_{mj}}{\partial z}\right]N_{ij}(z)\,dz$$

$$+\int_0^L \left[\xi\sum_{m=0}^{Mj}\frac{\partial C_{mj}(t)}{\partial t}N_{mj}(z)N_{ij}(z)\right] \qquad (4.20)$$

$$-\left[K_{zz}\sum_{m=0}^{Mj} C_{mj}(t)\frac{\partial N_{mj}(z)}{\partial(z)}N_{ij}(z)\right]_0^L = 0$$

Equation (4.20) represent the system of equations in matrix [A], [B] and [F] respectively and can be written as:

$$A_{ij} = \int_0^L K_{zz}\sum_{m=0}^{Mj} C_{mj}(t)\frac{\partial N_{mj}}{\partial z}\frac{\partial N_{ij}}{\partial z}\,dz \qquad (4.21a)$$

$$B_{ij} = \int_0^L \xi\sum_{m=0}^{Mj}\frac{\partial C_{mj}(t)}{\partial t}N_{mj}(z)\,N_{ij}(z)\,dz \qquad (4.21b)$$

$$F_i = \int_0^L K_{zz}\frac{\partial N_{mj}}{\partial z}N_{ij}(z)\,dz \qquad (4.21c)$$

The coefficient matrix [F] represented by equation (4.21c) will yield a typical value of zero since the derivative of the Fourier cosine shape function will produce a Fourier sine series that will converge to zero values. Therefore, the coefficient matrices represented by equations (4.21a) and (4.21b) will be treated numerically in this model.

4.5 FOURIER SERIES

The Fourier series represents a periodic waveform function as an infinite series of harmonically related sum of sines and cosines functions. For instance, the function f (z) over the interval $-L \le z \le L$ may be defined as: [adapted from Spiegel, 1974]

$$f(z) = \frac{a_0}{2} + \sum_{n=1}^{\infty} a_n \cos\left(\frac{n\pi z}{L}\right) + \sum_{n=1}^{\infty} b_n \sin\left(\frac{n\pi z}{L}\right) \tag{4.22}$$

If the function f (z) in equation (4.23) has the period 2L, the coefficients a_0, a_n and b_n can be calculated by the following formulas

$$a_n = \frac{1}{L}\int_{-L}^{L} f(z)\cos\frac{n\pi z}{L}\,dz \tag{4.23a}$$

$$b_n = \frac{1}{L}\int_{-L}^{L} f(z)\sin\frac{n\pi z}{L}\,dz \qquad n = 0,1,2\cdots\cdots \tag{4.23b}$$

$$\frac{a_0}{2} = \frac{1}{2L}\int_{-L}^{L} f(z)\,dz \tag{4.23c}$$

4.5.1 Fourier Orthogonal Properties

The Orthogonal properties of cosine series stated that the integral functions for one-dimensional region [0,L], is non-zero for m = n. Yet, it is equal to zero whenever m ≠ n. [adapted from Haberman, 1984].

The Orthogonal relationships of sine and cosine functions and their derivative will be applied to the Fourier cosine shape functions over the one-dimensional flow region [0,L] as:

$$\int_0^L \cos\left(\frac{n\pi z}{L}\right) \cos\left(\frac{m\pi z}{L}\right) dz = \begin{cases} 0 & m \neq n \\ \frac{L}{2} & m = n \neq 0 \\ L & m = n = 0 \end{cases} \tag{4.24a}$$

And

$$\int_0^L \sin\left(\frac{n\pi z}{L}\right) \sin\left(\frac{m\pi z}{L}\right) dz = \begin{cases} 0 & m \neq n \\ \frac{L}{2} & m = n \neq 0 \\ 0 & m = n = 0 \end{cases} \tag{4.24b}$$

$$\int_0^L \sin\left(\frac{n\pi z}{L}\right) \cos\left(\frac{m\pi z}{L}\right) dz = \{0 \quad m = n \tag{4.24c}$$

Where m and n are positive integers equal to the number of harmonics in the approximating function.

4.5.2 Fourier Cosine Shape Function

The Fourier cosine shape functions N_{mj} (z) for the one-dimensional discrete elements along the flow region is selected in such way to conform continuity conditions within and between elements and to satisfy the Fourier orthogonal properties as that the Fourier cosine functions is non-zero in the approximating function along the elements of equal number of m and n harmonics while it is equal to zero at all other elements.

4.6 CONTINUITY REQUIREMENTS AND BOUNDARY CONDITIONS

For illustration purposes, three elements and four nodes are selected to represent the one-dimensional sample problem domain of z global coordinate system, which varies over the range $0 \leq z \leq L$. The datum is at the water table where z equal zero with positive z upwards and the ground surface is located at z =L. The length of each element is $L_e = (z_j - z_i)$ and the nodal coordinates of the element are z_i and z_j.

4.6.1 Continuity Requirements

The head functions are required to be continuous at the element interfaces. For the one-dimensional vertical flow of water into unsaturated soil system illustrated in Figure 3-1, consider the continuity conditions exist at element j and j_{-1}, located at the interfaces between e_1, e_2 and e_3 respectively. The continuity of the head functions may be evaluated by equating their approximated values at the element interfaces as:

$$\hat{h}_{j-1}(z_i, t) = \hat{h}_j(z_{i+1}, t) \tag{4.25}$$

Substitution of the Fourier cosine series for both sides in equation (4.25) yields:

$$\hat{h}_{j-1}(z_i,t) = \sum_{m=0}^{M_j} C_{mj-1}(t)\, N_{mj}(z_i) \tag{4.26a}$$

and

$$\hat{h}_{j}(z_{i+1},t) = \sum_{m=0}^{M_j} C_{mj}(t)\, N_{mj}(z_{i+1}) \tag{4.26b}$$

Substitution of the Fourier cosine shape functions and evaluating equation (4.25) at $z = z_i$ and $z = z_{i+1}$ respectively yields

$$\hat{h}_{j-1}(z_i,t) = \sum_{m=0}^{M_j} C_{mj-1}(t)\cos m\pi\left(\frac{z_{i+1}-z_i}{z_{i+1}-z_i}\right) \tag{4.27a}$$

$$\hat{h}_{j}(z_{i+1},t) = \sum_{m=0}^{M_j} C_{mj}(t)\cos m\pi\left(\frac{z_{i+1}-z_{i+1}}{z_{i+1}-z_{i+1}}\right) \tag{4.27b}$$

$$\begin{aligned} i &= 1,2\cdots\cdots\cdots,nd \\ j &= 1,2\cdots\cdots\cdots,ne \\ m &= 0,1,2\cdots\cdots\cdots,M_j \end{aligned}$$

The quantity in brackets in equation (4.27a) is equal to 1 and Cosine (π) = -1. However, the quantity in brackets in equation (4.27b) is equal to zero. Cosine (0) = 1. Thus, equations (4.27a-b) can be written as:

$$\hat{h}_{j-1}(z_i,t) = \sum_{m=0}^{M_j} C_{mj-1}(t)\,(-1)^m \tag{4.28a}$$

$$\hat{h}_{j}(z_{i+1},t) = \sum_{m=0}^{M_j} C_{mj}(t)\,(1)^m \tag{4.28b}$$

Where m is the order of Fourier harmonics used in the Fourier cosine shape functions. Equation (4.28a-b) is used to solve for the Fourier cosine series coefficients within matrix [B] and then the continuity conditions at the element interfaces are evaluated.

4.6.2 Boundary Conditions

Two boundary conditions are to be satisfied in the Solution of the partial differential equations governing the flow of water into unsaturated soil system.

The first boundary condition is at the water table where Z is equal to zero and the specified pressure head $\Psi = 0$. The second boundary condition is at the ground surface where Ψ is also equal to zero and $h = Z$, if ponded water exists at the ground surface. Otherwise the pressure head $\Psi = h - Z$ and $h = \Psi + Z$.

4.6.2.1 Water Table

To approximate the hydraulic head at the water table, equation (4.7) has to be evaluated at first node in first element where $z = z_i$ as:

$$\hat{h}_j(z_i, t) = \sum_{m=0}^{M_j} C_{mj}(t)\, N_{mj}(z) \tag{4.29}$$

Substitution of the Fourier cosine shape functions in equation (4.30), the head may be approximated as

$$\hat{h}_{el}(z_i, t) = \frac{q_{small}}{Ks} \int_0^{le} C_{mj}(t) \left(\cos m\pi \frac{z_i - z_i}{z_{i+1} - z_i} \right) \tag{4.30a}$$

$$\hat{h}_{el}(z_i, t) = \frac{q_{small}}{Ks} \int_0^{le} C_{mj}(t)\, (1)^m \tag{4.30b}$$

Where q and Ks are constant values at the water table boundary and m is the order of Fourier harmonics used in the Fourier cosine shape functions.

4.6.2.2 Ground Surface

To approximate the hydraulic head at the ground surface, equation (4.29) has to be evaluated at last node in last element where $z = z_{i+1}$ as:

$$\hat{h}_j(z_{i+1},t) = \sum_{m=0}^{M_j} C_{mj}(t)\, N_{mj}(z) \tag{4.31}$$

Substitution of the Fourier cosine shape functions in equation (4.31), the head may be approximated as

$$\hat{h}_{e4}(z_{i+1},t) = \int_0^{le} C_{mj}(t)\left(\cos m\pi \frac{z_{i+1}-z_i}{z_{i+1}-z_i}\right) \tag{4.32a}$$

$$\hat{h}_{e4}(z_{i+1},t) = \int_0^{le} C_{mj}(t)\,(-1)^m \tag{4.32b}$$

The approximated head evaluated at the ground surface boundary depends on the condition at that boundary during the evaluation time.

If constant rate of infiltration applied at the ground surface then the infiltration rate q_I should be taken into consideration when the Fourier cosine series function applied to approximate the head at that location. Under such condition, equation (4.32b) may be expressed as:

$$\hat{h}_{e4}(z_{i+1},t)=\frac{q_1}{Ks}\int_0^{le}\frac{1}{k_{rw}}C_{mj}(t)(-1)^m \tag{4.33}$$

4.7 INITIAL HYDRAULIC HEAD DISTRIBUTION

The process of solving for the initial hydraulic head distribution along the one-dimensional vertical flow system requires the specification of a minute water flux at water table. Knowing that the initial hydraulic head h_o, and the initial pressure head Ψ_o, at the first node located at the water table are typically equal zero. However, the effective saturation S_e, and relative permeability k_{rw} are unity at that location.

The auxiliary equations describe the relationship between pressure head, effective saturation and relative permeability was applied to solve for the pressure head and initial hydraulic head distribution anywhere along the flow region [0,L].

4.7.1 Flow Specification

Analytical solution by means of steady state condition was applied to specify the flux and to calculate the initial hydraulic head distribution along the one-dimensional vertical flow system. Average harmonic for the saturated hydraulic conductivity was used to calculate the steady state flux as

$$q_{ss}=\bar{K}=\frac{\sum_{j=1}^{ne}L_j}{\sum_{j=1}^{ne}\frac{L_j}{K_j}} \tag{4.34}$$

Where:

q_{ss} = steady state flux, [L/T]

K_j = saturated hydraulic conductivity at element j, [L/T]

L_j = length of discrete element j, [L]

4.7.2 Initial Head Distribution

Darcy's law extended to unsaturated flow was applied to calculate the initial hydraulic head as

$$q = K_S k_{rw}\ (\psi)\frac{\partial h}{\partial z} \tag{4.35}$$

Rearrange and integrate along the unsaturated flow system. Thus, equation (4.35) becomes:

$$\int_0^z \left(\frac{q_{ss}}{K_S}\right)\left(\frac{1}{k_{rw}}\right)dz = \int_0^h dh \tag{4.36}$$

In equation (4.36), the flux and saturated hydraulic conductivity are both constant, while the relative permeability is variable at each nodal point along the unsaturated flow region.

Thus, equation (4.36) may be expressed as

$$h(z) = \left(\frac{q_{ss}}{K_S}\right)\int_0^z \left(\frac{1}{krw}\right)dz \tag{4.37}$$

Where:

h(z) = initial hydraulic head at any location z [L], and

Other variables as defined earlier.

4.7.3 Initial Fourier Series Coefficients

To determine the unknown initial Fourier series coefficients for the initial head distribution, an individual elemental coefficient matrix [aij], and a right hand-side matrix [bi], are formulated and solved for each element along the unsaturated flow region.

4.7.4 odd and even Fourier Series Coefficients

The numerical sum of the odd inverse squired Fourier series coefficients for M number of Fourier harmonics in the approximating function C_{mj} (0) are calculated at each individual elemental coefficient matrix $[a_{ij}]$. However, the sums of the even Fourier series coefficients are set to zero and the arbitrary coefficients a_o are set to be equal to average sum of the initial hydraulic head at water table and the interfaces of each element.

The numerical sum of the odd and even inverse squired Fourier series coefficients and the arbitrary coefficients a_o may be expressed as

$$\sum(a_{odd})_{ei} = \left[1+\frac{1}{(3)^2}+\frac{1}{(5)^2}+\frac{1}{(7)^2}+\frac{1}{(9)^2}+.....+\frac{1}{(M)^2_{(odd)}}\right] \tag{4.38a}$$

$$\sum(a_{even})_{ei} = \left[1+\frac{1}{(2)^2}+\frac{1}{(4)^2}+\frac{1}{(6)^2}+\frac{1}{(8)^2}+.....+\frac{1}{(M)^2_{(even)}}\right] \tag{4.38b}$$

Thus, the arbitrary coefficients a_o and odd and even Fourier series coefficients and the corresponding boundary conditions may be expressed as

$$(a_o)_{ei}+\sum(a_{odd})_{ei}+\sum(a_{even})_{ei} = h_o \tag{4.39a}$$

$$(a_o)_{ei}-\sum(a_{odd})_{ei}+\sum(a_{even})_{ei} = h_{L_j} \tag{4.39b}$$

Equations (4.39a) and (4.39b) represent the boundary conditions at water table and ground surface respectively. Wherein ho is the initial head at the first node located at

water table and h_{Lj} is the initial head at the last node of each element which may be replaced by the continuous head function at the element interfaces or by the initial head at the ground surface, depending on the number of elements to be used in the model.

The system of the matrix equations and the corresponding right hand-side may be simplified and expressed as

$$(a_o)_{ei} = \frac{(h_o)_{ei} + (h_{Lj})_{ei}}{2} \tag{4.40a}$$

$$\sum (a_{odd})_{ei} = \frac{(h_o)_{ei} - (h_{Lj})_{ei}}{2} \tag{4.40b}$$

$$\sum (a_{even})_{ei} = 0. \tag{4.40c}$$

The remaining systems of equations at each individual elemental coefficient matrix $[a_{ij}]$ are computed by the odd and even truncated Fourier cosine series. The initial hydraulic heads are used to formulate the right hand side of matrix [bi].

4.7.5 The Initial Odd, Even and Total Hydraulic Heads

Solving for the unknown initial Fourier series coefficients, the initial odd, even and total hydraulic heads may be calculated as

$$\begin{aligned} h_{odd}(z) &= (a_o) + a_1 \text{Cos}\frac{m\pi z}{Le} + a_3 \text{Cos}\frac{m\pi z}{Le} + \ldots\ldots\ldots + a_{n(odd)} \text{Cos}\frac{m\pi z}{Le} \\ h_{even}(z) &= a_2 \text{Cos}\frac{m\pi z}{Le} + a_4 \text{Cos}\frac{m\pi z}{Le} + \ldots\ldots\ldots + a_{n(even)} \text{Cos}\frac{m\pi z}{Le} \\ h_{total}(z) &= (a_o) + a_1 \text{Cos}\frac{m\pi z}{Le} + a_2 \text{Cos}\frac{m\pi z}{Le} + a_3 \text{Cos}\frac{m\pi z}{Le} \ldots\ldots\ldots + a_n \text{Cos}\frac{m\pi z}{Le} \end{aligned} \tag{4.41}$$

Wherein m in equation (4.41) is the corresponding odd or even number of Fourier harmonics in the approximating function, and other variables as defined earlier.

4.8 NUMERICAL INTEGRATION

4.8.1 Numerical Integration by Gauss-Legendre Quadrature Method

4.8.1.1 Integration of Matrix [A]

The numerical integration by Gauss-Legendre Quadrature method is applied to the global matrix coefficient A_{ij} given by equation (4.21a) and repeated below

$$A_{ij} = \int_0^L K_{zz} \sum_{m=0}^{Mj} C_{mj}(t) \frac{\partial N_{mj}}{\partial z} \frac{\partial N_{ij}}{\partial z} dz \tag{4.42}$$

The numerical integration of equation (4.42) is performed over the problem domain [0,L] in a piecewise manner on an elemental basis where C_{mj} are moved out from under the summation sign, as they do not depend on z. Equation (4.42) written on an elemental form as:

$$\begin{aligned} \int_0^L K_{zz} \frac{\partial N_{mj}}{\partial z} \frac{\partial N_{ij}}{\partial z} dz = & \int_0^{L_{e1}} K_{zz} \left(\frac{\partial N_{mj}(z)}{\partial z} \frac{\partial N_{ij}(z)}{\partial z} \right) dz \\ & + \int_{L_{e1}}^{L_{e2}} K_{zz} \left(\frac{\partial N_{mj}(z)}{\partial z} \frac{\partial N_{ij}(z)}{\partial z} \right) dz \\ & + \vdots \\ & + \int_{L_{e(n-1)}}^{L_{en}} K_{zz} \left(\frac{\partial N_{mj}(z)}{\partial z} \frac{\partial N_{ij}(z)}{\partial z} \right) dz \end{aligned} \tag{4.43}$$

Application of Fourier cosine shape functions over each element in equation (4.43) and differentiation inside the integrals yields

$$K_s \int_0^{L_{e1}} k_{rw}(z_j)\left[-\sin m\pi\left(\frac{z-z_o}{L_{e1}}\right)\frac{\partial}{\partial z}\frac{\pi(z-z_o)}{L_{e1}}\left(-\sin n\pi\left(\frac{z-z_o}{L_{e1}}\right)\frac{\partial}{\partial z}\frac{\pi(z-z_o)}{L_{e1}}\right)\right]dz$$

$$+K_s \int_{L_{e1}}^{L_{e2}} k_{rw}(z_j)\left[-\sin m\pi\left(\frac{z-z_o}{L_{e2}}\right)\frac{\partial}{\partial z}\frac{\pi(z-z_o)}{L_{e2}}\left(-\sin n\pi\left(\frac{z-z_o}{L_{e2}}\right)\frac{\partial}{\partial z}\frac{\pi(z-z_o)}{L_{e2}}\right)\right]dz$$

$$+$$

$$\vdots$$

$$+K_s \int_{L_{e(n-1)}}^{L_{en}} k_{rw}(z_j)\left[-\sin m\pi\left(\frac{z-z_o}{L_{en}}\right)\frac{\partial}{\partial z}\frac{\pi(z-z_o)}{L_{en}}\left(-\sin n\pi\left(\frac{z-z_o}{L_{en}}\right)\frac{\partial}{\partial z}\frac{\pi(z-z_o)}{L_{en}}\right)\right]dz \tag{4.44}$$

Where m and n are the number of Fourier harmonics in the approximating functions and ($z - z_o = zq$) is the global z-location of a node.

The numerical integration by Gauss-Legendre Quadrature Method is applied to approximate the integral functions in equation (4.44). The general elemental integral form of equation (4.44) may be written as:

$$K_s \int_0^{L_e} k_{rw}(z_q)\left(\frac{\pi}{L_e}\right)^2\left[\sin\left(\frac{m\pi z_q}{L_e}\right)\sin\left(\frac{n\pi z_q}{L_e}\right)\right]dz \tag{4.45}$$

The numerical approximation to the integral function in equation (4.46) is given by: [adapted from Applied numerical method, Carnahan et. al, 1975]

$$F(z_q) = C[wq_1 f(zq_1) + wq_2 f(zq_2) + \cdots\cdots + wq_n f(zq_n)] \qquad (4.46)$$

Wherein F (z_q) is the numerical approximation to the integral function given by equation (4.47), the quantity C is a constant determined by the limits of the integral, w_{qi} are the weight factors found in the course of the derivation and z_{qi} are the n unequally spaced points determined by the type and degree of the applied orthogonal polynomial.

The general Gauss-Legendre integral formula is of the form given on the interval $-1 \leq x \leq 1$.

$$\int_{-1}^{1} f(x)\,dx \approx \sum_{i=1}^{n} w_i\, f(x_i) \qquad (4.47)$$

Where xi are the nodes of the quadrature integration function and wi are weight factors.

4.8.1.2 Integral Transformation

To numerically evaluate the integral function in equation (4.46), the interval of the integration in equation (4.47) has to be changed from [-1,1] to [0,L_e] by a suitable transformation of variable. Let:

$$x_q = a \cdot z_q + b \qquad (4.48)$$

At the lower and upper interval limits xq are -1 and 1 respectively. Substitution for xq into equation (4.48) yield

$$-1 = a.(0) + b \quad \text{where} \quad b = -1$$
$$1 = a.(L_e) - 1 \quad \text{and} \quad a = \left(\frac{2}{L_e}\right) \tag{4.49}$$

Substitution for a and b into equation (4.49) yields:

$$x_q = \left(\frac{2}{L_e}\right).z_q + b \tag{4.50a}$$

$$z_q = \left(\frac{L_e}{2}\right).\left(x_q + 1\right) \tag{4.50b}$$

Rearranging the terms in equation (4.50b), the transformation formula can be expressed as:

$$z_q = \left(\frac{L_e}{2}\right) + \left(\frac{L_e}{2}\right).x_q \tag{4.51}$$

Wherein $\pm x_q$ are the n zeros of the nth degree Gauss-Legendre polynomials, w_q are the weight factors and l_e is the element length ($L_e = z_j - z_i$). Values of $\pm x_q$ and the corresponding weight factors w_q are tabulated for values of n ranging from n = 2 to n = 512.

The numerical approximation to the integral function in equation (4.45) is then given by

$$F(z_q) = K_s\, k_{rw}(z_q) \cdot \left(\frac{Le}{2}\right) \cdot \left(\frac{\pi}{Le}\right)^2 \sum_{q=1}^{n} w_q \cdot f(z_q) \qquad (4.52)$$

Wherein the approximating function f (zq) with nth-degree polynomials is defined as:

$$f(z_q) = \left[\sin\left(\frac{m\pi z_q}{Le}\right) \cdot \sin\left(\frac{n\pi z_q}{Le}\right)\right] \qquad (4.53)$$

The Fourier finite approximation method was applied on an element basis to approximate the pressure head Ψ at each node corresponding to $\pm x_q$ and w_q for a given number of harmonics m and n. Then, the constitutive relationships are applied to calculate the corresponding values of the volumetric water content θ, effective saturations S_e and relative permeability k_{rw}.

The Saturated hydraulic conductivity K_s in equation (4.52) is an input variable for each soil type.

The numerical integration of equation (4.53) is carried out with the nth-degree polynomials equal to 256 Gauss-Legendre quadrature points and the integral sum for each element matrices are formed to build up the global matrix [A].

4.8.1.3 Integration of Matrix [B]

The numerical integration by Gauss-Legendre Quadrature method is also applied to the matrix coefficient Bij to evaluate numerically the integral function for matrix [B] defined as:

$$B_{ij} = \int_0^L \xi \sum_{m=0}^{Mj} \frac{\partial C_{mj}(t)}{\partial t} N_{mj}(z)\, N_{ij}(z)\, dz \tag{4.54}$$

The integration of equation (4.54) is performed over the problem domain [0,L] in a piecewise manner on an elemental basis. Where C_{mj} in equation (4.16) and its first derivative with respect to t in equation (4.54) are kept under the summation sign, as they are depend on t. Equation (4.54) written on an elemental form as:

$$\int_0^L \xi \frac{\partial C_{mj}(t)}{\partial t}\left[N_{mj}(z)\, N_{ij}(z)\right] dz = \int_0^{L_{e1}} \xi_1 \frac{\partial C_{mj}(t)}{\partial t}\left[N_{mj}(z)\, N_{ij}(z)\right] dz$$

$$+ \int_{L_{e1}}^{L_{e2}} \xi_2 \frac{\partial C_{mj}(t)}{\partial t}\left[N_{mj}(z)\, N_{ij}(z)\right] dz \tag{4.55}$$

$$+ \vdots$$

$$+ \int_{L_{en-1}}^{L_{en}} \xi_n \frac{\partial C_{mj}(t)}{\partial t}\left[N_{mj}(z)\, N_{ij}(z)\right] dz$$

Application of Fourier cosine shape functions over each element in equation (4.54) yields

$$\int_0^L \xi \frac{\partial C_{mj}(t)}{\partial t}\left[N_{mj}(z)\, N_{ij}(z)\right]dz = \int_0^{L_{e1}} \xi_1 \frac{\partial C_{mj}(t)}{\partial t}\left[\cos m\pi\left(\frac{z-z_o}{L_{e1}}\right)\cos n\pi\left(\frac{z-z_o}{L_{e1}}\right)\right]dz$$

$$+\int_{L_{e1}}^{L_{e2}} \xi_2 \frac{\partial C_{mj}(t)}{\partial t}\left[\cos m\pi\left(\frac{z-z_o}{L_{e2}}\right)\cos n\pi\left(\frac{z-z_o}{L_{e2}}\right)\right]dz$$

$$+ \vdots$$

$$+\int_{L_{e n-1}}^{L_{en}} \xi_n \frac{\partial C_{mj}(t)}{\partial t}\left[\cos m\pi\left(\frac{z-z_o}{L_{en}}\right)\cos n\pi\left(\frac{z-z_o}{L_{en}}\right)\right]dz \qquad (4.56)$$

Wherein z – zo = zq and ξ as defined by equation (3.39). For simplicity the general elemental integral function of equation (4.56) can be expressed as:

$$\int_0^{L_e} \xi \frac{\partial C_{mj}(t)}{\partial t}\cdot\left[\cos\left(\frac{m\pi z_q}{L_e}\right)\cos\left(\frac{n\pi z_q}{L_e}\right)\right]dz \qquad (4.57)$$

The numerical approximation to the integral function in equation (4.57) is then given by:

$$F(z_q) = \xi . \left(\frac{Le}{2}\right) . \left(\frac{\pi}{Le}\right)^2 \sum_{q=1}^{n} w_q . f(z_q) \tag{4.58}$$

Where:

$$\xi = \phi e \frac{\partial Se}{\partial \psi} \left(1 - \frac{1}{\frac{\partial h}{\partial z}}\right) \tag{4.59}$$

Wherein the approximating function f (zq) with nth-degree polynomials is defined as:

$$f(z_q) = \left[\cos\left(\frac{m\pi z_q}{Le}\right) . \cos\left(\frac{n\pi z_q}{Le}\right)\right] \tag{4.60}$$

Before proceeding with the construction of the elemental matrix [b], it is still necessary to solve for the partial differential functions presented in equation (4.59).

Given the beginning and ending node location on the global z-coordinate on an element basis. The numerical solution by means of Gauss-Legendre Quadrature method is applied. The transformation formula developed in equation (4.51) and repeated below

$$z_q = \left(\frac{Le}{2}\right) + \left(\frac{Le}{2}\right) . x_q \tag{4.61}$$

Equation (4.61) used to approximate the upper and lower node locations for each quadrature point in the global z-coordinate along the unsaturated flow system.

The Fourier finite element method and the truncated Fourier cosine series are applied on an element basis to approximate the hydraulic head for each quadrature point at the upper and lower node locations. Given the locations of the upper and lower nodal points for each element along the global z-coordinate, the partial differential function for the hydraulic gradient in equation (4.59) is solved numerically. Nevertheless, The partial differential function for the change in the effective saturation with respect to the pressure head was solved analytically as presented in the previous chapter.

The final form of equation (4.59) may be expressed as

$$\xi = \phi e \frac{\partial Se}{\partial \psi}\left(1 - \frac{1}{\frac{h_{(z+\Delta z)} - h_z}{z_{(i+1)} - z_i}}\right) \tag{4.62}$$

Substituting equation (4.62), into equation (4.58) for ξ. The numerical integration presented in equation (4.58) is carried out with the nth-degree polynomials equal to 256 Gauss-Legendre quadrature points and the integral sum for each element matrices are formed to build up the global matrix [B].

4.9 TIME APPROXIMATION AND SOLUTION PROCEDURE

4.9.1 Time Approximation

Given the initial Fourier cosine series coefficients Cij(0) and the necessary initial, boundary and continuity conditions. Equations (4.44) and (4.55) for the global matrices [A] and [B] are solved for the new Fourier cosine series coefficients Cij(t).

