TR- 104 Vol 1
1980

Heat Transport in Groundwater Systems – Finite Element Model

E. K. Grubaugh
D.L. Reddell

Texas Water Resources Institute

Texas A&M University

RESEARCH PROJECT COMPLETION REPORT

Project Number A-039-TEX

(July 1, 1976 - December 31, 1978)

Agreement Numbers

14-34-0001-7091
14-34-0001-7092
14-34-0001-8046

HEAT TRANSPORT IN GROUNDWATER SYSTEMS
VOL. I - FINITE ELEMENT MODEL

Elston Kent Grubaugh

Donald L. Reddell

The work on which this publication is based was supported in part by funds provided by the Office of Water Research and Technology (Project A-039-TEX), U.S. Department of the Interior, Washington, D. C., as authorized by the Water Research and Development Act of 1978.

Technical Report No. 104-Vol. I
Texas Water Resources Institute
Texas A&M University

August 1980

Contents of this publication do not necessarily reflect the views and policies of the Office of Water Research and Technology, U. S. Department of the Interior, nor does mention of trade names or commercial products constitute their endorsement or recommendation for use by the U.S. Government.

ABSTRACT

Heat Transport in Groundwater Systems

Vol. I - Finite Element Model

by

Elston Kent Grubaugh

and

Donald L. Reddell

Solar energy is a possible alternate energy source for space heating. A method of economic long term solar energy storage is needed. Researchers have proposed storing solar energy by heating water using solar collectors and injecting the hot water into groundwater aquifers for long term energy storage. Of paramount importance to the success of such a system is the quality and the behavior of the aquifer used for hot water storage. In general, the problem is to obtain an accurate prediction of the response of an aquifer system and its basic components to the operation of a system of injection and pumping wells which are transporting water at a notably different temperature than the natural groundwater.

The injection of hot water into a groundwater storage system will have a pronounced effect on the specific storage and mass flow within the aquifer. These effects will result from differences in viscosity, density, specific heat, and thermal conductivity between the injected water and the natural groundwater. A complex system of energy and mass transport will result, making analytical solutions unattainable or very complex.

The objective of this study was to develop a numerical simulation which would predict the pressure and temperature of water in a groundwater system at any time in response to the pumping and injecting of hot and cold water. A numerical model was developed in which the groundwater flow equation and the energy transport equation are solved simultaneously using a finite difference approximation for the time derivative and three-dimensional Galerkin-finite element approximations for the space derivatives.

The use of a strict Galerkin approach led to unacceptable solution oscillations in sharp temperature front problems (i.e., problems where the temperature changes quickly over a small distance or time). Several techniques were tried in an attempt to correct the problem. Reduction of element and time step size proved ineffective in eliminating the sharp temperature front oscillation problem. An upstream weighting scheme corrected the oscillation problem, but resulted in an unacceptable smear of the sharp temperature front. A mass lumping scheme resulted in the best solution to sharp temperature front problems. The mass lumping scheme yielded solutions without the oscillation problem and with less smear than the upstream weighting scheme.

ACKNOWLEDGEMENTS

The authors wish to express appreciation to Dr. Walter E. Haisler, Department of Aerospace at Texas A&M University for his constant encouragement and constructive comments during this study. Thanks are also expressed to Dr. Jack R. Runkles, Dr. James H. Ruff, Dr. Terry A. Howell, and Dr. John L. Nieber for their review of the manuscript and helpful comments and corrections. A special thank you is extended to Dr. Jim Mercer with the U. S. Geological Survey in Reston, Virginia for visiting with the authors about this project, and for making many helpful comments and suggestions.

The authors are also grateful to the Department of Agricultural Engineering (Dr. E. A. Hiler, Head) and the Texas Water Resources Institute (Dr. J. R. Runkles, Director) at Texas A&M University for providing the financial assistance for this study.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

In an era of rising energy cost, the rationale for the development of alternate energy sources is obvious. The harsh winters of 1976-77 and 1977-78 clearly demonstrated the growing need for new or additional sources of energy for space and water heating.

As summarized by Smith and Hill (1976), the more abundant and less expensive energy alternatives of the immediate future included:

A. nuclear,

B. lowgrade coal,

C. geothermal,

D. fusion, and

E. solar.

The first four of these "abundant" energies are not directly applicable to space heating. They are more suitably used through large electrical power generation stations. Solar energy, however, is more conveniently used for space heating than for electrical generation.

Solar energy is not a "new" energy source. Solar evaporation of salt brines for salt recovery has been practiced for centuries, as has the solar drying of agricultural products. Hottel and Woertz (1942) mentioned that solar heat was used to pump water in 1885. Solar collectors were also used for domestic water heating in Florida and California prior to 1940. Solar energy has been experiencing a period of renewed and increased interest since the early 1970's as a result of the increasing cost of energy from traditional resources, and the problems of developing and refining fuels acceptable to current environmental standards. The mean solar insulation on a flat surface averages about 6.8 x 10^9 J m^{-2} yr^{-1} in most of the United States. With the current level of solar technology, probably 50 percent of this energy could be utilized in space heating. However, the design and application of solar heating systems presents some unique problems due to two factors;

A. heat is needed only in cold weather when conditions for the collection of solar energy are least favorable; that is, the intermittent nature of solar energy, and

B. the capital intensive nature of current hot water storage systems limits the use of solar energy for heating to only a few days, resulting in the need for a conventional standby heating system.

These factors could be overcome by developing a method of storing hot water over a period of seasons or years. Additional heat sources such as electrical generating plant coolant discharge could also be utilized in such a system.

Much of the earth's surface is underlain by groundwater aquifers. These aquifers vary greatly in depth, thickness, porosity, and water quality, but most are capable of yielding or receiving large volumes of water. Also, the earth is an excellent insulator and large volumes of hot water will stay hot for long periods of time when stored in groundwater aquifers. Considering these facts, one has the basis of a natural system for long term energy storage.

Objectives

In general, the problem is to obtain an accurate prediction of the response of a multiple aquifer system and its basic components to the operation of a system of injection and pumping wells which are transporting water at a notably different temperature than the natural groundwater. More specifically, the objectives are:

1) The development of a numerical model to evaluate the non-steady, areal, hydrostatic pressure distribution and water movement in a multiple aquifer groundwater system.
2) The development of a numerical model to evaluate the non-steady, areal, heat transfer and temperature distribution in a multiple aquifer groundwater system.
3) To combine the models developed in (1) and (2) above into a numerical simulation which will predict the water temperature and pressure throughout a multiple aquifer system at any time when hot and cold pumping and injection wells are operated in the system.

The resulting model will be of value in the design, operational evaluation, and environmental evaluation of well systems for the storage and subsequent recovery of industrial waste heat and seasonally collected solar heat.

Methods of Investigation

A three-dimensional model of heat and mass transport in a multiple aquifer groundwater system, with non-homogeneous and anisotropic components which are temperature and pressure dependent, can be developed from;

A. Darcy's Law,

B. Fourier's Law,

C. conservation of mass,

D. conservation of energy, and

E. equations of state, expressed as tabular data or empirical formulas.

The model takes the form of two coupled differential equations.

Two general methods have evolved for the numerical solution of partial differential equations. The more common finite difference method approximates the partial differential equations and boundary conditions with finite difference equations at discrete points in the solution domain. The second approach, known as the finite element method and described by Zienkiewiez (1971, 1977), Huebner (1975), Conner and Brebbia (1976) and Pinder (1977), is based on the application of variational calculus to find an alternate formulation of the problem. Unlike the finite difference method, which envisions the solution region as an array of grid points, the basic premise of the finite element method is that a solution region can be analytically modeled by replacing it with an assemblage of discrete elements.

In practice, finite difference methods have enjoyed greater popularity for generating approximate solutions to groundwater prolems than have finite element methods. Although the finite element approach has often been found to be more flexible than finite difference methods, the variational formulation is difficult. An alternative way of formulating a problem for finite element solution without using variational principles is offered by the Galerkin type methods which, as summarized by Douglas and Dupont (1970), have been used extensively in the petroleum industry for generating approximating equations.

In this work, use has been made of mixed, three-dimensional, isoparametric finite elements that can be deformed in the x, y and z directions. This approach allows economy of nodal location since higher order elements, and hence more nodes, can be used to describe complex boundaries, directional problems, and regions of rapid variation of the field variable, while lower order elements and fewer nodes can be used in other regions.

CHAPTER II

LITERATURE REVIEW

The concept of storing hot water in groundwater aquifers was suggested by Meyer and Todd (1973). Meyer and Todd's preliminary calculations for a 0.05 m^3 sec^{-1} well receiving 99°C water from an electrical power plant, indicated that after the fifth 90-day injection-recovery cycle, 86 percent of the injected heat could be recovered. The first through the fifth cycles were less efficient because some of the injected heat was required to bring the aquifer from its natural temperature of 16°c to equilibrium with the injected water. At the end of an injection period, hot water injected into a 33 m thick aquifer formed an inverted cone 200 m in radius. An equivalent insulated tank would be 30 m tall and 120 m in diameter. The amount of energy stored in a temperature range suitable for space heating was 2.4×10^{13} J, the equivalent of 5.9×10^3 m^3 of natural gas.

Aziz et al. (1973) considered a hot fluid industrial plant using deep lying groundwater. The water was cooled by heat exchangers and reinjected into another part of the aquifer. The efficiency of this process was shown to be correlated with pumping rates, the thermal conductivities of the aquifer layers, and the presence of thermal convection effects.

Rabbimov et al. (1974) conducted an experimental study of aquifer heating in solar energy accumulation. Gringarten and Sauty (1975) developed analytical solutions for the case of a rechargingdischarging well pair in an infinite, horizontal, homogeneous and isotropic aquifer.

Davison, Harris, and Martin (1975) proposed the "Solaterre" system for residential heating. Hot water produced by solar heaters in the summer would be stored in an aquifer for space heating use in winter. The water would be pumped from the aquifer in winter and circulated through a pipe network distribution system to provide heat for a large number of buildings. In the same system, the winter's cold could be used to chill water in a cooling pond. The chilled water would be stored in a separate part of the aquifer until summer, and then circulated through the same distribution system for air conditioning. After circulation, the water is returned and injected into a separate part of the aquifer, making the system regenerative. Ebeling and Reddell (1976) further discussed the Solaterre system and developed the basis for a three-dimensional model of heat and mass transfer in a non-homogeneous, anisotropic aquifer for a single pumping and injecting well.

Of paramount importance to the success of Solaterre and similar systems is the quality and the behavior of the aquifer used for pumping and injection. While the actual pumping and injection processes may directly involve only a single aquifer, it has been established over the last twenty years that if aquifer behavior is to be considered, then multiaquifer systems must be considered, also.

A multiple aquifer system is a succession of aquifers separated by aquitards and/or aquicludes of varying thicknesses and of relatively low permeability. If these intervening layers are sufficiently low in permeability, they act as no flow boundaries. However, as permeability increases, leakage and cross-flow become appreciable and flow is no longer restricted solely to the aquifer undergoing pumping or injection stress. The additional flow is attributed primarily to changes in storage in the aquitards and the adjoining unused aquifers. Since flow is not restricted to the stressed aquifer alone, the leaky aquifer must be considered as part of a complex multiple aquifer system where the behavior of each layer depends on the flow behavior of the entire system. Therefore, one cannot develop a complete understanding of flow in the stressed aquifer without analyzing the system as a whole.

A rigorous approach to flow in multiple aquifer systems involves complex partial differential equations and boundary conditions which create serious mathematical difficulties. As a result, analytical solutions are unattainable, or extremely complex. Recent work in this area has made extensive use of numerical simulations.

In early attempts at numerical simulation of multiple aquifer systems the mathematics of the problem were simplified by making two assumptions;

A. that vertical components of flow within the stressed aquifer are negligible, and

B. that horizontal components of flow in confining layers are negligible.

In considering steady-state problems, the assumption of vertical flow in the aquitards implies that the rate of leakage into or from a stressed aquifer is proportional to the change in potential across the leaky aquifer. Using these assumptions, Fayers and Sheldon (1962) considered the problem of evaluating the magnitude and direction of steady state flow in aquifer systems within a geologic basin. Freeze and Witherspoon (1966) applied both the analytical and numerical approach to the two-dimensional steady regional flow problem.

Newman and Witherspoon (1969a) realized that under transient conditions the assumption of leakage being proportional to the potential change across an aquitard is identical to ignoring the storage capacity of the aquitard. They concluded that the common assumptions could lead to serious errors unless the transmissibilities of the unstressed aquifers are significantly greater than that of the stressed aquifer.

Newman and Witherspoon (1969b) developed an analytic solution for flow to a well in a confined, infinite radial system composed of two aquifers separated by an aquitard. The analytic solutions were evaluated numerically for several cases, and the validity of these solutions were verified by the finite element method of simulation. Their solution differed from previous work in that the effects of storage in the aquitard and drawdown in the unstressed aquifer were both considered.

Using a two-dimensional finite element scheme, Javandel and Witherspoon (1969) analyzed drawdown in a two layer system with a permeability contrast of 100:1. A thirteen-layer multiple aquifer system was also evaluated and the results were found to be in general agreement with the two layer case.

Bredehoeft and Pinder (1970) developed a quasi-three-dimensional finite difference model of areal flow in a multiple aquifer system.

The model was based on the modified leaky aquifer theory of Hantush (1960). From the results of the model it was apparent that;

A. the rate of flux into or from a stressed aquifer changes over several orders of magnitude with time,

B. long times are necessary before steady flow can be established through a clay confining layer.

Bredehoeft and Pinder concluded that a three-dimensional model was needed to fully and accurately characterize a multiple aquifer system.

Freeze (1971) used finite difference techniques to develop a three-dimensional model for the treatment of saturated-unsaturated transient flow in small, non-homogeneous, anisotropic geologic basins. The unsaturated zone was included in a basin wide model with confined and unconfined saturated aquifers under both natural and developed conditions. This integrated analysis dispensed with the need for complex free surface calculations and the invocation of Dupuit-Forscheimer assumptions. The model assumed water density, water compressibility, soil specific permeability, porosity, and vertical soil compressibility as functions of potential.

Gupta et al. (1975) used finite element techniques to develop a three-dimensional model for the solution of both the fluid flow equation and the convection-dispersion equation for a multiple layer groundwater system. Gupta applied the model to a complex groundwater regime in Sutter Basin, California, to simulate rising connate waters through a vertical fault. The effects of layering of geologic formations and the vertical fault were analyzed for both water flow and mass transport of salts.

The injection of water into a groundwater storage system has a pronounced effect on the specific storage and mass flow within the aquifer. These affects are due to differences in viscosity, density, specific heat, and thermal conductivity between the injected water and the natural groundwater. A complex system of energy and mass transport results, making analytical solutions unattainable or very complex.

In recent years several attempts at numerical modeling of energy and mass transfer in porous media have been made. Chappelear and Volek (1969) considered the problem of prediction of fluid flow and temperature distribution in an initially cold-fluid-filled reservoir upon the injection of a similar hot fluid. They developed a two-dimensional finite difference model for heat and mass transport in a homogeneous and isotropic media, assuming viscosity to be temperature dependent. The media and fluid densities were considered constant. Comparison of numerical and experimental results were in reasonable agreement.

Mercer (1973) attempted to model the Wairokei, New Zealand, thermal system under both steady state and transient exploitation conditions. He obtained a numerical solution of the coupled partial differential equations for mass and energy transfer by Galerkin finite element techniques using mixed isoparametric two-dimensional finite elements. Ergataudis, Ivons, and Zienkiewicz (1968) first described this type of element in structural mechanics applications.

Wilhite and Wagner (1974) modeled a proposed waste heat disposal system consisting of a homogeneous semiconfined aquifer with a hydraulic connection to the bottom of a river. Heat and fluid flow were described by three-dimensional finite difference equations, which were solved by an alternating-direction iteration procedure. The porous media and fluid were considered compressible, but density was independent of temperature and viscosity was pressure independent. The model was applied to a prototype at Bonner Springs, Kansas, with 0.02 m^3 sec^{-1} injection wells and 32°c injection water. Ninety percent of the injected heat was predicted to enter the river.

Sorey (1976) analyzed heat and mass transfer associated with hot spring geothermal systems using both cylindrical and fault plane models. In the three-dimensional finite difference simulations, fluid density, viscosity, and specific heat were considered as functions of temperature and pressure. No mass source or sink terms were involved in the models.

Faust and Mercer (1977) presented a two-dimensional areal heat transport model capable of simulating both water and vapor dominated geothermal reservoirs. Finite difference techniques were used to solve for the dependent variables of pressure and enthalpy. The program was designed to simulate time dependent problems associated with geothermal reservoirs undergoing exploitation.

CHAPTER III

MATHEMATICAL MODEL

The solution of the hot water injection heat transport problem requires the simultaneous solution of a set of coupled equations;

A. the equation governing fluid flow, and

B. the equation governing the convective-conductive transport of energy.

A detailed development of these equations is given in Appendix A.

Equation of Fluid Flow

The equation for saturated, single phase groundwater fluid flow is written as

$$\nabla \cdot \left[\frac{kx_i \; \rho}{\mu} \left(\frac{\partial P}{\partial x_i} + \rho g \frac{\partial h}{\partial x_i}\right)\right] + Q = (\rho\alpha + \theta\rho_0\beta_p) \frac{\partial P}{\partial t} + \theta\rho_0 \; \beta_T \frac{\partial T}{\partial t} \qquad (1)$$

where

kx_i = intrinsic permeability in the x_i direction, L^2;

ρ = fluid density, $M \; L^{-3}$;

μ = fluid dynamic viscosity, FTL^{-2};

α = vertical compressibility of porous media, $L^2 \; F^{-1}$;

θ = porosity of porous media;

ρ_0 = reference density of fluid, $M \; L^{-3}$;

β_p = fluid compressibility coefficient, $L^2 \; F^{-1}$;

β_T = coefficient of thermal volume expansion (liquid), deg^{-1};

T = Temperature, deg, and

t = time

P = fluid pressure, $F\ L^{-2}$;

g = acceleration of gravity, $L\ T^{-2}$;

h = elevation above a datum, L;

Q = the strength of a mass source or sink term, $M\ T^{-1}\ L^{-3}$;

The source or sink term, Q, is defined by;

$$Q = \sum_{i=1}^{n} Q_w\ (x_i,\ y_i,\ z_i)\ \delta\ [(x - x_i),\ (y - y_i),\ (z - z_i)]$$

where

Q_w = net rate of mass flow per unit volume of porous media due to the operation of pumping and injecting wells, $M\ T^{-1}\ L^{-3}$;

δ = Dirac delta function

The Dirac delta function, $\delta(x - a)$ is defined by the equations

$$\delta(x - a) = 0 \text{ if } x \neq a$$

$$\int_{x_1}^{x_2} f(x) \qquad (x - a)\ dx = f(a)$$

for all x_1, x_2 such that $x_1 < a < x_2$ for an arbitrary integrable function $f(x)$.