The use of implicit finite difference scheme in time to solve the system of equations in matrix [A] and [B] yields

$$\left[A_{ij}\right]+\left[\frac{B_{ij}}{\Delta t}\right]C_{ij}^{t+\Delta t}=\left[\frac{B_{ij}}{\Delta t}\right]C_{ij}^{t} \tag{4.63}$$

or

$$\left[\left[A_{ij}\right]+\frac{\left[B_{ij}\right]}{\Delta t}\right]\left[\frac{B_{ij}}{\Delta t}\right]C_{ij}^{t}=C_{ij}^{t+\Delta t}$$

Wherein the superscript t indicates the previous time level and t+Δt denoting the new time level at which the updated Fourier cosine series coefficients C_{ij} are evaluated.

4.9.2 Solution Procedure

Given the updated Fourier cosine series coefficients C_{ij}. The solution for the approximated hydraulic head at any time level may be obtained by substituting for the Fourier coefficients into the Fourier cosine series approximating function given by equation (4.7) and repeated below.

$$\hat{h}_j(z,t)=\sum_{m=0}^{M_j}C_{mj}(t)\,N_{mj}(z) \tag{4.64}$$

The process of solving for the flux in the one-dimensional unsaturated soil system at any given time level may be obtained by substituting equation (4.64) for the approximated hydraulic head into the modified one-dimensional unsaturated flow equation given by (4.1) and repeated below.

$$\frac{\partial}{\partial z_j}\left[K_z k_{rw}(\psi)\left(\frac{\partial \hat{h}}{\partial z_j}\right)\right] = \phi_e \frac{\partial S_e}{\partial \hat{h}} \frac{\partial \hat{h}}{\partial t} \tag{4.65}$$

or

$$\frac{\partial}{\partial z_j}\left[K_{zj} k_{rw}(\psi)\left(\frac{\partial \hat{h}}{\partial z_j}\right)\right] = \xi \frac{\partial \hat{h}}{\partial t} \tag{4.66}$$

Where:

$$\xi = \phi e \frac{\partial Se}{\partial \psi}\left(1 - \frac{1}{\frac{\hat{h}_{(z+\Delta z)} - \hat{h}_z}{z_{(i+1)} - z_i}}\right) \tag{4.67}$$

In which $\hat{h}(z, t)$ is the hydraulic head approximated by the Fourier cosine series function and other variables as defined earlier. Repeating the process for the next time step to update the Fourier cosine series coefficients C_{ij} (t), equations (4.64), (4.65) and (4.67) may be used to evaluate the hydraulic head $\hat{h}(z, t)$ and the flux function q (z,t) at any time level t anywhere along the one-dimensional unsaturated flow region [0,L].

CHAPTER 5

FFUSAT, A COMPUTER PROGRAM FOR FLOW OF WATER INTO UNSATURATED POROUS MEDIA

5.1 COMPUTER PROGRAM STRUCTURE

5.1.1 Introduction

FFUSAT is a computer program structure uses the combination of the approximating Fourier cosine series and the finite element method to describe mathematically the vertical flow of water into unsaturated soil system.

In the FFUSAT model, the problem domain of a one-dimensional vertical system of length L extends from the water table to ground surface is discretized into a collection of nodal points and elements. Each element has a length of l_e consisted of two or more nodal points. The total number of nodal points is equal to the total number of elements plus one.

FFUSAT is written in FORTRAN 77 code and its complete listing is given in Appendix A. The main program determines subsequent time steps, controls the time incrementing feature and the frequency of input/output.

The input subroutine, however, directs calls to seven major subroutines, which, in turn, make calls to fourteen minor subroutines. Figure 5-1 presents the basic flow-chart for the main execution steps performed in FFUSAT model.

FFUSAT model consists of the main program and twenty-one subroutines; in order, they are: INPUT, ETYPMH, FICHYD, HOUT, BANAR, AUXEQS, GUASS, PIVOTCHANGE, QUAD, CUBIC, COEFFMAT, QUADPTS, HQUAD, HGRADZ, QUADINTG, MATXAB, HICHYD, QFLOWZ, QSTORAGE, WATBAL, HIHOUT, OUPUT.

5.1.2 Program Flow Chart

The flow chart presented in Figures (5-1) summarizes the general functions and describes the computational sequences of each subroutine. Each segment of the flow chart is representative of a subroutine in FFUSAT model. The chart also shows the order in which each subroutine is called by the main program or by other subroutines to perform a particular computational task.

5.1.3 Grid Parameters

Linear shape elements were used to design the problem domain of one-dimensional vertical flow of water into unsaturated soil system. The problem domain of total length L extended from water table to ground surface was replaced with a collection of nodal points and elements, each element consisted of two or more nodal points joined by straight-line. A cubic shape function employed near the ponded ground surface boundary where the pressure head is zero and the hydraulic head equal to L.

In the FFUSAT model, each element and nodal point assigned a unique number in escalating order from water table to ground surface. The number of Fourier series harmonics can be increased or decreased at each element depending on the element shape, location and the required degree of refinement.

5.1.4 Soil Properties

The physical soil properties for each soil layer were specified through the element identification array, which assigns the appropriate soil properties to each layer of soil.

Soil properties such as saturated hydraulic conductivity; residual volumetric water content and porosity were assumed to be constant for elements within the homogenous soil type. However, such soil properties were varies for elements within the heterogeneous soil layers and at element interfaces.

5.1.5 Initial and Boundary Conditions

The initial conditions describe the status of one-dimensional flow of water into unsaturated soil system at time equal zero. The initial head distribution was determined for steady state condition by specifying a minute water flux at water table.

Two boundary conditions were considered, specified head at water table, specified head and/or specified flux at the ground surface depending on the unsaturated flow condition at the time of evaluation.

5.1.6 Time Discretization

The time discretization is designed to maintain a high level of accuracy and minimum computational effort. The main program determines subsequent time steps and controls the time incrementing feature. Fine discretizations on time may be used to obtain accurate solution and to insure convergence. Given the necessary initial, boundary and continuity conditions of the unsaturated flow system and knowing the initial Fourier cosine series coefficients at time t, the condition of the system at the new time level $t + \Delta t$ may be evaluated by successive numerical iterations until the convergence criterion is satisfied.

The process of evaluating the hydraulic head and the flux functions may be obtained by updating the Fourier cosine series coefficients at any desire time level.

5.1.7 Program Units

The FFUSAT computer program of the Fourier finite element method for the flow of water into unsaturated soil system consists of the following units:

5.1.7.1 Main Program

The purpose of the main program is to control the calling order and the execution of subroutines according to the computational sequences. The main program calls three major subroutines to perform: reading of input data, formulation of the global matrices and updating the Fourier cosine series. Main program also determines subsequent time steps, controls the time incrementing feature and the frequency of input/output. The major subroutines, in turn, make calls to eight minor and function subroutines.

5.1.7.2 Subroutine INPUT

This subroutine is called by the main program to read all the input data and grid parameters including number of soil types, soil properties, the length of elements and calculates the number of elements, nodal points depending on the total depth to the water table. The input subroutine controls the element identification array, which assigns a unique number for each element along the unsaturated soil system depending on the element's soil types and properties. Subroutine input uses an identifying flag to control the application of the constitutive relationships.

Called from: main program

Subroutine called: QUADPTS, HICHYD, FICHYD, HGRADZ, HOUT, QFLOWZ and QSTORAGE.

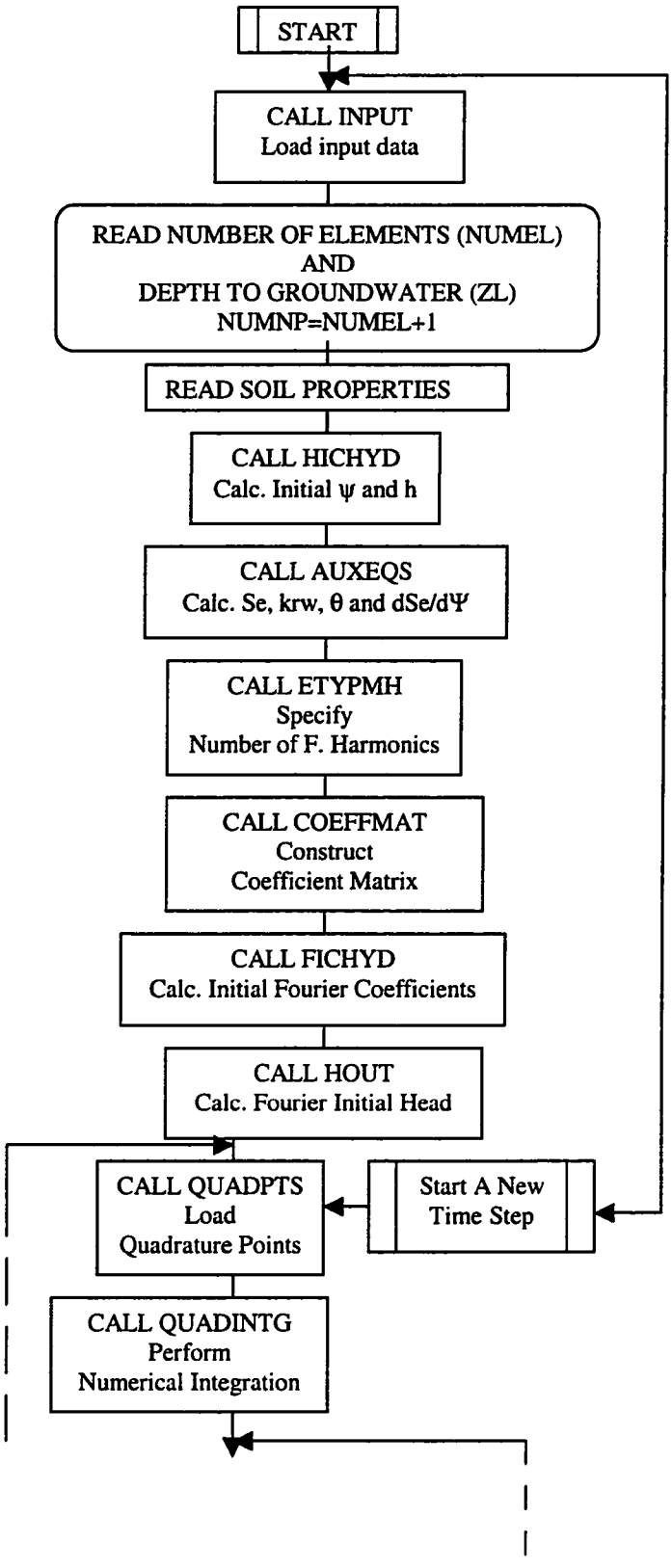
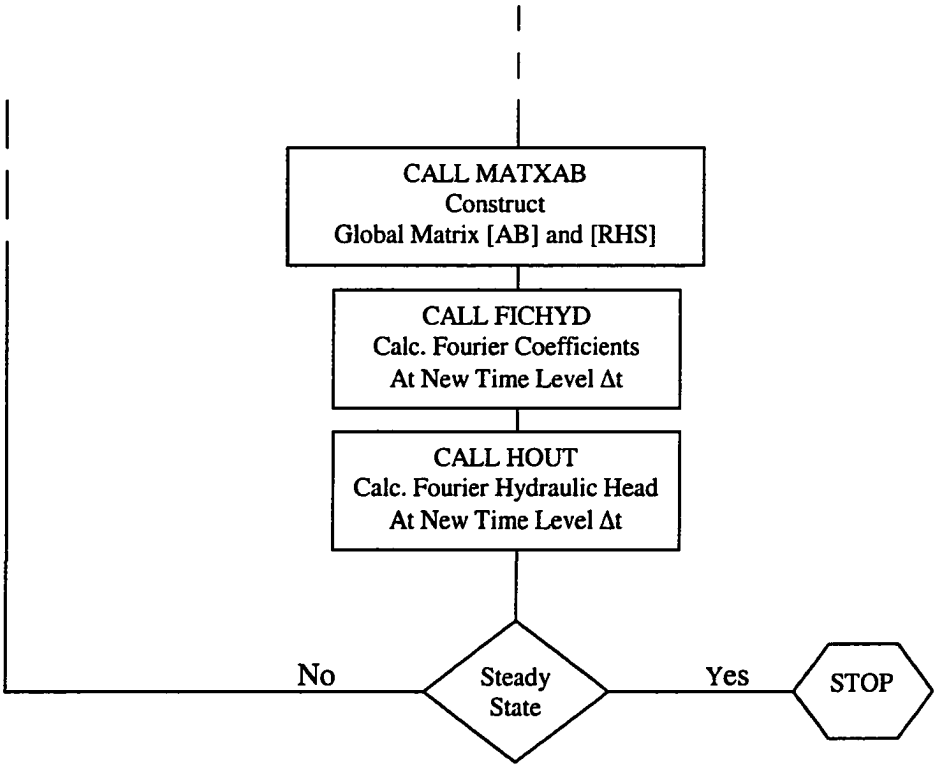


Figure 5.1 Simplified flow-chart for FFUSAT model



(Figure 5.1 continues)

5.1.7.3 Subroutine ETYPMH

The purpose of this subroutine is to determine the element type and to specify the required number of Fourier harmonics for the initial hydraulic head distributions.

Called from: HICHYD

Subroutine called: None

5.1.7.4 Subroutine FICHYD

This subroutine is called by subroutine INPUT at the beginning of each simulation to calculate the boundary conditions, the odd Fourier series coefficients and to construct the right hand side array. FICHYD, in turns, make call to subroutine COEFFMAT to calculate and to construct the initial elemental coefficient matrices along the unsaturated flow region [0,L]. Subroutine FICHYD also calls subroutine GAUSS to solve for the initial Fourier series coefficients. Subroutine FICHYD also updates and saves the Fourier cosine series coefficients at each new time level. This subroutine in turn, makes call to subroutine HOUT to calculate and print the approximated initial and updated hydraulic head distributions.

Called from: INPUT

Subroutine called: COEFFMAT, GUASS, and HOUT

5.1.7.5 Subroutine HOUT

Subroutine HOUT uses the computed Fourier cosine series coefficients to calculate the approximated hydraulic head distributions for each element along the unsaturated flow region [0,L].

Called from: INPUT, FICHYD

Subroutine called: None

5.1.7.6 Subroutine BANAR

This subroutine displays the program banar.

Called from: INPUT

Subroutine called: None

5.1.7.7 Subroutine AUXEQS

Given the approximated hydraulic head and pressure head values, this subroutine employs the commonly used constitutive relationships, Linear, Brooks-Corey and Van Genuchten to calculate relative permeability, volumetric water content, effective saturation, and the derivative function for the change of effective saturation with respect to pressure heads for any soil type at any element along the one-dimensional unsaturated flow region [0,L].

Called from: HICHYD, QUADINTG, QFLOWZ, QSTORAGE

Subroutine called: None

5.1.7.8 Subroutine GAUSS

This subroutine employs Gaussian elimination method to solve for the unknown Fourier coefficients by the backward substitution method. In subroutine GUASS, a system of order n linear algebraic equations, which represents a Fourier finite element coefficient matrix for m number of Fourier harmonics and the right hand side vectors, are solved simultaneously.

In the solution process, this subroutine calls subroutine PIVOTCHANGE to search for the largest absolute pivot value in the coefficient matrices and the right hand

side vectors to perform the necessary sequential changes between the smallest and the largest absolute pivot value within the same matrices. Subroutine GUASS, in turn, called by subroutines FICHYD to solve for the initial and updated Fourier series coefficients.

Called from: FICHYD

Subroutine called: PIVOTCHANGE

5.1.7.9 Subroutine PIVOTCHANGE

This subroutine is called by subroutine GAUSS to search through the upper triangular matrices and the right hand side vectors in order to find the largest absolute pivot value in the matrices and the right hand side vectors. PIVOTCHANG subroutine, in turn, performs the necessary sequential interchanges between the smallest and largest absolute pivot values.

Called from: GAUSS

Subroutine called: None

5.1.7.10 Subroutine CUBIC

This subroutine computes the required coefficients to construct the cubic shape function for the elements located at the vicinity of the ponded ground surface. CUBIC subroutine is called by subroutine HICHYD to calculate the initial hydraulic head distribution, which describes the cubic element shape function at that location.

Called from: HICHYD

Subroutine called: None

5.1.7.11 Subroutine COEFFMAT

This subroutine computes the odd and even Fourier cosine series coefficients and constructs the initial elemental coefficient matrices along the unsaturated flow region

[0,L]. Subroutine COEFFMAT is called once by subroutine FICHYD, at the beginning of the simulation, to compute the Fourier cosine series coefficients of the initial hydraulic head distributions.

Called from: FICHYD

Subroutine called: None

5.1.7.12 Subroutine QUADPTS

QUADPTS subroutine tabulates the nth zeros of the nth degree Gauss-Legendre polynomials $\pm x_q$ and the corresponding weight factors w_q requires for the higher order numerical integration by Gauss-Legendre Quadrature method.

Called from: INPUT

Subroutine called: None

5.1.7.13 Subroutine HQUAD

This subroutine employs the combination of the numerical Gauss-Legendre Quadrature method and the Fourier cosine series coefficients to approximate the hydraulic head at upper and lower quadrature point locations along the unsaturated flow region [0,L].

Called from: HGRADZ, QUADINTG,

Subroutine called: None

5.1.7.14 Subroutine HGRADZ

The main purpose of this subroutine is to approximate the hydraulic head gradient at any designated upper and lower quadrature point locations. Subroutine HGRADZ calculates the upper and lower node locations in the global z-coordinate along the one-

dimensional unsaturated flow region [0,L] and, in turn, makes call to subroutine HQUAD to approximate the hydraulic head at the designated upper and lower quadrature point locations. Given the approximated hydraulic heads at their quadrature point locations along the global z-coordinate, HGRADZ subroutine approximates the hydraulic head gradient at the specified node locations.

Called from: INPUT, QUADINTG,

Subroutine called: HQUAD

5.1.7.15 Subroutine QUADINTG

This subroutine applies the combination of the numerical integration by Gauss-Legendre Quadrature method and the Fourier finite element method to perform the numerical integration for the integral functions in the global coefficient matrices $[A_{ij}]$ and $[B_{ij}]$. Subroutine QUADINTG makes call to subroutine HQUAD to apply Gauss-Legendre Quadrature method and to approximate the hydraulic head and pressure head values at each quadrature point.

QUADINTG subroutine, in turn, makes call to subroutine HGRADZ to approximate the hydraulic head gradient at each quadrature point and also makes call to subroutine AUXEQS to calculate the derivative function for the change of effective saturation with respect to pressure heads. Given all the necessary parameters, subroutine QUADINTG is able to solve for the term ξ presnted by the partial differential equation and enters into the global coefficient matrix $[B_{ij}]$.

Called from: MATXAB

Subroutine called: HQUAD, HGRADZ, AUXEQS

5.1.7.16 Subroutine MATXAB

This subroutine setup the boundary and continuity conditions, computes the odd Fourier series coefficients and constructs the right hand side array. MATXAB subroutine is called once by the main program, at the beginning of each simulation, to construct the global coefficient matrices $[A_{ij}]$ and $[B_{ij}/\Delta t]$ by assembling the sum of the approximated numerical integral function corresponding to the elemental matrices $[a]_{ei}$ and $[b]_{ei}$ along the unsaturated flow region [0,L].

Called from: main program

Subroutine called: QUADINTG

5.1.7.16 Subroutine HICHYD

This subroutine specify a minute of water flux at water table, setup the boundary and initial conditions and calculate the initial hydraulic head distribution along the unsaturated flow region [0,L]. Subroutine HICHYD, makes call to subroutine ETYPMH, to determine the element type and to specify the element identification array, which assigns the appropriate soil properties for each element. HICHYD subroutine calls subroutine AUXEQS to calculate the average reciprocal values of relative permeability at each nodal point and also calls subroutine CUBIC to calculate the initial hydraulic head described by the cubic element shape function at the vicinity of the ponded ground surface.

Called from: INPUT

Subroutine called: EYPMH, AUXEQS, CUBIC

5.1.7.18 Subroutine QFLOWZ

Subroutine QFLOWZ is called from the main program at the end of each simulation to calculate the infiltration rate at the ground surface and to compute the hydraulic head and pressure head values by the approximated Fourier cosine series.

Subroutine QFLOWZ, in turn, makes call to subroutine AUXEQS to calculate the average reciprocal values of relative permeability at each nodal point and to calculate the water flux out to water table.

Called from: main program

Subroutine called: AUXEQS

5.1.7.19 Subroutine QSTORAGE

Subroutine QSTORAGE is called from the main program at the end of each simulation to approximate the hydraulic head by the computed Fourier cosine series, and calculate pressure head at each quadrature point. QSTORAGE subroutine makes call to subroutine AUXEQS to calculate the volumetric water content and employed the numerical integral function to calculate the total volume of water stored in the soil at any particular depth throughout any particular time period.

Called from: main program

Subroutine called: AUXEQS

5.1.7.20 Subroutine WATBAL

This subroutine is called from the main program at the end of each simulation to calculate the average volume of water entered through the ground surface and released to water table at any time t. Subroutine WATBAL also calculates the total amount of water

entered the ground surface and released by the system to water table at any particular time t. Subroutine WATBAL also calculates the cumulative water balance and the percentage of error associated with it at any particular time t.

Called from: main program

Subroutine called: QFLOWZ, QSTORAGE

5.1.7.21 Subroutine HIHOUT

This subroutine is called from the output subroutine to print out the initial points of the global-z coordinate system and the associated initial hydraulic head distributions.

Called from: OUTPUT

Subroutine called: None

5.1.7.22 Subroutine OUTPUT

This subroutine is called from the main program at the end of each simulation to print the results of the FFUSAT model. Subroutine OUTPUT prints the hydraulic head distribution, pressure head distribution, relative permeability, volumetric water content, effective saturation, average volume of water entered through the ground surface and released to water table, cumulative water balance and the percentage of error associated with it at any particular time t.

Called from: main program

Subroutine called: None

5.1.7.23 Subroutines PLOTTING

For plotting purposes, FFUSAT model also prepares special subroutines as: pressure head versus effective saturation (PSIVSE), relative permeability versus effective

saturation (RELPVSE), hydraulic head versus effective saturation (HVPSI) and hydraulic head versus the change of effective saturation with respect to pressure heads (HVDSDH)

CHAPTER 6

APPLICATION TO

ONE-DIMENSIONAL UNSATURATED FLOW PROBLEM

6.1 MODEL VERIFICATION

The FFUSAT model developed in this study was verified on one test problem. Owing to the non-linearity of the governing partial differential equation, closed form analytical solutions are not available for most problems to which the FFUSAT model might be applied. The governing one-dimensional partial differential equation in this model is developed with total hydraulic head as the dependent variable. For the one-dimensional test problem, only one discrete element of 10 ft length was used to describe the unsaturated flow region. The number of Fourier harmonics M_j was 51 harmonics. Starting at initial time t = 0, the solution for the total hydraulic head and pressure heads were evaluated at time steps Δt = 0.05, 0.1, 0.15 and 0.2 day at which the steady state solution was reached. The FFUSAT model has been verified against the UNSAT1D computer model (Celia et al, 1990) upon which 51 nodes were used to describe the problem domain. The initial pressure heads at time t = 0 were input as variable values within the domain. The solution for the total hydraulic head and pressure heads were evaluated at time steps Δt = 0.05, 0.1, 0.15 and 0.2 day at which the steady state solution

was reached. The input data used for the simulation are shown on tables 6-1 and 6-2 respectively.

6.2 Soil Characteristic Curve

Although, three different algebraic equations correspond to the soil characteristic curve are available for use in FFUSAT model, including linear, Brooks and Corey (1964) and Van Genuchten (1980). However, to be consistent with the application of UNSAT1D model, the Van Genuchten soil characteristic curve was applied to describe the unique relationship between pressure head and effective saturation and between pressure head and relative permeability.

6.2.1 Van Genuchten Soil Characteristic Curve

1. Saturation-Pressure Head Relationship

$$S_e(\psi) = \left[1 + |\alpha\psi|^n\right]^{-m} \qquad \text{for } \psi \le 0$$
$$S_e = 1 \qquad \text{for } \psi \ge 0 \tag{6.2a}$$

or

$$S(\psi) = S_r + \frac{1 - S_r}{\left[1 + |\alpha\psi|^n\right]^m} \qquad \text{for } \psi \le 0$$
$$S = 1 \qquad \text{for } \psi \ge 0 \tag{6.2b}$$

2. Relative permeability-Pressure Head Relationship

$$k_{rw}(\psi) = S_e^{1/2}\left[1 - \left(1 - S_e^{1/m}\right)^m\right]^2 \qquad \text{for } \psi \le 0$$
$$k_{rw} = 1 \qquad \text{for } \psi \ge 0 \tag{3.23}$$

Wherein: α is a positive parameter equal to the inverse of the air entry pore water pressure, $-\Psi_a$. m and n are empirically best-fitted parameters.

6.2.2 Domain and Time Discretization

FFUSAT MODEL	Value	UNSAT1D	Value
Element length	10 ft	Domain length	10 ft
No. Of Harmonics	51	No. Of Nodes	51
Time step	0.05 day	Time step	0.05 day
Max. Simulation	1 day	Max. Simulation	1 day
Max. Iterations	100	Max. Iterations	100

6.2.3 Soil Property Parameters

The selected soil property parameters for both FFUSAT and UNSAT1D models for the one-dimensional test problem, are as follow:

Table 6-3: Soil Properties (Van Genuchten)

Parameters	Value
ϕ	0.35
α	0.2
m	0.6
n	2.5
K_z	1.5 ft/day

6.3 Problem Description

The test involves a vertical downward flow of water into a 10 ft unsaturated soil column, the pressure head of zero is prescribed at the top and bottom of the soil column. The hydraulic head is zero at the bottom and equal 10 ft at the top. In the UNSAT1D computer model, 51 nodes were used to describe the problem domain; the nodes were 0.2 ft apart along the problem domain. The initial pressure heads at time t = 0 were input as variable values within the domain. The solution of the test problem was simulated for one day at time steps Δt of 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5, 0.55, 0.6, 0.65, 0.7, 0.75, 0.8, 0.85, 0.9, 0.95 and 1 day.

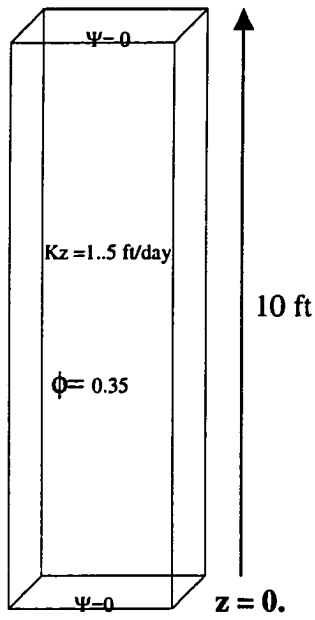


Figure 6-1 Schematic descriptions for the unsaturated flow test problem

The problem domain shown in Figure 6-1 is for one element test problem. FFUSAT model is tested for the solution of the pressure heads and hydraulic head. The obtained results of the simulation are presented on Table 6-3. The corresponding graphs for the test problem are shown in figures 6-2 and 6-3 respectively

6.4 Results and Discussion

Figures 6-2 shows the comparison between the pressure heads solution calculated by the FFUSAT model and the UNSAT1D model respectively. The comparison shows a reasonably good agreement between the two models. However, the pressure head profile computed by the FFUSAT model tends to be a head of that calculated by the UNSAT1D model at later time t =0.15 day. However, steady state condition was reached at t = 0.2 day.

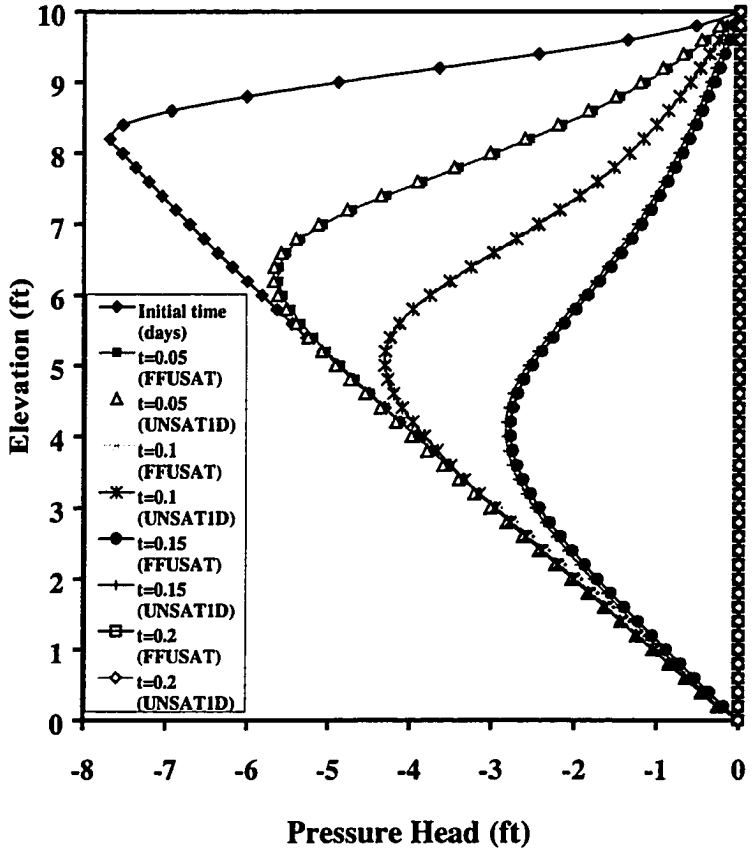


Figure 6-2: Comparison of the pressure head distribution calculated by FFUSAT and UNSAT1D models

Figures 6-3 shows the comparison between the hydraulic head solution calculated by the FFUSAT model and the UNSAT1D model respectively. The comparison shows a reasonably good agreement between the two models. However, the hydraulic head profile computed by the UNSAT1D model tends to be a head of that calculated by the FFUSAT model at time steps t = 0.05, 0.15, and 0.2 days. However, steady state condition was reached at t = 0.2 day.

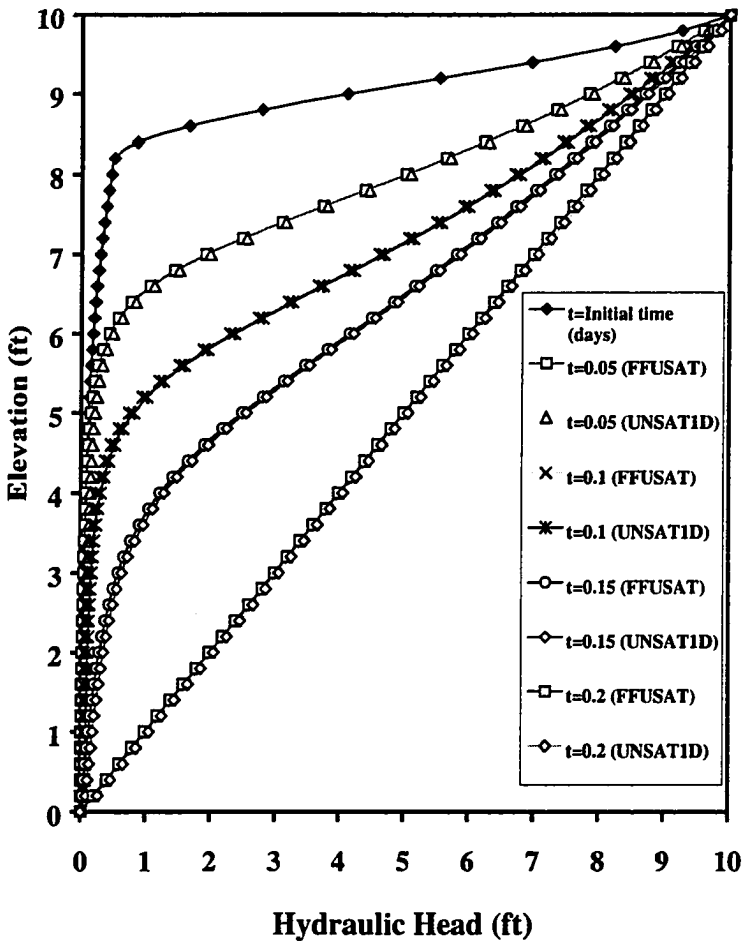


Figure 6-3: Comparison of the hydraulic head distribution calculated by FFUSAT and UNSAT1D models

CHAPTER 7

SUMMARY, CONCLUSIONS AND FUTURE RESEARCH

7.1 SUMMARY

A Fourier finite element method, which combines the efficiency of the Fourier series expansion as a refinement tool and the finite element method is developed and applied to the solution of the partial differential equation governing the unsaturated flow of water in porous media. The spatial domain is divided into discrete element and nodal points. A Fourier cosine series expansion of variable order over discrete elements is used as the approximating functions and the same denomination is applied for the numerical integration techniques using polynomials of a high order as approximating functions.

To estimate the hydraulic head and to solve for the flux functions that obey the governing one-dimensional unsaturated flow equation, the pressure head and hydraulic head solutions are required to be continuous at the element interfaces and the unknown Fourier cosine series coefficients are obtained as the solution of a system of linear algebraic equation.

The weighted residual principle is applied directly in terms of the governing unsaturated partial differential equation on a discrete element basis and the residual at each point along the problem domain is evaluated to determine the degree to which the

obtained Fourier cosine series coefficients are able to approximate the hydraulic head that satisfy the governing unsaturated partial differential equation. The solution to the partial differential equation is obtained by marching through time in discrete steps, beginning with a known solution of Fourier series coefficients at time zero.

Thus, the hydraulic head and the flux functions are evaluated at the end of each time step simulation at any point along the problem domain. FFUSAT is a Fourier finite element computer program, which was written to implement the new methodology with the truncated Fourier cosine series for the solution of the nonlinear partial differential equation governing the one-dimensional unsaturated flow of water in porous media. The nonlinear solution behavior was well simulated by the high order approximating functions. The results were compared with UNSAT1D program (Celia et al, 1990) and a reasonably good agreement was obtained.

7.2 CONCLUSIONS

On the basis of the results obtained from this research, the following conclusions were drawn:

1. The Fourier finite element method with the use of Fourier cosine series as an approximating function along with the numerical integration techniques by means of high order polynomials can be used for the numerical solution of the partial differential equation governing the unsaturated flow of water in porous media.

2. The hydraulic heads may be very well simulated with an appropriate number of Gauss-Legendre quadrature points if enough Fourier harmonics terms are utilized.
3. The new methodology is shown to yield an accurate description of the hydraulic heads and flux functions within a reasonable time frame and computer resources.
4. In addition to hydraulic head and flux functions, the FFUSAT model is capable of determining the hydraulic head gradient, the pressure head, effective saturation, volumetric water content, relative permeability, the change in effective saturation with respect to pressure heads and the change in effective saturation with respect to hydraulic head.

7.3 FUTURE RESEARCH

In light of the results obtained during the course of this research, the following future research and development of the Fourier finite element method are made:

1) Improve the Fourier finite element method for unsaturated flow in porous media to study a two-dimensional flow into very dry heterogeneous soil system.
2) Use the Fourier finite element method in connection with other methods or linked with simpler numerical models to simulate rapid change of the solution in particular directions.

3) Use the Fourier finite element method to study the effect of two-phase simultaneous flow, air and water, in the unsaturated soil system.

4) Use the Fourier finite element method to study the effect of succession wetting and drying events, hysteresis, in the unsaturated soil system.

BIBLIOGRAPHY

[1] Abdel-Meguid, A. M., Resolution Enhancement Of A Finite Element Groundwater Model By The Fast Adaptive Composite Grid Method. M.Sc. thesis, Colorado State University, Fort Collins, Colorado. 1994.

[2] Adjerid, S. and J. Flaherty, A Moving Finite Element Method with Error Estimation and Refinement for One-Dimensional Time Dependent Partial Differential Equations. SIAM J. Numer. Anal.23 (4), 778-796, 1986.

[3] Adjerid, S., J. Flaherty, P. Moore and Y. Wang. High-order Adaptive Methods for Parabolic Systems. Physica D. North-Holland, Elsevier Science Publishers. (60), 94-111, 1992.

[4] Al-Arawi, M. S., A three-dimensional Fast Adaptive Composite Grid Method for Groundwater Problems. Ph.D. Dissertation, Colorado State University, Fort Collins, Colorado. 1991.

[5] Altas, I. and J. Stephenson. A two-dimensional Adaptive Mesh Generation Method. J. of Comput. Physics, (94), 201-224. 1991.

[6] Amokrane, H. and J. P. Villeneuve, A Numerical Method for Solving the Water Flow Equation in Unsaturated Porous Media, Groundwater; 34(4), 666-674, 1996

[7] Axelsson, O. and I. Kaporin, Minimum Residual Adaptive Multilevel Finite Element Procedure for the Solution of Nonlinear Stationary Problems. SIAM J. Numer. Anal. 35(3), 1213-1229. 1998.

[8] Babel, M. et al. Simulation of stable flows in unsaturated homogeneous coarse sand. Hydrologic Sciences-Journal-des Sciences Hydrologiques, 42(1) 49-65, 1997.

[9] Bai, D. and A. Brandt. Local Mesh Refinement Multilevel Techniques. SIAM J. Sci. Statist. Comput. (8), 109-134. 1987.

[10] Bank, R. and T. Dupont. An Optimal Order Process for Solving Finite Element Equations, Math. Comp. 36:35-51.1977

[11] Barry, D. and G. Sander, Exact Solutions for Water infiltration With an Arbitrary Surface Flux or Nonlinear Solute Adsorption. Water Resources Res., 27(10), 2667-2680, 1991.

[12] Babuska, I. and W. Rheinboldt, Error Estimates for Adaptive Finite Element Computations. SIAM J. Numer. Anal.15 (4), 736-754, 1978.

[13] Babuska, I., J. Chandra, and J. Flaherty, Adaptive computational methods for partial differential equations. SIAM, Philadelphia, 144-164, 1983.

[14] Beckie, Roger, E. F. Wood, A. A. Aldama. Mixed Finite Element Simulation of Saturated Groundwater Flow Using a Multigrid Accelerated Domain Decomposition Technique., Water Resources Res., 29(9), 3145-3157., 1993.

[15] Berger, M. and P. Colella. Local Adaptive Mesh Refinement for Shock Hydrodynamics. J. of Comput. Physics, (82), 64-84. 1989.

[16] Brandt, A., Multilevel Adaptive Solutions to Boundary Value problems, Math.Comp.31: 333-390.1977

[17] Brandt, A., Algebraic Multigrid Theory: The Symmetric Case, Appl. Math. Comp.19: 23-56. 1986.

[18] Bramble, H., R. Ewing, J. Pasciak. and Schatz. A Preconditioning Technique for the Efficient Solution of Problems with Local Grid Refinement. Comput. Meths. Appl. Mech. Engrg. 67:149-159,1988.

[19] Brass, D. and W. Hackbusch, A New Convergence Proof for the Multigrid Method including the V-cycle, SIAM J. Numer. Anal. 20 (5), 967-975. 1983.

[20] Broadbridge, et al., Constant rate rainfall in a bounded profile: Solutions of a nonlinear mode, Soil Sci. Soc. Am. J., 52, 1526-1533, 1988.

[21] Broadbridge, P. and I. White, Constant rate rainfall infiltration: A versatile nonlinear model, 1. Analytical Solution, Water Resources Res., 24(1), 145-154. 1988.

[22] Burie, J. and M. Marion. Adaptive Multilevel Methods in Space and Time for Parabolic Problems-The Periodic Case. Mathematics of Computation. pp. 1-35, Electronically Published. March 3,1999.

[23] Cao, W., W. Huang. and R. Russell. An r-Adaptive Finite Element Method Based upon Moving Mesh Partial Differential Equations. J. of Comput. Physics. (149), 221-244. 1999.

[24] Carlenzoli, C. and A. Quarteroni, Adaptive Domain Decomposition Methods for Advection-Diffusion Problems. Proceeding of the IMA Workshop on Mesh Adaptivity. 1993.

[25] Celia, M.et al. A general mass-conservative numerical solution for the unsaturated flow equation. Water Resour. Res., 26(7) 1483-1496, 1990.

[26] Celia, M.et al. A mass conservative numerical solution for two-phase flow in porous media with application to unsaturated flow. Water Resour. Res., 28(10) 2819-2828, 1992.

[27] Chehata, M. Spectral Domain Decomposition Method for the Solution of the Groundwater Flow Equation. Ph.D. thesis, Colorado State University, Fort Collins, Colorado. 1988.

[28] Choi, S. Resolution Enhancement of a Finite Difference Groundwater Model by Spectral Decomposition Method. . Ph.D. thesis, Colorado State University, Fort Collins, Colorado. 1989.

[29] Costa, B. and L. Dettori, Fourier Collocation Splitting for Partial Differential Equations. J. of Comput. Physics, (142), 562-580. 1998.

[30] Davis, S. and J. Flaherty, An adaptive finite element method for initial boundary value problems for partial differential equations. SIAM J Sci. Stat. Comput. 3:6-27, 1982.

[31] Devloo, P., J. Oden, and P. Pattani. An h-p Adaptive Finite Element Method for the Numerical Simulation of Compressible Flow. Comput. Methods. Appl. Mech. Engrg. 70:203-235. 1988.

[32] Dunbar, W. S., A. D. Woodbury, B. Nour-Omid, G. Gambolati, Comment 'On time Integration of Groundwater flow Equations by Spectral Methods. Water Resources Res., 30(7), 2347-2352, 2353- 2355, 1994.

[33] Eriksson, K. and C. Johnson. Adaptive Finite Element Methods for Parabolic Problems I: A Linear Model Problem. SIAM J. Numer. Anal.28 (1), 43-77, 1991.

[34] Ewing, R. E. A Survey of Domian Decomposition Techniques and their Implementation. Advances in Water Resources, 13(3), 117-125, 1990.

[35] Ewing, R., R. Lazarov. and P. Vassilevski. Local Refinement Techniques for Elliptic Problems on Cell-Centered Grids. I. Error Analysis. Math. Of Comput. 56(194), 437-461, 1991.

[36] Flaherty, J. J. Coyle, R. Ludwing and S. Davis, Adaptive finite element methods. SIAM, Philadelphia, 1983.

[37] Freeze, R. and J. Cherry, Groundwater, Prentice Hall, Englewood Cliffs, NJ. 07632, USA, 1979.

[38] Gambolati, G., On time Integration of Groundwater flow Equations by Spectral Methods., Water Resources Res., 29(4), 1275-1267, 1993.

[39] Guymon, G., Unsaturated Zone Hydrology, Prentice Hall, Englewood Cliffs, NJ. 07632, USA, 1994.

[40] Gottardi, G. and M. Venutelli, Moving Finite Element Model for one-dimensional Infiltration in Unsaturated Soil, Water Resources Res., 28(12), 3259-3267, 1992.

[41] Gottardi, G. and M. Venutelli, One-dimensional moving finite-element model of solute transport. Groundwater, 32(4), 645-649, 1994.

[42] Haberman, R., Elementary Applied Partial Differential Equations: With Fourier Series and Boundary Value Problems, 2nd edition. Prentice-Hall, Englewood Cliffs, New Jersey, 1987.

[43] Hamzeh, O. N. the Fast Adaptive Composite Grid Method in Finite Element Applications. M.Sc. thesis, Colorado State University, Fort Collins, Colorado. 1990.

[44] Heinrich, B. The Fourier-Finite-Element Method for Poisson's Equation in Axisymmetric Domains with Edges. SIAM J. Numer. Anal. 33 (5), 1885-1911, 1996.

[45] Heroux, M. A. the Fast Adaptive Composite Grid Method for Time Dependent Problems. Ph.D. thesis, Colorado State University, Fort Collins, Colorado. 1989.

[46] Hart, L., S. McCormick, A. O'Gallagher, and J. Thomas. The Fast Adaptive Composite Grid Method (FAC): Algorithms for Advance Computers. Appl. Math. and Comput., 19:103-125, 1986.

[47] Haverkamp, R., et al. A comparison of numerical simulation models for one-dimensional infiltration, Soil Sci. Soc. Am. J., 41:285-294, 1977

[48] Harter, T. and T.Yeh, An Efficient Method for Simulating Steady Unsaturated Flow in Random Porous Media: Using an Analytical Perturbation Solution as Initial Guess to a Numerical Model. Water Resources Res., 29(12), 4139-4149, 1993.

[49] Hillel, D., Fundamental of Soil Physics, Academic, San Diego, California, USA, 1980.

[50] Hillel, D. and R. Baker, A descriptive theory of fingering during infiltration into layered soils. Soil Sci. 146(1).51-56, 1988.

[51] Heuer, N., E. Stephan and T. Tran. Multilevel Additive Schwarz Method for The h-p Version of the Galerkin Boundary Element Method. Math. Of Comput. 67(222).501-518, 1998.

[52] Hoppe, R. and B. Wohlmuth. Adaptive Multilevel Techniques for Mixed Finite Element Discretizations of Elliptic Boundary Value Problems. SIAM J. Numer. Anal. 34(4), 1658-1681,1997.

[53] Huang, W., Y. Ren. and R. Russel. Moving Mesh Partial Differential Equations (MMPDES) Based on the Equidistribution Principle. SIAM J. Numer. Anal. 31 (3), 709-730, 1994.

[54] Huang, W., and R. Russell. Moving Mesh Strategy Based on A Gradient Flow Equation for two-dimensional Problems. SIAM J. Sci. Comput. 20(3), 998-1015, 1999.

[55] Istok, J., Groundwater Modeling by the Finite Element Method. American Geophysical Union, Water Resources Monograph 13, 1989.

[56] Japhet, C., Optimisation of Interface Conditions in Domain Decomposition Methods: Application to Convection-Diffusion Problems. Ninth International Conference on Domain Decomposition Methods. P. Bjørstad, M. Espedal and D. Keyes, editors, Bergen, Norway. 3-8 June, 1997

[57] Kirkland, M. et al. Algorithms for solving Richards' equation for variably saturated flow. Water Resources Res., 28(8), 2049-2058,1992.

[58] Lee, June-Yub. and L. Greengard. A Fast Adaptive Numerical Method for Stiff Two-Point Boundary Value Problems. SIAM J. Sci. Comput., 18(2), 403-429. 1997.

[59] Li, S. and L. Petzold. Moving Mesh Methods with Upwinding Schemes for Time-Dependent Partial Differential Equations. J. of Comput. Physics, (131), 368-377. 1997.

[60] Lions, P.L., On the Schwarz Alternating Method III: A variant for Nonoverlapping Subdomains, Third International Symposium on Domain Decomposition Methods for Partial Differential Equations, SIAM. pp.202-223, 1989.

[61] Liu, A. and B. Joe. Quality Local Refinement of Tetrahedral Meshes Based on 8-Subtetrahedron Subdivision. Math. Of Comput. 65(215), 1183-1200. 1996.

[62] Maidment, D., Handbook of hydrology, McGraw-Hill, New York, USA. 1992.

[63] Marino, M. and J. Luthin, Seepage and Groundwater, Elsevier, New York, 1982.

[64] McKay, S. Adaptive Methods Applied to the Fast Adaptive Composite Grid Method. Ph.D. thesis, Colorado State University, Fort Collins, Colorado. 1990.

[65] McKeon, T. and Wen-Sen Chu, A Multigrid Model for Steady Flow in Partially Saturated Porous Media. Water Resources Res., 23(4), 542-550, 1987.

[66] McKinney, D. and Han-Lin Tsai, Multigrid Methods in GIS Grid-Cell-Based Modeling Environment. J. of Computing in Civil Engr. 10(1), 25-30, 1996.

[67] McCormick, S. F., Multilevel Adaptive Methods for Partial Differential Equations. SIAM, Philadelphia, USA. 1989.

[68] McCormick, S. and J. Thomas. The Fast Adaptive Composite Gird Method. (FAC) for Elliptic Boundary Value Problems, Math. Comp. (46), 439-456. 1986.

[69] McWhorter, D., and D. K. Sunada, Groundwater Hydrology and Hydraulics, Water Resources Publications, Fort Collins, Colorado, 1977.

[70] Miller, C. T. and G. A. Williams, Robust solution of Richards' Equation for Non-uniform porous media, Water Resources Res., 34(10), 2599-2610, 1998.

[71] Miller, K. and R. Miller, Moving finite elements I. SAIM J. Numer. Anal. 18(6), 1019-1032, 1981.

[72] Miller, K., Moving finite elements. I I. SAIM J. Numer. Anal. 18(6), 1033-1057, 1981

[73] Milly, P., A Mass-conservative Procedure for Time-stepping in Models of Unsaturated Flow. Adv. Water Res., 8:32-36, 1985.

[74] Milly, P. Advances In the Modeling of Water in the Unsaturated Zone. Transp. Porous Media, 3:491-518, 1988.

[75] Mirabzadeh, et al., Numerical solution of water flow in the unsaturated zone using a dynamic programming. Groundwater: An Endangered Resource. Proceeding of Theme C. San Francisco, CA. August 10-15, 1997.

[76] Oosterlee, C. and T. Washio, On the Use of Multigrid as a Preconditioner. Ninth International Conference on Domain Decomposition Methods. P. Bjørstad , M. Espedal and D. Keyes, editors, Bergen, Norway. 3-8 June 1997.

[77] Pan, L. and J. Wierenga, A transformed pressure head-based approach to solve Richards' equation for variably saturated soils. Water Resources Res., 31(4), 925-931,1995.

[78] Paniconi, C., A. A. Aldama, and E. F. Wood, Numerical Evaluation of Iterative and Noniterative Methods for the Solution of the Nonlinear Richards Equation., Water Resources Res., 27(6), 1147-1163, 1991.

[79] Parlange, J. -Y., Water Transport in Soils. Annual Review of Fluid Mechanics. 12:77-102, 1980.

[80] Parlange, J. -Y. and D. E. Hill, Air and water movement in porous media: compressibility effects. Soil Sci. 127: 257-263, 1979

[81] Philip, J. "The Theory of infiltration" 1. The infiltration equation and its solution. Soil Sci., 83:345-357, 1957.

[82] Philip, J. "Theory of infiltration". Advance Hydrosci.5: 215-290,1969.

[83] Pinkus, A. and S. Zafrany, Fourier Series and Integral Transforms. Cambridge University Press,1997

[84] Quarteroni, A., Fourier Spectral Methods for Pseudo-Parabolic Equations, SIAM J. Numer. Anal., 24(2), 323-335, 1987.

[85] Richards, L. A., Capillary conduction of liquids through porous mediums, Physics, 1, 318-333, 1931.

[86] Rogers, C. et al. On two-phase filtration under gravity and with boundary infiltration: Application of a Backlund transformation, Nonlinear Anal. Theory Methods Appl. 7:785-799, 1983.

[87] Romano, N., et al., Numerical Analysis of one-dimensional Unsaturated flow in layered Soils, Advance in water resources, 21(4), 315-324, 1998.

[88] Ross, P., Efficient numerical methods for infiltration using Richards' equation. Water Resources Res., 26(2), 279-290, 1990.

[89] Salvucci, G., Series solution for Richards' equation under concentration boundary conditions and uniform initial conditions. Water Resources Res., 32(8), 2401-2407, 1996.

[90] Sanders, C. et al., Exact Solution for Nonlinear, Nonhysteretic Redistribution in Vertical Soil of Finite Depth. Water Resources Res., 27(7), 1529-1536, 1991.

[91] Scheidegger, A., The Physics of Flow through Porous Media, 3rd ed. University of Toronto, Ont.1974.

[92] Schwarz, H. A., Gesammelte Mathmatische Abhandlungen, Springler, Berlin, 2, 1890.

[93] Serrano, S. Analytical decomposition of nonlinear unsaturated flow equation. Water Resources Res., 34(3), 397-407, 1998.

[94] Salvucci, G. D., Series solution for Richards' Equation under concentration boundary conditions and uniform initial conditions, Water Resources Res., 32(8), 2401-2407, 1996.

[95] Slattery, J. E., The Finite Strip Method in Groundwater Hydrology, Master Thes. Colorado State University, Fort Collins, Colorado, 1986.

[96] Sposito, G., The "Physics" of Soil Water Physics, Water Resources Res., 22(9), 83s-88s, 1986.

[97] Stals, L., Implementation of Multigrid on Parallel Machines Using Adaptive Finite Element Methods. Ninth International Conference on Domain Decomposition Methods. P. Bjørstad, M. Espedal and D. Keyes, editors, Bergen, Norway. 3-8 June,1997

[98] Spiegel, M. R., Fourier Analysis with Application to Boundary Value Problems. Schaum's Outlines Series. McGraw-Hill, 1974.

[99] Stephens, D., Vadose Zone Hydrology, CRC Press, Inc. Lewis Publishers, New York, USA.1996.

[100] Sumner, et al., Effect of Unsaturated Zone on Groundwater Mounding. J. Hydrologic Engineering. 4(1) 65-69, 1999.

[101] Tan, K. and M. Borsboom, On Generalized Schwarz Coupling Applied to Advection-Dominated Problems, 180:125-130, 1994.

[102] Totoev, Y. and P. Kleeman, An infiltration model to predict suction changes in the soil profile. Water Resources Res., 34(7), 1617-1622,1998.

[103] Todd, D. K., Groundwater Hydrology, 2nd Ed., John Wiley and Sons, 1980.

[104] Van Genuchten, M. A comparison of numerical solutions of one-dimensional unsaturated-saturated flow and mass transport equations. Adv. Water Res., 5:47-55, 1982.

[105] Vereecken, H., O. Neuendrof, G. Lindenmary, and A. Basermann. A Schwarz Domain Decomposition Method for Solution of Transient Unsaturated water Flow on Parallel Computers., Ecological Modelling, 93(3), 275-289, Dec. 1996.

[106] Vichnevetsky, R., and J. B. Bowles, Fourier Analysis of Numerical Approximations of Hyperbolic Equations, Studies in Appl. Math. Philadelphia, Penn., 1982.

[107] Voigt, R. G., D. Gottlieb, and M. Y. Hussanini, Spectral Methods for Partial Differential Equations, SIAM, Philadelphia, 1984.

[108] Wang, H. F., and M. P. Anderson, Introduction to Groundwater Modeling: Finite Difference and Finite Element Methods, Freeman and Company, San Fransisco, California, 1982.

[109] Warrick, A. et al., An analytical solution to Richards' equation for a draining soil profile, Water Resources Res., 26(2), 253-258, 1990.

[110] Warrick, A. et al., An analytical solution to Richards' equation for time-varying infiltration, Water Resources Res., 27(5), 763-766, 1991.

[111] Wildenschild, D. and K. Jensen, Numerical modeling of observed effective flow behavior in unsaturated heterogeneous sands, Water Resources Res. 35(1), 29-42, 1999.

[112] Yeh, T. C., One-Dimensional Steady State Infiltration in Homogeneous Soils. Water Resources Res.,25(10), 2149-2158, 1989.

[113] Zienkiewicz, O. C., The Finite Element Method. 3rd expanded and revised edition., McGraw-Hill, London, 1977.