Equation (1) is a nonlinear, second order, partial differential equation describing the spatial distribution of pressure in a volume of saturated porous medium and the changes in that pressure distribution with time. The first term in equation (1) represents the net inflow of the fluid per unit volume of porous media due to flow in response to the potential term. The next term represents the net rate of mass flow added or withdrawn per unit volume of porous media due to the operation of source and sink terms. The third term represents the rate of gain of the fluid per unit volume and the last term represents the rate of mass change of fluid per unit volume due to temperature changes.

Equation of Heat Transport

The convective-conductive equation for heat transport in a porous media may be written as

$$\frac{\partial}{\partial x}(K_x \frac{\partial T}{\partial x}) + \frac{\partial}{\partial y}(K_y \frac{\partial T}{\partial y}) + \frac{\partial}{\partial z}(K_z \frac{\partial T}{\partial z}) - C_p(v_x \frac{\partial T}{\partial x} + v_y \frac{\partial T}{\partial y} + v_z \frac{\partial T}{\partial z})$$

$$+ C_p \dot{Q}_{in}(T_{inj} - T) + C_p \dot{Q}_{out} (T - T_p)$$

$$= (\rho_f C_f (1 - \theta) + \rho_w C_p \theta) \frac{\partial T}{\partial t} + \Phi \qquad (2)$$

where

K = effective thermal conductivity, $F\ T^{-1}\ deg^{-1}$;

v = mass fluid flux, $M\ L^{-2}\ T^{-1}$;

C_f = specific heat of media material, $F\ L\ M^{-1}$;

C_p = fluid specific heat, $F\ L\ M^{-1}\ deg^{-1}$;

ρ_f = density of media material, $M\ L^{-3}$;

ρ_w = density of fluid, $M\ L^{-3}$;

$\dot{Q}_{in}$ = rate of mass inflow from source terms, per unit volume of porous media, $M\ T^{-1}\ L^{-3}$;

$\dot{Q}_{out}$ = rate of mass outflow due to sink terms, per unit volume of porous media, $M\ T^{-1}\ L^{-3}$;

T = aquifer temperature, deg;

T_{inj} = temperature of injected fluid, deg,

T_p = temperature of fluid removed by sink terms deg; and

Φ = an artificial dissipation term, $F\ L\ T^{-1}\ L^{-3}$

Equation (2) is a nonlinear, second order, hyperbolic partial differential equation which describes the temperature distribution, both spatial and temporal, in a porous media. The first term represents the rate of energy input per unit volume by effective thermal conduction, the second term represents the rate of energy input per unit volume by fluid convection. The next two terms describe the rate of energy inputs and withdrawals due to mass source and sink terms. The next term represents the net rate of gain of energy for both the solid phase of the porous media and the fluid. In order to insure continuity between the convective term and the heat storage term, an artificial dissipation term is added. In this case, the term is a differential mass storage term and is derived in Appendix A. In the development of the energy flow equation, terms allowing for vertical aquifer compressibility canceled out. The artificial dissipation term is based on the principle that the integral of the divergence of the fluid mass flux is equal to the time rate of change of mass storage for a differential volume element. The term is added to allow for the energy content of fluid stored in a domain during a given time step.

Assumptions

The following major assumptions were made in the derivation of equations (1) and (2).

A. Both Darcy's Law and Fourier's Law are valid.

B. Fluid density can be expanded about ρ_o, a reference density, in a Taylor series truncated with first order terms in temperature and pressure.

C. Instantaneous thermal equilibrium is assumed between the fluid and solid phases at any point in the porous media.

According to (C), the temperature represents an average of the fluid and solid temperatures. Heat transfer by all mechanisms other than convection is approximated by the concept of "effective thermal conductivity." All other assumptions made in the derivation of equations (1) and (2) are given in the appendix.

Fluid Mass Flux

The energy equation is coupled to the fluid flow equation by the fluid mass flux. The components of the fluid mass flux may be obtained from Darcy's Law expressed as

$$v_{x_i} = - \frac{k x_i \rho}{\mu} \left(\frac{\partial P}{\partial x_i} + \rho g \frac{\partial h}{\partial x_i}\right). \tag{3}$$

Fluid Properties and Effective Thermal Conductivity

Accurate prediction of fluid and energy transfer requires accurate values of fluid viscosity, specific heat, and thermal conductivity. Vargaftick (1975) presents the data for water in tabular form. The specific heat of typical aquifer constituent elements are available from Kelly (1949). Assad (1955) presented values for the thermal conductivity of rock solids. Assad (1955) also presented a correlation between rock and fluid properties and the thermal conductivity of fluid saturated rock as;

$$\frac{K}{k_1} = \left(\frac{k_2}{k_1}\right)^{\theta}$$

where

K = effective thermal conductivity; $F\ T^{-1}\ \mathrm{deg}^{-1}$

k_1 = thermal conductivity of rock solids; $F\ T^{-1}\ \mathrm{deg}^{-1}$

k_2 = thermal conductivity of the saturating fluid; $F\ T^{-1}\ \mathrm{deg}^{-1}$

θ = porosity.

CHAPTER IV

NUMERICAL MODEL

Partial differential equations (1) and (2) presented in Chapter III are generally impossible to solve analytically, and therefore a numerical scheme is required. As discussed in Chapter I, the Finite Element Method was selected for use in this study. Three-dimensional, mixed, isoparametric finite elements were used.

In any continuum problem, the field variable under consideration possesses infinitely many values as a result of the variable being a function of each point in the solution region. The finite element method reduces the problem to one with a finite number of unknowns by dividing the solution region into a finite number of elements, and by expressing the unknown field variable in terms of specified approximating functions defined over each element. These approximating functions, or basis functions, are defined in terms of values of the field variable at specified nodal points. The nodal value of the field variable and the basis functions for an element define the behavior of the field variable within the element. For a finite element representation of a problem, the nodal values of the field variables are the unknowns.

Basis Functions

The first step of any finite element analysis is to divide the solution region into elements and locate nodal points appropriate for the type of elements used. Huebner (1975), Pinder (1977), and

Zienkiewicz (1977) present discussions of the various elements in common use.

After a solution domain has been divided into elements, the unknown field variable in each element can be represented as a linear combination of nodal values of the field variable and linearly independent basis functions. This is a standard finite element methodology. Hence, the unknown field variable within each element (e) can be represented as

$$\phi^{(e)} = N_1\phi_1 + N_2\phi_2 + \ldots + N_L\phi_L$$

where

N_i = basis function for node "i",

ϕ_i = nodal value of the field variable,

L = number of nodes in element.

In matrix notation

$$\phi^{(e)} = \sum_{i=1}^{L} N_i\phi_i = [N^{(e)}] \{\phi^{(e)}\}$$

where $[N^{(e)}]$ is a row vector of basis functions for the element and $\{\phi^{(e)}\}$ is a column vector of unknown nodal values of ϕ associated with the element.

If the domain is divided into n elements, the complete representation of the field variable over the entire solution region is given by

$$\phi = \sum_{e=1}^{n} \phi^{(e)} = \sum_{e=1}^{n} [N^{(e)}] \{\phi^{(e)}\}$$

Hence, the unknown exact solution of the field variable can be approximated by a function $\bar{\phi}$ such that

$$\phi \simeq \bar{\phi} = \sum_{e=1}^{n} \sum_{i=1}^{L} N_i \phi_i = \sum_{j=1}^{M} N_j \phi_j$$

where M = number of nodes in the domain.

In this work, piecewise continuous polynomial basis functions are used with three-dimensional, mixed, isoparametric, quadrilateral elements. The polynomial may be linear, quadratic, or cubic where there are two, three, or four nodes on a side, respectively. To facilitate integration over the volume of an element, a dimensionless local (ξ,η,ζ) coordinate system is introduced where the element appears as a cube with the side nodes located at midpoints (quadratic sides) or 1/3 points (cubic sides). The essential idea underlying the use of this type of element centers on mapping simple geometric shapes in a local coordinate system into distorted shapes in the global Cartesian coordinate system, and then evaluating the element equations for the distorted elements that result. Figure 1 shows elements in local coordinates with node locations for linear, quadratic and cubic elements. Figure 2 presents a deformed, mixed, isoparametric element in global and local coordinates. The basis functions are defined with the condition that N_i is unity at node i and zero at all other nodes associated with the element (Figure 3). Under this condition the undetermined nodal values of the field variables are equal to the approximating functions $\bar{\phi}_i$ at the M node points.

The basis functions used here are presented in Table 1. These functions fulfill both the compatibility and completeness requirements

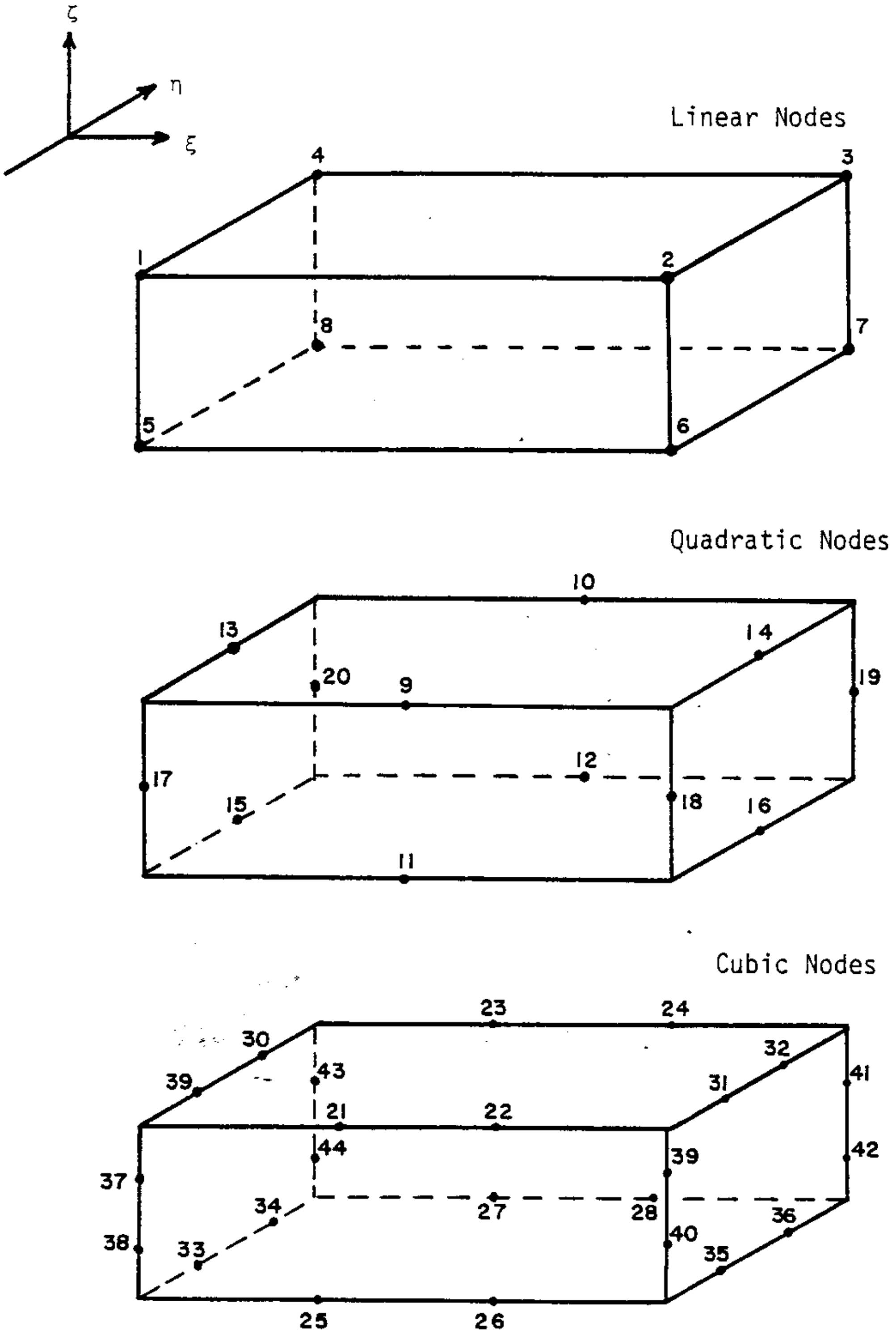
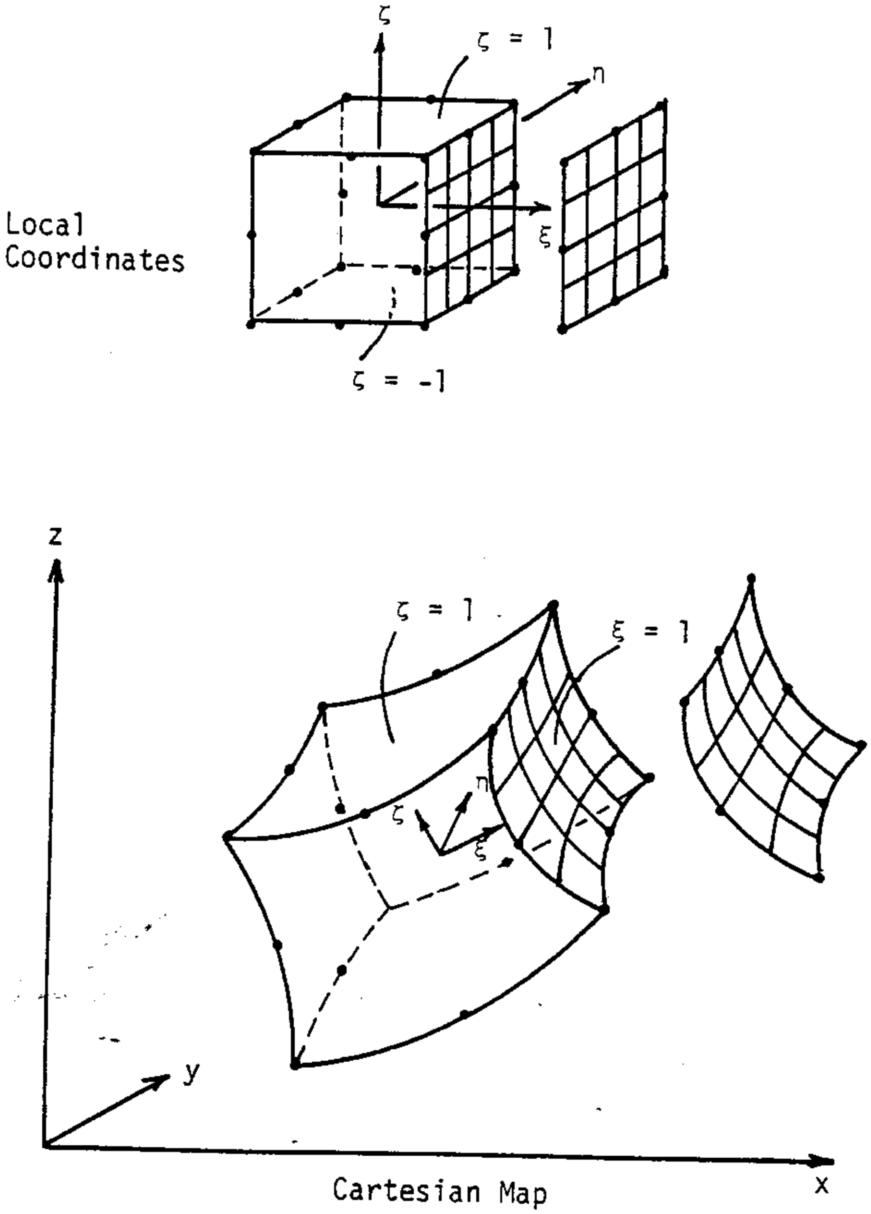


Figure 1. Nodal locations in local coordinates.



Local Coordinates

Cartesian Map

Figure 2. Isoparametric elements in Cartesian and local coordinates.

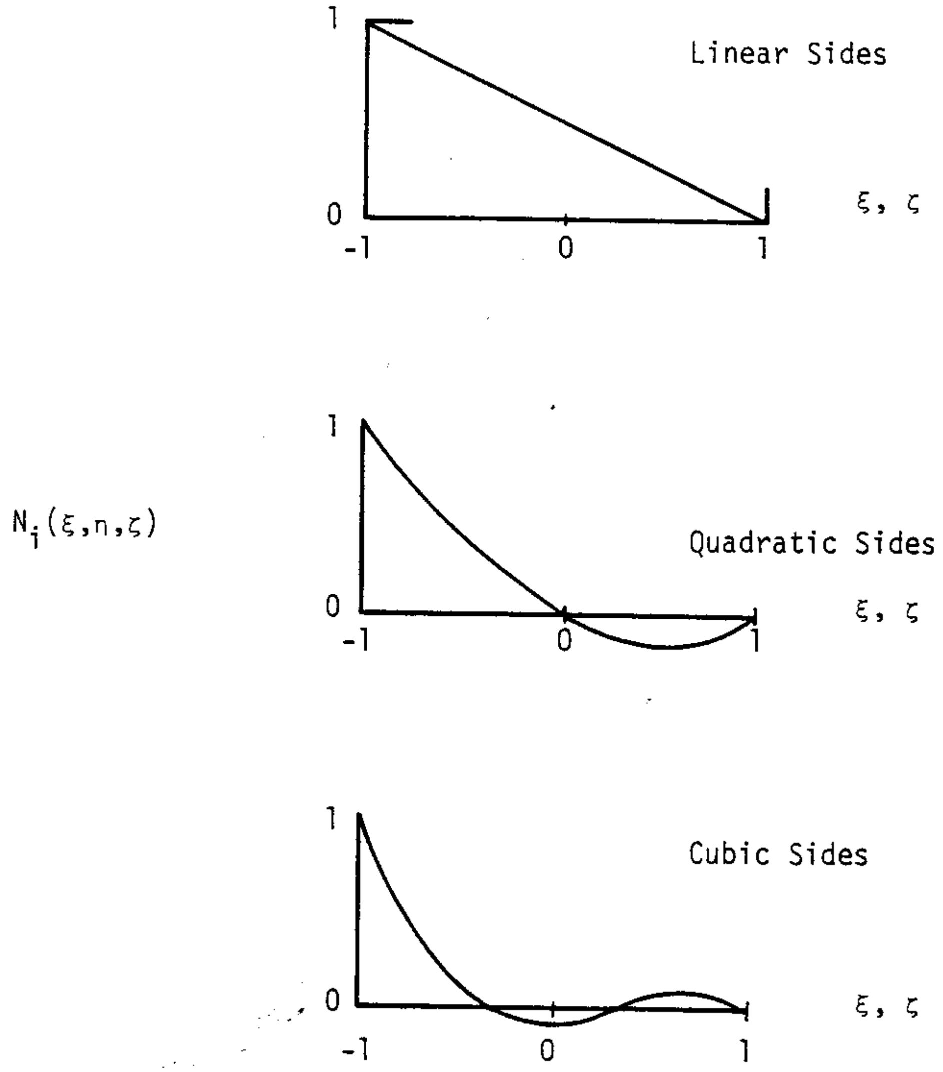


Figure 3. Basis function along $\eta = -1$ for corner node (-1, -1, -1) with adjacent sides of indicated order.