APPENDIX A

COMPUTER PROGRAM LISTING

(FFUSAT)
FOURIER FINITE ELEMENT MODEL
FOR
UNSATURATED FLOW IN POROUS MEDIA

APPENDIX (A)

COMPUTER PROGRAM LISTING
FOURIER FINITE ELEMENT MODEL
FOR
UNSATURATED FLOW IN POROUS MEDIA
(FFUSAT)

```
C*************************************************************
      PROGRAM FFUSAT
C*************************************************************
      PARAMETER (NEL=3,NNP=NEL+1,MH=200,NZMAX=MH*10,NQP=256)
C
      COMMON /BLOCK1/NUMNP,NUMEL,Z(NNP),MHARM(NEL),ZL,ZLE(NEL)
      COMMON /BLOCK2/IFLAG(10)
      COMMON /BLOCK3/JFLAG(20),SATK(20),PHI(20),THETAR(20),
     +      PSIR(20),BCLAMDA(20),BCPB(20),VGALPHA(20),VGM(20),VGN(20)
      COMMON /BLOCK4/IDELEM(NEL)
      COMMON /BLOCK5/TIME,DTIME,TOTIME
      COMMON /BLOCK6/PSII(NNP),PSI(NNP),HI(NNP),H(NNP),PSICON
      COMMON /BLOCK7/AOLD(NEL,0:MH),ANEW(NEL,0:MH)
      COMMON /BLOCK8/XF(3)
      COMMON /BLOCK9/CM(MH,MH),RHS(MH),X(MH),XOLD(MH),NR,NC
      COMMON /BLOCK10/DXQ(NQP),DWQ(NQP),HQP(NEL,NQP),DHDZ(NEL,NQP)
      COMMON /BLOCK12/QIN,QOUT,TOTQIN,TOTQOUT,QSTORI
      COMMON /BLOCK13/HIH(NEL,MH),ZGHIH(NEL,MH),HZ(NEL,MH)
C
      DOUBLE PRECISION DXQ,DWQ,DSUMA,DSUMB,Z
C
      TIME=0.
      DTIME=.05
      TOTIME=.15
C
      CALL INPUT
C
  100 TIME=TIME+DTIME
C
      CALL MATXAB (I,M,N,DSUMA,DSUMB)
C
      CALL GAUSS (CM,RHS,NR,X)
C
C SOLUTION RETURN IN X-SET AOLD=X
C
      ANEW(I,0)=X(1)
      ANEW(I,1)=X(2)
C
C CALCULATE ODD COEFFICIENTS
C
```

```
   40 ANEW(I,J)=ANEW(I,1)/J/J
C
C SAVE EVEN COEFFICIENTS
C
      SUM=0
      DO 50 J=2,MHARM(I),2
      ANEW(I,J)=X(J/2+2)
   50 SUM=SUM+ANEW(I,J)
      WRITE(9,99)TIME,SUM
   99 FORMAT(/'TIME=',F10.3,/'SUM EVEN=',F20.6)
C
C TEST IF TOTAL TIME REACHED
C
      IF(TIME.LT.TOTIME)THEN
C
C PROCEED TO NEXT TIME STEP
C
C
C SAVE FOURIER COEFFICIENTS IN ANEW
C
      DO 60 I=1,NUMEL
      DO 60 J=1,MHARM(I)
   60 AOLD(I,J-1)=ANEW(I,J-1)
      GOTO 100
      ENDIF
C
      WRITE(8,65)TIME
   65 FORMAT(/'TIME=',F10.3)
C
      DO 80 I=1,NUMEL
      DO 80 J=1,MHARM(I)
      WRITE(8,70)ANEW(I,J-1)
   70 FORMAT(F10.6)
   80 CONTINUE
C
      CALL HOUT
      GOTO 999
C     CALL WATBAL (1)
C        GOTO 100
C     ENDIF
C     CALL HOUT
C
  999 WRITE(*,10)
   10 FORMAT(' NORMAL PROGRAM EXECUTION TERMINATED')
      STOP
      END
C
C*********************************************************************
C*********************************************************************
C
      SUBROUTINE INPUT
C
**********************************************************************
C
C THIS SUBROUTINE READS AND WRITES (ECHOS) ALL INPUT DATA
C
      PARAMETER (NEL=3,NNP=NEL+1,MH=200,NZMAX=MH*10,NQP=256)
```

```
C
      COMMON /BLOCK1/NUMNP,NUMEL,Z(NNP),MHARM(NEL),ZL,ZLE(NEL)
      COMMON /BLOCK2/IFLAG(10)
      COMMON /BLOCK3/JFLAG(20),SATK(20),PHI(20),THETAR(20),
     +      PSIR(20),BCLAMDA(20),BCPB(20),VGALPHA(20),VGM(20),VGN(20)
      COMMON /BLOCK4/IDELEM(NEL)
      COMMON /BLOCK5/TIME,DTIME,TOTIME
      COMMON /BLOCK6/PSII(NNP),PSI(NNP),HI(NNP),H(NNP),PSICON
      COMMON /BLOCK7/AOLD(NEL,0:MH),ANEW(NEL,0:MH)
      COMMON /BLOCK11/ZG(500),DHDZ(500)
      COMMON /BLOCK12/QIN,QOUT,TOTQIN,TOTQOUT,QSTORI
      COMMON /BLOCK13/HIH(NEL,MH),ZGHIH(NEL,MH),HZ(NEL,MH)
C
      DOUBLE PRECISION ZG,Z
C
C
C
C*********************************************************************
      CHARACTER TITLE*80, COMMENT*80
C
C     OPEN FILES
C
      OPEN(UNIT=5,FILE='FFUSAT.DAT',STATUS='OLD')
      OPEN(UNIT=6,FILE='FFUSAT.ECH',STATUS='UNKNOWN')
      OPEN(UNIT=7,FILE='FFUSAT.OUT',STATUS='UNKNOWN')
      OPEN(UNIT=8,FILE='DEBUG.OUT',STATUS='UNKNOWN')
      OPEN(UNIT=9,FILE='PLOT.DAT',STATUS='UNKNOWN')
C
*********************************************************************
C
C  DISPLAY BANNER
C
      CALL BANNER
C
*********************************************************************
****
C
C  READ TITLE OF PROBLEM
C
      READ(5,10) TITLE
  10  FORMAT(A)
      WRITE(6,10) TITLE
      WRITE(*,15) TITLE
  15  FORMAT(///,1X,A)
C
C*********************************************************************
C
C
      WRITE(6,25)
   25 FORMAT(//////,
     +25X,' I N P U T     D A T A  '/,
     +25X,'_____________________ '//)

C  READ GRID DATA
C
C  NUMEL = NUMBER OF ELEMENTS
C  ZL = DEPTH TO GROUNDWATER
```

```
C  NUMNP = NUMBER OF NODAL POINTS = NUMEL+1
C
      READ(5,10)COMMENT
      READ(5,20)NUMEL,ZL
   20 FORMAT(I5,F10.0)
      NUMNP = NUMEL+1
      WRITE(6,30)NUMEL,ZL
   30 FORMAT(///
     1'     GRID DATA              ',/
     2'     ___________            ',//
     3'     NUMEL (NUMBER OF ELEMENTS)    = ',4X,I5/
     4'     ZL    (DEPTH TO GROUNDWATER) = ',F10.1)
C*****************************************************************************
C
C READ Z COORDINATES OF NODAL POINTS
C
C*****************************************************************************
C
C IFLAG(1) = 1 (UNIFORM ELEMENT LENGTH)
C IFLAG(1) NOT= 1 (VARIABLE ELEMENT LENGTH)
C IF IFLAG = 1 THEN PROGRAM WILL CALCULATE Z CORRDINATE OF NODAL POINTS
C OTHERWISE IF IFLAG NOT=1 THEN READ Z COORDINATES FROM DATA FILE
C
      READ(5,10)COMMENT
      READ(5,40)IFLAG(1)
   40 FORMAT(I2)
      IF(IFLAG(1) .EQ. 1) THEN
         DELTAZ = ZL/NUMEL
         Z(1) = 0.
         DO 50 I=2,NUMNP
   50    Z(I)=Z(I-1)+DELTAZ
      ELSEIF(IFLAG(1) .NE. 1) THEN
         READ(5,10)COMMENT
         READ(5,60) (NP,Z(NP),I=1,NUMNP)
   60    FORMAT(5(I5,F10.0))
      ENDIF
C
C PRINT OUT NODAL CORDINATES
C
      WRITE(6,25)
      WRITE(6,70)
   70 FORMAT(///,
     +4X,' NODAL COORDINATES '/,
     +4X,'_____________________'//)
      WRITE (6,80)
   80 FORMAT(/,
     +3(' NODE      Z(NP)    ')/
     +3('______    _______   '))
      WRITE(6,90)(I,Z(I),I=1,NUMNP)
   90 FORMAT(3(I3,F12.2,6X))
C*****************************************************************************
C
C CALCULATE ELEMENT LENGTH
C
C*****************************************************************************
C
      DO 100 I=1,NUMEL
```

```
      ZLE(I)=Z(I+1)-Z(I)
  100 CONTINUE
C
C  PRINT OUT ELEMENT LENGTH
C
      WRITE(6,25)
      WRITE(6,110)
  110 FORMAT(///,
     +4X,' ELEMENT LENGTH '/,
     +4X,'________________'//)
      WRITE (6,120)
  120 FORMAT(/,
     +1(' ELEMENT (I)    Z(I)       Z(I+1)      ZLE (I)  ')/
     +1('_____________   _____      ______      _______  '))
      WRITE(6,130)(I,Z(I),Z(I+1),ZLE(I),I=1,NUMEL)
  130 FORMAT(I6,3F13.2,5X)
C
*************************************************************************
C
C READ SOIL PROPERTIES
C
C*************************************************************************
C
C NSTYPE = NUMBER OF SOIL TYPES
C
      READ(5,10)COMMENT
      READ(5,140)NSTYPE
  140 FORMAT(I5)
C
C  READ PROPERTIES FOR EACH SOIL TYPE
C
C  ISTYPE = SOIL TYPE"I"
C  F1 = SATK = SATURATED HYDRAULIC CONDUCTIVITY
C  F2 = PHI = POROSITY
C  F3 = THETAR = RESIDUAL VOLUMETRIC WATER CONTENT
C  J  = JFLAG IDENTIFYING AUXILIARY EQUATIONS TO BE USED
C       J = 1 = LINEAR
C       J = 2 = BROOKS COREY
C       J = 3 = VAN GENUCHTEN
C
C  LINEAR RELATIONSHIP
C
C  F4 = PSIR = RESIDUAL PRESSURE HEAD
C  F4 = 0 FOR BROOKS/COREY AND VAN GENUCHTEN RELATIONSHIPS
C
C  BROOKS COREY RELATIONSHIP
C
C  F5 = BCLAMDA = PORE SIZE INDEX
C  F6 = BCPB = BUBBLING PRESSURE
C  F5=F6=0 FOR LINEAR AND VAN GENUCHTEN RELATIONSHIPS
C
C  VAN GENUCHTEN RELATIONSHIP
C
C  F7 = VGALPHA = FITTING PARAMETER = CONSTANT
C  F8 = VGM     = FITTING PARAMETER = CONSTANT
C  F9 = VGN     = FITTING PARAMETER = CONSTANT
C  F7=F8=F9=0= FOR LINEAR AND BROOKS COREY RELATIONSHIPS
```

```
C
      READ(5,10)COMMENT
      DO 160 I =1,NSTYPE
      READ(5,150)ISTYPE,F1,F2,F3,J,F4,F5,F6,F7,F8,F9
  150 FORMAT(I5,F10.0,2F5.0,I5,6F5.0)
      SATK(ISTYPE) = F1
      PHI(ISTYPE) = F2
      THETAR(ISTYPE) = F3
      JFLAG(ISTYPE) = J
      PSIR(ISTYPE) = F4
      BCLAMDA(ISTYPE) = F5
      BCPB(ISTYPE) = F6
      VGALPHA(ISTYPE) = F7
      VGM(ISTYPE) = F8
      VGN(ISTYPE) = F9
  160 CONTINUE
C
      WRITE(6,170)
  170 FORMAT(//////,
     +5X,'SOIL TYPES AND PROPERTIES'/,
     +5X,'_____'//)
      WRITE(6,180)
  180 FORMAT(//,
     +('SOIL TYPE',4X,'SATK',7X,'PHI',7X,'THETAR',6X,'JFLAG',5X,'PSIR',
     +7X,'BCLAMDA',7X,'BCPB',7X,'VGALPHA',7X,'VGM',9X,'VGN')/
     +'___________',3X,'____',7X,'___',7X,'_______',6X,'_____',5X,'____',
     +7X,'_______',7X,'____',7X,'_______',6X,'____',8X,'____')
      WRITE(6,190)(I,SATK(I),PHI(I),THETAR(I),JFLAG(I),PSIR(I),
     +BCLAMDA(I), BCPB(I),VGALPHA(I),VGM(I),VGN(I),I=1,NSTYPE)
  190 FORMAT(I5,3F11.2,I11,6F12.2)
C*********************************************************************
C
C  READ SOIL TYPE FOR EACH ELEMENT
C
C*********************************************************************
C
      READ(5,10)COMMENT
      READ(5,200)(ISTYPE,IDELEM(ISTYPE),I=1,NUMEL)
  200 FORMAT(16I5)
      WRITE(6,220)
  220 FORMAT(//,
     +8(1X'ELEM',2X,'SOIL')/
     +8(1X'NUMB',2X,'TYPE')/
     +8(1X'____',2X,'____'))
      WRITE(6,230)(I,IDELEM(I),I=1,NUMEL)
  230 FORMAT(8(2I4,3X))
C*********************************************************************
*******
C
C  READ TIME AND ITERATION PARAMETERS
C
C*********************************************************************
*******
C
C      READ(5,10)COMMENT
C      READ(5,240)MXTSTP,MXITER,DLTAT,DLTAI,TDMAX
C  240 FORMAT(2I5,3F10.2)
```

```
C      WRITE (6,250)MXTSTP,MXITER,DLTAT,DLTAI,TDMAX
C  250 FORMAT(///,25X,'TIME AND ITERATION PARAMETERS'/
C     +25X,           '_____________________________'///
C     +10X,'MXTSTP (MAXIMUM NUMBER OF TIME STEPS)=',2X,I5/
C     +10X,'MXITER (MAXIMUM NUMBER OF ITERATIONS)=',2X,I5/
C     +10X,'DLTAT  (TIME FOR EACH STEP          )=',2X,F5.3/
C     +10X,'DLTAI  (INITIAL TIME STEP           )=',2X,F5.3/
C     +10X,'TDMAX  (MAXIMUM TIME STEP IN DAYS   )=',2X,F5.3)
C*********************************************************************
*******
C
C READ INITIAL HEAD DISTRIBUTION
C
C IF IFLAG(2) = 1 THEN HYDROSTATIC INITIAL CONDITIONS
C
C*********************************************************************
C
      READ(5,10)COMMENT
      READ(5,260) IFLAG(2)
  260 FORMAT(I2)
C
      IF (IFLAG(2) .EQ. 1) THEN
C
C   HYDROSTATIC INITIAL HEAD DISTRIBUTION
C
      WRITE(6,270)
  270 FORMAT(//////,
     +25X,'I N P U T     D A T A'/,
     +25X,'_____________________'//)
      WRITE(6,280)
  280 FORMAT(///,
     +4X,' HYDROSTATIC INITIAL CONDITIONS '/,
     +4X,'_____________________________________'//)
C
C*********************************************************************
C
C QUADPTS PERFORMS NUMERICAL APPROXIMATION TO THE INTGRAL FUNCTION
C
      CALL QUADPTS
C
C SPECIFY QSMALL AND CALCULATE THE INITIAL HYDROSTATIC HEAD DISTRIBUTION
C
      CALL HICHYD
C
C CALCULATE THE INITIAL VALUES OF FOURIER COEFFICIENTS
C
      CALL FICHYD
C
C CALCULATE THE HYDRAULIC HEAD GRADIENT (DH/DZ)
C
      CALL HGRADZ
C
C PRINT THE INITIAL HYDRAULIC HEAD DISTRIBUTIONS (HIHZ)
C
      CALL HOUT
C
      ELSE
```

```
      STOP
      ENDIF
C*********************************************************************
C
C DETERMINE INITIAL WATER FLOWIN/FLOWOUT AND THE INITIAL WATER STORED
C
C*********************************************************************
C
C QIN  = INFILTRATION RATE INTO SOIL AT THE GROUND SURFACE
C QOUT = WATER FLOW THROUGH THE UNSATURATED ZONE TO THE WATER TABLE
C QSTORI = INITIAL AMOUNT OF WATER STORED IN SOIL
C TOTQIN = TOTAL AMOUNT OF WATER FLOW INTO THE SOIL
C TOTQOUT = TOTAL AMOUNT OF WATER RELEASED BY THE SOIL
C
C DETERMINE INITIAL QIN AND QOUT
C
      CALL QFLOWZ
C
C DETERMINE INITIAL WATER IN STORAGE
C
      CALL QSTORAGE (QSTORI)
C
C INITIALIZE WATER BALANCE
C
      TOTQIN=0.
      TOTQOUT=0.
C
      RETURN
      END
C*********************************************************************
      SUBROUTINE ETYPMH
C*********************************************************************
C
C  THIS SUBROUTINE DETERMINES ELEMENT TYPE,SPECIFY NUMBER OF HARMONICS
C  FOR HYDROSTATIC, CONSTANT AND USER DEFINED CONDITIONS FOR INITIAL
C  HEAD DISTRIBUTIONS IN FFVSAT MODEL
C
C*********************************************************************
C
      PARAMETER (NEL=3,NNP=NEL+1,MH=200,NZMAX=MH*10,NQP=256)
C
      COMMON /BLOCK1/NUMNP,NUMEL,Z(NNP),MHARM(NEL),ZL,ZLE(NEL)
      COMMON /BLOCK2/IFLAG(10)
      COMMON /BLOCK6/PSII(NNP),PSI(NNP),HI(NNP),H(NNP),PSICON
      COMMON /BLOCK7/AOLD(NEL,0:MH),ANEW(NEL,0:MH)
      COMMON /BLOCK8/XF(3)
      COMMON /BLOCK9/CM(MH,MH),RHS(MH),X(MH),XOLD(MH),NR,NC
C
      DOUBLE PRECISION Z
C*********************************************************************
C
C  MHARM(I) = NUMBER OF HARMONICS + 1 FOR ELEMENT I
C  IFLAG(2) = 1 IS HYDROSTATIC INITIAL HEAD DISTRIBUTION
C  IFLAG(2) = 2 IS CONSTANT INITIAL HEAD DISTRIBUTION
C  IFLAG(2) = 3 IS USER DEFINED INITIAL HEAD DISTRIBUTION
C
      DO 500 I=1, NUMEL
```

```
C
C DETERMINE NUMBER OF HARMONICS MHARM(I) FOR ELEMENT I.
C
         IF(I.LT.NUMEL) THEN
            MHARM(I)=33
         ELSE
              MHARM(I)=101
         ENDIF
C
  500 CONTINUE
C
      RETURN
      END
C*********************************************************************
      SUBROUTINE FICHYD
C*********************************************************************
C
C THIS SUBROUTINE CALCULATES THE INITIAL VALUES OF FOURIER COEFFICIENTS
C
C*********************************************************************
C
      PARAMETER (NEL=3,NNP=NEL+1,MH=200,NZMAX=MH*10,NQP=256)
C
      COMMON /BLOCK1/NUMNP,NUMEL,Z(NNP),MHARM(NEL),ZL,ZLE(NEL)
      COMMON /BLOCK6/PSII(NNP),PSI(NNP),HI(NNP),H(NNP),PSICON
      COMMON /BLOCK7/AOLD(NEL,0:MH),ANEW(NEL,0:MH)
      COMMON /BLOCK8/XF(3)
      COMMON /BLOCK9/CM(MH,MH),RHS(MH),X(MH),XOLD(MH),NR,NC
      COMMON /BLOCK10/DXQ(NQP),DWQ(NQP),HQP(NEL,NQP),DHDZ(NEL,NQP)
      COMMON /BLOCK13/HIH(NEL,MH),ZGHIH(NEL,MH),HZ(NEL,MH)
C
      DOUBLE PRECISION DXQ,DWQ,Z
C
C*********************************************************************
C
C MHARM(I) = NUMBER OF HARMONICS + 1 FOR ELEMENT I
C ZGLOBAL  = GLOBAL COORDINATE Z
C ZLOCAL   = LOCAL COORDINATE Z
C DELZG    = INCREMENT FOR Z IN GLOBAL COORDINATE SYSTEM
C DELZL    = INCREMENT FOR Z IN LOCAL COORDINATE SYSTEM
C Z(I)     = BEGINING NODE FOR ELEMENT I (GLOBAL SYSTEM)
C Z(I+1)   = ENDING NODE FOR ELEMENT I (GLOBAL SYSTEM)
C ZLE      = LENGTH OF ELEMENT I
C HIH(I,J) = INITIAL HYDROSTATIC HEAD DISTRIBUTION FOR STEADY-STATE FLOW
C
      DPI=3.14159265358979323846264 3
      WRITE(*,15)
   15 FORMAT(5X,'ENTER FICHYD')
C
C  CREATE THE COEFFICIENT MATRIX (CM)
C
      DO 100 I=1,NUMEL
C
      CALL COEFFMAT (I)
C
C  CALCULATE THE RHS ARRAY
C
```

```
      NDIV=MHARM(I)/2
      HO=HIH(I,1)
      HL=HIH(I,NDIV+1)
      WRITE(9,25)HO,HL
   25 FORMAT(2F10.6)
C
      RHS(1)=(HO+HL)/2.
      RHS(2)=(HO-HL)/2.
      RHS(3)=0.
C
      DO 50 J=2,NDIV
   50 RHS(J+2)=HIH(I,J)
C
C SOLVE FOR FOURIER COEFFICIENTS
C
      CALL GAUSS (CM,RHS,NR,X)
C
C SOLUTION RETURN IN X-SET AOLD=X
C
      AOLD(I,0)=X(1)
      AOLD(I,1)=X(2)
C
      DO 60 J=2,MHARM(I),2
   60 AOLD(I,J)=X(J/2+2)
C
C CALCULATE ODD COEFFICIENTS
C
      DO 70 J=3,MHARM(I)-1,2
   70 AOLD(I,J)=AOLD(I,1)/J/J
  100 CONTINUE
C
C SAVE FOURIER COEFFICIENTS IN ANEW
C
      DO 110 I=1,NUMEL
      DO 110 J=1,MHARM(I)
  110 ANEW(I,J-1)=AOLD(I,J-1)
C
      DO 80 I=1,NUMEL
      DO 80 J=1,MHARM(I)
      WRITE(9,90)AOLD(I,J-1)
   90 FORMAT(F10.6)
   80 CONTINUE
C
      CALL HOUT
C
      WRITE(*,30)
   30 FORMAT(5X,'EXIT FICHYD')
C
      RETURN
      END
C
C*************************************************************************
      SUBROUTINE HOUT
C*************************************************************************
C
C THIS SUBROUTINE PRINTS THE HYDRAULIC HEADS AND PRESSURE HEADS AT THE
C SPECIFIED SPACE INTERVAL.
```

```
C
*************************************************************************
C
      PARAMETER (NEL=3,NNP=NEL+1,MH=200,NZMAX=MH*10,NQP=256)
C
      COMMON /BLOCK1/NUMNP,NUMEL,Z(NNP),MHARM(NEL),ZL,ZLE(NEL)
      COMMON /BLOCK5/TIME,DTIME,TOTIME
      COMMON /BLOCK7/AOLD(NEL,0:MH),ANEW(NEL,0:MH)
      COMMON /BLOCK10/DXQ(NQP),DWQ(NQP),HQP(NEL,NQP),DHDZ(NEL,NQP)
      COMMON /BLOCK13/HIH(NEL,MH),ZGHIH(NEL,MH),HZ(NEL,MH)
C
      DOUBLE PRECISION DPI,DXQ,DWQ,Z
C
C
*************************************************************************
****
C
      DPI=3.14159265358979323846264 3
      WRITE(*,5)
    5 FORMAT(5X,'ENTER HOUT')
C
      WRITE(8,10)TIME
   10 FORMAT(/,20X,'HYDRAULIC HEAD AND PRESSURE HEAD AT TIME=',F5.3,
     +'(DAYS)'/,20X,'_____________________________________________________'
     +,'___')
C
      WRITE(8,20)
   20 FORMAT(/,3(7X,'ZH',8X,'HEAD',6X,'PSI',3X))
C
      DO 200 I=1,NUMEL
C
C  CALCULATE HYDRAULIC HEAD (HZ)
C
      NDIV=MHARM(I)/2
C
      DO 150 K=1,NDIV+1
      ZLOCAL=(ZGHIH(I,K)-Z(I))/ZLE(I)
C
      HZ(I,K)=ANEW(I,0)
      DO 100 J=1,MHARM(I)-1
      HZ(I,K)=HZ(I,K)+ANEW(I,J)*COS(J*DPI*ZLOCAL)
  100 CONTINUE
      IF(HZ(I,K).LT.0.)HZ(I,K)=0.
  150 CONTINUE
C
C     WRITE(8,25)(ZGHIH(I,J),HIH(I,J),J=1,NDIV+1)
C   25      FORMAT(8F10.6)
      WRITE(9,30)(ZGHIH(I,J),HIH(I,J),HZ(I,J),J=1,NDIV+1)
   30 FORMAT(3F10.6)
  200 CONTINUE
C
C     CALL WATBAL (2)
C
      WRITE(*,15)
   15 FORMAT(5X,'EXIT HOUT')
      RETURN
      END
```

```
C
C*********************************************************************
      SUBROUTINE BANNER
C*********************************************************************
C
C THIS SUBROUTINE DISPLAYS THE PROGRAM BANNER
C
*********************************************************************
      WRITE(*,100)
      WRITE(7,100)
  100 FORMAT(////////,
     +12X,'*************** Colorado State University ***************'/,
     +12X,'*             Department of Civil Engineering           *'/,
     +12X,'*                                                       *'/,
     +12X '*          GROUNDWATER/ENVIRONMENTAL HYDROGEOLOGY       *'/,
     +12X '*                    PROGRAM                            *'/,
     +12X '*               FFUSAT MODEL (v 1.0)                    *'/,
     +12X '*                                                       *'/,
     +12X,'*                                                       *'/,
     +12X,'*             A FOURIER FINITE ELEMENT MODEL            *'/,
     +12X,'*                       FOR                             *'/,
     +12X,'*            UNSATURATED FLOW IN POROUS MEDIA           *'/,
     +12X,'*                                                       *'/,
     +12X,'*                    WRITTEN BY:                        *'/,
     +12X,'*          MOHAMED ALI SALEH DHBEEL AL-DHAMARI          *'/,
     +12X,'*                                                       *'/,
     +12X,'*********************************************************'/,
     +12X,///////,
     +12X,'             (Press ENTER to continue)'\)
C

      READ(*,*)
C
      RETURN
      END

*********************************************************************
*********************************************************************
C
      SUBROUTINE AUXEQS (PSI,ISTYPE,RELP,THETA,SE,DSDPSI)
C
C*********************************************************************
C  FOR SOIL TYPE "I", GIVEN PSI THIS SUBROUTINE CALCULATES:
C  (1) EFFECTIVE SATURATION.(SE)
C  (2) RELATIVE PERMEABILITY.(krw)
C  (3) VOLUMETERIC WATER CONTENT.(THETA)
C
C*********************************************************************
C
      PARAMETER (NEL=3,NNP=NEL+1,MH=200,NZMAX=MH*10,NQP=256)
C
      COMMON /BLOCK1/NUMNP,NUMEL,Z(NNP),MHARM(NEL),ZL,ZLE(NEL)
      COMMON /BLOCK3/JFLAG(20),KSAT(20),PHI(20),THETAR(20),
     +      PSIR(20),BCLAMDA(20),BCPB(20),VGALPHA(20),VGM(20),VGN(20)
      COMMON /BLOCK4/IDELEM(NEL)
      COMMON /BLOCK7/AOLD(NEL,0:MH),ANEW(NEL,0:MH)
C
```

```
      DOUBLE PRECISION Z
C
          IF(JFLAG(ISTYPE).EQ.1)THEN
C
C  LINEAR RELATIONSHIP
C
            IF(PSI.LT. 0.)THEN
               SE=1.-(PSI/PSIR(ISTYPE))
               THETA=(SE*(PHI(ISTYPE)-THETAR(ISTYPE))+THETAR(ISTYPE))
               RELP= SE
            ELSEIF(PSI.GE. 0.)THEN
               SE=1.
                 THETA=PHI(ISTYPE)
                 RELP=1.
            ENDIF
         ELSEIF(JFLAG(ISTYPE).EQ.2)THEN
C
C  BROOKS-COREY RELATIONSHIP
C
            IF(PSI.LT. BCPB(ISTYPE))THEN
               SE=(BCPB(ISTYPE)/PSI)**BCLAMDA(ISTYPE)
               THETA=(SE*(PHI(ISTYPE)-THETAR(ISTYPE))+THETAR(ISTYPE))
               RELP=(PSI/BCPB(ISTYPE))**(-2-3*BCLAMDA(ISTYPE))
            ELSEIF(PSI.GE.BCPB(ISTYPE))THEN
               SE=1.
                 THETA=PHI(ISTYPE)
               RELP=1.
            ENDIF
         ELSEIF(JFLAG(ISTYPE).EQ.3)THEN
C
C  VAN GENUCHTEN RELATIONSHIP
C
            IF(PSI.LT. 0.)THEN
               A=(ABS(VGALPHA(ISTYPE)*PSI))**VGN(ISTYPE)
               SE=1./(1.+A)**VGM(ISTYPE)
               THETA=(SE*(PHI(ISTYPE)-THETAR(ISTYPE))+THETAR(ISTYPE))
               B= (1.-(SE**(1./VGM(ISTYPE))))**VGM(ISTYPE)
               RELP = SE**0.5 *(1-B)**2.
               DSDPSI=VGM(ISTYPE)*VGN(ISTYPE)*A*
     +         (SE)**((VGM(ISTYPE)+1.)/VGM(ISTYPE))/ABS(PSI)
            ELSEIF(PSI.GE.0.)THEN
               SE=1.
               THETA=PHI(ISTYPE)
               RELP=1.
               DSDPSI=0.
            ENDIF
         ENDIF
C
      RETURN
      END
C*****************************************************************************
C*****************************************************************************
      SUBROUTINE GAUSS(C,D,N,X)
C*****************************************************************************
C
C THIS SUBROUTINE SOLVES A SET OF SIMULATANEOUS EQUATION CD=X BY GUASS
C ELIMINATION AND RETURN THE RESULT IN MATRIX [X]
```

```
C
      PARAMETER (NEL=3,NNP=NEL+1,MH=200,NZMAX=MH*10,NQP=256)
C
      DIMENSION C(MH,MH),D(MH),X(MH)
C
C UPPER TRAINGULAR MATRIX [C]
C
      DO 30 I=1,N-1
      DO 20 J=I+1,N
      CALL PIVOTCHANGE (C,D,N,I)
      FCTR=C(J,I)/C(I,I)
      D(J)=D(J)-FCTR*D(I)
      DO 10 K=1,N
   10 C(J,K)=C(J,K)-FCTR*C(I,K)
   20 CONTINUE
   30 CONTINUE
C
C SOLVE FOR [X] BY BACK SUBSTITUTION
C
      X(N)=D(N)/C(N,N)
      DO 50 I=N-1,1,-1
         SUM=0.
      DO 40 J=I+1,N
   40 SUM=SUM+X(J)*C(I,J)
   50 X(I)=(D(I)-SUM)/C(I,I)
C
      RETURN
      END
C*********************************************************************
C*********************************************************************

      SUBROUTINE PIVOTCHANGE(C,D,N,IROW)
C*********************************************************************
C
C THIS SUBROUTINE FINDS A ROW WITH LARGEST ABS PIVOT VALUE IN MATRIX [D]
C AND REPLACE IT WITH THE LOWEST ABSOLUTE VALUE.
C
      PARAMETER (NEL=3,NNP=NEL+1,MH=200,NZMAX=MH*10,NQP=256)
C
      DIMENSION C(MH,MH),D(MH),TEMP(MH)
C
C
C FIND THE ABSOLUTE PIVOT VALUE IN MATRIX [C]
C
      PIVOT=0.
      DO 100 J=IROW,N
          IF(ABS(C(J,IROW)).GE.PIVOT)THEN
             PIVOT=ABS(C(J,IROW))
             JROW=J
         ENDIF
  100 CONTINUE
C
C INTERCHANGE IROW AND JROW IN MATRIX [C]
C
              DO 150 J=1,N
                 TEMP(J)=C(IROW,J)
                 C(IROW,J)=C(JROW,J)
```

```
                  C(JROW,J)=TEMP(J)
  150 CONTINUE
C
C INTERCHANGE IROW AND JROW IN MATRIX [D]
C
                  DUMP=D(IROW)
                  D(IROW)=D(JROW)
                  D(JROW)=DUMP
C
      RETURN
      END
C*************************************************************************
C*************************************************************************
      SUBROUTINE QUAD (Z1,H1,Z2,H2,DHDZ1,A,B,C)
C*************************************************************************
C
C THIS SUBROUTINE CALCULATES THE COEFFICIENTS FOR A QUADRATIC EQUATION
C POINTS (Z1,H1) AND (Z2,H2) AND THE DERIVATIVE (SLOPE) DHDZ AT POINT1.
C POINT1= BEGINNING POINT
C POINT2= ENDING POINT
C Z1= ZLOCAL FOR POINT1
C H1= INITIAL HEAD FOR POINT1
C Z2= ZLOCAL FOR POINT2
C H2= INITIAL HEAD FOR POINT2
C DHDZ1= SLOPE OF DH/DZ AT POINT1
C
      DOUBLE PRECISION H2
C
C
C CALCULATE COEFFICIENTS OF THE QUADRATIC EQUATION
C
C     WRITE (*,10) Z1,H1,Z2,H2,DHDZ1
C   10      FORMAT (6F10.5)
      D=(H2-H1)/(Z2-Z1)
      C=(D-DHDZ1)/(Z2-Z1)
      B=D-(Z2+Z1)*C
      A=H1-(B*Z1)-(C*Z1**2)
C      WRITE(*,*)A,B,C
C
      RETURN
      END
C*************************************************************************
C*************************************************************************
      SUBROUTINE CUBIC (Z1,H1,Z2,H2,DH1DZ,DH2DZ,A,B,C,D)
C*************************************************************************
C
C THIS SUBROUTINE CALCULATES THE COEFFICIENTS FOR A CUBIC EQUATION
C POINTS (Z1,H1) AND (Z2,H2) AND THE SLOPE DERIVATIVES DH1DZ AND DH2DZ
C POINT1= BEGINNING POINT
C POINT2= ENDING POINT
C Z1= ZLOCAL FOR POINT1
C H1= INITIAL HEAD FOR POINT1
C Z2= ZLOCAL FOR POINT2
C H2= INITIAL HEAD FOR POINT2
C DH1DZ= SLOPE OF DH/DZ AT POINT1
C DH2DZ= SLOPE OF DH/DZ AT POINT2
C
```

```
C     DOUBLE PRECISION H2
C
C CALCULATE COEFFICIENTS OF THE CUBIC EQUATION
C
      E=(Z2-Z1)
      F=((H2-H1)/E**2)-(DH1DZ/E)
      G=(DH2DZ/E)-((H2-H1)/E**2)
      D=(G-F)/E
C
      P=2*Z2+Z1
      R=2*Z1+Z2
      C=(G*R-F*P)/(R-P)
C
      B=DH2DZ-3.*D*Z2**2-2.*C*Z2
      A=H2-D*Z2**3-C*Z2**2-B*Z2
C
      RETURN
      END
C*********************************************************************
C*********************************************************************
      SUBROUTINE COEFFMAT (I)
C*********************************************************************
C
C THIS SUBROUTINE CALCULATES THE COEFFICIENTS MATRIX (CM)
C
C*********************************************************************
C
      PARAMETER (NEL=3,NNP=NEL+1,MH=200,NZMAX=MH*10,NQP=256)
C
      COMMON /BLOCK1/NUMNP,NUMEL,Z(NNP),MHARM(NEL),ZL,ZLE(NEL)
      COMMON/ BLOCK9/CM(MH,MH),RHS(MH),X(MH),XOLD(MH),NR,NC
C
      DOUBLE PRECISION DPI,Z
C
      DPI=3.14159265358979323846264 3
      WRITE(*,15)
   15 FORMAT(5X,'ENTER COFFMAT')
C
C  I=ELEMENT NUMBER
C  M=HARMONICS M
C  N=HARMONICS N
C  MHARM(I)=NUMBER OF HARMONICS FOR ELEMENT I
C
      NR=(MHARM(I)/2)+2
      NC=NR
C
C INITIALIZE MATRIX [CM] AND [RHS]
C
      DO 20 J=1,NR
      RHS(J)=0.
      DO 20 K=1,NC
   20 CM(J,K)=0.
C
C CONSTRUCT COEFFICIENT MATRIX [CM]
C
C
C CONSTRUCT FIRST THREE ROWS OF [CM] FOR ELEMENT (I)
```

```
C
      SUM=1.
      DO 35 J=3,MHARM(I)-1,2
   35 SUM=SUM+1./(J**2)
C
      CM(1,1)=+1.
      CM(2,2)=SUM
C
      DO 45 J=3,NR
   45 CM(3,J)=+1.
C
C CONSTRUCT THE REMAINING ROWS OF [CM] FOR ELEMENT (I)
C
      NDIV=MHARM(I)/2
      DELZG=ZLE(I)/NDIV/1.01
      ZGLOBAL=Z(I)+DELZG
C
      DO 55 K=4,NR
      ZLOCAL=(ZGLOBAL-Z(I))/ZLE(I)
C     WRITE(9,99)ZGLOBAL,ZLOCAL
C   99      FORMAT(2F10.7)
C
      SUM=0.
      DO 60 J=1,MHARM(I)-1,2
   60 SUM=SUM+(1./J**2)*COS(J*DPI*ZLOCAL)
C
      CM(K,1)=+1
      CM(K,2)=SUM
      KK=0
      DO 65 J=3,NC
      KK=KK+2
   65 CM(K,J)=COS(KK*DPI*ZLOCAL)
      ZGLOBAL=ZGLOBAL+DELZG
   55 CONTINUE
C
C DEBUG-WRITE MATRIX [CM]
C
C     DO 110 J=1,NR
C     WRITE(8,95)J
C   95 FORMAT(/'LINE NO.=',I3)
C     WRITE(8,120)(CM(J,K),K=1,NC)
C  120      FORMAT(18F10.6)
C  110      CONTINUE
C
      WRITE(*,25)
   25 FORMAT(5X,'EXIT COFFMAT')
C
      RETURN
      END
C*********************************************************************
C*********************************************************************
      SUBROUTINE QUADPTS
C*********************************************************************
C
C THIS SUBROUTINE TABULATES THE ABSCISSAS DXQ AND WEIGHT FACTORS DWQ
C*********************************************************************
C
```

```
      PARAMETER (NEL=3,NNP=NEL+1,MH=200,NZMAX=MH*10,NQP=256)
C
      COMMON /BLOCK10/DXQ(NQP),DWQ(NQP),HQP(NEL,NQP),DHDZ(NEL,NQP)
      DOUBLE PRECISION DXQ,DWQ
      WRITE(*,10)
   10 FORMAT(5X,'ENTER QUADPTS')
C
      DXQ(1)=-0.99995605001899223073480121016 8D+00
      DXQ(2)=-0.99976843740926318610487858 5512D+00
      DXQ(3)=-0.99943093746626140824085418 1711D+00
      DXQ(4)=-0.99894352584340885655502628 9786D+00
      DXQ(5)=-0.99830626647300644405550048 2418D+00
      DXQ(6)=-0.99751925275672082756340875 8638D+00
      DXQ(7)=-0.99658260202338154043050442 4270D+00
      DXQ(8)=-0.99549645448109635659264705 2789D+00
      DXQ(9)=-0.99426097292240966496287754 8324D+00
      DXQ(10)=-0.99287634260882211714353380 2322D+00
      DXQ(11)=-0.99134277120758308692218851 0456D+00
      DXQ(12)=-0.98966048874506521831924374 7264D+00
      DXQ(13)=-0.98782974756486060891648771 2732D+00
      DXQ(14)=-0.98585082228612595647924512 3868D+00
      DXQ(15)=-0.98372400976031549616668611 7821D+00
      DXQ(16)=-0.98144962902546440576930311 4157D+00
      DXQ(17)=-0.97902802125762203882423803 5283D+00
      DXQ(18)=-0.97645954971923415562101065 1750D+00
      DXQ(19)=-0.97374459970437040526607855 7889D+00
      DXQ(20)=-0.97088357848074302932092329 9728D+00
      DXQ(21)=-0.96787691522848945490900377 7710D+00
      DXQ(22)=-0.96472506097570643093261227 9685D+00
      DXQ(23)=-0.96142848853073214400640746 8736D+00
      DXQ(24)=-0.95798769241117812936579044 6493D+00
      DXQ(25)=-0.95440318876971624176444794 1140D+00
      DXQ(26)=-0.95067551531662827636385212 4711D+00
      DXQ(27)=-0.94680523123912748137205172 5052D+00
      DXQ(28)=-0.94279291711746244318307614 0631D+00
      DXQ(29)=-0.93863917483781480498192608 4106D+00
      DXQ(30)=-0.93434462750200309429247654 2768D+00
      DXQ(31)=-0.92990991933400564118024555 0881D+00
      DXQ(32)=-0.92533571558331620287273029 1856D+00
      DXQ(33)=-0.92062270242514649550504708 4323D+00
      DXQ(34)=-0.91577158685749038452666962 6561D+00
      DXQ(35)=-0.91078309659506501189090720 3179D+00
      DXQ(36)=-0.90565797996014464708268190 5800D+00
      DXQ(37)=-0.90039700577030354477161999 5416D+00
      DXQ(38)=-0.89500096322308457744122280 0002D+00
      DXQ(39)=-0.88947066177761088882867658 0534D+00
      DXQ(40)=-0.88380693103315828485982618 1734D+00
      DXQ(41)=-0.87801062060470654398643486 7952D+00
      DXQ(42)=-0.87208259999548828913004589 9567D+00
      DXQ(43)=-0.86602375846655451929751543 1123D+00
      DXQ(44)=-0.85983500490337635069617309 7846D+00
      DXQ(45)=-0.85351726767950296507303553 5714D+00
      DXQ(46)=-0.84707149451729620718707243 2785D+00
      DXQ(47)=-0.84049865234576271389506799 7292D+00
      DXQ(48)=-0.83379972715550489434844389 9986D+00
      DXQ(49)=-0.82697572385081251428909289 8047D+00
      DXQ(50)=-0.82002766609891706740347808 1822D+00
```

```
DXQ(51)=-0.81295659617643154313641043849 5D+00
DXQ(52)=-0.80576357481299862325738910468 7D+00
DXQ(53)=-0.79844968103217075878254286054 9D+00
DXQ(54)=-0.79101601198954599454670749798 5D+00
DXQ(55)=-0.78346368280818382075067020673 9D+00
DXQ(56)=-0.77579382641132573913205257458 2D+00
DXQ(57)=-0.76800759335244563597589060072 9D+00
DXQ(58)=-0.76010615164265545494190679754 0D+00
DXQ(59)=-0.75209068657549205958752972836 8D+00
DXQ(60)=-0.74396240054911156845568312157 5D+00
DXQ(61)=-0.73572251288591783462037285023 5D+00
DXQ(62)=-0.72737225964965212658689439814 8D+00
DXQ(63)=-0.71891289345997144837263985024 8D+00
DXQ(64)=-0.71034568330454331339456631697 5D+00
DXQ(65)=-0.70167191434868515940608354988 3D+00
DXQ(66)=-0.69289288774257696010534160635 6D+00
DXQ(67)=-0.68400992042607595312487710735 2D+00
DXQ(68)=-0.67502434493116276385591873153 8D+00
DXQ(69)=-0.66593750918204855990640841387 7D+00
DXQ(70)=-0.65675077629297322188750023539 6D+00
DXQ(71)=-0.64746552436372486261701622882 9D+00
DXQ(72)=-0.63808314627291136866868858338 3D+00
DXQ(73)=-0.62860504946901497543220986776 7D+00
DXQ(74)=-0.61903265575926121943096763429 1D+00
DXQ(75)=-0.60936740109633393952231084466 0D+00
DXQ(76)=-0.59961073536296832173038822663 6D+00
DXQ(77)=-0.58976412215445430078578614391 2D+00
DXQ(78)=-0.57982903855908294492183171211 9D+00
DXQ(79)=-0.56980697493656875905766753546 5D+00
DXQ(80)=-0.55969943469448114513690743112 1D+00
DXQ(81)=-0.54950793406271855704242688438 8D+00
DXQ(82)=-0.53923400186605918112793624407 5D+00
DXQ(83)=-0.52887917929482226195147641396 4D+00
DXQ(84)=-0.51844501967367447622166170963 1D+00
DXQ(85)=-0.50793308822861603623192491903 3D+00
DXQ(86)=-0.49734496185218147711951238535 3D+00
DXQ(87)=-0.48668222886689035010362141941 0D+00
DXQ(88)=-0.47594648878698330639073752309 0D+00
DXQ(89)=-0.46513935207847931364557045048 1D+00
DXQ(90)=-0.45426243991758999877445520109 4D+00
DXQ(91)=-0.44331738394752735721692575276 0D+00
DXQ(92)=-0.43230582603374130995344107214 7D+00
DXQ(93)=-0.42122941801762382497681237185 2D+00
DXQ(94)=-0.41008982146871655000643357484 2D+00
DXQ(95)=-0.39888707435459127713463212760 D+00
DXQ(96)=-0.38762775619451558363798458808 4D+00
DXQ(97)=-0.37630865699871639028305571292 1D+00
DXQ(98)=-0.36493310782365401853346488052 0D+00
DXQ(99)=-0.35350281511296998953779015303 8D+00
DXQ(100)=-0.34201949352237163648072967415 7D+00
DXQ(101)=-0.33048486566241697622918703647 3D+00
DXQ(102)=-0.31890066184010627563168338215 6D+00
DXQ(103)=-0.30726861979931907625861025422 0D+00
DXQ(104)=-0.29559048446013561456378679563 3D+00
DXQ(105)=-0.28386800765708174179976576517 1D+00
DXQ(106)=-0.27210294787633660950524472329 6D+00
DXQ(107)=-0.26029706999194254197856085902 4D+00
```

```
DXQ(108)=-0.24845214500105444483324268886OD+00
DXQ(109)=-0.23656994975828401847750840886 9D+00
DXQ(110)=-0.22465226670913196714787831631 5D+00
DXQ(111)=-0.21270088362262595793704015346 3D+00
DXQ(112)=-0.20071759332312667006800073061 4D+00
DXQ(113)=-0.18870419342138882646150357223 4D+00
DXQ(114)=-0.17666248604490199740372181525 5D+00
DXQ(115)=-0.16459427756755384982928450755 5D+00
DXQ(116)=-0.15250137833865639537460684066 9D+00
DXQ(117)=-0.14038560241137588591302487564 1D+00
DXQ(118)=-0.12824876727060709474204959621 2D+00
DXQ(119)=-0.11609269356033280494073486388 7D+00
DXQ(120)=-0.10391920481050940363919689655 9D+00
DXQ(121)=-0.09173012716351955203114560059 78D+00
DXQ(122)=-0.07952728910023296590327027068 4566D+00
DXQ(123)=-0.06731252116571640024229028881 70D+00
DXQ(124)=-0.05508765569463398410456141642 48D+00
DXQ(125)=-0.04285452653637909838124228868 20D+00
DXQ(126)=-0.03061496877997902936627858132 14D+00
DXQ(127)=-0.01837081847881366511792629206 05D+00
DXQ(128)=-0.00612391237518952950117016496227D+00
DXQ(129)=0.00612391237518952950117016496227D+00
DXQ(130)=0.01837081847881366511792629206 05D+00
DXQ(131)=0.03061496877997902936627858132 14D+00
DXQ(132)=0.04285452653637909838124228868 20D+00
DXQ(133)=0.05508765569463398410456141642 48D+00
DXQ(134)=0.06731252116571640024229028881 70D+00
DXQ(135)=0.07952728910023296590327027068 4566D+00
DXQ(136)=0.09173012716351955203114560059 78D+00
DXQ(137)=0.10391920481050940363919689655 9D+00
DXQ(138)=0.11609269356033280494073486388 7D+00
DXQ(139)=0.12824876727060709474204959621 2D+00
DXQ(140)=0.14038560241137588591302487564 1D+00
DXQ(141)=0.15250137833865639537460684066 9D+00
DXQ(142)=0.16459427756755384982928450755 5D+00
DXQ(143)=0.17666248604490199740372181525 5D+00
DXQ(144)=0.18870419342138882646150357223 4D+00
DXQ(145)=0.20071759332312667006800073061 4D+00
DXQ(146)=0.21270088362262595793704015346 3D+00
DXQ(147)=0.22465226670913196714787831631 5D+00
DXQ(148)=0.23656994975828401847750840886 9D+00
DXQ(149)=0.24845214500105444483324268886OD+00
DXQ(150)=0.26029706999194254197856085902 4D+00
DXQ(151)=0.27210294787633660950524472329 6D+00
DXQ(152)=0.28386800765708174179976576517 1D+00
DXQ(153)=0.29559048446013561456378679563 3D+00
DXQ(154)=0.30726861979931907625861025422 0D+00
DXQ(155)=0.31890066184010627563168338215 6D+00
DXQ(156)=0.33048486566241697622918703647 3D+00
DXQ(157)=0.34201949352237163648072967415 7D+00
DXQ(158)=0.35350281511296998953779015303 8D+00
DXQ(159)=0.36493310782365401853346488052 0D+00
DXQ(160)=0.37630865699871639028305571292 1D+00
DXQ(161)=0.38762775619451558363798458808 4D+00
DXQ(162)=0.39888870743545912771346321276 0D+00
DXQ(163)=0.41008982146871655000643357484 2D+00
DXQ(164)=0.42122941801762382497681237185 2D+00
```

```
DXQ(165)=0.432305826033741309953441072147D+00
DXQ(166)=0.443317383947527357216925752760D+00
DXQ(167)=0.454262439917589998774455201094D+00
DXQ(168)=0.465139352078479313645570450481D+00
DXQ(169)=0.475946488786983306390737523090D+00
DXQ(170)=0.486682228866890350103621419410D+00
DXQ(171)=0.497344961852181477119512385353D+00
DXQ(172)=0.507933088228616036231924919033D+00
DXQ(173)=0.518445019673674476221661709631D+00
DXQ(174)=0.528879179294822261951476413964D+00
DXQ(175)=0.539234001866059181127936244075D+00
DXQ(176)=0.549507934062718557042426884388D+00
DXQ(177)=0.559699434694481145136907431121D+00
DXQ(178)=0.569806974936568759057667535465D+00
DXQ(179)=0.579829038559082944921831712119D+00
DXQ(180)=0.589764122154454300785786143912D+00
DXQ(181)=0.599610735362968321730388226636D+00
DXQ(182)=0.609367401096333939522310844660D+00
DXQ(183)=0.619032655759261219430967634291D+00
DXQ(184)=0.628605049469014975432209867767D+00
DXQ(185)=0.638083146272911368668688583383D+00
DXQ(186)=0.647465524363724862617016228829D+00
DXQ(187)=0.656750776292973221887500235396D+00
DXQ(188)=0.665937509182048559906408413877D+00
DXQ(189)=0.675024344931162763855918731538D+00
DXQ(190)=0.684009920426075953124877107352D+00
DXQ(191)=0.692892887742576960105341606356D+00
DXQ(192)=0.701671914348685159406083549883D+00
DXQ(193)=0.710345683304543313394566316975D+00
DXQ(194)=0.718912893459971448372639850248D+00
DXQ(195)=0.727372259649652126586894398148D+00
DXQ(196)=0.735722512885917834620372850235D+00
DXQ(197)=0.743962400549111568455683121575D+00
DXQ(198)=0.752090686575492059587529728368D+00
DXQ(199)=0.760106151642655454941906797540D+00
DXQ(200)=0.768007593352445635975890600729D+00
DXQ(201)=0.775793826411325739132052574582D+00
DXQ(202)=0.783463682808183820750670206739D+00
DXQ(203)=0.791016011989545994546707497985D+00
DXQ(204)=0.798449681032170758782542860549D+00
DXQ(205)=0.805763574812998623257389104687D+00
DXQ(206)=0.812956596176431543136410438495D+00
DXQ(207)=0.820027666098917067403478081822D+00
DXQ(208)=0.826975723850812514289092898047D+00
DXQ(209)=0.833799727155504894348443899986D+00
DXQ(210)=0.840498652345762713895067997292D+00
DXQ(211)=0.847071494517296207187072432785D+00
DXQ(212)=0.853517267679502965073035535714D+00
DXQ(213)=0.859835004903376350696173097846D+00
DXQ(214)=0.866023758466554519297515431123D+00
DXQ(215)=0.872082599995488289130045899567D+00
DXQ(216)=0.878010620604706543986434867952D+00
DXQ(217)=0.883806931033158284859826181734D+00
DXQ(218)=0.889470661777610888828676580534D+00
DXQ(219)=0.895000963223084577441222800002D+00
DXQ(220)=0.900397005770303544771619995416D+00
DXQ(221)=0.905657979960144647082681905800D+00
```

```
DXQ(222)=0.91078309659506501189090720317 9D+00
DXQ(223)=0.91577158685749038452666962656 1D+00
DXQ(224)=0.92062270242514649550504708432 3D+00
DXQ(225)=0.92533571558331620287273029185 6D+00
DXQ(226)=0.92990991933400564118024555088 1D+00
DXQ(227)=0.93434462750200309429247654276 8D+00
DXQ(228)=0.93863917483781480498192608410 6D+00
DXQ(229)=0.94279291711746244318307614063 1D+00
DXQ(230)=0.94680523123912748137205172505 2D+00
DXQ(231)=0.95067551531662827636385212471 1D+00
DXQ(232)=0.95440318876971624176444794114 0D+00
DXQ(233)=0.95798769241117812936579044649 3D+00
DXQ(234)=0.96142848853073214400640746873 6D+00
DXQ(235)=0.96472506097570643093261227968 5D+00
DXQ(236)=0.96787691522848945490900377771 0D+00
DXQ(237)=0.97088357848074302932092329972 8D+00
DXQ(238)=0.97374459970437040526607855788 9D+00
DXQ(239)=0.97645954971923415562101065175 0D+00
DXQ(240)=0.97902802125762203882423803528 3D+00
DXQ(241)=0.98144962902546440576930311415 7D+00
DXQ(242)=0.98372400976031549616668611782 1D+00
DXQ(243)=0.98585082228612595647924512386 8D+00
DXQ(244)=0.98782974756486060891648771273 2D+00
DXQ(245)=0.98966048874506521831924374726 4D+00
DXQ(246)=0.99134277120758308692218851045 6D+00
DXQ(247)=0.99287634260882211714353380232 2D+00
DXQ(248)=0.99426097292240966496287754832 4D+00
DXQ(249)=0.99549645448109635659264705278 9D+00
DXQ(250)=0.99658260202338154043050442427 0D+00
DXQ(251)=0.99751925275672082756340875863 8D+00
DXQ(252)=0.99830626647300644405550048241 8D+00
DXQ(253)=0.99894352584340885655502628978 6D+00
DXQ(254)=0.99943093746626140824085418171 1D+00
DXQ(255)=0.99976843740926318610487858551 2D+00
DXQ(256)=0.99995605001899223073480121016 8D+00
DWQ(1)=0.00011278901782227217551253887725 0D+00
DWQ(2)=0.00026253494429644590628745756249 9D+00
DWQ(3)=0.00041246325442617632843218583773 6D+00
DWQ(4)=0.00056234895403140980281523674759 3D+00
DWQ(5)=0.00071215416347332066690898915112 3D+00
DWQ(6)=0.00086185370142008903781409341625 1D+00
DWQ(7)=0.00101142439320844045260581284143D+00
DWQ(8)=0.00116084355756772472397059811346D+00
DWQ(9)=0.00131008868190250445783168042706D+00
DWQ(10)=0.00145913733331073320108838649958D+00
DWQ(11)=0.00160796713074932724244993956898D+00
DWQ(12)=0.00175655573633072999360691452952D+00
DWQ(13)=0.00190488085349971840441914117458D+00
DWQ(14)=0.00205292022796614317454878184920D+00
DWQ(15)=0.00220065164983991049968488341886D+00
DWQ(16)=0.00234805295632731201700646090869D+00
DWQ(17)=0.00249510203470370685083953543720D+00
DWQ(18)=0.00264177682542749056412082925158D+00
DWQ(19)=0.00278805532532770688057476107625D+00