Table I

Three-Dimensional Mixed Element Basis Functions

For corner nodes

$$N_i\ (\xi,\ \eta,\ \zeta) = \alpha_i\ \beta_i$$

where $\alpha_i = \frac{1}{8}\,(1 + \xi\xi_i)(1 + \eta\eta_i)(1 + \zeta\zeta_i)$

and $\beta_i = \beta_\xi + \beta_\eta + \beta_\zeta$.

order of side	β_ξ	β_η	β_η
Linear	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$
Quadratic	$\xi\xi_i - \frac{2}{3}$	$\eta\eta_i - \frac{2}{3}$	$\zeta\zeta_i - \frac{2}{3}$
Cubic	$\frac{9}{8}\,\xi^2 - \frac{19}{24}$	$\frac{9}{8}\,\eta^2 - \frac{19}{24}$	$\frac{9}{8}\,\zeta^2 - \frac{19}{24}$

Nodes along the sides of an element; a typical midpoint of a quadratic side

$$N_i = 1/4(1 - \xi^2)(1 + \eta\eta_i)(1 + \zeta\zeta_i) \text{ for } \xi_i = 0,\ \eta_i = \pm 1,\ \zeta_i = \pm 1$$

$$N_i = 1/4(1 + \xi\xi_i)(1 - \eta^2)(1 + \zeta\zeta_i) \text{ for } \xi_i = \pm 1,\ \eta_i = 0,\ \zeta_i = \pm 1$$

$$N_i = 1/4(1 + \xi\xi_i)(1 + \eta\eta_i)(1 - \zeta^2) \text{ for } \xi_i = \pm 1,\ \eta_i = \pm 1,\ \zeta_i = 0$$

A typical side node of a cubic side

$$N_i = 9/64(1 - \xi^2)(1 + 9\ \xi\xi_i)\ (1 + \eta\eta_i)(1 + \zeta\zeta_i) \text{ for } \xi_i = \pm 1/3,\quad \eta_i = \pm 1,\ \zeta_i = \pm 1$$

$$N_i = 9/64(1 + \xi\xi_i)(1 - \eta^2)(1 + 9\eta\eta_i)(1 + \xi\xi_i) \text{ for } \xi_i = \pm 1,\ \eta_i = \pm 1/3,\quad \zeta_i = \pm 1$$

$$N_i = 9/64(1 + \xi\xi_i)(1 + \eta\eta_i)(1 - \zeta^2)(1 + 9\ \zeta\zeta_i) \text{ for } \xi_i = \pm 1,\ \eta_i = \pm 1,\quad \zeta_i = \pm 1/3$$

for basis functions as set forth by Huebner (1975). The basis functions are written in terms of the local coordinates and are selected such that in addition to meeting requirements of a basis function they also relate the global and local coordinate systems.

Derivation of Element Equations by Galerkin's Method

Once the elements and basis functions have been selected, the next step in a finite element analysis is to determine the matrix equations expressing the properties of the individual elements. This study uses the Galerkin criterion for setting up the element equations.

Galerkin's method is one technique of the method of weighted residuals. The method of weighted residuals has no connection with the finite element method, other than providing a means for formulating the finite element equations for a given problem.

The method of weighted residuals is a technique for obtaining approximate solutions to linear and nonlinear partial differential equations. Application of the method involves two basic steps. First, the general functional behavior of the dependent field variable is assumed so as to approximately satisfy the given differential equation and any prescribed boundary conditions. Substitution of this approximation into the original differential equation and boundary conditions results in some residual error. This residual is required to vanish in some average sense over the entire solution domain.

For example, suppose that an approximate functional representation is to be found for a field variable ϕ governed by the differential equation

$$I(\phi) - f = 0 \tag{4}$$

where I is a differential operator. The solution domain is bounded by a surface with properly prescribed boundary conditions, and the function f is a known function of the independent variables, for instance spatial coordinates.

The unknown exact solution ϕ is approximated by $\overline{\phi}$ as

$$\phi \simeq \overline{\phi} = \sum_{i=1}^{M} N_i C_i \tag{5}$$

where the N_i are the assumed functions and the C_i are the unknown parameters. When $\overline{\phi}$ is substituted into (4), the equation will probably not be satisfied, that is

$$I(\overline{\phi}) - f \neq 0.$$

In fact

$$I(\overline{\phi}) - f = R$$

where R is the residual error that results from having approximated ϕ by $\overline{\phi}$. The method of weighted residuals seeks to determine the M unknowns, C_i, in a manner such that the error R over the whole solution domain is small. The desired result is accomplished by forming a weighted average of the residual error and specifying that this weighted average vanish over the solution domain. Therefore, M linearly independent weighting functions, W_i, are selected and if

$$\int_D [I(\overline{\phi}) - f]\, W_i \, dD = \int R \, W_i \, d\,D = 0, \; i - 1, 2; M, \qquad (6)$$

integrated over the solution domain D, then R is approximately equal to zero in some average sense.

The exact form of the error distribution principle expressed above depends on the choice for weighting functions. Once the weighting functions W_i, i = 1, 2, ..., M, are specified, equation (6) represents a set of M equations, either algebraic equations or ordinary differential equations, to be solved for the undetermined C_i, i = 1, 2, ..., M. The second step of the method of weighted residuals is to solve (6) for the required coefficients and thereby obtain an approximate representation of the unknown field variable ϕ through equation (5).

As reviewed by Finlayson and Scriven (1966), there exist a wide variety of weighted residual techniques because of the broad choice of weighting functions. The error distribution principle most often used to derive finite element equations is known as Galerkin's method. According to Galerkin's method, the weighting functions are chosen to be the same as the approximating functions used to represent ϕ, that is $W_i = N_i$, i =-1, 2, ..., M. Therefore, Galerkin's method requires that

$$\int_D [I(\overline{\phi}) - f]\, N_i dD = 0, \; i = 1, 2, \ldots, M \qquad (7)$$

Pinder and Frind (1972) noted that equation (7) is equivalent to the requirement of the orthogonality of $[I(\overline{\phi}) - f]$ to the basis functions N_i, i=1, 2, ..., M.

To apply Galerkin's method to equation (1) for fluid flow define a function F(P) of pressure such that

$$F(P) = \frac{\partial}{\partial x}\left(\frac{k_x\rho}{\mu}\frac{\partial P}{\partial x}\right) + \frac{\partial}{\partial y}\left(\frac{k_y\rho}{\mu}\frac{\partial P}{\partial y}\right) + \frac{\partial}{\partial z}\left(\frac{k_z\rho}{\mu}\frac{\partial P}{\partial y}\right)$$

$$+ \frac{\partial}{\partial x}\left(\frac{k_x\rho^2 g}{\mu}\frac{\partial h}{\partial x}\right) + \frac{\partial}{\partial y}\left(\frac{k_y\rho^2 g}{\mu}\frac{\partial h}{\partial y}\right) + \frac{\partial}{\partial z}\left(\frac{k_z\rho^2 g}{\mu}\frac{\partial h}{\partial z}\right)$$

$$+ Q - \theta\rho_0\, \beta_T \frac{\partial T}{\partial t} - \left(\rho\alpha + \theta\rho_0\, \beta_p \frac{\partial P}{\partial t}\right) = 0$$

Next assume a pressure solution of the form

$$\overline{P}(x,y,z,t) = \sum_{i=1}^{M} P_i\ (t)\ N_i\ (x,y,z)$$

where N_i, i=1, 2, ..., M is a set of nodal basis functions.

For any given period of time (timestep), the unknown pressure coefficients, P_i, are assumed constant at specified points (nodes) in the solution domain and to vary within the solution domain according to the order of the basis functions.

Applying Galerkin's criterion over the solution domain

$$\iiint_D F(\overline{P})N_i\ dD = 0$$

and substituting the assumed pressure solution gives

$$\iiint_D F\left(\sum_{j=1}^{M} N_j\ P_j\right) N_i\ dD = 0$$

or, expanding,

$$\iiint \left[\left[\frac{\partial}{\partial x}\left(\frac{k_x\rho}{\mu}\frac{\partial}{\partial x}\right) + \frac{\partial}{\partial y}\left(\frac{k_y\rho}{\mu}\frac{\partial}{\partial y}\right) + \frac{\partial}{\partial z}\left(\frac{k_z\rho}{\mu}\frac{\partial}{\partial z}\right)\right] \left(\sum_{j=1}^{M} P_j\ N_j\right)\right.$$

$$+ [\frac{\partial}{\partial x}(\frac{k_x \rho^2 g}{\mu} \frac{\partial}{\partial x}) + \frac{\partial}{\partial y}(\frac{k_y \rho^2 g}{\mu} \frac{\partial}{\partial y}) + \frac{\partial}{z}(\frac{k_z \rho^2 g}{\partial} \frac{\partial}{\partial z})](\sum_{j=1}^{M} h_j N_j)$$

$$- (\rho\alpha + \theta\rho_o \beta_p) \frac{\partial}{\partial t}(\sum_{j=1}^{M} P_j N_j) + Q - \theta\rho_o \beta_T \frac{\partial T}{\partial t}] N_i dD = 0.$$

Rearranging results in

$$\iiint_D [\sum_{j=1}^{M} [N_i \frac{\partial}{\partial x}(\frac{k_x \rho}{\mu} \frac{\partial N_j}{\partial x}) + N_i \frac{\partial}{\partial y}(\frac{k_y \rho}{\mu} \frac{\partial N_j}{\partial y}) + N_i \frac{\partial}{\partial z}(\frac{k_z \rho}{\mu} \frac{\partial N_j}{\partial z})] P_j$$

$$+ \sum_{j=1}^{M} [N_i \frac{\partial}{\partial x}(\frac{k_x \rho^2 g}{\mu} \frac{\partial N_j}{\partial x}) + N_i \frac{\partial}{\partial y}(\frac{k_y \rho^2 g}{\mu} \frac{\partial N_j}{\partial y}) + N_i \frac{\partial}{\partial z}(\frac{h_z \rho^2 g}{\mu} \frac{\partial N_j}{\partial z})] h_j$$

$$- N_i (\rho\alpha + \theta\rho_o \beta_p) \sum_{j=1}^{M} \frac{\partial P_j}{\partial t} N_j$$

$$+ (Q - \theta\rho_o \beta_T \frac{\partial T}{\partial t}) N_i] dD = 0$$

To eliminate the second derivatives, which impose unnecessary continuity requirements between elements, Gauss's theorem (integration by parts in three dimensions) is applied. Integration by parts also offers a convenient way to introduce natural boundary conditions. The form of Gauss's theorem applied is an extension of a form of Green's theorem (integration by parts in two dimensions) presented by Weinstock (1952) and is

$$\iiint_D[a \frac{\partial^2 b}{\partial x} + a \frac{\partial^2 b}{\partial y} + a \frac{\partial^2 b}{\partial z}] dD$$

$$= - \iiint_D[\frac{\partial b}{\partial x} \frac{\partial a}{\partial x} + \frac{\partial b}{\partial y} \frac{\partial a}{\partial y} + \frac{\partial b}{\partial z} \frac{\partial a}{\partial z}] dD + \iint_s \frac{\partial b}{\partial m} dS$$

and $\frac{\partial b}{\partial m} = \frac{\partial b}{\partial x} \ell_x + \frac{\partial b}{\partial y} \ell_y + \frac{\partial b}{\partial z} \ell_z$

where $\frac{\partial b}{\partial m}$ is the outward normal derivative of the function b(x,y,z) and ℓ_x, ℓ_y, and ℓ_z are direction cosines of the outward surface normal. Introducing Gauss's theorem and distributing the integral sign results in

$$\iiint_D \sum_{j=1}^{m} \left(\frac{k_x\rho}{\mu} \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + \frac{k_y\rho}{\mu} \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} + \frac{k_z\rho}{\mu} \frac{\partial N_i}{\partial z} \frac{\partial N_j}{\partial z}\right) P_j \, dD$$

$$- \iiint_D \sum_{j=1}^{m} \left(\frac{k_x\rho^2 g}{\mu} \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + \frac{k_y\rho^2 g}{\mu} \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} + \frac{k_z\rho^2 g}{\mu} \frac{\partial N_i}{\partial z} \frac{\partial N_j}{\partial z}\right) h_j \, dD$$

$$+ \iint_S N_i \sum_{j=1}^{m} \left(\frac{k_x\rho}{\mu} \frac{\partial N_j}{\partial x} \ell_x + \frac{k_y\rho}{\mu} \frac{\partial N_j}{\partial y} \ell_y + \frac{k_z\rho}{\mu} \frac{\partial N_j}{\partial z} \ell_z\right)(P_j + Pgh_j) \, dS$$

$$- \iiint_D (\rho\alpha + \theta\rho_o \beta_p) N_i \sum_{j=1}^{m} \frac{\partial P_j}{\partial t} N_j \, dD$$

$$\iiint_D \left(Q - \theta\rho_o \beta_T \frac{\partial T}{\partial t}\right) N_i \, dD = 0$$

Writing in matrix form

$$[K]\{P\} + [K_H]\{h\} + [K_T]\frac{\partial P}{\partial t} + \{J\} = 0 \qquad (8)$$

where

$$[K]_{ij} = \iiint_D \left(\frac{k_x\rho}{\mu} \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + \frac{k_y\rho}{\mu} \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} + \frac{k_z\rho}{\mu} \frac{\partial N_i}{\partial z} \frac{\partial N_j}{\partial z}\right) dD$$

$$[K_H]_{ij} = \iiint_D \left(\frac{k_x\rho^2 g}{\mu} \frac{N_i}{\partial x} \frac{N_j}{\partial x} + \frac{k_y\rho^2 g}{\mu} \frac{N_i}{\partial y} \frac{N_j}{\partial y} + \frac{k_z\rho^2 g}{\mu} \frac{N_i}{\partial z} \frac{N_j}{\partial z}\right) dD$$

$$[K_t]_{ij} = \iiint_D (\rho\alpha + \theta\rho_o\ \beta_p)\ N_i\ N_j\ dD$$

$$\{J\}_i \quad = -\iiint_D (Q - \theta\rho_o\ \beta_T \frac{\partial T}{\partial t})\ N_i\ dD$$

$$- \iint_S N_i \sum_{j=1}^{m} (\frac{k_x\rho}{\mu} \frac{\partial N_j}{\partial x} \ell_x + \frac{k_y\rho}{\mu} \frac{\partial N_j}{\partial y} \ell_y + \frac{k_z\rho}{\mu} \frac{\partial N_j}{\partial z} \ell_z)$$

$$\cdot (P_j + \rho g h_j)\ dS$$

The well discharge component is easily handled using finite elements. Because of the properties of the Dirac delta function, the integral $\int_V Q\ N_i\ dV$ is equal to the well discharge at node i and is simply added directly to $\{J\}_i$.

The equation of energy transport may be written as,

$$L(T) = \frac{\partial}{\partial x}(K_x \frac{\partial T}{\partial x}) + \frac{\partial}{\partial y}(K_y \frac{\partial T}{\partial y}) + \frac{\partial}{\partial z}(K_z \frac{\partial T}{\partial z})$$

$$- C_p\ (v_x \frac{\partial T}{\partial x} + v_y \frac{\partial T}{\partial y} + v_z \frac{\partial T}{\partial z})$$

$$+ C_p\ \dot{Q}_{in}\ (T_{inj} - T) + C_p\ \dot{Q}_{out}\ (T - T_p)$$

$$- (\rho_f\ C_f\ (1 - \theta) + \rho_w\ C_p\ \theta) \frac{\partial T}{\partial t} - \Phi = 0$$

Following the same procedure used for the fluid flow equation, the following matrix equation may be developed for the temperature equation,

$$[M]\ \{T\} + [M_t] \frac{\partial T}{\partial t} + \{J\} = 0 \qquad (9)$$

where

$$[M]_{ij} = \iiint_D [K_x \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + K_y \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} + K_z \frac{\partial N_i}{\partial z} \frac{\partial N_j}{\partial z}$$

$$+ C_p N_i (v_x \frac{\partial N_j}{\partial x} + v_y \frac{\partial N_j}{\partial y} + v_z \frac{\partial N_j}{\partial z}) - \Phi$$

$$+ (C_p Q_{in} - C_p \dot{Q}_{out})] N_i N_j dD$$

$$[M_t]_{ij} = \iiint_D (\rho_f C_f (1 - \theta) + \rho_w \theta \ C_p) N_i N_j dD$$

and

$$\{J\}_i = - \iiint_D (C_p Q_{in} T_{inj} - C_p Q_{out} T_p) N_i dD$$

$$- \iint_s N_i \sum_{j=1}^{m} (K_x \frac{\partial N_j}{\partial x} \ell_x + K_y \frac{\partial N_j}{\partial y} \ell_y + K_z \frac{\partial N_j}{\partial z} \ell_z) T_j dS$$

The last terms in equations (8) and (9) incorporate the Newmann boundary condition;

$$\frac{k\rho}{\mu} (\frac{\partial P}{\partial m} + \rho g \frac{\partial h}{\partial m}) = q$$

and

$$K \frac{\partial T}{\partial m} = q_H$$

where q is the volumetric flux of a fluid into the element per unit area of boundary and q_H is the conductive flux of heat into the element per unit area of boundary. These terms are formed when q and q_H are non-zero in which case they take the form

$$\iint_S N_i \ q \ dS \quad \text{and} \quad \iint_s N_i \ q_H \ dS$$

At nodes where the Dirichlet (constant pressure or constant temperature) conditions exist, equations (8) and (9) are not generated since they are not needed. Thus [K], $[K_H]$, $[K_T]$, [M], and $[M_t]$ are reduced to (m-n) x (m-n) matrices, n being the number of passive nodes due to Dirichlet boundary conditions.

Recurrence Formula

After the matrices in equations (8) and (9) have been generated for each element and assembled for the entire solution domain, the undetermined coefficients $P_i(t)$ and $T_i(t)$ must be evaluated at selected points in time. This may be done by using a finite difference recurrence formula. Using a forward difference in time recurrence formula in equation (8) gives:

$$[K]\ \{P_{t+\Delta t}\} + [K_H]\ \{h\} + [K_t]\ (\{P_{t+\Delta t}\} - \{P_t\})/\Delta t + \{J\} = 0$$

or

$$([K] + [K_t]/\Delta t)\ \{P_{t+\Delta t}\} = [K_t]/\Delta t\ \{P_t\} - [K_H]\ \{h\} - \{J\}\ .$$

For equation (9), the result is

$$([M] + [M_t]/\Delta t)\ \{T_{t+\Delta t}\} = ([M_t]/\Delta t)\ \{T_t\} - \{J\}$$

Integration

Integration is done over each element separately, and the contribution of each element is accounted for in assembling the matrices [K], $[K_H]$, $[K_t]$, [M] and $[M_t]$. As mentioned previously, the element is transformed to a dimensionless coordinate system in which the element appears as a 2 x 2 square centered at the origin; thus, the limits of integration are from -1 to +1 for all three coordinate directions. The integration is performed numerically using a Gaussian quadrature scheme.

To integrate equations (8) and (9), the basis function derivatives, $\frac{\partial N_i}{\partial x}$, $\frac{\partial N_i}{\partial y}$, and $\frac{\partial N_i}{\partial z}$, are obtained as follows. First, consider the set of local coordinates ξ, η, and ς and a corresponding set of global coordinates x, y, and z. By the usual rules of partial differentiation, the derivatives may be written as

$$\frac{\partial N_i}{\partial \xi} = \frac{\partial N_i}{\partial x}\frac{\partial x}{\partial \xi} + \frac{\partial N_i}{\partial y}\frac{\partial y}{\partial \xi} + \frac{\partial N_i}{\partial z}\frac{\partial z}{\partial \xi} \quad .$$

Performing the same differentiation with respect to the other coordinates and writing in matrix form,

$$\begin{bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \\ \frac{\partial N_i}{\partial \zeta} \end{bmatrix} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial y}{\partial \xi} & \frac{\partial z}{\partial \xi} \\ \frac{\partial x}{\partial \eta} & \frac{\partial y}{\partial \eta} & \frac{\partial z}{\partial \eta} \\ \frac{\partial x}{\partial \zeta} & \frac{\partial y}{\partial \zeta} & \frac{\partial z}{\partial \zeta} \end{bmatrix} \begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} \end{bmatrix} = [J] \begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} \end{bmatrix} \quad .$$

The left hand side can be evaluated since the basis functions are specified in local coordinates. The matrix [J], known as the "Jacobian matrix", can also be formed in terms of local coordinates. Inverting [J] the global derivatives may be found as

$$\begin{bmatrix} \frac{\partial N_i}{\partial x} \\ \frac{\partial N_i}{\partial y} \\ \frac{\partial N_i}{\partial z} \end{bmatrix} = [J]^{-1} \begin{bmatrix} \frac{\partial N_i}{\partial \xi} \\ \frac{\partial N_i}{\partial \eta} \\ \frac{\partial N_i}{\partial \zeta} \end{bmatrix}$$

In terms of the basis functions defining the coordinate transformation, which are identical to the basis functions N_i when isoparametric formulation is used, J may be written as,

$$[J] = \begin{bmatrix} \Sigma \frac{\partial N_i}{\partial \xi} x_i & \Sigma \frac{\partial N_i}{\partial \xi} y_i & \Sigma \frac{\partial N_i}{\partial \xi} z_i \\ \Sigma \frac{\partial N_i}{\partial \eta} x_i & \Sigma \frac{\partial N_i}{\partial \eta} y_i & \Sigma \frac{\partial N_i}{\partial \eta} z_i \\ \Sigma \frac{\partial N_i}{\partial \zeta} x_i & \Sigma \frac{\partial N_i}{\partial \zeta} y_i & \Sigma \frac{\partial N_i}{\partial \zeta} z_i \end{bmatrix} .$$

The final relationship needed to perform the integrations in local coordinates is to replace an element of domain volume by

$$dD = dxdydz = \det[J]\, d\xi \; d\eta \; d\zeta \; .$$

After applying the above transformations and assuming the domain D to be one element,

$$[K]_{ij} = \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} \left(\frac{k_x\rho}{\mu} \frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + \frac{k_y\rho}{\mu} \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} + \frac{k_z\rho}{\mu} \frac{\partial N_i}{\partial z} \frac{\partial N_j}{\partial z}\right) \det[J]\, d\xi d\eta d\zeta .$$

Similar expressions are developed for the remaining terms of equations (8) and (9).

Mass Flux Equations

Solution of the convective-conductive heat transfer equation requires values of the mass flux at each nodal point. The mass flux may be obtained from the volumetric flux (q_x). The required volumetric fluxes may be obtained from Darcy's Law in the form,

$$q_x = - \frac{k_x}{\mu} (\frac{\partial P}{\partial x} + \rho g \frac{\partial h}{\partial x})$$

$$q_y = - \frac{k_y}{\mu} (\frac{\partial P}{\partial y} + \rho g \frac{\partial h}{\partial x})$$

$$q_z = - \frac{k_z}{\mu} (\frac{\partial P}{\partial z} + \rho g \frac{\partial h}{\partial z})$$

Looking first at only the x direction, a function L(q) may be defined such that,

$$L(q_x) = q_x + \frac{k_x}{\mu} (\frac{\partial P}{\partial x} + \rho g \frac{\partial h}{\partial x}) = 0 \qquad (10)$$

Following standard finite element methology a functional representation of q_x may be defined as

$$\overline{q}_x = \sum_{j=1}^{m} q_{xj} N_j$$

where N_j, j = 1, 2, ... m is a set of basis functions. Applying Galerkin's method to equation (10), and substituting $\overline{q}_x$ for q_x results in the following equations;

$$\iiint_D L(\overline{q}_x) N_i \, dD = 0 \qquad i = 1, 2, \ldots, m.$$

or

$$\iiint_D L(\sum_{j=1}^{m} q_{xj} N_j) N_i \, dD = 0$$

or

$$\iiint_D \left(\sum_{j=1}^{m} q_{xj} N_j + \frac{k_x}{\mu} \left(\frac{\partial P}{\partial x} + \rho g \frac{\partial h}{\partial x}\right)\right) N_i \, dD = 0. \tag{11}$$

Defining a functional definition for the pressure and the vertical position vector derivatives as,

$$\frac{\partial P}{\partial x} = \sum_{j=1}^{m} P_j \frac{\partial N_j}{\partial x} \text{ and } \frac{\partial h}{\partial x} = \sum_{j=1}^{m} h_j \frac{\partial N_j}{\partial x} \quad ,$$

and substituting into equation (11) give

$$\iiint_D [(\sum_{j=1}^{m} q_{xj} N_j) + \frac{k_x}{\mu} (\sum_{j=1}^{m} P_j \frac{\partial N_j}{\partial x} + \rho g \sum_{j=1}^{m} h_j \frac{\partial N_j}{\partial x})] N_i \, dD = 0$$

or

$$\iiint_D [N_i \sum_{j=1}^{m} q_{xj} N_j) + \frac{k_x}{\mu} (N_i \sum_{j=1}^{m} P_j \frac{\partial N_j}{\partial x} + N_i \rho g \sum_{j=1}^{m} h_j \frac{\partial v_j}{\partial x})] \, dD = 0,$$

Writing in matrix form, after transforming coordinates,

$$[A] \{Q_x\} = \{B\} \tag{12}$$

where $[A]_{ij} = \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} N_i N_j \det[J] \, d\xi \, d\eta \, d\zeta,$

$\{Q_x\} \equiv$ column vector of nodal fluxes, and

$$\{B\} = \int_{-1}^{1}\int_{-1}^{1}\int_{-1}^{1} [\sum_{j=1}^{m} \frac{k_x}{\mu} (P_j + \rho g h_j) N_i \frac{\partial N_j}{\partial x} \det[J] \, d\xi \, d\eta \, d\zeta.$$

A similar development can be made for the y and z coordinate directions. Note that [A] remains constant for all directions, however {B} changes for each direction. This method of solving for nodal fluxes insures continuity of flux, both within each element and at boundaries between elements.

Boundary Flow Calculations

From equation (9) the boundary flow attributable to a given node (i) is

$$\iint_A N_i \left[\sum_{j=1}^{m} (q_x \ell_x + q_y \ell_y + q_z \ell_z)\right] dA$$

where $\ell_{x,y,z}$ are the direction cosines of the outward directed surface normal. In general dA will lie on a surface where one of the coordinates, for example ζ, is constant. If ξ and η are some curvilinear coordinates (ζ = constant) then the vectors in 3-D space,

$$d\bar{\xi} = \begin{Bmatrix} \partial x/\partial\xi \\ \partial y/\partial\xi \\ \partial z/\partial\xi \end{Bmatrix} d\xi \quad \text{and} \quad d\bar{\eta} = \begin{Bmatrix} \partial x/\partial\eta \\ \partial y/\partial\eta \\ \partial z/\partial\eta \end{Bmatrix} d\eta$$

defined from the relationship between the Cartesian and curvilinear coordinates, are vectors directed tangentially to the ξ = constant and η = constant surfaces. The length of the vector resulting from a cross product of $d\bar{\xi}$ and $d\bar{\eta}$ ($d\bar{\xi}$ x $d\bar{\eta}$) is equal to the area of dA (the elementary "transformed" parallelogram).

Considering dA as a vector oriented in the direction normal to the surface, one would form the vector cross product

$$dA = \begin{Bmatrix} \partial x/\partial\xi \\ \partial y/\partial\xi \\ \partial z/\partial\xi \end{Bmatrix} \times \begin{Bmatrix} \partial x/\partial\eta \\ \partial y/\partial\eta \\ \partial z/\partial\eta \end{Bmatrix} d\xi \, d\eta$$

Taking the magnitude of dA, substituting with appropriate limits for ξ and η, the boundary flow is obtained as

$$\int_{-1}^{1}\int_{-1}^{1} N_i \left[\sum_{j=1}^{m} (q_x \ell_x + q_y \ell_y + q_z \ell_z)\right] \zeta \left| \begin{Bmatrix} \partial x/\partial \xi \\ \partial y/\partial \xi \\ \partial z/\partial \xi \end{Bmatrix} \times \begin{Bmatrix} \partial x/\partial \eta \\ \partial y/\partial \eta \\ \partial z/\partial \eta \end{Bmatrix} \right| d\xi \, d\eta$$

Description of the Computer Program

The numerical model was programmed in FORTRAN IV for the AMDAHL 470 at the Texas A&M University Data Processing Center. The model consists of three separate programs. A listing of each program is included in Appendix C.

Program 1 reads all input data regarding material properties, node coordinates, information for each element and boundary conditions. All nodes are given new sequential numbers for matrix formulation while the nodal numbers of the user are used for output. Internal numbering of nodes provides the user a choice of node-numbering schemes without affecting core storage requirements or computational cost. Input data is written on tape for use in subsequent programs. Subroutine COUNT determines which nodes have pressure and temperature as unknowns and lists them. Subroutine BANWDH estimates the bandwidth along with the upper and lower limits of the diagonals of a sparse system matrix. The maximum bandwidth for subroutine EQSOLV is also estimated.

In program 2, Gaussian Quadrature integration is carried out. Parameters required for the subsequent program are estimated and organized on tape so that the numerical integration data are available element by element. The matrices in equations (8), (9) and (12) are formed by simply reading element by element the integration parameters and multiplying by the necessary constants or known variables.

Program 3 performs the actual simulation. The matrix equation (8) for pressure is usually solved first and the new pressures are used to compute flux values for the temperature equation. The resulting new temperatures and pressures are then used to update fluid properties to be used in the next time step. Thus the program moves through time solving the equations of motion and energy transport at each time step. A basic flow chart of the program is included in Appendix B.

Subroutine RDPROP reads and prints a table of fluid dynamic viscosity, specific heat and thermal conductivity as functions of pressure and temperature. Subroutine UPDATE calculates new values of fluid properties at the end of each time step. It was assumed that fluid density could be expanded about ρ_0, a reference density, in a Taylor series truncated with first order terms in temperature and pressure.

Subroutine WATBAL carries out boundary flow calculations and performs a mass balance on the fluid flow. Subroutine FLUX calculates the mass flux values for each nodal point based on current values of pressure and fluid properties.

Subroutine EQSOLV is used to solve the matrix equations for pressure and temperature. EQSOLV uses compressed matrices storing non-zero coefficients in one matrix and their corresponding column identifications in another matrix. The compressed matrices are decomposed into an upper triangular form, followed by bases substitution. The compressed matrix approach was chosen because the Galerkin approximation of the convection-conduction equation results in an unbanded, nonsymmetric system of equations.

CHAPTER V

RESULTS AND DISCUSSION

Introduction

Operational tests of the numerical model were made using two problems for which analytical solutions could be obtained. Both problems involved hot water injection into a confined aquifer. In each case, the injected water was at a much higher temperature than the natural groundwater. The resulting mass fluxes were relatively large resulting in a sharp thermal front problem; that is, a problem with temperature changing rapidly over space and time. Due to the nature of the two test problems, it was felt that if the model performed satisfactorily, it could be used for general field type problems with a high degree of confidence.

Test Problems

The first test problem consisted of the 28 node convection-conduction problem shown in Figure 4. Three-dimensional linear and mixed isoparametric elements with quadratic and cubic nodal configurations in the principal flow direction were used. In all simulation runs, linear nodal configurations were used in all directions other than the principal flow direction. Physical parameters for this test problem, termed the "grid" problem, are shown in Table II. The top and bottom boundaries of the solution region ($z = \text{constant}$) were considered impervious to fluid flow and insulated to heat flow. Identical boundary conditions were used for the y = constant boundaries. For

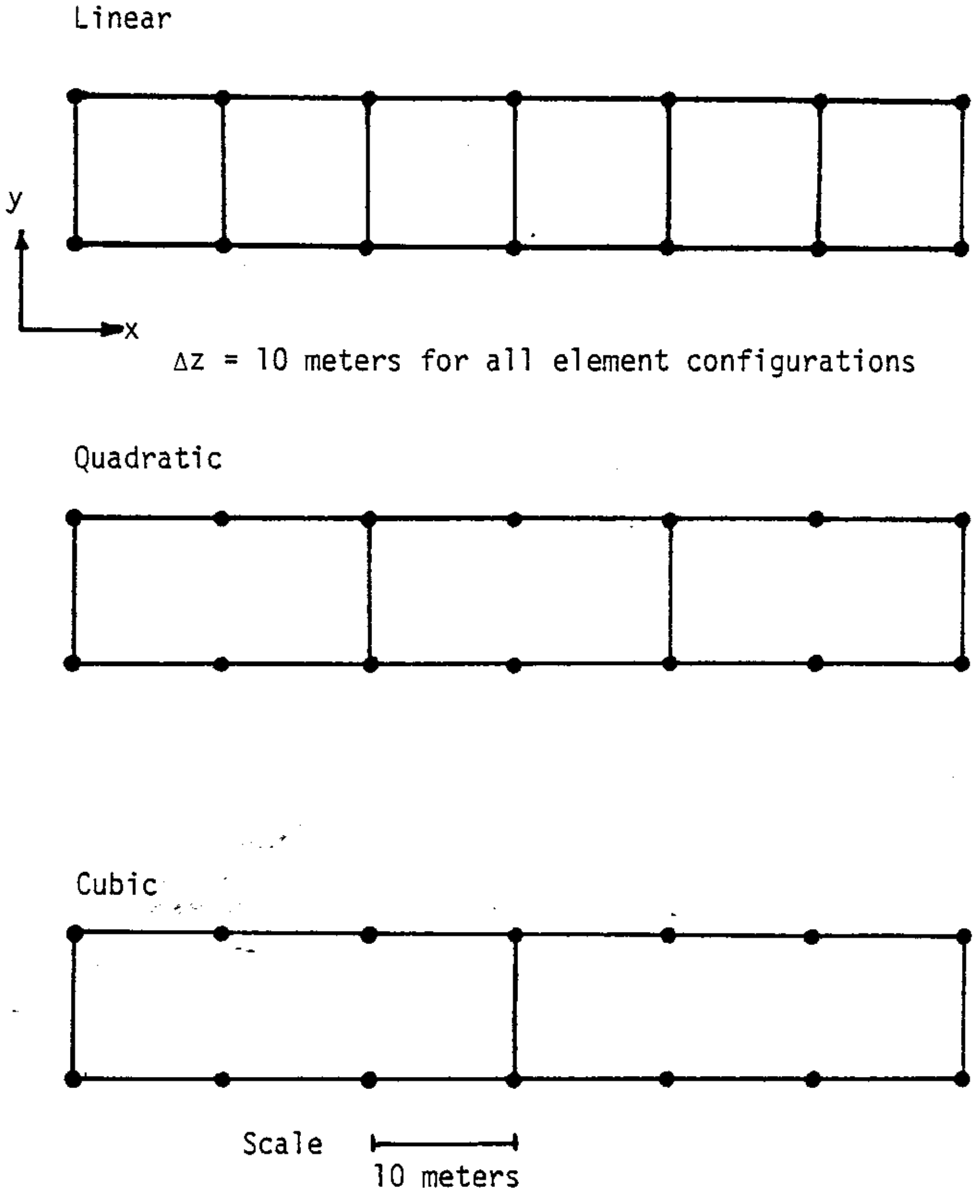


Figure 4. Element configurations for the grid test problem.