DWQ(20)=0.00293391559082971664601232541422D+00
DWQ(21)=0.00307933574119933758320535283158D+00
DWQ(22)=0.00322429396179419815701071342688D+00
```

```
DWQ(23)=0.00336876850731555101201910624893D+00
DWQ(24)=0.00351273770505630733097105498443D+00
DWQ(25)=0.00365617995814250216938924130522D+00
DWQ(26)=0.00379907374876625799811701920824D+00
DWQ(27)=0.00394139764140883362772903498400D+00
DWQ(28)=0.00408313028605266840859977592119D+00
DWQ(29)=0.00422425042138153627235650490599D+00
DWQ(30)=0.00436473687796805668156842006212D+00
DWQ(31)=0.00450456858144789706864179231592D+00
DWQ(32)=0.00464372455568006031397909235247D+00
DWQ(33)=0.00478218392589269137293173404476D+00
DWQ(34)=0.00491992592181386566955877655649D+00
DWQ(35)=0.00505692988078684238755781607621D+00
DWQ(36)=0.00519317525086928093032875362964D+00
DWQ(37)=0.00532864159391593031708111147879D+00
DWQ(38)=0.00546330858864431027757053185663D+00
DWQ(39)=0.00559715603368291007755144525719D+00
DWQ(40)=0.00573016385060143717738441755539D+00
DWQ(41)=0.00586231208692265306066159880109D+00
DWQ(42)=0.00599358091911533822112769687036D+00
DWQ(43)=0.00612395065556793254238908118660D+00
DWQ(44)=0.00625340173954240127206364597525D+00
DWQ(45)=0.00638191475210788057037516427494D+00
DWQ(46)=0.00650947041505366026780989995138D+00
DWQ(47)=0.00663604959378106504459003835507D+00
DWQ(48)=0.00676163330017379878092786110788D+00
DWQ(49)=0.00688620269544632034671332377453D+00
DWQ(50)=0.00700973909296982262123443619368D+00
DWQ(51)=0.00713222396107539007167242298552D+00
DWQ(52)=0.00725363892583391378382913721418D+00
DWQ(53)=0.00737396577381234643757244069469D+00
DWQ(54)=0.00749318645480588335859976113332D+00
DWQ(55)=0.00761128308454565946161871961804D+00
DWQ(56)=0.00772823794738155563111019495784D+00
DWQ(57)=0.00784403349893971186681031615122D+00
DWQ(58)=0.00795865236875434835361316122694D+00
DWQ(59)=0.00807207736287349950094697480413D+00
DWQ(60)=0.00818429146643826993561976100372D+00
DWQ(61)=0.00829527784623522542517141255288D+00
DWQ(62)=0.00840501985322153575618030169816D+00
DWQ(63)=0.00851350102502249069383835478970D+00
DWQ(64)=0.00862070508840101430536883841018D+00
DWQ(65)=0.00872661596169880714033663221736D+00
DWQ(66)=0.00883121775724875002531827268512D+00
DWQ(67)=0.00893449478375820754840841708495D+00
DWQ(68)=0.00903643154866287368022777557239D+00
DWQ(69)=0.00913701276045080640200047221868D+00
DWQ(70)=0.00923622333095630268737871671386D+00
DWQ(71)=0.00933404837762326971246601448636D+00
DWQ(72)=0.00943047322573775274735276448237D+00
DWQ(73)=0.00952548341062928481182968575443D+00
DWQ(74)=0.00961906467984072785716216440052D+00
DWQ(75)=0.00971120299526627996424967049581D+00
DWQ(76)=0.00980188453525732782549880024998D+00
DWQ(77)=0.00989109569669582860263068380918D+00
DWQ(78)=0.00997882309703491012473394949458D+00
DWQ(79)=0.01006505357630638330946097892960+00
```

```
DWQ(80)=0.010149774199094865654634066041 9D+00
DWQ(81)=0.010232972256478219656954857160 4D+00
DWQ(82)=0.010314635267934015068260713997 3D+00
DWQ(83)=0.010394750983211728997101725205 3D+00
DWQ(84)=0.010473307384170403003569566926 7D+00
DWQ(85)=0.010550292686581481517533575536 0D+00
DWQ(86)=0.010625695341896561133961681801 4D+00
DWQ(87)=0.010699504038979785603048200582 7D+00
DWQ(88)=0.010771707705804626636653631927 2D+00
DWQ(89)=0.010842295511114795995293477058 4D+00
DWQ(90)=0.010911256866049039700796847787 6D+00
DWQ(91)=0.010978581425729570637988203447 7D+00
DWQ(92)=0.011044259090813901263517571043 6D+00
DWQ(93)=0.011108280009009843630460815450 7D+00
DWQ(94)=0.011170634576553449462710881938 3D+00
DWQ(95)=0.011231313439649668572656802083 4D+00
DWQ(96)=0.011290307495875509508367594121 3D+00
DWQ(97)=0.011347607895545491941625714296 8D+00
DWQ(98)=0.011403206043039185964847059552 3D+00
DWQ(99)=0.011457093598090639152334392298 0D+00
DWQ(100)=0.011509262477039497958586392438 7D+00
DWQ(101)=0.011559704854043635772668656950 3D+00
DWQ(102)=0.011608413162253105722084706677 2D+00
DWQ(103)=0.011655380094945242121298939730 3D+00
DWQ(104)=0.011700598606620740288189823359 2D+00
DWQ(105)=0.011744061914060550305376732758 6D+00
DWQ(106)=0.011785763497343426181690117626 9D+00
DWQ(107)=0.011825697100823977771160737958 1D+00
DWQ(108)=0.011863856734071078731904572907 6D+00
DWQ(109)=0.011900236672766489754287204237 1D+00
DWQ(110)=0.011934831459563562255873201696 4D+00
DWQ(111)=0.011967635904905893729007282669 9D+00
DWQ(112)=0.011998645087805811934536710070 8D+00
DWQ(113)=0.012027854356582571161267533497 6D+00
DWQ(114)=0.012055259329560149814347085327 4D+00
DWQ(115)=0.012080855895724544655975183975 5D+00
DWQ(116)=0.012104640215340463097757829736 4D+00
DWQ(117)=0.012126608720527321034718492204 5D+00
DWQ(118)=0.012146758115794459815559837664 2D+00
DWQ(119)=0.012165085378535502061307291838 9D+00
DWQ(120)=0.012181587759481772174047585032 0D+00
DWQ(121)=0.012196262783114713518180974196 5D+00
DWQ(122)=0.012209108248037240407514094370 7D+00
DWQ(123)=0.012220122227303969191708737227 3D+00
DWQ(124)=0.012229303068710278904146266083 2D+00
DWQ(125)=0.012236649395040158109242574767 4D+00
DWQ(126)=0.012242160104272800769728083260 2D+00
DWQ(127)=0.012245834369747920142463857549 9D+00
DWQ(128)=0.012247671640289755904070326489 7D+00
DWQ(129)=0.012247671640289755904070326489 7D+00
DWQ(130)=0.012245834369747920142463857549 9D+00
DWQ(131)=0.012242160104272800769728083260 2D+00
DWQ(132)=0.012236649395040158109242574767 4D+00
DWQ(133)=0.012229303068710278904146266083 2D+00
DWQ(134)=0.012220122227303969191708737227 3D+00
DWQ(135)=0.012209108248037240407514094370 7D+00
DWQ(136)=0.012196262783114713518180974196 5D+00
```

```
DWQ(137)=0.012181587759481772174047585032OD+00
DWQ(138)=0.012165085378535502061307291838 9D+00
DWQ(139)=0.012146758115794459815559837664 2D+00
DWQ(140)=0.012126608720527321034718492204 5D+00
DWQ(141)=0.012104640215340463097757829736 4D+00
DWQ(142)=0.012080855895724544655975183975 5D+00
DWQ(143)=0.012055259329560149814347085327 4D+00
DWQ(144)=0.012027854356582571161267533497 6D+00
DWQ(145)=0.011998645087805811934536710070 8D+00
DWQ(146)=0.011967635904905893729007282669 9D+00
DWQ(147)=0.011934831459563562255873201696 4D+00
DWQ(148)=0.011900236672766489754287204237 1D+00
DWQ(149)=0.011863856734071078731904572907 6D+00
DWQ(150)=0.011825697100823977771160737958 1D+00
DWQ(151)=0.011785763497343426181690117626 9D+00
DWQ(152)=0.011744061914060550305376732758 6D+00
DWQ(153)=0.011700598606620740288189823359 2D+00
DWQ(154)=0.011655380094945242121298939730 3D+00
DWQ(155)=0.011608413162253105722084706677 2D+00
DWQ(156)=0.011559704854043635772668656950 3D+00
DWQ(157)=0.011509262477039497958586392438 7D+00
DWQ(158)=0.011457093598090639152334392298 0D+00
DWQ(159)=0.011403206043039185964847059552 3D+00
DWQ(160)=0.011347607895545491941625714296 8D+00
DWQ(161)=0.011290307495875509508367594121 3D+00
DWQ(162)=0.011231313439649668572656802083 4D+00
DWQ(163)=0.011170634576553449462710881938 3D+00
DWQ(164)=0.011108280009009843630460815450 7D+00
DWQ(165)=0.011044259090813901263517571043 6D+00
DWQ(166)=0.010978581425729570637988203447 7D+00
DWQ(167)=0.010911256866049039700796847787 6D+00
DWQ(168)=0.010842295511114795995293477058 4D+00
DWQ(169)=0.010771707705804626636653631927 2D+00
DWQ(170)=0.010699504038979785603048200582 7D+00
DWQ(171)=0.010625695341896561133961681801 4D+00
DWQ(172)=0.010550292686581481517533575536 0D+00
DWQ(173)=0.010473307384170403003569566926 7D+00
DWQ(174)=0.010394750983211728997101725205 3D+00
DWQ(175)=0.010314635267934015068260713997 3D+00
DWQ(176)=0.010232972256478219656954857160 4D+00
DWQ(177)=0.010149774199094865654634066041 9D+00
DWQ(178)=0.010065053576306383309460978929 6D+00
DWQ(179)=0.009978823097034910124733949494 58D+00
DWQ(180)=0.009891095696695828602630683809 18D+00
DWQ(181)=0.009801884535257327825498800249 98D+00
DWQ(182)=0.009711202995266279964249670495 81D+00
DWQ(183)=0.009619064679840727857162164400 52D+00
DWQ(184)=0.009525483410629284811829685754 43D+00
DWQ(185)=0.009430473225737752747352764482 37D+00
DWQ(186)=0.009334048377623269712466014486 36D+00
DWQ(187)=0.009236223330956302687378716713 86D+00
DWQ(188)=0.009137012760450806402000472218 68D+00
DWQ(189)=0.009036431548662873680227775572 39D+00
DWQ(190)=0.008934449478375820754840841708 495D+00
DWQ(191)=0.008831217757248750025318272685 12D+00
DWQ(192)=0.008726615961698807140336632217 36D+00
DWQ(193)=0.008620705088401014305368838410 18D+00
```

```
DWQ(194)=0.008513501025022490693838354789700+00
DWQ(195)=0.008405019853221535756180301698160+00
DWQ(196)=0.008295277846235225425171412552880+00
DWQ(197)=0.008184291466438269935619761003720+00
DWQ(198)=0.008072077362873499500946974804130+00
DWQ(199)=0.007958652368754348353613161226940+00
DWQ(200)=0.007844033498939711866810316151220+00
DWQ(201)=0.007728237947381555631110194957840+00
DWQ(202)=0.007611283084545659461618719618040+00
DWQ(203)=0.007493186454805883358599761133320+00
DWQ(204)=0.007373965773812346437572440694690+00
DWQ(205)=0.007253638925833913783829137214180+00
DWQ(206)=0.007132223961075390071672422985520+00
DWQ(207)=0.007009739092969822621234436193680+00
DWQ(208)=0.006886202695446320346713323774530+00
DWQ(209)=0.006761633300173798780927861107880+00
DWQ(210)=0.006636049593781065044590038355070+00
DWQ(211)=0.006509470415053660267809899951380+00
DWQ(212)=0.006381914752107880570375164274940+00
DWQ(213)=0.006253401739542401272063645975250+00
DWQ(214)=0.006123950655567932542389081186600+00
DWQ(215)=0.005993580919115338221127696870360+00
DWQ(216)=0.005862312086922653060661598801090+00
DWQ(217)=0.005730163850601437177384417555390+00
DWQ(218)=0.005597156033682910077551445257190+00
DWQ(219)=0.005463308588644310277570531856630+00
DWQ(220)=0.005328641593915930317081111478790+00
DWQ(221)=0.005193175250869280930328753629640+00
DWQ(222)=0.005056929880786842387557816076210+00
DWQ(223)=0.004919925921813865669558776556490+00
DWQ(224)=0.004782183925892691372931734044760+00
DWQ(225)=0.004643724555680060313979092352470+00
DWQ(226)=0.004504568581447897068641792315920+00
DWQ(227)=0.004364736877968056681568420062120+00
DWQ(228)=0.004224250421381536272356504905990+00
DWQ(229)=0.004083130286052668408599775921190+00
DWQ(230)=0.003941397641408833627729034984000+00
DWQ(231)=0.003799073748766257998117019208240+00
DWQ(232)=0.003656179958142502169389241305220+00
DWQ(233)=0.003512737705056307330971054984430+00
DWQ(234)=0.003368768507315551012019106248930+00
DWQ(235)=0.003224293961794198157010713426880+00
DWQ(236)=0.003079335741199337583205352831580+00
DWQ(237)=0.002933915590829716646012325414220+00
DWQ(238)=0.002788055325327706880574761076250+00
DWQ(239)=0.002641776825427490564120829251580+00
DWQ(240)=0.002495102034703706850839535437200+00
DWQ(241)=0.002348052956327312017006460908690+00
DWQ(242)=0.002200651649839910499684883418860+00
DWQ(243)=0.002052920227966143174548781849200+00
DWQ(244)=0.001904880853499718404419141174580+00
DWQ(245)=0.001756555736330729993606914529520+00
DWQ(246)=0.001607967130749327242449939568980+00
DWQ(247)=0.001459137333310733201088386499580+00
DWQ(248)=0.001310088681902504457831680427060+00
DWQ(249)=0.001160843557567724723970598113460+00
DWQ(250)=0.001011424393208440452605812841430+00
```

```
      DWQ(251)=0.00086185370142008903781409341625 1D+00
      DWQ(252)=0.00071215416347332066690898915112 3D+00
      DWQ(253)=0.00056234895403140980281523674759 3D+00
      DWQ(254)=0.00041246325442617632843218583773 6D+00
      DWQ(255)=0.00026253494429644590628745756249 9D+00
      DWQ(256)=0.00011278901782227217551253887725 0D+00
C
C WRITE THE QUADPOINTS DATA IN THE ECHO FILE
C
      DO 20 I=1,NQP
      WRITE(6,15) DXQ(I), DWQ(I)
   15 FORMAT(1X,2(D40.31,2X))
   20 CONTINUE
C
      WRITE(*,25)
   25 FORMAT(5X,'EXIT QUADPTS')

      RETURN
      END
C*********************************************************************
C*********************************************************************
      SUBROUTINE HQUAD
C*********************************************************************
C
C THIS SUBROUTINE APPLYS QUADRATURE POINTS TO CALCULATE THE HYDRAULIC
C HEAD
C
C*********************************************************************
C
      PARAMETER (NEL=3,NNP=NEL+1,MH=200,NZMAX=MH*10,NQP=256)
C
      COMMON /BLOCK1/NUMNP,NUMEL,Z(NNP),MHARM(NEL),ZL,ZLE(NEL)
      COMMON /BLOCK3/JFLAG(20),SATK(20),PHI(20),THETAR(20),
     +      PSIR(20),BCLAMDA(20),BCPB(20),VGALPHA(20),VGM(20),VGN(20)
      COMMON /BLOCK4/IDELEM(NEL)
      COMMON /BLOCK6/PSII(NNP),PSI(NNP),HI(NNP),H(NNP),PSICON
      COMMON /BLOCK7/AOLD(NEL,0:MH),ANEW(NEL,0:MH)
      COMMON /BLOCK8/XF(3)
      COMMON /BLOCK9/CM(MH,MH),RHS(MH),X(MH),XOLD(MH),NR,NC
      COMMON /BLOCK10/DXQ(NQP),DWQ(NQP),HQP(NEL,NQP),DHDZ(NEL,NQP)
      COMMON /BLOCK13/HIH(NEL,MH),ZGHIH(NEL,MH),HZ(NEL,MH)
C
      DOUBLE PRECISION DPI,DXQ,DWQ,DZQ,Z
C
      DPI=3.14159265358979323846264 3
C
      DO 500 I=1,NUMEL
C
         A=Z(I)
         B=Z(I+1)
         NDIV=MHARM(I)/2
C
      DO 500 K=1,NQP
         DZQ=(B+A)/2.+(B-A)/2.*DXQ(K)
         DO 400 J=1,NDIV+1
C
         IF(DZQ.LT.ZGHIH(I,J)) THEN
```

```
C
C CALCULATE HEAD AT LOWER LOCATION
C
            ZHQL=ZGHIH(I,J-1)
            ZLOCAL=(ZHQL-Z(I))/ZLE(I)
C
            HQL=AOLD(I,0)
            DO 300 JL=1,MHARM(I)-1
  300       HQL=HQL+AOLD(I,JL)*COS(JL*DPI*ZLOCAL)
C
C CALCULATE HEAD AT UPPER LOCATION
C
            ZHQU=ZGHIH(I,J)
            ZLOCAL=(ZHQU-Z(I))/ZLE(I)
C
            HQU=AOLD(I,0)
            DO 310 JU=1,MHARM(I)-1
  310       HQU=HQU+AOLD(I,JU)*COS(JU*DPI*ZLOCAL)
C
            DENOM=ZGHIH(I,J)-ZGHIH(I,J-1)
            DELZ=DZQ-ZGHIH(I,J-1)
            IF(HQP(I,K).LT.0.)HQP(I,K)=0.
            IF(HQP(I,K).GT.ZL)HQP(I,K)=ZL
            HQP(I,K)=(HQU-HQL)*DELZ/DENOM+HQL
C           WRITE(9,99)HQL,HQP(I,K),HQU,ZGHIH(I,J-1),DZQ,ZGHIH(I,J)
C   99            FORMAT(6F10.6)
         GOTO 500
         ENDIF
  400 CONTINUE
  500 CONTINUE
C
      RETURN
      END
C*********************************************************************
C*********************************************************************
      SUBROUTINE HGRADZ
C*********************************************************************
C
C THIS SUBROUTINE CALCULATES THE HYDRAULIC GRADIENT AT ANY DESIGNATED
C LOCATION
C
C*********************************************************************
C
      PARAMETER (NEL=3,NNP=NEL+1,MH=200,NZMAX=MH*10,NQP=256)
C
      COMMON /BLOCK1/NUMNP,NUMEL,Z(NNP),MHARM(NEL),ZL,ZLE(NEL)
      COMMON /BLOCK3/JFLAG(20),SATK(20),PHI(20),THETAR(20),
     +      PSIR(20),BCLAMDA(20),BCPB(20),VGALPHA(20),VGM(20),VGN(20)
      COMMON /BLOCK4/IDELEM(NEL)
      COMMON /BLOCK6/PSII(NNP),PSI(NNP),HI(NNP),H(NNP),PSICON
      COMMON /BLOCK7/AOLD(NEL,0:MH),ANEW(NEL,0:MH)
      COMMON /BLOCK8/XF(3)
      COMMON /BLOCK9/CM(MH,MH),RHS(MH),X(MH),XOLD(MH),NR,NC
      COMMON /BLOCK10/ DXQ(NQP),DWQ(NQP),HQP(NEL,NQP),DHDZ(NEL,NQP)
C     COMMON /BLOCK11/ZG(500)
C
      DOUBLE PRECISION DXQ,DWQ,ZGU,ZGL,Z
```

```
C
C I = ELEMENT NUMBER
C Z(I) = BEGINING NODE FOR ELEMENT I (GLOBAL SYSTEM)
C Z(I+1) = ENDING NODE FOR ELEMENT I (GLOBAL SYSTEM)
C ZLE = LENGTH OF ELEMENT I
C  ZG =ZGLOBAL
C ZGU = ZGLOBAL AT ANY DESIGNATED LOCATION (UPPER)
C ZG  = ZGLOBAL AT ANY DESIGNATED LOCATION (CENTER)
C ZGL = ZGLOBAL AT ANY DESIGNATED LOCATION (LOWER)
C ZG  = ZGLOBAL AT ANY DESIGNATED LOCATION (CENTER)
C HQP = HYDRAULIC HEAD
C HQU = HYDRAULIC HEAD AT ANY DESIGNATED LOCATION (UPPER)
C HQL = HYDRAULIC HEAD AT ANY DESIGNATED LOCATION (LOWER)
C PSIQ =PRESSURE HEAD
C PSIQU = PRESSURE HEAD AT ANY DESIGNATED LOCATION (UPPER)
C PSIQL = PRESSURE HEAD AT ANY DESIGNATED LOCATION (LOWER)
C SE =EFFECTIVE SATURATION
C SEU =  EFFECTIVE SATURATION AT ANY DESIGNATED LOCATION (UPPER)
C SEL =  EFFECTIVE SATURATION AT ANY DESIGNATED LOCATION (LOWER)
C
      CALL HQUAD
C
      DO 500 I=1,NUMEL
C
      A=Z(I)
      B=Z(I+1)
C
      DO 400 K=2,NQP-1
C
C  CALCULATE ZGLOBAL UPPER AND LOWER
C
      ZGL=(B+A)/2.+(B-A)/2.*DXQ(K-1)
      ZGU=(B+A)/2.+(B-A)/2.*DXQ(K+1)
C
C CALCULATE HYDRAULIC HEAD (HQ) UPPER AND LOWER
C
      HQU=HQP(I,K+1)
      HQL=HQP(I,K-1)
C
C COMPUTE THE HYDRAULIC HEAD GRADIENT (DH/DZ)
C
      DHDZ(I,K)=(HQU-HQL)/(ZGU-ZGL)
      IF(DHDZ(I,K).LT..001)DHDZ(I,K)=.001
C
  400 CONTINUE
C
C FIRST VALUE
C
      ZGU=(B+A)/2.+(B-A)/2.*DXQ(2)
      ZGL=(B+A)/2.+(B-A)/2.*DXQ(1)
C
      HQU=HQP(I,2)
      HQL=HQP(I,1)
C
      DHDZ(I,1)=(HQU-HQL)/(ZGU-ZGL)
      IF(DHDZ(I,1).LT..001)DHDZ(I,1)=.001
C
```

```
C LAST VALUE
C
      ZGU=(B+A)/2.+(B-A)/2.*DXQ(NQP)
      ZGL=(B+A)/2.+(B-A)/2.*DXQ(NQP-1)
C
      HQU=HQP(I,NQP)
      HQL=HQP(I,NQP-1)
C
      DHDZ(I,NQP)=(HQU-HQL)/(ZGU-ZGL)
      IF(DHDZ(I,NQP).LT..001)DHDZ(I,NQP)=.001
C
  500 CONTINUE
C
      DO 700 I=1,NUMEL
C
C SMOOTH DHDZ 100 TIMES
C
      DO 650 KTR=1,100
         DHDZ(I,1)=(DHDZ(I,1)+DHDZ(I,2))/2.
      DO 600 K=2,NQP-1
  600    DHDZ(I,K)=(DHDZ(I,K-1)+DHDZ(I,K)+DHDZ(I,K+1))/3.
         DHDZ(I,NQP)=(DHDZ(I,NQP-1))+(DHDZ(I,NQP))/2.
  650 CONTINUE
C
  700 CONTINUE
      RETURN
      END
C*********************************************************************
C*********************************************************************
      SUBROUTINE QUADINTG (I,M,N,DSUMA,DSUMB)
C*********************************************************************
C
C THIS SUBROUTINE PERFORMS NUMERICAL INTEGRATION
C
C*********************************************************************
C
      PARAMETER (NEL=3,NNP=NEL+1,MH=200,NZMAX=MH*10,NQP=256)
C
      COMMON /BLOCK1/NUMNP,NUMEL,Z(NNP),MHARM(NEL),ZL,ZLE(NEL)
      COMMON /BLOCK3/JFLAG(20),SATK(20),PHI(20),THETAR(20),
     +      PSIR(20),BCLAMDA(20),BCPB(20),VGALPHA(20),VGM(20),VGN(20)
      COMMON /BLOCK4/IDELEM(NEL)
      COMMON /BLOCK7/AOLD(NEL,0:MH),ANEW(NEL,0:MH)
      COMMON /BLOCK6/PSII(NNP),PSI(NNP),HI(NNP),H(NNP),PSICON
      COMMON /BLOCK10/ DXQ(NQP),DWQ(NQP),HQP(NEL,NQP),DHDZ(NEL,NQP)
C
      DOUBLE PRECISION DPI,DXQ,DWQ,DZQ,DFA,DFB,DSUMA,DSUMB,Z
C
C I = ELEMENT NUMBER
C A = THE LOWER LIMIT OF THE INTEGRAL= Z(I)
C B = THE UPPER LIMIT OF THE INTEGRAL= Z(I+1)
C M = NUMBER OF HARMONICS M= MHARM(I)
C N = NUMBER OF HARMONICS N= NHARM(I)
C ZLE=B-A= LENGTH OF ELEMENT I
C DZQ = THE TRANSFORMATION VALUES FOR THE LIMITS OF THE INTEGRAL I.
C DFA = THE APPROXIMATED INTEGRAL I.
C DFB = THE APPROXIMATED INTEGRAL I.
```

```
C DSUMA = SUM OF THE APPROXIMATED INTEGRAL I.
C DSUMB = SUM OF THE APPROXIMATED INTEGRAL I.
C
      IF(I.EQ.1)THEN
C
      CALL HQUAD
      CALL HGRADZ
      ENDIF
C
      DPI=3.14159265358979323846 2643
C
      A=Z(I)
      B=Z(I+1)
C
C PERFORM THE NUMRICAL INTEGRATION
C
      DSUMA=0.
      DSUMB=0.
C
      DO 50 K=1,NQP
         DZQ=(B+A)/2.+(B-A)/2.*DXQ(K)
         ZLOCAL=(DZQ-Z(I))/ZLE(I)
C
C CALCULATE HYDRAULIC HEAD (HQ)
C
      HQ=HQP(I,K)
C
C CALCULATE PRESSURE HEAD AT EACH QUADRATURE POINT
C
      PSIQ=HQ-DZQ
C
C CALL SUBROUTINE AUXQS TO CALCULATE krw AND DSDPSI
C
      CALL AUXEQS(PSIQ,IDELEM(I),RELP,THETA,SE,DSDPSI)
C
C CALCULATE SQ
C
      SQ=PHI(IDELEM(I))*DSDPSI*(1.-1./DHDZ(I,K))
C
C     WRITE(9,99)DZQ,HQ,PSIQ,DSDPSI
C   99      FORMAT(1X,4(F10.6,1X))
C
      DFA=SIN(M*DPI*ZLOCAL)*SIN(N*DPI*ZLOCAL)
      DFA=RELP*SATK(IDELEM(I))*DFA*DWQ(K)
      DSUMA=DSUMA+DFA
      DFB=COS(M*DPI*ZLOCAL)*COS(N*DPI*ZLOCAL)
      DFB=SQ*DFB*DWQ(K)
      DSUMB=DSUMB+DFB
   50 CONTINUE
C
      DSUMA=DSUMA*(B-A)/2.*(M*N)*(DPI/ZLE(I))**2
      DSUMB=DSUMB*(B-A)/2.
C
      RETURN
      END
C*********************************************************************
C*********************************************************************
```

```
      SUBROUTINE MATXAB (I,JJ,KK,DSUMA,DSUMB)
C*********************************************************************
C
C THIS SUBROUTINE FORMS THE GLOBAL MATRIX[CM]
C
C*********************************************************************
C
      PARAMETER (NEL=3,NNP=NEL+1,MH=200,NZMAX=MH*10,NQP=256)
C
      COMMON /BLOCK1/NUMNP,NUMEL,Z(NNP),MHARM(NEL),ZL,ZLE(NEL)
      COMMON /BLOCK5/TIME,DTIME,TOTIME
      COMMON /BLOCK6/PSII(NNP),PSI(NNP),HI(NNP),H(NNP),PSICON
      COMMON /BLOCK7/AOLD(NEL,0:MH),ANEW(NEL,0:MH)
      COMMON /BLOCK9/CM(MH,MH),RHS(MH),X(MH),XOLD(MH),NR,NC
      COMMON /BLOCK10/DXQ(NQP),DWQ(NQP),HQP(NEL,NQP),DHDZ(NEL,NQP)
      COMMON /BLOCK13/HIH(NEL,MH),ZGHIH(NEL,MH),HZ(NEL,MH)
C
      DOUBLE PRECISION DXQ,DWQ,DSUMA,DSUMB,Z
C
C  I=ELEMENT NUMBER
C  M=HARMONICS M
C  N=HARMONICS N
C  MHARM(I)=NUMBER OF HARMONICS FOR ELEMENT I
C
      DPI=3.14159265358979323846264 3
      WRITE(*,5)
    5 FORMAT(5X,'ENTER MATXAB')
C
      NR=0
      DO 10 I=1,NUMEL
   10 NR=NR+(MHARM(I)/2)+2
      NC=NR
C
C INITIALIZE MATRIX [CM] and [RHS]
C
      DO 20 J=1,NR
      RHS(J)=0.
      DO 20 K=1,NC
   20 CM(J,K)=0.
C
C CONSTRUCT COEFFICIENT MATRIX [CM]
C
C
C CONSTRUCT THE FIRST TWO ROWS OF [CM] FOR ELEMENT (I)
C
      I=1
      SUM=1.
      DO 30 J=3,MHARM(I)-1,2
   30 SUM=SUM+1./(J**2)
C
      CM(1,1)=+1.
      CM(2,2)=SUM
C
C CONSTRUCT THE REMAINING ROWS OF MATRIX [CM]=[A]+[B]/DTIME
C
      DO 50 I=1,NUMEL
C
```

```
C CONSTRUCT MATRIX [CM]=[A]+[B]/DTIME
C
      DO 40 J=3,NR
      DO 40 K=3,NC
      IF(ABS(J-K).GE.2)GOTO 40
      JJ=(J-2)*2
      KK=(K-2)*2
C
      CALL QUADINTG (I,JJ,KK,DSUMA,DSUMB)
C
C CALCULATE [CM]=[A]+[B]/DTIME
C
      CM(J,K)=DSUMA+DSUMB/DTIME
      RHS(J)=(DSUMB/DTIME)*AOLD(I,KK)+RHS(J)
C
C     WRITE(9,99)J,K,JJ,KK,DSUMA,DSUMB
C   99      FORMAT(4I3,2F10.6,5X)
   40 CONTINUE
   50 CONTINUE
C
C MODIFY THE RHS ARRAY
C
      I=1
      NDIV=MHARM(I)/2
C     WRITE(*,*)NDIV
      HO=HIH(I,1)
      HL=HIH(I,NDIV+1)
C
      RHS(1)=(HO+HL)/2.
      RHS(2)=(HO-HL)/2.
      RHS(3)=0.
C
C SUM OF EVEN COEFF.=0.
C
      DO 60 J=3,NC
      CM(3,J)=+1
   60 CONTINUE
C
C DEBUG-WRITE MATRIX [CM]
C
      DO 100 J=1,NR
      WRITE(8,90)J
   90 FORMAT(/'LINE NO.=',I3)
      WRITE(8,95)(CM(J,K),K=1,NC)
   95 FORMAT(18F20.9)
  100 CONTINUE
C
C DEBUG-WRITE MATRIX [RHS]
C
      DO 120 J=1,NR
      WRITE(8,110)RHS(J)
  110 FORMAT(F20.9)
  120 CONTINUE
C
      WRITE(*,25)
   25 FORMAT(5X,'EXIT MATXAB')
C
```

```
      RETURN
      END
C*********************************************************************
      SUBROUTINE HICHYD
C*********************************************************************
C
C THIS SUBROUTINE CALCULATES THE INITIAL HYDRAULIC HEAD DISTRIBUTION
C
C*********************************************************************
C
      PARAMETER (NEL=3,NNP=NEL+1,MH=200,NZMAX=MH*10,NQP=256)
C
      COMMON /BLOCK1/NUMNP,NUMEL,Z(NNP),MHARM(NEL),ZL,ZLE(NEL)
      COMMON /BLOCK3/JFLAG(20),SATK(20),PHI(20),THETAR(20),
     +      PSIR(20),BCLAMDA(20),BCPB(20),VGALPHA(20),VGM(20),VGN(20)
      COMMON /BLOCK4/IDELEM(NEL)
      COMMON /BLOCK6/PSII(NNP),PSI(NNP),HI(NNP),H(NNP),PSICON
      COMMON /BLOCK7/AOLD(NEL,0:MH),ANEW(NEL,0:MH)
      COMMON /BLOCK8/XF(3)
      COMMON /BLOCK9/CM(MH,MH),RHS(MH),X(MH),XOLD(MH),NR,NC
      COMMON /BLOCK10/DXQ(NQP),DWQ(NQP),HQP(NEL,NQP),DHDZ(NEL,NQP)
      COMMON /BLOCK13/HIH(NEL,MH),ZGHIH(NEL,MH),HZ(NEL,MH)
C
      DOUBLE PRECISION DXQ,DWQ,Z
C
C*********************************************************************
*******
C
C MHARM(I) = NUMBER OF HARMONICS + 1 FOR ELEMENT I
C Z(I)=Z1 = BEGINING NODE FOR ELEMENT I (GLOBAL SYSTEM)
C Z(I+1)=Z2 = ENDING NODE FOR ELEMENT I (GLOBAL SYSTEM)
C ZLE      = LENGTH OF ELEMENT I
C DELZG    = INCREMENT FOR Z IN GLOBAL COORDINATE SYSTEM
C DELZL    = INCREMENT FOR Z IN LOCAL COORDINATE SYSTEM
C HIH1 = FIRST GEUSS OF INITIAL HYDROSTATIC HEAD VALUE AT WATER TABLE
C HIH2 = SECOND GEUSS OF INITIAL HYDROSTATIC HEAD VALUE
C RELP12 = AVERAGE RECIPROCAL VALUE OF RELATIVE PERMEABILITY
C HIHNEW2 = INITIAL HYDROSTATIC HEAD DISTRIBUTION FOR STEADY-STATE FLOW
C ZLOCAL FOR CUBIC POINT=0.6
C
      DPI=3.14159265358979323846264 3
      WRITE(*,15)
   15 FORMAT(5X,'ENTER HICHYD')
C
C CALL SUBROUTINE ETYPMH TO DETERMINE ELEMENT TYPE AND NUMBER OF
C HARMONICS
C
      CALL ETYPMH
C
      DENOM=0.
      DO 100 I=1,NUMEL
  100 DENOM=DENOM+(ZLE(I)/SATK(IDELEM(I)))
      QSS=ZL/DENOM
C
      QSMALL=QSS/250
C     WRITE(*,*)QSS,QSMALL
C
```

```
C INITIAL HEAD AND PRESSURE HEADS ARE ZERO AT WATER TABLE
C
      HIH1=0.
      PSI1=0.
      ZGHIH(1,1)=0.
      HIH(1,1)=0.
      RELP1=1.
      SE1=1.
C
C START CALCULATIONS FOR HIH
C
      DO 500 I=1, NUMEL
C
      IF(I.GE.2)HIH(I,1)=HIH(I-1,NDIV+1)
      NDIV=MHARM(I)/2
      WRITE(*,*)NDIV
      DELZG=ZLE(I)/NDIV/1.01
      ZA=Z(I)
      ZLOCAL=(ZA-Z(I))/ZLE(I)
C
         DO 450 J=1,NDIV
         ZB=ZA+DELZG
         ZGHIH(I,J+1)=ZB
         IF(J.EQ.NDIV)THEN
         ZB=Z(I+1)
         ZGHIH(I,NDIV+1)=Z(I+1)
         DELZG=ZB-ZA
         ENDIF
C
C  GUESS VALUE OF HIH2 (HIH1 = KNOWN)
C
      HIH2=HIH1
  440 PSI2=HIH2-ZB
C
C  CALCULATE RELATIVE PERMEABILITY AND EFFECTIVE SATURATION
C
      CALL AUXEQS (PSI2,IDELEM(I),RELP2,THETA2,SE2,DSDPSI)
C
      RELP12=(RELP1+RELP2)/(2*RELP1*RELP2)
      HIHNEW2=(QSMALL/SATK(IDELEM(I)))*RELP12*DELZG+HIH1
C
      IF(ABS(HIHNEW2-HIH2).GT..0001) THEN
      HIH2=HIHNEW2
      GOTO 440
      ELSE
      HIH(I,J+1)=HIHNEW2
      HIH1=HIHNEW2
      PSI1=HIHNEW2-ZB
      ZA=ZB
      ZLOCAL=(ZA-Z(I))/ZLE(I)
C
C  CALCULATE EFFECTIVE SATURATION
C
      CALL AUXEQS (PSI1,IDELEM(I),RELP1,THETA1,SE1,DSDPSI)
C
      ENDIF
      WRITE(8,25)ZA,ZLOCAL,HIH1,PSI1
```

```
   25 FORMAT(4(1X,F10.7,2X))
C
  450 CONTINUE
  500 CONTINUE
C
C  CALCULATE THE CUBIC EQUATION FOR INITIAL HEAD DISTRIBUTIONS
C
      NDIV=MHARM(NUMEL)/2.
      DELZG=ZLE(NUMEL)/NDIV/1.01
      KTR=(.801)*NDIV+1
      ZLOCAL=KTR*DELZG/ZLE(NUMEL)
C
      CALL CUBIC (ZLOCAL,HIH(NUMEL,KTR+1),1.,ZL,5.,10.,A,B,C,D)
C
      DO 550 K=KTR+1,NDIV+1
      ZGLOBAL=ZGHIH(NUMEL,K)
      ZLOCAL=(ZGLOBAL-Z(NUMEL))/ZLE(NUMEL)
  550 HIH(NUMEL,K)=A+B*ZLOCAL+C*ZLOCAL**2+D*ZLOCAL**3
C     WRITE(9,99)A,B,C,D
C   99      FORMAT(F20.9)
      HIH(NUMEL,NDIV+1)=ZL
C
C     DO 80 I=1,NUMEL
C     DO 80 J=1,MHARM(I)
C     PSI3=HIH(I,J)-ZGHIH(I,J)
C
C      CALL AUXEQS (PSI3,IDELEM(I),RELP3,THETA3,SE3,DSDPSI)
C
C     WRITE(9,99)ZGHIH(I,J),HIH(I,J),PSI3
C   99      FORMAT(3(1X,F10.6,2X))
C   80      CONTINUE
C
      WRITE(*,20)
   20 FORMAT(5X,'EXIT HICHYD')
C
      CALL HIHOUT
C
      RETURN
      END
C
C*********************************************************************
C*********************************************************************
      SUBROUTINE PSIVSE
**********************************************************************
C
C THIS SUBROUTINE CALLS SUBROUTINE AUXEQS TO CALCULATE XPSI AND SE
C
      PARAMETER (NEL=3,NNP=NEL+1,MH=200,NZMAX=MH*10,NQP=256)
C
      COMMON /BLOCK1/NUMNP,NUMEL,Z(NNP),MHARM(NEL),ZL,ZLE(NEL)
      COMMON /BLOCK2/IFLAG(10)
      COMMON /BLOCK3/JFLAG(20),SATK(20),PHI(20),THETAR(20),
     +      PSIR(20),BCLAMDA(20),BCPB(20),VGALPHA(20),VGM(20),VGN(20)
      COMMON /BLOCK4/IDELEM(NEL)
      COMMON /BLOCK6/PSII(NNP),PSI(NNP),HI(NNP),H(NNP),PSICON
      COMMON /BLOCK7/AOLD(NEL,0:MH),ANEW(NEL,0:MH)
      COMMON /BLOCK8/XF(3)
```

```
C
      DOUBLE PRECISION Z
C
      XPSI=0.
C
   99 CALL AUXEQS (XPSI,3,RELP,THETA,SE,DSDPSI)
      WRITE(9,15)SE,XPSI,DSDPSI
   15 FORMAT(1X,3(F12.6,2X))
      XPSI=XPSI-.1
      IF(XPSI.GT.-20) GOTO 99
C
      RETURN
      END
C
C*********************************************************************
C*********************************************************************
      SUBROUTINE RELPVSE
C*********************************************************************
C
C THIS SUBROUTINE CALLS SUBROUTINE AUXEQS TO CALCULATE XPSI AND SE
C
      PARAMETER (NEL=3,NNP=NEL+1,MH=200,NZMAX=MH*10,NQP=256)
C
      COMMON /BLOCK1/NUMNP,NUMEL,Z(NNP),MHARM(NEL),ZL,ZLE(NEL)
      COMMON /BLOCK2/IFLAG(10)
      COMMON /BLOCK3/JFLAG(20),SATK(20),PHI(20),THETAR(20),
     +       PSIR(20),BCLAMDA(20),BCPB(20),VGALPHA(20),VGM(20),VGN(20)
      COMMON /BLOCK4/IDELEM(NEL)
      COMMON /BLOCK6/PSII(NNP),PSI(NNP),HI(NNP),H(NNP),PSICON
      COMMON /BLOCK7/AOLD(NEL,0:MH),ANEW(NEL,0:MH)
      COMMON /BLOCK8/XF(3)
C
      DOUBLE PRECISION Z
C
      SE=0.
      ISTYPE=3
      DO 100 K=1,101
C
   99 B= (1.-(SE**(1./VGM(ISTYPE))))**VGM(ISTYPE)
      RELP = SE**0.5 *(1-B)**2.
C
      WRITE(9,15)SE,RELP
   15 FORMAT(1X,2(F12.6,2X))
      SE=SE+.01
C
      IF(SE.GT.1.01) GOTO 99
  100 CONTINUE
C
      RETURN
      END
C*********************************************************************
C*********************************************************************
      SUBROUTINE HANDPSI
C*********************************************************************
C THIS SUBROUTINE USES FOURIER SERIES TO CALCULATE THE HYDRAULIC HEAD
C*********************************************************************
C
```

```
      PARAMETER (NEL=3,NNP=NEL+1,MH=200,NZMAX=MH*10,NQP=256)
C
      COMMON /BLOCK1/NUMNP,NUMEL,Z(NNP),MHARM(NEL),ZL,ZLE(NEL)
      COMMON /BLOCK3/JFLAG(20),SATK(20),PHI(20),THETAR(20),
     +       PSIR(20),BCLAMDA(20),BCPB(20),VGALPHA(20),VGM(20),VGN(20)
      COMMON /BLOCK4/IDELEM(NEL)
      COMMON /BLOCK6/PSII(NNP),PSI(NNP),HI(NNP),H(NNP),PSICON
      COMMON /BLOCK7/AOLD(NEL,0:MH),ANEW(NEL,0:MH)
      COMMON /BLOCK8/XF(3)
      COMMON /BLOCK9/CM(MH,MH),RHS(MH),X(MH),XOLD(MH),NR,NC
C
      DOUBLE PRECISION ZG,Z
C
C I = ELEMENT NUMBER
C Z(I) = BEGINING NODE FOR ELEMENT I (GLOBAL SYSTEM)
C Z(I+1) = ENDING NODE FOR ELEMENT I (GLOBAL SYSTEM)
C ZLE = LENGTH OF ELEMENT I
C  ZG =ZGLOBAL COORDINATE (Z)
C  ZL = ZLOCAL
C  XH = HYDRAULIC HEAD
C  XPSI =PRESSURE HEAD
C  SE =EFFECTIVE SATURATION
C
      DPI=3.14159265358979323846264 3
C
      DO 500 I=1,NUMEL
         ZG=Z(I)
         DELZ=ZLE(I)/100
      DO 500 II=1,101
C
C CALCULATE ZLOCAL
C
      ZLOCAL=(ZG-Z(I))/ZLE(I)
C
C  CALCULATE HYDRAULIC HEAD (H)
C
C
C HYDRAULIC HEAD AT LOCATION (ZG)
C
      XH=AOLD(I,0)
      DO 10 J=1,MHARM(I)-1
   10 XH=XH+AOLD(I,J)*COS(J*DPI*ZLOCAL)
C
C  CALCULATE PRESSURE HEAD (PSI)
C
C  PRESSURE HEAD AT UPPER LOCATION (PSIU)
C
      XPSI=XH-ZG
C
C  CALCULATE EFFECTIVE SATURATION (SE)
C
C EFFECTIVE SATURATION AT LOCATION (ZG)
C
      CALL AUXEQS (XPSI,IDELEM(I),RELP,THETA,SE,DSDPSI)
C
      WRITE(9,15)ZG,THETA,SE
   15 FORMAT(1X,3(F12.6,2X))
```

```
C
      ZG=ZG+DELZ
  500 CONTINUE
C
      RETURN
      END
C*********************************************************************
C*********************************************************************
      SUBROUTINE HQVDSDH
C*********************************************************************
C
C THIS SUBROUTINE CALLS SUBROUTINE HQUAD TO CALCULATE THE HYDRAULIC
C HEAD AND PRESSURE HEAD USING QUADRATURE POINTS AND COMPUTES SQ
C
C*********************************************************************
C
      PARAMETER (NEL=3,NNP=NEL+1,MH=200,NZMAX=MH*10,NQP=256)
C
      COMMON /BLOCK1/NUMNP,NUMEL,Z(NNP),MHARM(NEL),ZL,ZLE(NEL)
      COMMON /BLOCK3/JFLAG(20),SATK(20),PHI(20),THETAR(20),
     +      PSIR(20),BCLAMDA(20),BCPB(20),VGALPHA(20),VGM(20),VGN(20)
      COMMON /BLOCK4/IDELEM(NEL)
      COMMON /BLOCK6/PSII(NNP),PSI(NNP),HI(NNP),H(NNP),PSICON
      COMMON /BLOCK7/AOLD(NEL,0:MH),ANEW(NEL,0:MH)
      COMMON /BLOCK8/XF(3)
      COMMON /BLOCK9/CM(MH,MH),RHS(MH),X(MH),XOLD(MH),NR,NC
      COMMON /BLOCK10/ DXQ(NQP),DWQ(NQP),HQP(NEL,NQP),DHDZ(NEL,NQP)
C
      DOUBLE PRECISION DXQ,DWQ,DZQ,Z
C
C I=ELEMENT NUMBER
C Z(I)=BEGINING NODE FOR ELEMENT I (GLOBAL SYSTEM)
C Z(I+1)=ENDING NODE FOR ELEMENT I (GLOBAL SYSTEM)
C HQP=HYDRAULIC HEAD AT EACH QUADRATURE POINT
C HQ=HYDRAULIC HEAD
C PSIQ=PRESSURE HEAD
C
      DO 110 I=1,NUMEL
C
         IF(I.EQ.1)THEN
C
         CALL HQUAD
         CALL HGRADZ
         ENDIF
C
         A=Z(I)
         B=Z(I+1)
C
         DO 100 K=1,NQP
         DZQ=(B+A)/2.+(B-A)/2.*DXQ(K)
C
C CALCULATE HYDRAULIC HEAD (HQ)
C
      HQ=HQP(I,K)
C
C CALCULATE PRESSURE HEAD (PSIQ)
C
```

```
      PSIQ=HQ-DZQ
C
C CALCULATE SLOPE DERIVATIVE (DSDPSI) FOR PSIQ
C
  CALL AUXEQS (PSIQ,IDELEM(I),RELP,THETA,SE,DSDPSI)
C
C CALCULATE (SQ)
C
      SQ=PHI(IDELEM(I))*DSDPSI*(1.-1./DHDZ(I,K))
C
      DSDH=SQ/PHI(IDELEM(I))
C
      WRITE(*,*)DSDPSI
      WRITE(9,15)DZQ,HQ,PSIQ
   15 FORMAT(3F10.6)
C
  100 CONTINUE
  110 CONTINUE
C
      RETURN
      END
C*********************************************************************
      SUBROUTINE QFLOWZ
C*********************************************************************
C
C   THIS SUBROUTINE CALCULATES THE FLOW RATE AT THE GROUND SURFACE(QIN)
C   AND THE FLOWOUT AT WATER TABLE (QOUT)
C
C*********************************************************************
C
C
      PARAMETER (NEL=3,NNP=NEL+1,MH=200,NZMAX=MH*10,NQP=256)
C
      COMMON /BLOCK1/NUMNP,NUMEL,Z(NNP),MHARM(NEL),ZL,ZLE(NEL)
      COMMON /BLOCK3/JFLAG(20),SATK(20),PHI(20),THETAR(20),
     +      PSIR(20),BCLAMDA(20),BCPB(20),VGALPHA(20),VGM(20),VGN(20)
      COMMON /BLOCK4/IDELEM(NEL)
      COMMON /BLOCK6/PSII(NNP),PSI(NNP),HI(NNP),H(NNP),PSICON
      COMMON /BLOCK7/AOLD(NEL,0:MH),ANEW(NEL,0:MH)
      COMMON /BLOCK8/XF(3)
      COMMON /BLOCK9/CM(MH,MH),RHS(MH),X(MH),XOLD(MH),NR,NC
      COMMON /BLOCK11/ZG(500),DHDZ(500)
      COMMON /BLOCK12/QIN,QOUT,TOTQIN,TOTQOUT,QSTORI
C
      DOUBLE PRECISION DPI,Z,ZG
C
C*********************************************************************
C
C QIN = INFILTRATION RATE AT GROUND SURFACE
C QOUT = FLOWOUT RATE AT WATER TABLE
C SATK = SATURATED HYDRAULIC CONDUCTIVITY Ksat
C DHDZ = HYDRAULIC GRADIENT
C
      DPI=3.14159265358979323846264 3
C
      QIN=SATK(IDELEM(NUMEL))*DHDZ(301)
C
```

```
C INITIAL HEAD AND PRESSURE HEAD ARE ZERO AT WATER TABLE
C
      HWT=0.
      PSIWT=0.
      RELPWT=1.
      SEWT=1.
C
C CALCULATE ZLOCAL
C
      DELZ =ZLE(1)/(MHARM(1)-1)
      ZLOCAL = DELZ/ZLE(1)
C
C CALCULATE HYDRAULIC HEAD VALUE AT Z=DELZ
C
      I=1
      HZ=AOLD(I,0)
      DO 10 J=1,MHARM(I)-1
   10 HZ=HZ+AOLD(I,J)*COS(J*DPI*ZLOCAL)
      PSIZ=HZ-DELZ
C
C CALCULATE RELATIVE PERMEABILITY AND EFFECTIVE SATURATION
C
      CALL AUXEQS (PSIZ,IDELEM(1),RELPZ,THETA,SEZ,DSDPSI)
C
      RELPAV=(RELPWT+RELPZ)/(2*RELPWT*RELPZ)
C
      QOUT=(HZ*SATK(IDELEM(1)))/(RELPAV*DELZ)
C     WRITE(*,*)QOUT,QIN
      WRITE(9,20)QOUT,QIN
   20 FORMAT(1X,1(F10.8,2X))
C
      RETURN
      END
C*********************************************************************
C*********************************************************************
      SUBROUTINE QSTORAGE (DSUMT)
C*********************************************************************
C
C THIS SUBROUTINE CALCULATES THE VOLUME OF WATER STORED IN THE SOIL AT
C ANY SPECIFIC DEPTH DURING ANY SPECIFIED TIME PERIOD.
C
C*********************************************************************
C
      PARAMETER (NEL=3,NNP=NEL+1,MH=200,NZMAX=MH*10,NQP=256)
C
      COMMON /BLOCK1/NUMNP,NUMEL,Z(NNP),MHARM(NEL),ZL,ZLE(NEL)
      COMMON /BLOCK3/JFLAG(20),SATK(20),PHI(20),THETAR(20),
     +      PSIR(20),BCLAMDA(20),BCPB(20),VGALPHA(20),VGM(20),VGN(20)
      COMMON /BLOCK4/IDELEM(NEL)
      COMMON /BLOCK7/AOLD(NEL,0:MH),ANEW(NEL,0:MH)
      COMMON /BLOCK6/PSII(NNP),PSI(NNP),HI(NNP),H(NNP),PSICON
      COMMON /BLOCK10/ DXQ(NQP),DWQ(NQP),HQP(NEL,NQP),DHDZ(NEL,NQP)
C
      DOUBLE PRECISION DPI,DXQ,DWQ,DZQ,DFT,Z
C
C I = ELEMENT NUMBER
C A = THE LOWER LIMIT OF THE INTEGRAL= Z(I)
```

```
C  B = THE UPPER LIMIT OF THE INTEGRAL= Z(I+1)
C  ZLE=B-A= LENGTH OF ELEMENT I
C  DZQ = THE TRANSFORMATION VALUES FOR THE LIMITS OF THE INTEGRAL I.
C  DFT = THE APPROXIMATED INTEGRAL I.
C  DSUMT = SUM OF THE APPROXIMATED INTEGRAL I.
C
      DPI=3.14159265358979323846264 3
      DSUMT=0.
C
      DO 200 I=1,NUMEL
C
         A = Z(I)
         B = Z(I+1)
C
C  PERFORM THE NUMRICAL INTEGRATION
C
      DO 100 K=1,NQP
         DZQ=(B+A)/2.+(B-A)/2.*DXQ(K)
         ZLOCAL=(DZQ-Z(I))/ZLE(I)
C
C  CALCULATE HYDRAULIC HEAD (HT)
C
      HT=AOLD(I,0)
      DO 50 J=1,MHARM(I)-1
   50 HT=HT+AOLD(I,J)*COS(J*DPI*ZLOCAL)
C
C  CALCULATE PRESSURE HEAD AT EACH QUADRATURE POINT
C
      IF(HT.LT.0.)HT=0.
      PSIT=HT-DZQ
C
C CALL SUBROUTINE AUXQS TO CALCULATE THETA
C
      CALL AUXEQS(PSIT,IDELEM(I),RELP,THETA,SE,DSDPSI)
C
C CALCULATE THE INTEGRAL FUNCTION
C
      DFT=THETA*DWQ(K)
      DSUMT=DSUMT+DFT
C
  100 CONTINUE
  200 CONTINUE
      DSUMT=DSUMT*(B-A)/2.
C     WRITE(*,*)DSUMT
      WRITE(9,25)DSUMT
   25 FORMAT(F12.8)
C
      RETURN
      END
C*********************************************************************
C*********************************************************************
      SUBROUTINE WATBAL (ICALL)
C*********************************************************************
C
C THIS SUBROUTINE CALCULATES THE AMOUNT OF WATER STORED IN THE SOIL AT
C ANY PARTICULAR DEPTH DURING ANY SPECIFIED TIME PERIOD.
C
```

```
C
C**********************************************************************
C
      PARAMETER (NEL=3,NNP=NEL+1,MH=200,NZMAX=MH*10,NQP=256)
C
      COMMON /BLOCK1/NUMNP,NUMEL,Z(NNP),MHARM(NEL),ZL,ZLE(NEL)
      COMMON /BLOCK4/IDELEM(NEL)
      COMMON /BLOCK5/TIME,DTIME,TOTIME
      COMMON /BLOCK6/PSII(NNP),PSI(NNP),HI(NNP),H(NNP),PSICON
      COMMON /BLOCK7/AOLD(NEL,0:MH),ANEW(NEL,0:MH)
      COMMON /BLOCK12/QIN,QOUT,TOTQIN,TOTQOUT,QSTORI
C
      DOUBLE PRECISION Z
C
C QIN=INFILTRATION RATE INTO SOIL AT THE GROUND SURFACE
C QOUT=WATER FLOW THROUGH THE UNSATURATED ZONE TO THE WATER TABLE
C QSTORI=INITIAL AMOUNT OF WATER STORED IN SOIL
C QINAVG=AVERAGE AMOUNT OF WATER INFILTRATE INTO THE SOIL
C QOUTAVG=AVERAGE AMOUNT OF WATER RELEASED FROM THE SOIL
C TOTQIN=TOTAL AMOUNT OF WATER FLOW INTO THE SOIL
C TOTQOUT=TOTAL AMOUNT OF WATER RELEASED BY THE SOIL
C DLTSTOR=CHANGE IN STORAGE
C DTIME=TIME FOR EACH STEP
C
      QINBEG=QIN
      QOUTBEG=QOUT
C
C CALCULATE FLOWIN AND FLOWOUT
C
      CALL QFLOWZ
C
      IF(TIME.EQ.0.)RETURN
C
C CALCULATE AND PRINT CUMULATIVE WATER BALANCE AT TIME=T
C
      WRITE(8,10)TIME
   10 FORMAT(/,20X,'WATER BALANCES AT TIME =',F5.3,'(DAYS)'/
     +20X,          '_____________________________________'/)
C
      WRITE(8,20)QIN,QOUT
   20 FORMAT(///,22X,'INSTANTANEOUS WATER BALANCE'/
     +22X,           '_____________________________'///
     +10X,'QIN (INFILTRATION RATE AT GROUND SURFACE)=',2X,F10.6/
     +10X,'QOUT(FLOW OUT TO WATER TABLE)',12X,'=',2X,F10.6)
C
C CALCULATE AVERAGE AMOUNT OF FLOW INTO THE SOIL DURING THE TIME STEP
C
      QINAVG=(QIN+QINBEG)/2.
C
C CALCULATE AVERAGE AMOUNT OF FLOW TO THE WATER TABLE DURING TIME STEP
C
      QOUTAVG=(QOUT+QOUTBEG)/2.
C
C CALCULATE TOTAL AMOUNT OF WATER FLOW INTO THE SOIL AT TIME=T
C
      TOTQIN=TOTQIN+QINAVG*DTIME
C
```

```
C CALCULATE TOTAL AMOUNT OF WATER RELEASED BY THE SOIL AT TIME=T
C
      TOTQOUT=TOTQOUT+QOUTAVG*DTIME
C
      IF(ICALL.EQ.1)RETURN
C
      CALL QSTORAGE (QSTOR)
C
      DLTSTOR=QSTOR-QSTORI
C
C CALCULATE WATER BALANCE ERROR
C
      ERROR=TOTQIN-TOTQOUT-DLTSTOR
C
      TOTQINERR=(ERROR/TOTQIN)*100.
      TOTQOUTERR=(ERROR/TOTQOUT)*100.
      DLTSTOR=.1
      DLTSERR=(ERROR/DLTSTOR)*100.
C
      WRITE(8,30)TOTQIN,TOTQOUT,QSTORI,QSTOR,ERROR
   30 FORMAT(///,22X,'CUMULATIVE WATER BALANCE'/
     +22X,           '_____________________'///
      +10X,'TOTQIN (TOTAL INFILTRATION)',14X,'=',2X,F10.6/
      +10X,'TOTQOUT(TOTAL OUTFLOW TO WATER TABLE)',4X,'=',2X,F10.6/
      +10X,'QSTORI (INITIAL WATER IN STORAGE)',8X,'=',2X,F10.6/
      +10X,'QSTOR  (CURRENT WATER IN STORAGE)',8X,'=',2X,F10.6/
      +10X,'ERROR  (WATER BALANCE RESIDUAL)',10X,'=',2X,F10.6)
C
C WRITE CUMULATIVE WATER BALANCE ERRORS AS A PERCENT
C
      WRITE(8,60)TOTQINERR,TOTQOUTERR,DLTSERR
   60 FORMAT(/,
      +10X,'TOTQINERR (ERROR AS PERCENT OF TOTQIN)',3X,'=',2X,F10.6/
      +10X,'TOTQOUTERR(ERROR AS PERCENT OF TOTQUT)',3X,'=',2X,F10.3/
      +10X,'DLTSERR   (ERROR AS PERCENT OF DLTSTOR)',2X,'=',2X,F10.6)
C
      RETURN
      END
C
C*********************************************************************
C*********************************************************************
      SUBROUTINE HIHOUT
C*********************************************************************
C
C THIS SUBROUTINE PRINTS THE CALCULATED HYDRAULIC HEAD AND PRESSURE
C HEADS
C
*********************************************************************
C
      PARAMETER (NEL=3,NNP=NEL+1,MH=200,NZMAX=MH*10,NQP=256)
C
      COMMON /BLOCK1/NUMNP,NUMEL,Z(NNP),MHARM(NEL),ZL,ZLE(NEL)
      COMMON /BLOCK5/TIME,DTIME,TOTIME
      COMMON /BLOCK7/AOLD(NEL,0:MH),ANEW(NEL,0:MH)
      COMMON /BLOCK9/CM(MH,MH),RHS(MH),X(MH),XOLD(MH),NR,NC
      COMMON /BLOCK13/HIH(NEL,MH),ZGHIH(NEL,MH),HZ(NEL,MH)
C
```

```
      DOUBLE PRECISION Z
C
C
**********************************************************************
C
C
      WRITE(*,5)
    5 FORMAT(5X,'ENTER HIHOUT')
C
      WRITE(8,10)
   10 FORMAT(/,20X,'INITIAL HYDRAULIC HEAD AND PRESSURE HEAD'
     +'(DAYS)'/,20X,'_____________________________________________'
     +,'___')
C
      WRITE(8,20)
   20 FORMAT(/,4(5X,'ZH',7X,'HEAD',2X))
C
      DO 200 I=1,NUMEL
C
C  WRITE ZGHIH AND HIH
C
      NDIV=MHARM(I)/2
      WRITE(8,25)(ZGHIH(I,J),HIH(I,J),J=1,NDIV+1)
   25 FORMAT(8F10.6)
C     WRITE(9,30)(ZGHIH(I,J),HIH(I,J),J=1,NDIV+1)
C   30      FORMAT(2F10.6)
  200 CONTINUE
C
      WRITE(*,15)
   15 FORMAT(5X,'EXIT HIHOUT')
C
      RETURN
      END
```