Table II

Grid Test Problem Physical Parameters

Specific permeability	1.0×10^{-9} M^2
Porosity	30%
Vertical compressibility (α)	6×10^{-8} M^2/N
Specific rock density	2500 Kg/M^3
Rock material heat capacity	1095 J/Kg-°C
Rock thermal conductivity	2.15 W/M-°C
Fluid thermal volume expansion	-4.167×10^{-4} $°C^{-1}$
Fluid compressibility	5.0×10^{-10} M^2/N
Initial aquifer temperature	15°C

the remaining sides, boundary conditions were as follows;

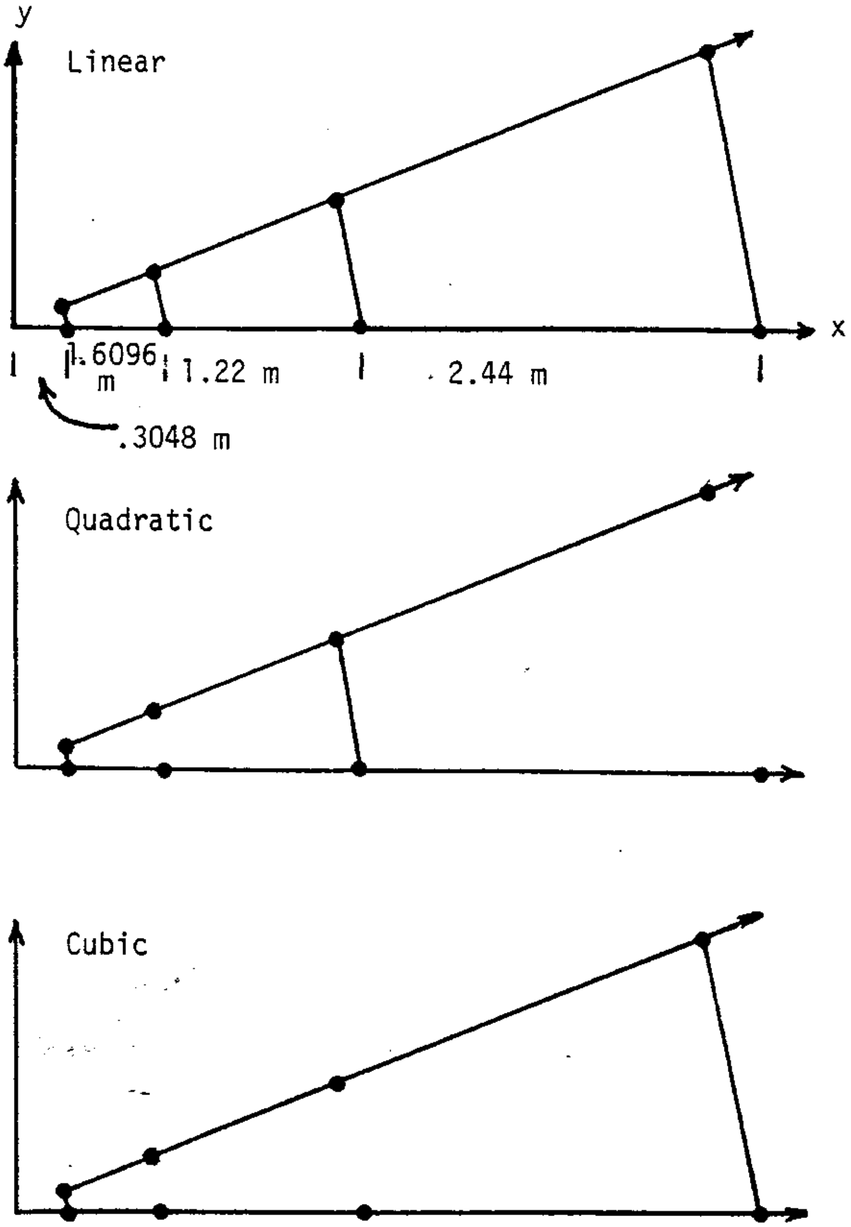
a) for the upstream side, the temperature was set at a prescribed value with a constant nodal mass flux of .025 Kg M^{-2} sec^{-1}, and

b) the downstream side fluid pressures were set at constant prescribed values (constant pressure boundary conditions are commonly known as "Dirchlet" boundary conditions).

For the transient solution of this problem, equal time steps of 50,000 seconds were used.

The second test problem consisted of the 22.5° wedge shaped radial well problem depicted in Figure 5. The z = constant and lateral boundaries were considered impervious and insulated, as was the outer radial boundary. Water was injected into a 20°C aquifer at the rate of 2 Kg sec^{-1}. Fifty-six nodes were used in linear and mixed element configurations. Quadratic and cubic nodal configurations were used in the radial direction, while a linear configuration was used in the vertical direction and the direction perpendicular to fluid flow. The outermost element (the element at greatest radius from the well) remained linear in all test runs. This large linear element served the purpose of protecting the solution region from boundary effects. The same nodes are used for the linear and mixed quadratic and cubic element configurations. Each successive nodal space, proceeding along the x axis is twice the preceding one. The aquifer thickness is 10 meters. Physical parameters for the radial test problem are shown in Table III.

Temperature solutions are compared to the analytical solutions



Δz = 10 meters for all element configurations.

Figure 5. Element configurations for the radial test problem.

Table III

Radial Test Problem Physical Parameters

Specific permeability	9.5×10^{-10} M^2
Porosity	40%
Vertical compressibility (α)	$.5 \times 10^{-8}$ M^2/N
Specific rock density	2500 Kg/M^3
Rock material heat capacity	1095 J/Kg-°C
Rock thermal conductivity	3.81 W/M-°C
Fluid thermal volume expansion	-4.167×10^{-4} $°C^{-1}$
Fluid compressibility	5.0×10^{-10} M^2/N
Initial aquifer temperature	20°C

of Avdonin (1964). Avdonin presented an analytical solution for the linear flow of a hot fluid through a confined, homogenous, isotropic aquifer with constant fluid properties as;

$$U(X,\ t) = \frac{X}{\sqrt{\pi\tau}} \int_0^1 \{\exp[-(s\gamma\sqrt{\tau} - \frac{X}{2s\sqrt{\tau}})^2]\ \mathrm{erfc}\ (\frac{\alpha s^2\sqrt{\tau}}{2\sqrt{1-s^2}})\frac{ds}{s^2}$$

where $X = \frac{2x}{b}$;

$\tau = \frac{4\ K_t\ t}{c_t\ \rho_t\ b^2}$

$\gamma = \frac{Q\ C_w\ \rho_w}{4\ K_t}$

$\alpha = \frac{K_r\ C_r\ \rho_r}{K_t\ C_t\ \rho_t}$

x = position; L,

$U = \frac{T - T_o}{T_i - T_o}$, dimensionless temperature

b = flow layer thickness; L,

K_t = horizontal thermal conductivity in the flow layer; $F\ T^{-1}\ deg^{-1}$

t = time; T,

C = specific heat $F\ L\ M^{-1}\ deg^{-1}$

ρ = density; $M\ L^{-3}$,

Q = injection rate; $L^3\ T^{-1}$,

K = thermal conductivity; $F\ T^{-1}\ deg^{-1}$, and

s = an integration parameter

The subscripts refer to the rock, r, water, w, and the total (rock and water), t. Where the subscript r is used, it refers to the confining strata above and below the flow layer.

Avdonin (1964) also presented the solution for the temperature distribution in a radial system as;

$$U\ (\omega,\tau) = \frac{1}{\Gamma(\nu)}\ [\frac{\omega^2}{4\tau}]^{\nu} \int_0^1 \{\exp\ (-\ \frac{\omega^2}{4\gamma s})\ \mathrm{erfc}\ (\frac{\alpha s\sqrt{\tau}}{2\sqrt{1-s}})\}\ \frac{ds}{s^{\nu+1}}$$

where $\omega = \frac{2R}{b}$, R is the radial distance from the injecting well; L,

$$\nu = \frac{Q\ C_w\ \rho_w}{4\pi\ b\ K_t}$$

and all other parameters are as previously defined.

In developing his solutions, Avdonin (1964) made the following assumptions;

a) there is no convective heat transport in the confining layers,

b) only vertical thermal conductivity exists in the confining layers,

c) a finite non-zero horizontal thermal conductivity exists in the flow layer, and

d) the flow layer has an infinite vertical thermal conductivity.

Assuming that the aquifer has an infinite vertical thermal conductivity results in no temperature gradient vertically throughout the aquifer. Therefore the temperatures reported for the test problems are averages over the aquifer thickness. For initial runs, a strict Galerkin finite element approximation was used.

Pressure Simulation

To evaluate the pressure simulation part of the model, initial runs for each problem were made with the injection temperature equal to the aquifer temperature (no heat transport). Results of the grid test problem are shown in Figure 6 after 27.7 hours. At this time the aquifer pressures had effectively reached steady state. A steady state analytical solution for the linear problem may be formulated in the "x" direction as

$$P_x = P_o + \rho g\ h_d - \frac{v_x \mu}{\rho k} (x - x_o)$$

where P_x = pressure at a downstream distance "x"; $F\ L^{-2}$,

P_o = reference pressure defined at the top of the flow layer at x_o; $F\ L^{-2}$,

x = downstream distance; L,

x_o = distance to pressure reference point; L,

ρ = fluid density; $M\ L^{-3}$

g = acceleration due to gravity $L\ T^{-2}$

h_d = depth below top of flow layer; L,

v_x = fluid mass flux; $M\ L^{-2}\ T^{-1}$

μ = fluid dynamic viscosity; $F\ T\ L^{-2}$, and

k = specific permeability; L^2 .

Elements with linear, quadratic, and cubic sides in the x direction agree very well with the analytical solution for the grid test problem.

Results from the radial flow problem are shown in Figure 7. Theis (1935) developed an analytical solution for a pumping or in-

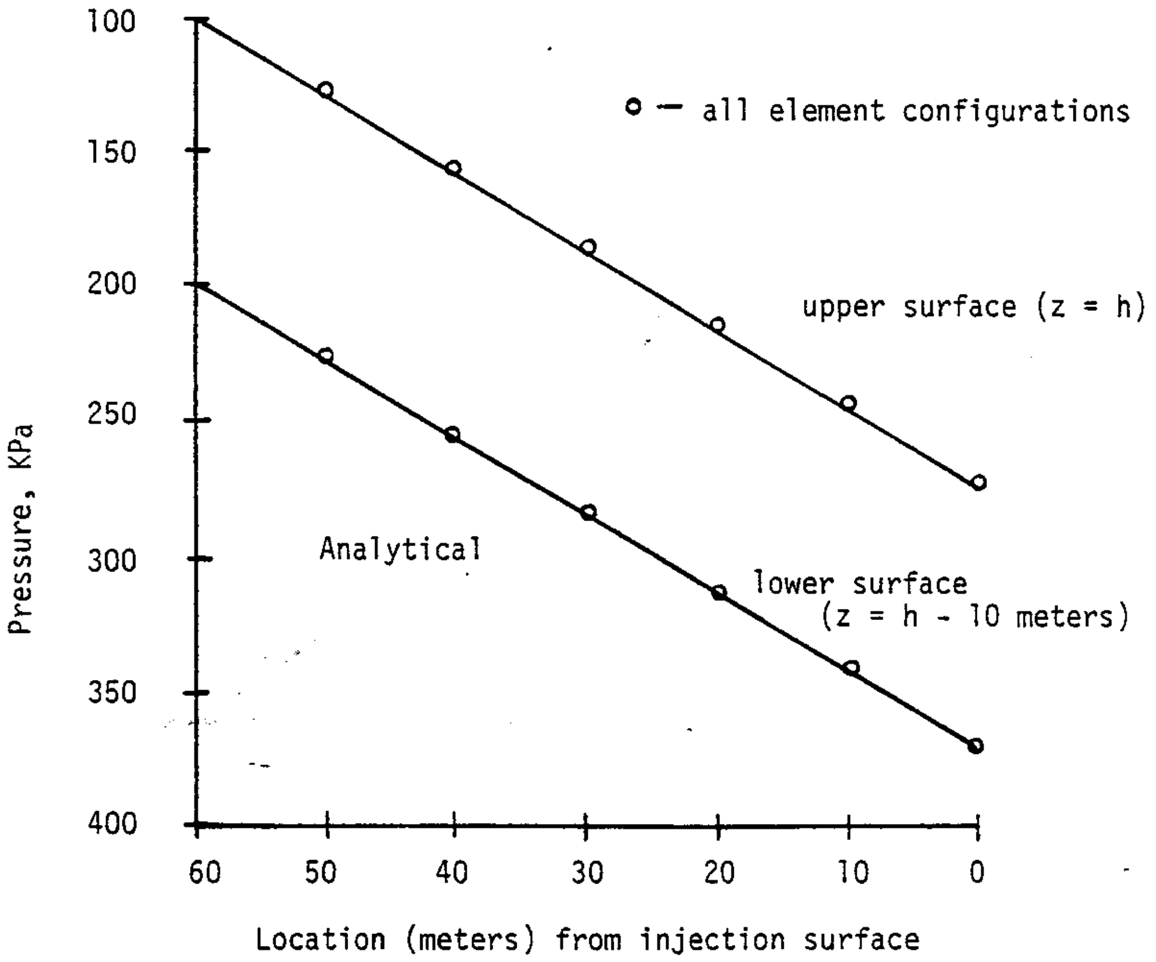


Figure 6. Comparison of the grid test problem pressure distribution without heat transport to an analytical solution.

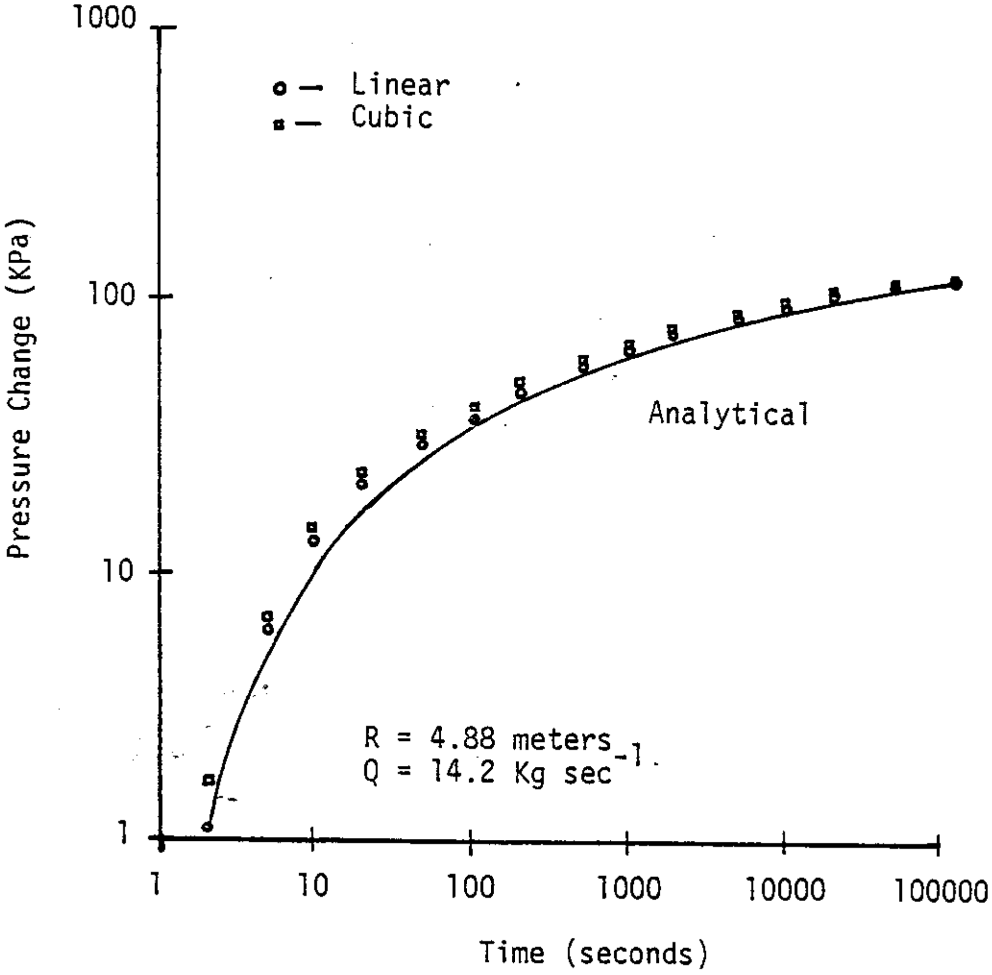


Figure 7. Comparison of the radial test problem transient pressure distribution without heat transport to Theis' solution.

jecting well at the center of a confined, homogeneous, isotropic aquifer. The Theis solution may be presented as

$$\Delta P = \frac{24500}{\pi T} \int_u^\infty \frac{e^{-u}}{u}\, du$$

where $u = \frac{R^2 S}{4T_m t}$

with R = distance from pumping or injecting well to observation point,

S = coefficient of storage,

T_m = transmissivity; $L^2\ T^{-1}$,

t = time; T,

Q = well discharge (+) or recharge (-); $L^3\ T^{-1}$,

ΔP = change in fluid pressure; $F\ L^{-2}$.

The radial finite element test problem without heat transport meets these conditions. For comparison, a Theis solution at a radius of 4.88 meters is plotted and compared in Figure 7 with simulation results for the same data. For this problem (fluid injection without heat transport), time steps were increased exponentially. It should be noted that the injection rate was increased to 14.2 Kg sec^{-1} for this simulation. Late in the simulation, all element configurations converge to the Theis solution. Early in the simulation, however, the larger number of linear elements seems to give somewhat better results at 4.88 meters than the fewer, higher order nodes. This behavior appears to be opposite to that reported by Pinder and Frind (1972). This behavior indicates that, with a given set of nodes, the order of the basis functions for representing a transient groundwater problem may vary both as a function of time and physical parameters (exponentially increasing time steps and geometrically increasing element size were used).

Heat Transport Simulation

In order to test the heat transport section of the model for the grid problem, the upstream end nodes were set at a constant temperature of 45°c. For the radial well problem, water at 75°c was injected into the aquifer. While excellent results were obtained from the pressure section of the model (without heat transport), the temperature section failed to give an acceptable result.

As previously mentioned for the grid problem, water at 45°c was injected into an aquifer initially at 15°c. Hence, the fluid temperature should not exceed the 15°c to 45°c temperature range. As shown in Figure 8, particularly at t = 300,000 seconds, the numerical solution overshoots both the analytical solution and the upper temperature front. At the downstream edge of the temperature front, the numerical solution undershoots both the analytical solution and the lower temperature limit (15°c). The resulting oscillations about the sharp thermal front grew in time until the simulation failed. The order of the element basis functions had no effect on the end result. Similar problems occurred with the radial test problem (Figure 9). The analytical solution of Avdonin (1964) is plotted for comparison.

The approximation of sharp solution fronts from hyperbolic differential equations by finite element analysis seems to be an especially difficult task. Strang and Fix (1973) discussed briefly the

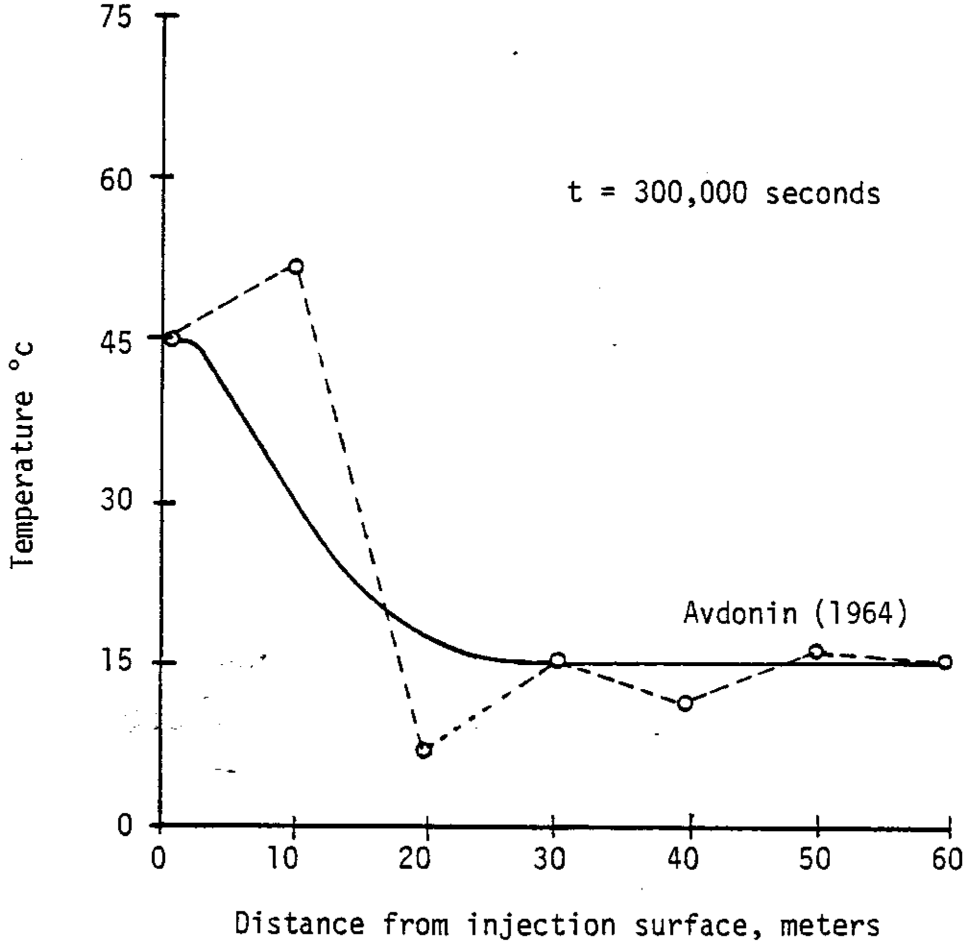


Figure 8. Comparison of the grid test problem transient temperature distribution using a strict Galerkin approach to the solution of Avdonin (Quadratic elements).

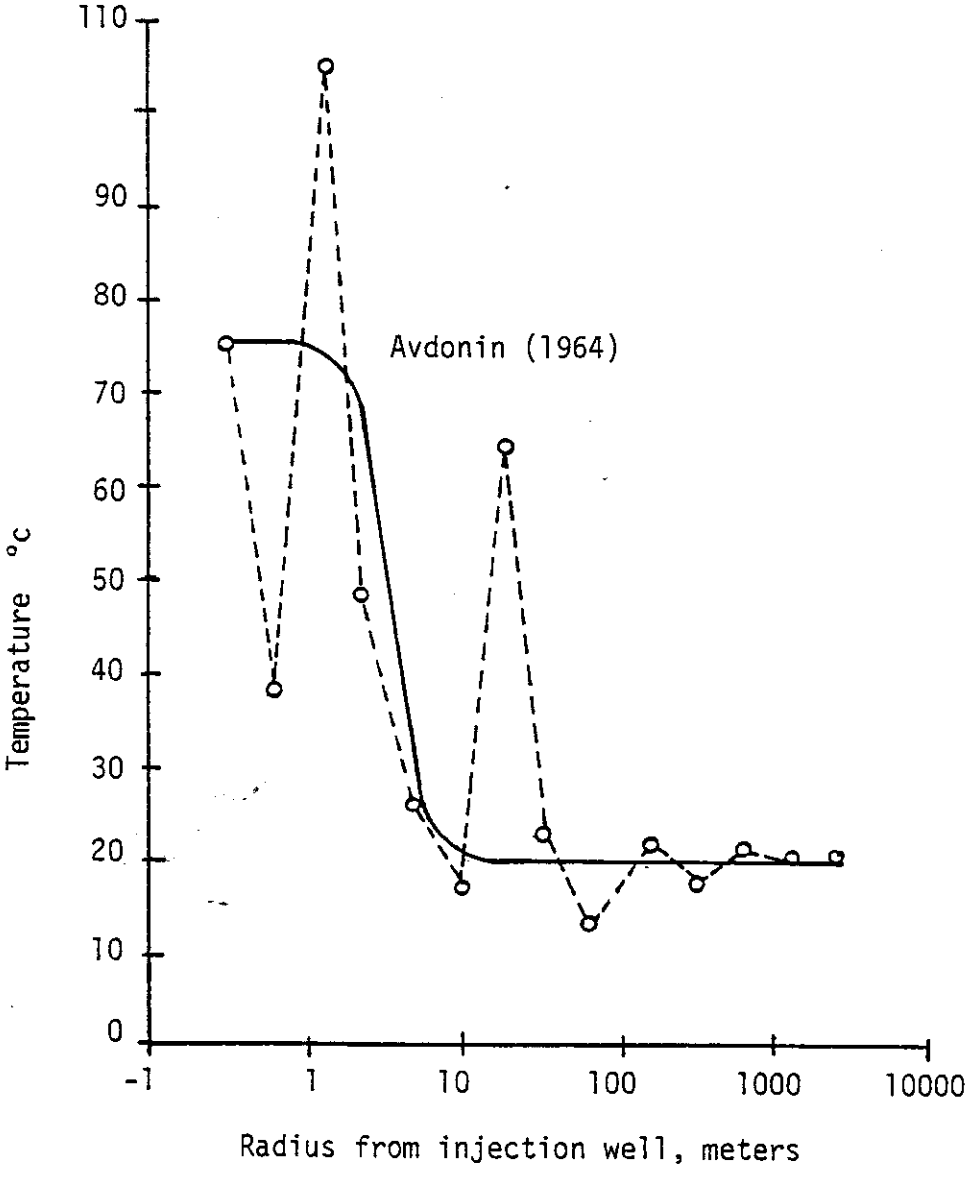


Figure 9. Comparison of the radial test problem transient temperature distribution using a strict Galerkin approach to the solution of Avdonin (Quadratic elements).

application of Galerkin finite-element procedures to hyperbolic equations. They indicated two major difficulties: (1) oscillations of the predicted solution surface about the real solution surface, and (2) convergence to incorrect solutions for certain nonlinear problems. The oscillation problem seems to be common, while the convergence problem was reported by Finlayson (1972) to only apply to a special case of equations.

Faust and Mercer (1976) stated that the oscillations which occur when solving hyperbolic problems are caused by truncation errors in the approximation of both space and time derivatives, and that difficulties associated with the spacial approximations appear to be more severe. A detailed search of the literature showed that several techniques have been suggested for reducing oscillations [Gupta et al. (1975), Faust and Mercer (1976), Jennings (1969), Langsrud (1976), Zeinkiewicz (1977)].

The most commonly suggested method was to decrease the element size on the time step size. For the grid test problem, the element spacing was decreased to one meter and time steps were reduced to twenty seconds with no appreciable improvement in results. Judging by results from the literature and results from the simulation runs made in this study, it appears that these suggested methods are most suitable for problems which demonstrate a natural diffusion mechanism. But for hyperbolic problems, where the convection term is much larger than the diffusion term, these techniques are essentially ineffective in reducing numerical oscillation.

Another commonly suggested method for reducing oscillations was to smear the sharp thermal front using upstream weighting [Zienkiewicz (1977) and Huyacorn et al. (1977)]. The method of upstream weighting, based on a modified basis function as presented by Huyacorn et al. (1977), was applied to the grid test problem using only linear elements. As shown in Figure 10, the oscillations were sufficiently damped; however, the sharp thermal front was spread across the solution region. This is evidenced in Figure 10 by the larger than expected downstream temperatures, and the smooth slope of the solution curve at t = 700,000 seconds. This degree of "smearing" of the sharp thermal front is unacceptable.

For the radial test problem with linear elements and upstream weighting, the front was smeared throughout the spatial boundaries of the simulation, as shown in Figure 11. This was probably caused by the doubling of the nodal spacing with increasing radius. Due to the disappointing results and the large amount of time necessary to program the upstream weighting method for linear elements, no attempt was made to develop a general method for mixed elements.

Heat Transport Simulation Using Mass Lumping

Zienkiewicz (1977) discussed the use of "mass matrix lumping" for computational convenience and and improved solutions in finite element solutions of structural mechanics problems. The effect of "lumping" is to diagonalize the mass matrix in the hyperbolic heat transfer equation (in this case matrix M_t of equation 9).

Newman (1974) used a lumping approach for solving a problem of two-dimensional, unsaturated flow under a dam. He solved the pressure

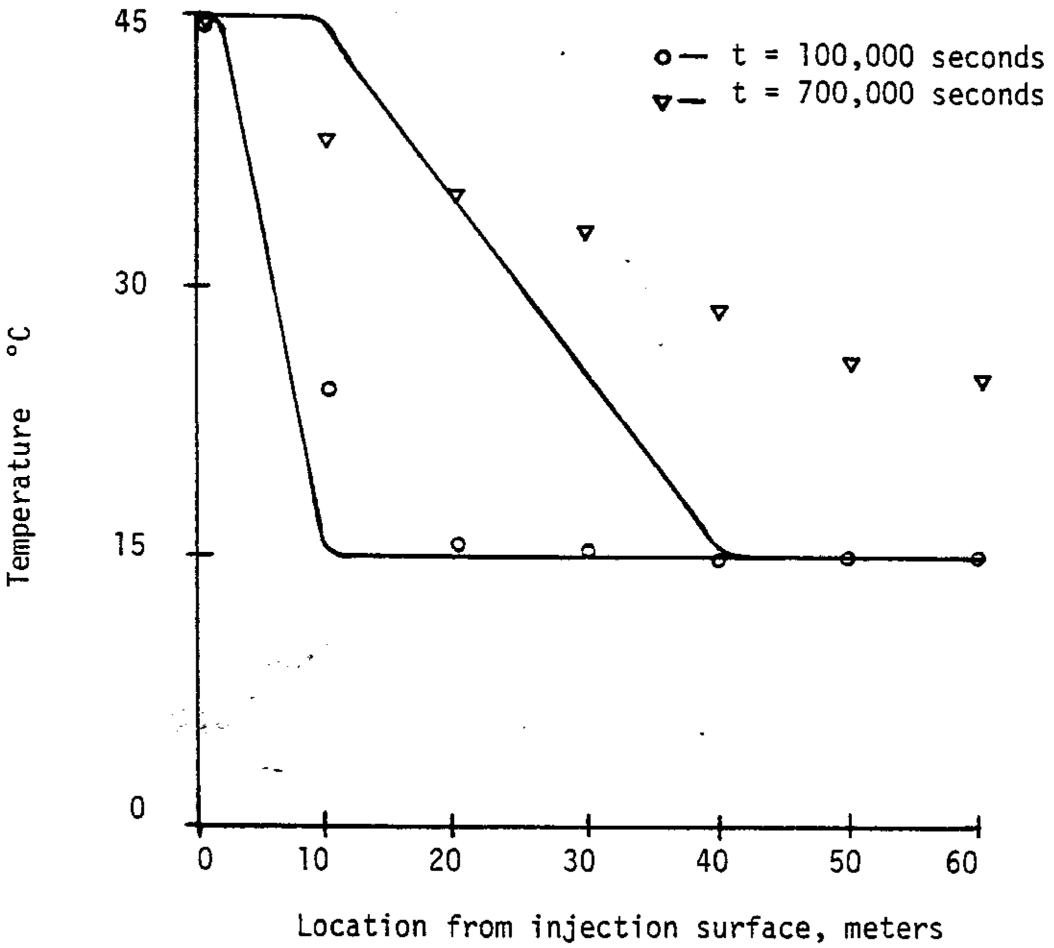


Figure 10. Comparison of the grid test problem transient temperature distribution using upstream weighting to the solution of Avdonin (linear elements).

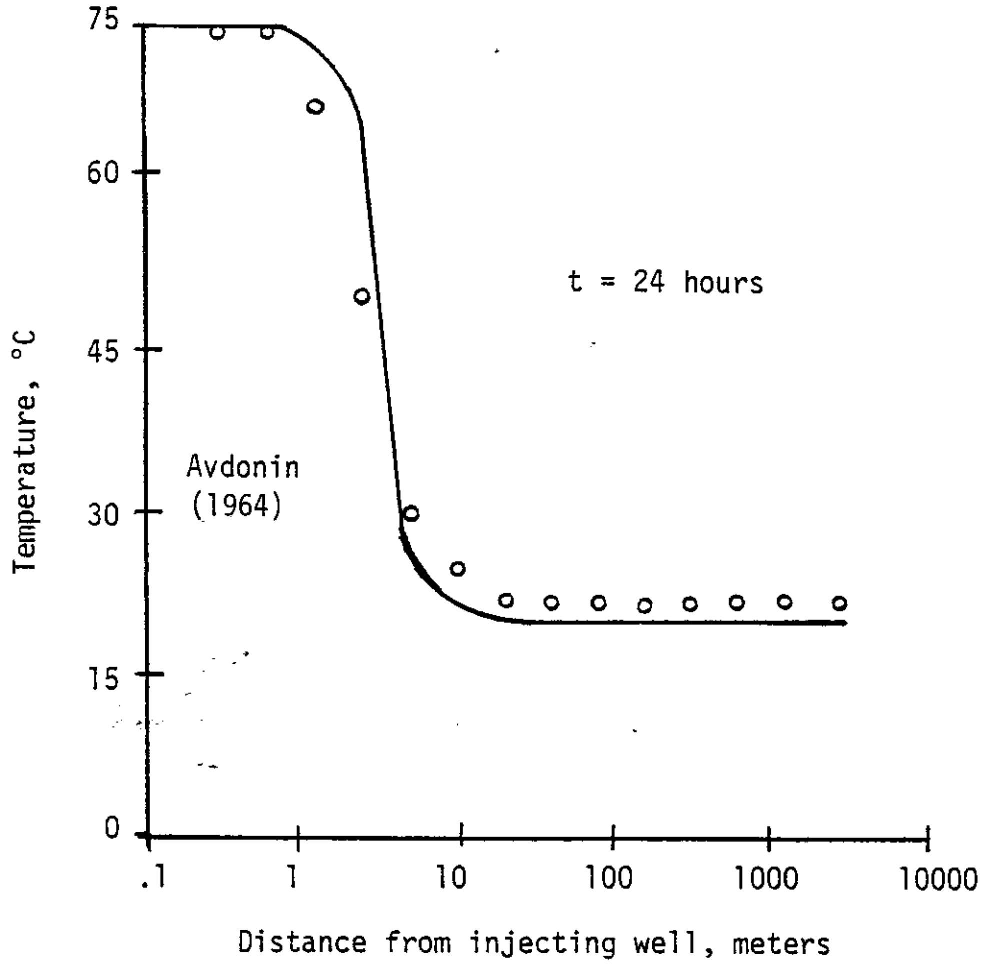


Figure 11. Comparison of the radial test problem transient temperature profile using upstream weighting to the solution of Avdonin (linear elements).

equation using a Galerkin approximation with triangular elements. Instead of using the Galerkin method in the time derivative (mass or "time" matrix), he used nodal values of the time derivatives which were weighted averages of the time derivatives over the entire flow region. This was done to avoid oscillations.

Langsrud (1976) used a scheme similar to Newman's, which he refers to as "capacity lumping", to reduce oscillations. Langsrud (1976) applied the Galerkin finite element method to a waterflood problem, solving for pressure and saturation simultaneously. In his lumping process, he added the off-diagonal matrix elements in the mass (time) matrix to the diagonal elements. Such a process is applicable to fully linear, quadratic, or cubic elements but cannot be generalized for mixed isoparametric elements.

It appears from the literature that diagonalization (lumping) of the mass (time) matrix is a possible solution to the oscillation problem. However, any proposed scheme must satisfy the finite element criteria of completeness and integrability; and total mass must be conserved in any lumping process. One such process was demonstrated by Hinton et al. (1976) for two dimensional isoparametric elements as follows.

Lumping may be considered as a process in which a different basis function, $\overline{N}$, is used to describe the unknown temperature distribution in equation 2 for those terms giving rise to the mass matrix. For example, $\overline{T} = \sum_{i=1}^{m} T_i N_i$ could be used for terms arising from the conductive terms of equation 2, which lead to the matrix [M] in equation 9. Simultaneously, $\overline{T} = \sum_{i=1}^{m} T_i \overline{N}_i$ could be used for terms leading to the mass matrix. If the basis function $\overline{N}$ are piecewise constants such that $\overline{N}_i = 1$ in a certain part of the element and zero elsewhere, and if such parts are not overlapping, the mass matrix $[M_T]$ of equation 9 becomes diagonal as;

$$\iiint_D (\rho_f \, C_f \, (1 - \theta) + \rho_w \, \theta \, C_p) \, \overline{N}_i \, \overline{N}_j = 0, \quad (i \neq j).$$

If the diagonal terms of the mass matrix are scaled so as to preserve the total element mass on the diagonal of the matrix, the above lumping method of Hinton is obtained identically without the introduction of additional basis functions. This lumping scheme was programmed into the model and tested using the grid and radial test problems.

Figure 12 presents a representative temperature profile for the grid test problem at t = 900,000 seconds. With the lumping technique applied to the mass matrix of the heat transport equation, the oscillation problem has disappeared. Both quadratic and cubic elements give favorable agreement with the analytical solution of Avdonin (1964). Figure 13 presents the temperature progression at x = 20 meters. Again, favorable results are obtained using mixed elements with the quadratic and cubic sides oriented in the flow direction. The use of linear elements tends to over-predict the temperature until late in the simulation. Both Figures 12 and 13 indicate that a small amount of undershoot existed as the thermal front approached x = 20 meters. This undershoot did not exceed 1°c at any time during the simulation.