APPENDIX B

INPUT DATA FILE
FOR
SAMPLE OUTPUT

APPENDIX (B)

INPUT DATA FOR SAMPLE OUTPUT FOURIER FINITE ELEMENT MODEL (FFUSAT MODEL)

DEBUGING DATA SET FOR PROGRAM FFUSAT X
NUMEL,ZL (I5,F10.0)
1 10.
IFLAG(1)=1 (UNIFORM SPACING OF NODAL POINTS) ELSE IFLAG(1) NOT= 1
READ Z COORDINATES (I2)
1
NSTYPE (I5)
20
ISTYPE,SATK,PHI,THETAR,JFLAG,PSIR,BCLAMDA,BCPB,VGALPHA,VGM,VGN
(I5,F10.0,2F10.0,I5,6F5.0)

1	.1388	.3	.15	1	-5.	.0	.0	.0	.0	.0
2	.1440	.3	.03	2	.0	2.0	-4.0	.0	.0	.0
3	1.5000	.3	.15	3	.0	.0	.0	.2	.6	2.5
4	.1238	.5	.13	1	-3.	.0	.0	.0	.0	.0
5	.1132	.2	.11	2	.0	1.5	-1.1	.0	.0	.0
6	.3000	.4	.22	3	.0	.0	.0	.02	.29	1.41
7	.2213	.1	.34	1	-8.	.0	.0	.0	.0	.0
8	.2532	.3	.33	2	.0	1.3	-3.3	.0	.0	.0
9	.1426	.7	.21	3	.0	.0	.0	.4	1.2	.3
10	.2213	.9	.11	1	-7.	.0	.0	.0	.0	.0
11	.1212	.8	.22	2	.0	1.0	-2.5	.0	.0	.0
12	.1121	.7	.23	3	.0	.0	-1.5	.3	1.3	.6
13	.0012	.2	.17	1	-5.	.0	.0	.0	.0	.0
14	.2010	.7	.14	2	.0	1.3	-3.3	.0	.0	.0
15	.4121	.4	.11	3	.0	.0	.0	.6	1.3	.3
16	.3114	.9	.03	1	-5.	.0	.0	.0	.0	.0
17	.2213	.8	.32	2	.0	1.7	-2.1	.0	.0	.0
18	.5515	.5	.35	3	.0	.0	.0	.7	1.5	.6
19	.6361	.7	.31	1	-9.	.0	.0	.0	.0	.0
20	.4562	.5	.29	2	.0	1.5	-4.1	.0	.0	.0

SOIL TYPE ARRAY (IDELEM) (5I5)
1 3 2 3 3 3
IFLAG(2)=1 (HYDROSTATIC), IFLAG(2)=2 (CONSTANT PSI), IFLAG(2)=3 (USER DEFINED) (I2)

APPENDIX C

ECHO DATA FILE
FOR
SAMPLE OUTPUT

APPENDIX (C)

ECHO DATA FOR SAMPLE OUTPUT FOURIER FINITE ELEMENT MODEL (FFUSAT MODEL)

DEBUGING DATA SET FOR PROGRAM FFUSAT X

INPUT DATA

GRID DATA

NUMEL (NUMBER OF ELEMENTS) = 1
ZL (DEPTH TO GROUNDWATER) = 10.0

INPUT DATA

NODAL COORDINATES

NODE	Z(NP)	NODE	Z(NP)
1	.00	2	10.00

INPUT DATA

ELEMENT LENGTH

ELEMENT (I)	Z(I)	Z(I+1)	ZLE (I)
1	.00	10.00	10.00

SOIL TYPES AND PROPERTIES

SOIL TYPE	SATK	PHI	THETAR	JFLAG	PSIR	BCLAMDA	BCPB	VGALPHA	VGM	VGN
1	.14	.30	.15	1	-5.00	.00	.00	.00	.00	.00
2	.14	.30	.03	2	.00	2.00	-4.00	.00	.00	.00
3	1.50	.30	.15	3	.00	.00	.00	.20	.60	2.50
4	.12	.50	.13	1	-3.00	.00	.00	.00	.00	.00
5	.11	.20	.11	2	.00	1.50	-1.10	.00	.00	.00
6	.30	.40	.22	3	.00	.00	.00	.00	2.20	9.40
7	.22	.10	.34	1	-8.00	.00	.00	.00	.00	.00
8	.25	.30	.33	2	.00	1.30	-3.30	.00	.00	.00
9	.14	.70	.21	3	.00	.00	.00	.40	1.20	.30
10	.22	.90	.11	1	-7.00	.00	.00	.00	.00	.00
11	.12	.80	.22	2	.00	1.00	-2.50	.00	.00	.00
12	.11	.70	.23	3	.00	.00	-1.50	.30	1.30	.60
13	.15	.20	.17	1	-5.00	.00	.00	.00	.00	.00
14	.20	.70	.14	2	.00	1.30	-3.30	.00	.00	.00
15	.41	.40	.11	3	.00	.00	.00	.60	1.30	.30
16	.31	.90	.03	1	-5.00	.00	.00	.00	.00	.00
17	.22	.80	.32	2	.00	1.70	-2.10	.00	.00	.00
18	.55	.50	.35	3	.00	.00	.00	.70	1.50	.60
19	.64	.70	.31	1	-9.00	.00	.00	.00	.00	.00
20	.46	.50	.29	2	.00	1.50	-4.10	.00	.00	.00

HYDROSTATIC INITIAL CONDITIONS

-.99995605001899220000000000000000D+00	.11278901782227220000000000000000D-03
-.99976843740926310000000000000000D+00	.26253494429644590000000000000000D-03
-.99943093746626140000000000000000D+00	.41246325442617630000000000000000D-03
-.99894352584340880000000000000000D+00	.56234895403140980000000000000000D-03
-.99830626647300650000000000000000D+00	.71215416347332070000000000000000D-03
-.99751925275672080000000000000000D+00	.86185370142008900000000000000000D-03
-.99658260202338160000000000000000D+00	.10114243932084410000000000000000D-02
-.99549645448109640000000000000000D+00	.11608435575677250000000000000000D-02
-.99426097292240970000000000000000D+00	.13100886819025050000000000000000D-02
-.99287634260882210000000000000000D+00	.14591373333107330000000000000000D-02
-.99134277120758310000000000000000D+00	.16079671307493270000000000000000D-02
-.98966048874506520000000000000000D+00	.17565557363307300000000000000000D-02
-.98782974756486060000000000000000D+00	.19048808534997190000000000000000D-02
-.98585082228612600000000000000000D+00	.20529202279661430000000000000000D-02
-.98372400976031550000000000000000D+00	.22006516498399110000000000000000D-02
-.98144962902546440000000000000000D+00	.23480529563273120000000000000000D-02
-.97902802125762200000000000000000D+00	.24951020347037070000000000000000D-02
-.97645954971923420000000000000000D+00	.26417768254274900000000000000000D-02
-.97374459970437040000000000000000D+00	.27880553253277070000000000000000D-02
-.97088357848074300000000000000000D+00	.29339155908297170000000000000000D-02
-.96787691522848950000000000000000D+00	.30793357411993380000000000000000D-02
-.96472506097570640000000000000000D+00	.32242939617941980000000000000000D-02
-.96142848853073220000000000000000D+00	.33687685073155510000000000000000D-02
-.95798769241117810000000000000000D+00	.35127377050563070000000000000000D-02
-.95440318876971620000000000000000D+00	.36561799581425020000000000000000D-02
-.95067551531662830000000000000000D+00	.37990737487662580000000000000000D-02

-.946805231239127500000000000000000D+00	.394139764140883300000000000000000D-02
-.942792917117462500000000000000000D+00	.408313028605266900000000000000000D-02
-.938639174837814800000000000000000D+00	.422425042138153600000000000000000D-02
-.934344627502003100000000000000000D+00	.436473687796805700000000000000000D-02
-.929909919334005700000000000000000D+00	.450456858144789700000000000000000D-02
-.925335715583316200000000000000000D+00	.464372455568006000000000000000000D-02
-.920622702425146500000000000000000D+00	.478218392589269100000000000000000D-02
-.915771586857490400000000000000000D+00	.491992592181386600000000000000000D-02
-.910783096595065100000000000000000D+00	.505692988078684300000000000000000D-02
-.905657979960144600000000000000000D+00	.519317525086928100000000000000000D-02
-.900397005770303600000000000000000D+00	.532864159391593100000000000000000D-02
-.895000963223084500000000000000000D+00	.546330858864431000000000000000000D-02
-.889470661777610900000000000000000D+00	.559715603368291100000000000000000D-02
-.883806931033158300000000000000000D+00	.573016385060143700000000000000000D-02
-.878010620604706600000000000000000D+00	.586231208692265300000000000000000D-02
-.872082599995488300000000000000000D+00	.599358091911533900000000000000000D-02
-.866023758466554600000000000000000D+00	.612395065556793300000000000000000D-02
-.859835004903376400000000000000000D+00	.625340173954240100000000000000000D-02
-.853517267679503000000000000000000D+00	.638191475210788100000000000000000D-02
-.847071494517296200000000000000000D+00	.650947041505366100000000000000000D-02
-.840498652345762700000000000000000D+00	.663604959378106500000000000000000D-02
-.833799727155504900000000000000000D+00	.676163330017379900000000000000000D-02
-.826975723850812500000000000000000D+00	.688620269544632000000000000000000D-02
-.820027666098917100000000000000000D+00	.700973909296982300000000000000000D-02
-.812956596176431600000000000000000D+00	.713222396107539000000000000000000D-02
-.805763574812998700000000000000000D+00	.725363892583391400000000000000000D-02
-.798449681032170700000000000000000D+00	.737396577381234600000000000000000D-02
-.791016011989546000000000000000000D+00	.749318645480588400000000000000000D-02
-.783463682808183800000000000000000D+00	.761128308454566000000000000000000D-02
-.775793826411325800000000000000000D+00	.772823794738155600000000000000000D-02
-.768007593352445600000000000000000D+00	.784403349893971300000000000000000D-02
-.760106151642655500000000000000000D+00	.795865236875434800000000000000000D-02
-.752090686575492100000000000000000D+00	.807207736287349900000000000000000D-02
-.743962400549111600000000000000000D+00	.818429146643827000000000000000000D-02
-.735722512885917800000000000000000D+00	.829527784623522600000000000000000D-02
-.727372259649652100000000000000000D+00	.840501985322153700000000000000000D-02
-.718912893459971500000000000000000D+00	.851350102502249100000000000000000D-02
-.710345683304543300000000000000000D+00	.862070508840101500000000000000000D-02
-.701671914348685100000000000000000D+00	.872661596169880800000000000000000D-02
-.692892887742577000000000000000000D+00	.883121775724875000000000000000000D-02
-.684009920426076000000000000000000D+00	.893449478375820700000000000000000D-02
-.675024344931162800000000000000000D+00	.903643154866287300000000000000000D-02
-.665937509182048500000000000000000D+00	.913701276045080600000000000000000D-02
-.656750776292973200000000000000000D+00	.923622333095630200000000000000000D-02
-.647465524363724800000000000000000D+00	.933404837762327100000000000000000D-02
-.638083146272911400000000000000000D+00	.943047322573775300000000000000000D-02
-.628605049469015000000000000000000D+00	.952548341062928500000000000000000D-02
-.619032655759261300000000000000000D+00	.961906467984072700000000000000000D-02
-.609367401096333900000000000000000D+00	.971120299526628000000000000000000D-02
-.599610735362968300000000000000000D+00	.980188453525732800000000000000000D-02
-.589764122154454300000000000000000D+00	.989109569669582900000000000000000D-02
-.579829038559083000000000000000000D+00	.997882309703491100000000000000000D-02
-.569806974936568700000000000000000D+00	.100650535763063800000000000000000D-01
-.559699434694481100000000000000000D+00	.101497741990948700000000000000000D-01
-.549507934062718600000000000000000D+00	.102329722564782200000000000000000D-01
-.539234001866059200000000000000000D+00	.103146352679340100000000000000000D-01

-.52887917929482230000000000000000D+00	.10394750983211730000000000000000D-01
-.51844501967367450000000000000000D+00	.10473307384170400000000000000000D-01
-.50793308822861610000000000000000D+00	.10550292686581480000000000000000D-01
-.49734496185218150000000000000000D+00	.10625695341896560000000000000000D-01
-.48668222886689030000000000000000D+00	.10699504038979790000000000000000D-01
-.47594648878698330000000000000000D+00	.10771707705804630000000000000000D-01
-.46513935207847930000000000000000D+00	.10842295511114800000000000000000D-01
-.45426243991759000000000000000000D+00	.10911256866049040000000000000000D-01
-.44331738394752730000000000000000D+00	.10978581425729570000000000000000D-01
-.43230582603374130000000000000000D+00	.11044259090813900000000000000000D-01
-.42122941801762380000000000000000D+00	.11108280009009840000000000000000D-01
-.41008982146871650000000000000000D+00	.11170634576553450000000000000000D-01
-.39888870743545910000000000000000D+00	.11231313439649670000000000000000D-01
-.38762775619451560000000000000000D+00	.11290307495875510000000000000000D-01
-.37630865699871640000000000000000D+00	.11347607895545490000000000000000D-01
-.36493310782365400000000000000000D+00	.11403206043039190000000000000000D-01
-.35350281511297000000000000000000D+00	.11457093598090640000000000000000D-01
-.34201949352237170000000000000000D+00	.11509262477039500000000000000000D-01
-.33048486566241700000000000000000D+00	.11559704854043640000000000000000D-01
-.31890066184010630000000000000000D+00	.11608413162253110000000000000000D-01
-.30726861979931910000000000000000D+00	.11655380094945240000000000000000D-01
-.29559048446013560000000000000000D+00	.11700598606620740000000000000000D-01
-.28386800765708180000000000000000D+00	.11744061914060550000000000000000D-01
-.27210294787633660000000000000000D+00	.11785763497343430000000000000000D-01
-.26029706999194250000000000000000D+00	.11825697100823980000000000000000D-01
-.24845214500105450000000000000000D+00	.11863856734071080000000000000000D-01
-.23656994975828400000000000000000D+00	.11900236672766490000000000000000D-01
-.22465226670913200000000000000000D+00	.11934831459563560000000000000000D-01
-.21270088362262600000000000000000D+00	.11967635904905890000000000000000D-01
-.20071759332312670000000000000000D+00	.11998645087805810000000000000000D-01
-.18870419342138880000000000000000D+00	.12027854356582570000000000000000D-01
-.17666248604490200000000000000000D+00	.12055259329560150000000000000000D-01
-.16459427756755390000000000000000D+00	.12080855895724550000000000000000D-01
-.15250137833865640000000000000000D+00	.12104640215340460000000000000000D-01
-.14038560241137590000000000000000D+00	.12126608720527320000000000000000D-01
-.12824876727060710000000000000000D+00	.12146758115794460000000000000000D-01
-.11609269356033280000000000000000D+00	.12165085378535500000000000000000D-01
-.10391920481050940000000000000000D+00	.12181587759481770000000000000000D-01
-.91730127163519550000000000000000D-01	.12196262783114710000000000000000D-01
-.79527289100232960000000000000000D-01	.12209108248037240000000000000000D-01
-.67312521165716400000000000000000D-01	.12220122227303970000000000000000D-01
-.55087655694633980000000000000000D-01	.12229303068710280000000000000000D-01
-.42854526536379100000000000000000D-01	.12236649395040160000000000000000D-01
-.30614968779979030000000000000000D-01	.12242160104272800000000000000000D-01
-.18370818478813670000000000000000D-01	.12245834369747920000000000000000D-01
-.61239123751895290000000000000000D-02	.12247671640289760000000000000000D-01
.61239123751895290000000000000000D-02	.12247671640289760000000000000000D-01
.18370818478813670000000000000000D-01	.12245834369747920000000000000000D-01
.30614968779979030000000000000000D-01	.12242160104272800000000000000000D-01
.42854526536379100000000000000000D-01	.12236649395040160000000000000000D-01
.55087655694633980000000000000000D-01	.12229303068710280000000000000000D-01
.67312521165716400000000000000000D-01	.12220122227303970000000000000000D-01
.79527289100232960000000000000000D-01	.12209108248037240000000000000000D-01
.91730127163519550000000000000000D-01	.12196262783114710000000000000000D-01
.10391920481050940000000000000000D+00	.12181587759481770000000000000000D-01
.11609269356033280000000000000000D+00	.12165085378535500000000000000000D-01

.128248767270607100000000000000000D+00	.121467581157944600000000000000000D-01
.140385602411375900000000000000000D+00	.121266087205273200000000000000000D-01
.152501378338656400000000000000000D+00	.121046402153404600000000000000000D-01
.164594277567553900000000000000000D+00	.120808558957245500000000000000000D-01
.176662486044902000000000000000000D+00	.120552593295601500000000000000000D-01
.188704193421388800000000000000000D+00	.120278543565825700000000000000000D-01
.200717593323126700000000000000000D+00	.119986450878058100000000000000000D-01
.212700883622626000000000000000000D+00	.119676359049058900000000000000000D-01
.224652266709132000000000000000000D+00	.119348314595635600000000000000000D-01
.236569949758284000000000000000000D+00	.119002366727664900000000000000000D-01
.248452145001054500000000000000000D+00	.118638567340710800000000000000000D-01
.260297069991942500000000000000000D+00	.118256971008239800000000000000000D-01
.272102947876336600000000000000000D+00	.117857634973434300000000000000000D-01
.283868007657081800000000000000000D+00	.117440619140605500000000000000000D-01
.295590484460135600000000000000000D+00	.117005986066207400000000000000000D-01
.307268619799319100000000000000000D+00	.116553800949452400000000000000000D-01
.318900661840106300000000000000000D+00	.116084131622531100000000000000000D-01
.330484865662417000000000000000000D+00	.115597048540436400000000000000000D-01
.342019493522371700000000000000000D+00	.115092624770395000000000000000000D-01
.353502815112970000000000000000000D+00	.114570935980906400000000000000000D-01
.364933107823654000000000000000000D+00	.114032060430391900000000000000000D-01
.376308656998716400000000000000000D+00	.113476078955454900000000000000000D-01
.387627756194515600000000000000000D+00	.112903074958755100000000000000000D-01
.398888707435459100000000000000000D+00	.112313134396496700000000000000000D-01
.410089821468716500000000000000000D+00	.111706345765534500000000000000000D-01
.421229418017623800000000000000000D+00	.111082800090098400000000000000000D-01
.432305826033741300000000000000000D+00	.110442590908139000000000000000000D-01
.443317383947527300000000000000000D+00	.109785814257295700000000000000000D-01
.454262439917590000000000000000000D+00	.109112568660490400000000000000000D-01
.465139352078479300000000000000000D+00	.108422955111148000000000000000000D-01
.475946488786983300000000000000000D+00	.107717077058046300000000000000000D-01
.486682228866890300000000000000000D+00	.106995040389797900000000000000000D-01
.497344961852181500000000000000000D+00	.106256953418965600000000000000000D-01
.507933088228616100000000000000000D+00	.105502926865814800000000000000000D-01
.518445019673674500000000000000000D+00	.104733073841704000000000000000000D-01
.528879179294822300000000000000000D+00	.103947509832117300000000000000000D-01
.539234001866059200000000000000000D+00	.103146352679340100000000000000000D-01
.549507934062718600000000000000000D+00	.102329722564782200000000000000000D-01
.559699434694481100000000000000000D+00	.101497741990948700000000000000000D-01
.569806974936568700000000000000000D+00	.100650535763063800000000000000000D-01
.579829038559083000000000000000000D+00	.997882309703491100000000000000000D-02
.589764122154454300000000000000000D+00	.989109569669582900000000000000000D-02
.599610735362968300000000000000000D+00	.980188453525732800000000000000000D-02
.609367401096333900000000000000000D+00	.971120299526628000000000000000000D-02
.619032655759261300000000000000000D+00	.961906467984072700000000000000000D-02
.628605049469015000000000000000000D+00	.952548341062928500000000000000000D-02
.638083146272911400000000000000000D+00	.943047322573775300000000000000000D-02
.647465524363724800000000000000000D+00	.933404837762327100000000000000000D-02
.656750776292973200000000000000000D+00	.923622333095630200000000000000000D-02
.665937509182048500000000000000000D+00	.913701276045080600000000000000000D-02
.675024344931162800000000000000000D+00	.903643154866287300000000000000000D-02
.684009920426076000000000000000000D+00	.893449478375820700000000000000000D-02
.692892887742577000000000000000000D+00	.883121775724875000000000000000000D-02
.701671914348685100000000000000000D+00	.872661596169880800000000000000000D-02
.710345683304543300000000000000000D+00	.862070508840101500000000000000000D-02
.718912893459971500000000000000000D+00	.851350102502249100000000000000000D-02

.72737225964965210000000000000000D+00	.84050198532215370000000000000000D-02
.73572251288591780000000000000000D+00	.82952778462352260000000000000000D-02
.74396240054911160000000000000000D+00	.81842914664382700000000000000000D-02
.75209068657549210000000000000000D+00	.80720773628734990000000000000000D-02
.76010615164265550000000000000000D+00	.79586523687543480000000000000000D-02
.76800759335244560000000000000000D+00	.78440334989397130000000000000000D-02
.77579382641132580000000000000000D+00	.77282379473815560000000000000000D-02
.78346368280818380000000000000000D+00	.76112830845456600000000000000000D-02
.79101601198954600000000000000000D+00	.74931864548058840000000000000000D-02
.79844968103217070000000000000000D+00	.73739657738123460000000000000000D-02
.80576357481299870000000000000000D+00	.72536389258339140000000000000000D-02
.81295659617643160000000000000000D+00	.71322239610753900000000000000000D-02
.82002766609891710000000000000000D+00	.70097390929698230000000000000000D-02
.82697572385081250000000000000000D+00	.68862026954463200000000000000000D-02
.83379972715550490000000000000000D+00	.67616333001737990000000000000000D-02
.84049865234576270000000000000000D+00	.66360495937810650000000000000000D-02
.84707149451729620000000000000000D+00	.65094704150536610000000000000000D-02
.85351726767950300000000000000000D+00	.63819147521078810000000000000000D-02
.85983500490337640000000000000000D+00	.62534017395424010000000000000000D-02
.86602375846655460000000000000000D+00	.61239506555679330000000000000000D-02
.87208259999548830000000000000000D+00	.59935809191153390000000000000000D-02
.87801062060470660000000000000000D+00	.58623120869226530000000000000000D-02
.88380693103315830000000000000000D+00	.57301638506014370000000000000000D-02
.88947066177761090000000000000000D+00	.55971560336829110000000000000000D-02
.89500096322308450000000000000000D+00	.54633085886443100000000000000000D-02
.90039700577030360000000000000000D+00	.53286415939159310000000000000000D-02
.90565797996014460000000000000000D+00	.51931752508692810000000000000000D-02
.91078309659506510000000000000000D+00	.50569298807868430000000000000000D-02
.91577158685749040000000000000000D+00	.49199259218138660000000000000000D-02
.92062270242514650000000000000000D+00	.47821839258926910000000000000000D-02
.92533571558331620000000000000000D+00	.46437245556800600000000000000000D-02
.92990991933400570000000000000000D+00	.45045685814478970000000000000000D-02
.93434462750200310000000000000000D+00	.43647368779680570000000000000000D-02
.93863917483781480000000000000000D+00	.42242504213815360000000000000000D-02
.94279291711746250000000000000000D+00	.40831302860526690000000000000000D-02
.94680523123912750000000000000000D+00	.39413976414088330000000000000000D-02
.95067551531662830000000000000000D+00	.37990737487662580000000000000000D-02
.95440318876971620000000000000000D+00	.36561799581425020000000000000000D-02
.95798769241117810000000000000000D+00	.35127377050563070000000000000000D-02
.96142848853073220000000000000000D+00	.33687685073155510000000000000000D-02
.96472506097570640000000000000000D+00	.32242939617941980000000000000000D-02
.96787691522848950000000000000000D+00	.30793357411993380000000000000000D-02
.97088357848074300000000000000000D+00	.29339155908297170000000000000000D-02
.97374459970437040000000000000000D+00	.27880553253277070000000000000000D-02
.97645954971923420000000000000000D+00	.26417768254274900000000000000000D-02
.97902802125762200000000000000000D+00	.24951020347037070000000000000000D-02
.98144962902546440000000000000000D+00	.23480529563273120000000000000000D-02
.98372400976031550000000000000000D+00	.22006516498399110000000000000000D-02
.98585082228612600000000000000000D+00	.20529202279661430000000000000000D-02
.98782974756486060000000000000000D+00	.19048808534997190000000000000000D-02
.98966048874506520000000000000000D+00	.17565557363307300000000000000000D-02
.99134277120758310000000000000000D+00	.16079671307493270000000000000000D-02
.99287634260882210000000000000000D+00	.14591373333107330000000000000000D-02
.99426097292240970000000000000000D+00	.13100886819025050000000000000000D-02
.99549645448109640000000000000000D+00	.11608435575677250000000000000000D-02
.99658260202338160000000000000000D+00	.10114243932084410000000000000000D-02

.997519252756720800000000000000000D+00	.861853701420089000000000000000000D-03
.998306266473006500000000000000000D+00	.712154163473320700000000000000000D-03
.998943525843408800000000000000000D+00	.562348954031409800000000000000000D-03
.999430937466261400000000000000000D+00	.412463254426176300000000000000000D-03
.999768437409263100000000000000000D+00	.262534944296445900000000000000000D-03
.999956050018992200000000000000000D+00	.112789017822272000000000000000000D-03