Figures 12 and 13 also indicate that the sharp thermal front is still smeared. However, the mass matrix lumping technique exhibited

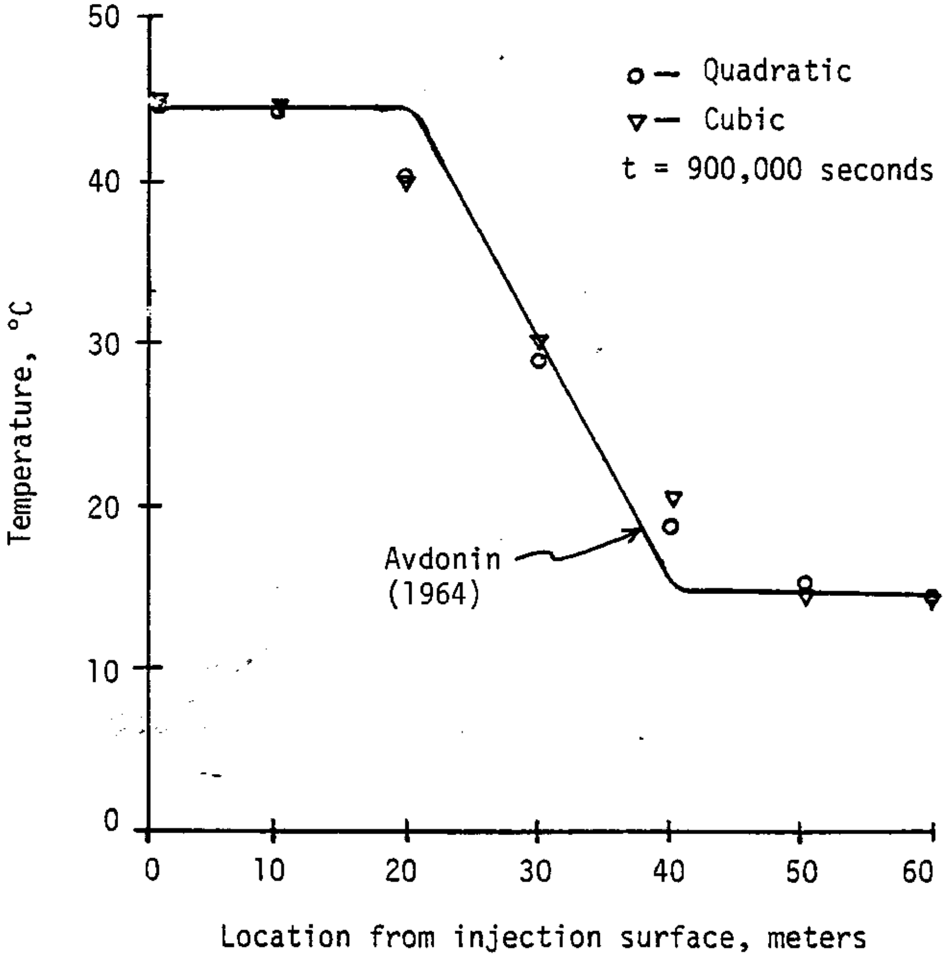


Figure 12. Comparison of the grid problem transient temperature profile using mass lumping to the solution of Avdonin.

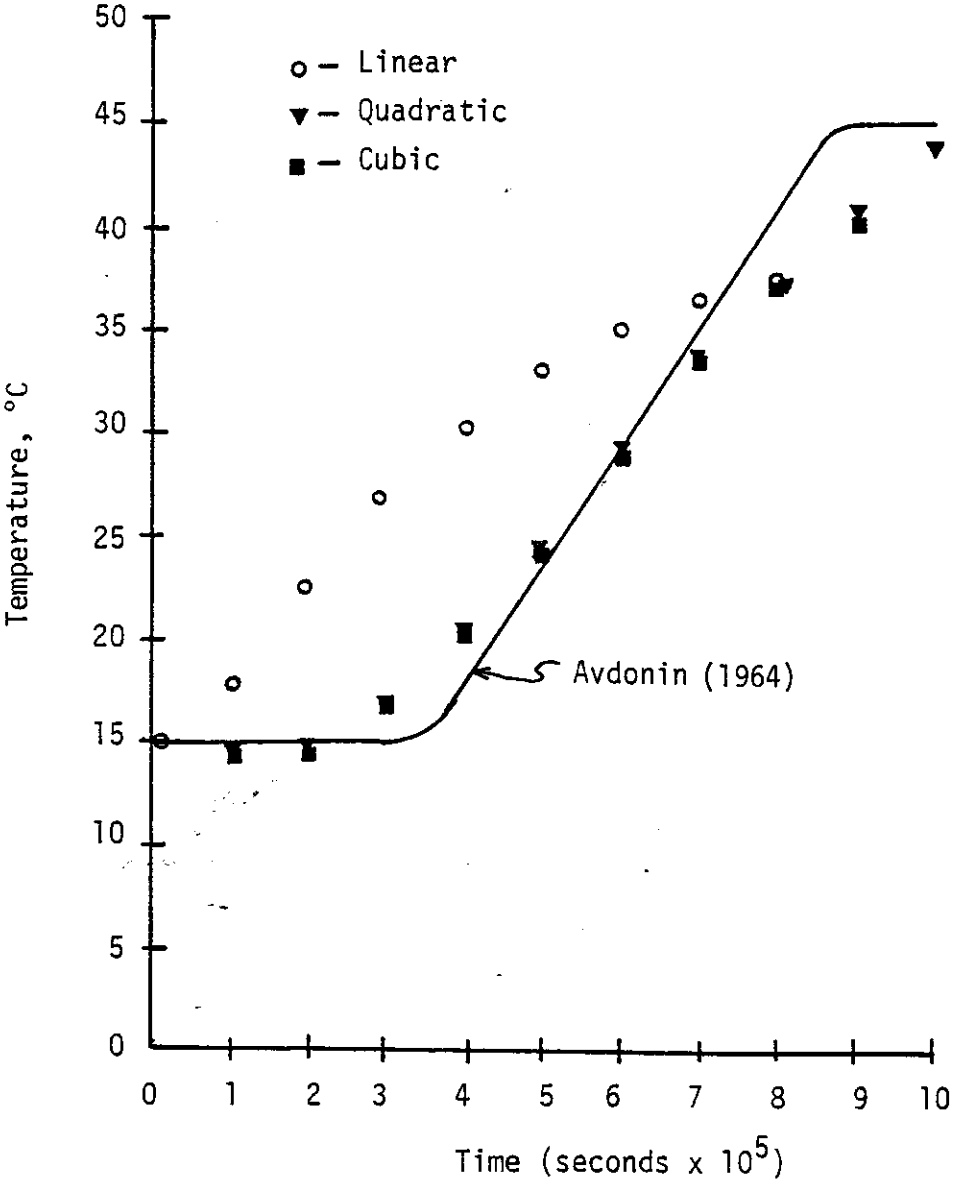


Figure 13. Comparison of the grid problem temperature progression using mass lumping to the solution of Avdonin at x = 20 meters.

much less smearing of the sharp thermal front than did the upstream weighting technique. As shown by the comparisons with Avdonin's solution, the location of the sharp front is easily determined with the mass lumping technique; which was not the case with the upstream weighting technique.

Figure 14 presents the temperature progression at a radius of 4.88 meters for the radial test problem. The use of mixed elements with cubic sides in the principal flow direction gave the closest agreement with Avdonin's solution. Again, undershoot never exceeded 1°c. However, the smearing problem was still evident. The numerical smear present with the use of the mass matrix lumping technique is due to spatial truncation. It is strongly possible that with lumping a decreased element size would decrease the smear of the sharp thermal front.

Effects of Increasing Aquifer Temperature on Fluid Pressure Distribution

Generally, when water is injected into an aquifer, the fluid pressures are expected to increase with time. Figure 15 indicates the effects of increasing aquifer temperature on water pressures. In Figure 15, pressures at the top of the flow layer are plotted for the grid test problem. As indicated by Figure 15, fluid pressures decreases as the aquifer temperature increases. This effect may be explained by examining the effects of temperature on hydraulic conductivity. Hydraulic conductivity is defined as;

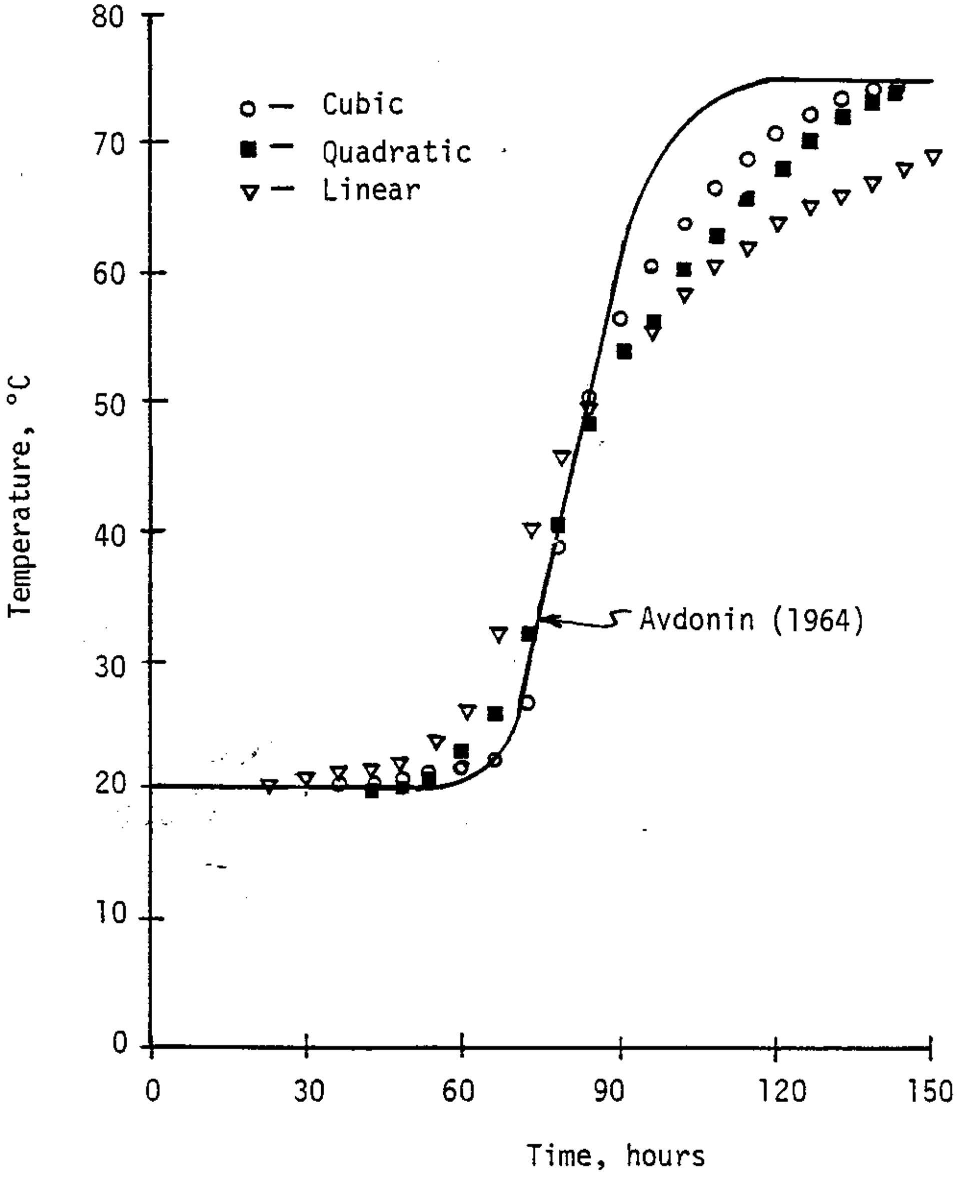


Figure 14. Comparison of the radial well problem temperature progression at a radius of 4.88 meters to the solution of Avdonin.

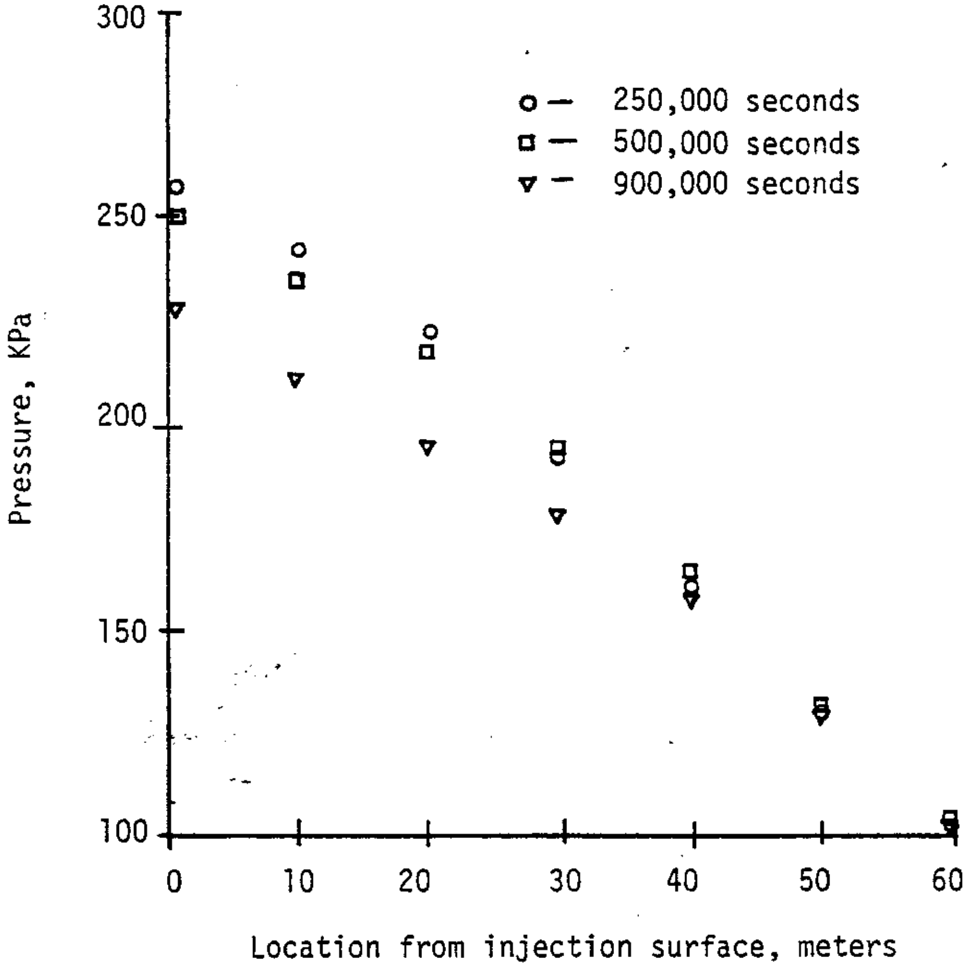


Figure 15. Effect of increasing aquifer temperature on the grid test problem transient pressure distribution.

$$K_c = \frac{k \rho g}{\mu}$$

where K_c = hydraulic conductivity; $L\ T^{-1}$,

k = specific permeability; L^2,

ρ = fluid density; $M\ L^{-3}$,

g = acceleration of gravity; $L\ T^{-2}$, and

μ = fluid dynamic viscosity; $F\ T\ L^{-2}$.

Fluid density and viscosity are temperature dependent in the model. As the temperature at a point increases, both density and viscosity decrease. However, viscosity decreases to a greater extent than density for a given change in temperature; resulting in an increase in hydraulic conductivity with an increase in temperature. For example, in the grid problem, the flow layer was initially at a temperature of 15°c. The viscosity and density of water were 0.115×10^{-2} Kg m^{-1} sec^{-1} and 993.4 Kg m^{-3}, respectively. The specific permeability of the porous media was specified as 1.0×10^{-11} m^2. The hydraulic conductivity was therefore 8.47×10^{-5} m sec^{-1}. With the passing of the temperature front, the temperature of the flow layer increased to 45°c. At 45°c, the viscosity and density of water are 0.597×10^{-3} Kg m^{-1} sec^{-1} and 987.3 Kg m^{-3}, respectively, and the resulting hydraulic conductivity is 1.62×10^{-4} m sec^{-1}. Thus, a ninety-one percent increase in hydraulic conductivity occurred because of an increase in water temperature from 15°c to 45°c. For this reason, the hydraulic conductivity in the simulation increased with time, and the fluid pressures declined as the thermal front moved through the aquifer. Figure 16 indicates similar results for the radial test problem. Although no analytical model exists to verify the pressure curves in this circumstance, Reed (1979) obtained similar results with a physical laboratory model.

Mass Balance

A mass balance was performed on each test problem after the lumping technique was added to the program. The mass balance was defined as $1 - \frac{\text{inflows-outflows}}{\text{change in storage}}$. The mass balance for the grid problem (mixed cubic elements) was 0.14×10^{-4}, and for the radial problem, 0.22×10^{-3}. In the radial problem, element volumes ranged from less than 1 m^3 to slightly less than 10^7 m^3. The larger mass balance error for the radial problem was probably due to numerical difficulties created by the large differential in element size.

Discussion

The results presented for the grid and radial finite element test problems indicate the degree of accuracy that may be expected from the model as it now stands. Although both the grid and radial test problems show that a degree of undershoot and smearing of the sharp thermal front still occur, even with the use of a mass lumping scheme, the model agrees closely with analytical solutions. Fluid pressures are accurately predicted, and the location of the sharp thermal front is easily determined. The numerical undershoot and smear that remain are due to spatial truncation and could probably be improved by the use of a smaller nodal spacing. However, the objective

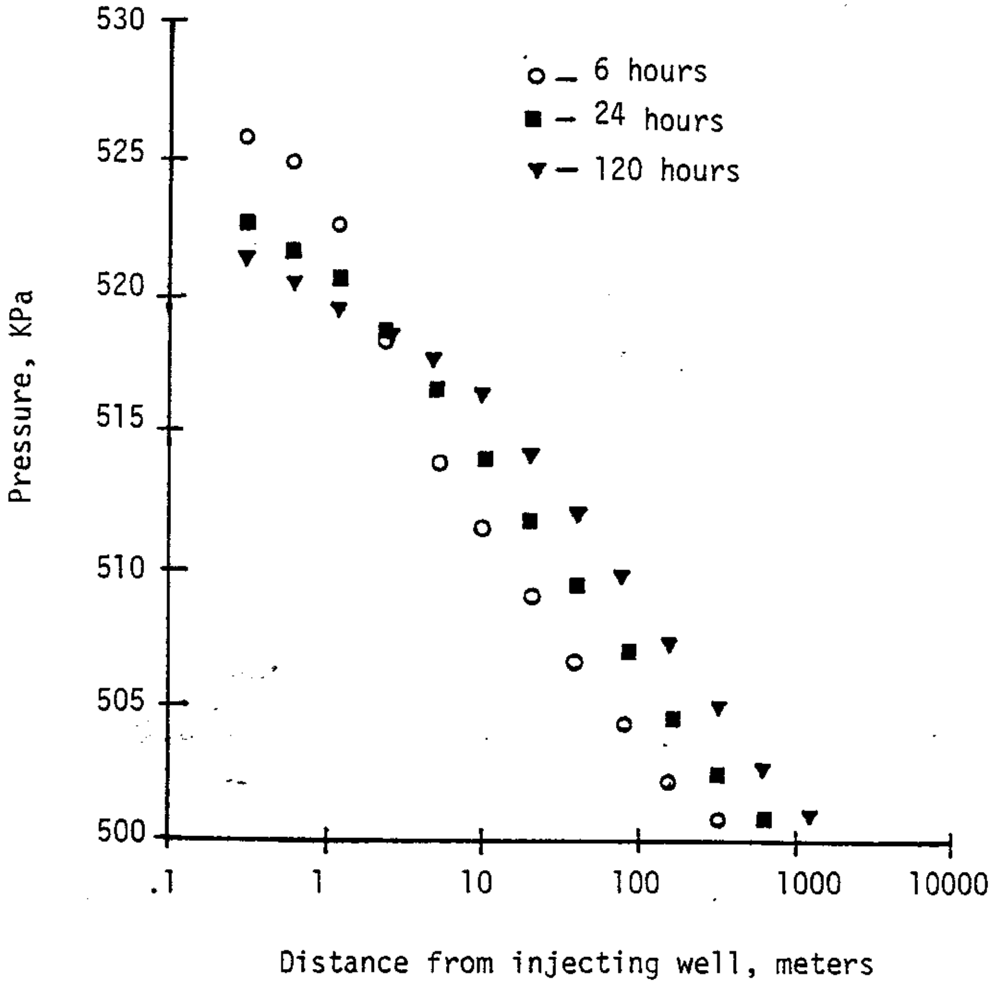


Figure 16. Effect of increasing aquifer temperature on the radial test problem transient pressure distributor.

of this study was the development of a model applicable to areal field type problems. In such problems, elements can be expected to vary greatly in size. While additional testing of the model is needed to establish relationships between element size, nodal spacing, and accuracy, some degree of numerical difficulties will have to be tolerated in large, areal type problems. The model should be tested on such problems to judge the effects of numerical smear and undershoot on their solutions.

The three-dimensional simulation model developed does not consider two phase flow and is therefore limited to saturated groundwater systems. The model was not designed to handle geothermal systems with temperature and pressure ranges allowing steam formation.

Often the cost of running a simulation model affects its use to almost the same degree as the accuracy of the model. As can be seen in Table IV, solution (CPU) time naturally increases as the number of equations to be solved (nodes) increase. For a given number of nodes solution time increases as higher ordered elements are used. This results from an increasing solution matrix bandwidth with more complex elements. Although element integration is performed only once and the resulting parameters stored on tape, a relatively large amount of CPU time is required even for small three-dimensional conduction-convection problems. Approximately sixty percent of the CPU time used is required to estimate the nodal mass fluxes in three directions. Given a number of nodes n for a problem, a banded n x n matrix, with the bandwidth dependent on element configuration, must be solved for each direction in order to determine the nodal mass fluxes. Although

Table IV

Required CPU Time for Problem Solution

Seconds/Time Step

Nodal order in principal flow direction	Grid problem (28 nodes)	Radial problem (56 nodes)
Linear	4.5	5.1
Quadratic	4.9	7.0
Cubic	6.0	10.5

a very efficient matrix solving routine is used in the model, perhaps additional work in this area would be of benefit. In any case, however, large amounts of computer time will be required for the simulation of moderate sized three-dimensional finite element problems.

The theoretical development of the Galerkin method of finite element approximation is mathematically abstract, as is the Hinton mass lumping technique. Once developed and programmed, however, the method may be applied to a wide range of convection-conduction groundwater problems without major modification. The model is now capable of acceptably handling sharp thermal front problems. It should be applicable to general field type problems with a good degree of confidence, although the degree of error for such problems needs to be quantified.

CHAPTER VI

SUMMARY AND CONCLUSIONS

A mathematical model was developed in which the groundwater flow equation and the energy transport equation are solved simultaneously using a finite difference approximation for the time derivative and Galerkin-finite element approximations for the space derivatives. An operational test of the numerical model was made using two problems for which analytical solutions could be obtained. Both problems involved hot water injection into a confined aquifer. In each case, the injected water was at a much higher temperature than the natural groundwater. The resulting mass fluxes were relatively large resulting in a sharp thermal front problem (i.e., a problem with temperature changing rapidly over time or space). Due to the nature of the two test problems, it was felt that if the model performed satisfactorily, it could be used for general field type problems with a high degree of confidence.

The first test problem consisted of a 28 node grid type convection conduction problem with fluid being injected through the end surface of a three-dimensional strip of elements. The second test problem consisted of a 56 node, 22.5° radial well problem. Linear and mixed three-dimensional isoparametric elements with quadratic and cubic nodal configurations in the principal flow directions were used with a constant nodal arrangement for each problem.

Initially, each test problem was simulated with mass injection, but without heat transport, in order to evaluate the pressure part of the model. Predicted fluid pressures closely matched analytical solutions for both test problems. While excellent results were obtained from the pressure section of the model (without heat transport), when heat transport was included, the temperature section failed to give an acceptable result. Overshoot and undershoot oscillations occurred about the sharp thermal front. These oscillations grew in time until the simulation failed.

Several techniques were used in attempts to remove the oscillations. Reducing element size and using smaller time steps did not improve the solution. An upstream weighting scheme corrected the oscillation problem, but an unacceptable amount of smear was introduced to the sharp thermal front. A mass lumping scheme also corrected the oscillation problem with some undershoot, and it also provided an improved approximation to the sharp thermal front (smear was reduced).

It now appears that the numerical model gives acceptable results to the test problems used in this analysis. Fluid pressures are accurately predicted, and the location of the sharp thermal front is easily determined. The numerical undershoot and smear that remain are due to spatial truncation and could probably be improved by the use of a smaller element size. The objective of this study was the development of a model applicable to areal field type conduction-convection problems. In such problems, elements would be expected to vary greatly in size. While additional testing of the model is needed to establish relationships between element size, nodal configurations, and accuracy, some degree of numerical difficulties will have to be tolerated in large areal type problems. The model should be tested on such problems to quantify the effects of numerical smear and undershoot on their solutions.

The theoretical development of the Galerkin method of finite element approximation is mathematically abstract, as is the Hinton mass lumping technique. Once developed and programmed the method may be applied to a wide range of convection-conduction groundwater problems without major modification. The model is now capable of acceptably handling sharp thermal front problems. It should be applicable to general field type problems with a good degree of confidence.

Future research should include;

A) simulating additional radial and grid type test problems in order to quantify the effects of element size and time step size on solution accuracy,

B) simulation of moderate size field problems in order to quantify the numerical difficulties of smear and undershoot for such problems,

C) a sensitivty analysis to quantify the effects of changing parameters on the final solution,

D) laboratory physical verification to evaluate the ultimate accuracy of the numerical solution, and

E) the development of problem size - CPU required relationships to establish the size of problem that can be reasonably modeled for given cost constraints.

LIST OF REFERENCES

1. Assad, Youssi. 1955. A study of the thermal conductivity of fluid bearing porous rocks. Ph.D. thesis. University of California.

2. Avdonin, N. A. 1964. Some formulas for calculating the temperature field of a stratum subject to thermal injection. Neft' i Gaz 7(3):37.

3. Aziz, K., S. A. Bories, and M. A. Combarnous. 1973. The influence of natural convection in gas, oil, and water reservoirs. Journal of Canadian Petroleum Technology. 12:41-47.

4. Bredoehoeft, J. D. and G. F. Pinder. 1970. Digital analysis of areal flow in multiaquifer groundwater systems: a quasi-three-dimensional model. Water Resources Research. 6(3):883-888.

5. Chappelear, J. E. and C. W. Volek. 1969. The injection of a hot liquid into a porous medium. Society of Petroleum Engineers Journal. 9(1):100-114.

6. Connor, J. J. and C. A. Brebbia. 1976. Finite Element Techniques for Fluid Flow. Butterworth, Inc., Boston.

7. Davis, S. N. and R. J. N. DeWiest. 1967. Hydrogeology. John Wiley and Sons, New York.

8. Davison, R. R., W. B. Harris and J. H. Martin. 1975. Storing sunlight underground. Chemical Technology. 5:736-741.

9. Douglas, J. and T. Dupont. 1970. Galerkin methods for parabolic equations. Journal of Numerical Analysis. 7(4):575-626.

10. Ebeling, L. L. and D. L. Reddell. 1976. Energy (hot water) storage in-groundwater aquifers. American Society of Agricultural Engineers, Technical Paper No. 76-2540.

11. Ergatoudis, I., B. M. Irons and O. C. Zienkiewicz. 1968. Curved, isoparametric, "quatrilateral" elements for finite element analysis. International Journal of Solids and Structures. 4:31-42.

12. Faust, C. R. and J. W. Mercer. 1976. An analysis of finite difference and finite element techniques for geothermal reservoir simulation. Fourth Symposium of Numerical Simulation of Reservoir Performance. Society of Petroleum Engineers.

13. Faust, C. R. and J. W. Mercer. 1977. Theoretical analysis of fluid flow and energy transport in hydrothermal systems. U.S. Geological Survey Open-File Report 77-60.

14. Fayers, F. J. and J. W. Sheldon. 1967. The use of a high speed digital computer in the study of the hydrodynamics of geologic basins. Journal of Geophysical Research. 67(6):2421-2431.

15. Finlayson, B. A. and L. E. Scriven. 1966. The method of weighted residuals - a review. Applied Mechanics Review 1969.

16. Finlayson, B. A. 1972. The method of weighted residuals and variational principles. Academic Press, New York.

17. Freeze, R. A. and P. A. Witherspoon. 1966. Theoretical analysis of regional groundwater flow, 1., analytical and numerical solutions to the mathematical model. Water Resources Research. 2(4):641-656.

18. Freeze, R. A. 1971. Three-dimensional, transient, saturated-unsaturated flow in a groundwater basin. Water Resources Research. 7(2):347-366.

19. Gringarten, A. C. and J. P. Sauty. 1975. A theoretical study of heat extraction from aquifers with uniform regional flow. Journal of Geophysical Research. 80(35):4956-4962.

20. Gupta, S. K., K. K. Tanji and J. N. Luthin. 1975. A three-dimensional finite element groundwater model. Contribution No. 152, California Water Resources Center, University of California at Davis.

21. Hantush, M. S. 1960. Modification of the theory of leaky aquifers. Journal of Geophysical Research. 65(11):3713-3725.

22. Hinton, E., A. Rock and O. C. Zienkiewicz. 1976. A note on mass lumping in related process in the finite element method. Int. Journal of Earthquake Engr. and Structural Dynamics. 4:245-249.

23. Hottel, H. C. and B. B. Woertz. 1942. The performance of flat-plate solar heat collectors. Transactions of the A.S.M.E. 64: 91-104.

24. Huebner, K. H. 1975. The finite element method for engineers. John Wiley and Sons, New York.

25. Huyakorn, P. S., J. C. Heinrich and O. C. Zienkiewicz. 1977. An upwind finite element scheme for two dimensional convective transport equation. Int. Journal for Numerical Methods in Engr. 11:131-143.

26. Javandel, I. and P. A. Witherspoon. 1969. A method of analyzing transient fluid flow in multilayered aquifers. Water Resources Research. 5(4):856-870.

27. Jennings, J. 1969. The application of variational methods for calculating two-dimensional immiscible displacement in porous media. Ph.D. thesis, University of Pittsburgh, Pa.

28. Kelly, K. K. 1949. Contributions to the data on theoretical metallurgy. Bulletin of the United States Bureau of Metals.

29. Langsrud, O. 1976. Simulation of two-phase flow by finite element element methods. Fourth S.P.E. Symposium on Numerical Simulation of Reservoir Performance. Paper No. 5725.

30. Mercer, J. W. 1973. Finite element approach to the modeling of hydrothermal systems. Ph.D. thesis. University of Illinois.

31. Meyer, C. F. and D. K. Todd. 1973. Conserving energy with heat storage wells. Environmental Science and Technology. 7(6): 512-518.

32. Neuman, S. P. and P. A. Witherspoon. 1969a. Transient flow of ground water to wells in multiple aquifer systems. Geotechnical Engineering Report. University of California.

33. Neuman, S. P. and P. A. Witherspoon. 1969b. Theory of flow in a two aquifer system. Water Resources Research. 5(4):803-816.

34. Neuman, S. P. 1974. Saturated-unsaturated seepage by finite elements. Journal of the Hydraulic Division, American Society of Civil Engineers. Vol. 99, No. HY12, 2233-2250.

35. Pinder, G. F. and E. O. Frind. 1972. Application of Galerkin's procedure to aquifer analysis. Water Resources Research. 8(1): 108-120.

36. Pinder, G. F. and W. G. Gray. 1977. Finite element simulation in surface and subsurface hydrology. Academic Press. New York.

37. Rabbimov, R. T., G. A. Umorov and R. A. Zakhidov. 1974. Experimental study of aquifer heating in solar energy accumulation. Geliotekhnika. 10(2):20-27.

38. Reed, D. B. 1979. Modeling of thermal storage in groundwater aquifers. M.S. thesis. Texas A&M University. Dept. of Agricultural Engineering.

39. Smith, N. and R. C. Hill. 1976. A solar assisted heat pump system. American Society of Agricultural Engineers Technical Paper No. 76-4018.

40. Sorey, M. L. 1976. Numerical modeling of liquid geothermal systems. United States Geological Survey, Open File Report 75-613.

41. Strang, G. and G. J. Fix. 1973. An analysis of the finite element method. Prentice-Hall, Englewood Cliffs, N.J.

42. Theis, C. V. 1935. The relation between the lowering of the piezometric surface and the rate and duration of discharge of a well using groundwater storage. EOS Transactions AGU, 16, 519-524.

43. Vargaftick, N. B. 1975. Tables on the thermophysical properties of liquids and gases. John Wiley and Sons, New York.

44. Walton, W. C. 1970. Groundwater resource evaluation. McGraw-Hill, New York.

45. Weinstock, R. 1952. Calculus of variations. McGraw-Hill, New York.

46. Wilhite, G. P. and J. Wagner. 1974. Disposal of heated water through groundwater systems. Kansas Water Resources Research Institute.

47. Zienkiewicz, O. C. and C. J. Parker. 1970. Transient field problems: two and three dimensional analysis by isoparametric finite elements. Int. Journal of Numerical Methods in Engineering. 2:61-71.

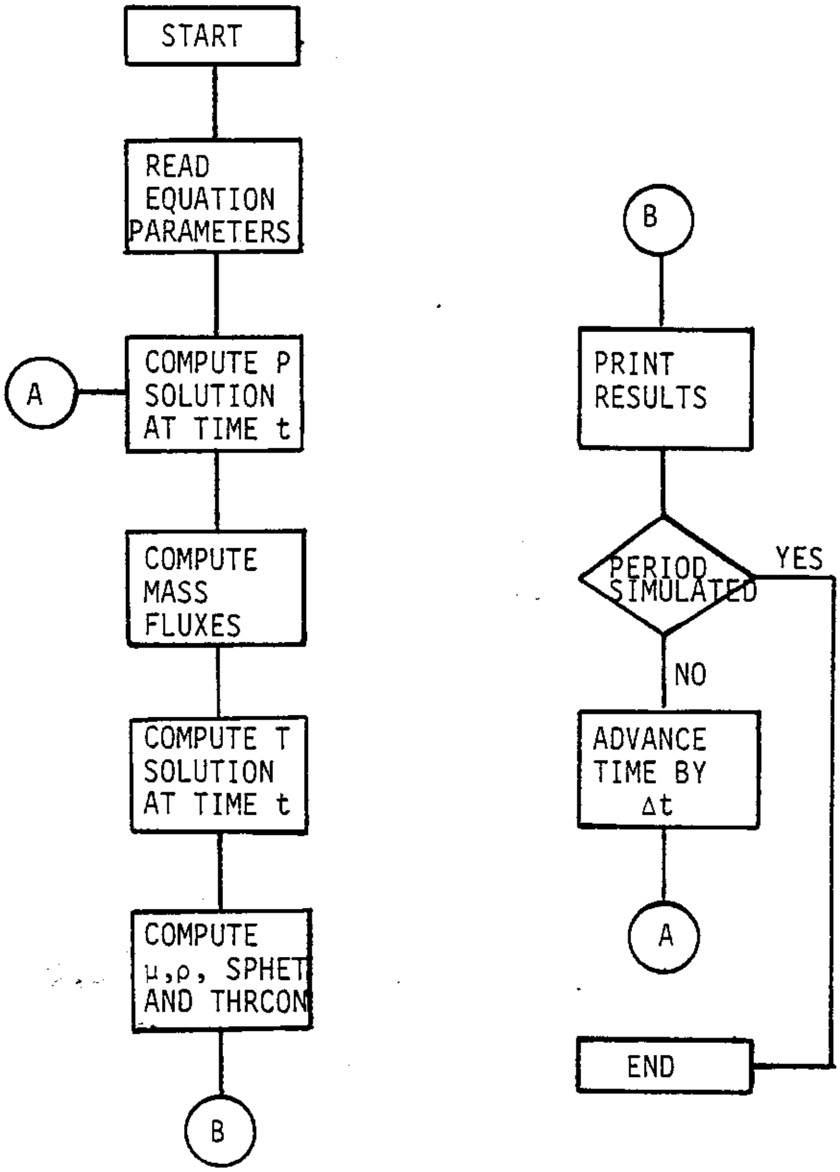
48. Zienkiewicz, O. C. 1971. The finite element method. McGraw-Hill, London. 2nd Ed.

49. Zienkiewicz, O. C. 1977. The finite element method. McGraw-Hill, London. 3rd. Ed.

APPENDIX A

SIMULATION FLOW CHART

SIMULATION FLOW CHART



APPENDIX B

DERIVATION OF EQUATIONS

DERIVATION OF EQUATIONS

Equation of Groundwater Flow

The equation of groundwater flow must obey

a) Darcy's Law, and

b) Conservation of mass.

Darcy's Law is stated as

$$q = -K_c \frac{\partial h'}{\partial L}$$

where q = flux, volumetric flow rate per unit area; $L^3\ L^{-2}\ T^{-1}$,

K_c = hydraulic conductivity; $L\ T^{-1}$,

h' = hydraulic head; L,

L = length of flow path; L.

K, the hydraulic conductivity may be written as

$$K = \frac{k\ \rho\ g}{\mu}$$

where k = specific permeability of the porous media; L^2,

ρ = fluid density; $M\ L^{-3}$,

g = acceleration of gravity; $L\ T^{-2}$, and

μ = fluid dynamic viscosity; $F\ T\ L^{-2}$.

The conservation of mass requires that

mass in - mass out = mass accumulation.

Referring to the following figure;

inflows

$$\rho q_x|_x \, \Delta y \Delta z \Delta t + \rho q_y|_y \, \Delta x \Delta z \Delta t + \rho q_z|_z \, \Delta x \Delta y \Delta t + Q_{in} \, \Delta t$$

outflows

$$\rho q_x|_{x+\Delta x} \, \Delta y \Delta z \Delta t + \rho q_y|_{y+\Delta y} \, \Delta x \Delta z \Delta t + \rho q_z|_{z+\Delta z} \, \Delta x \Delta y \Delta t + Q_{out} \, \Delta t$$

accumulation

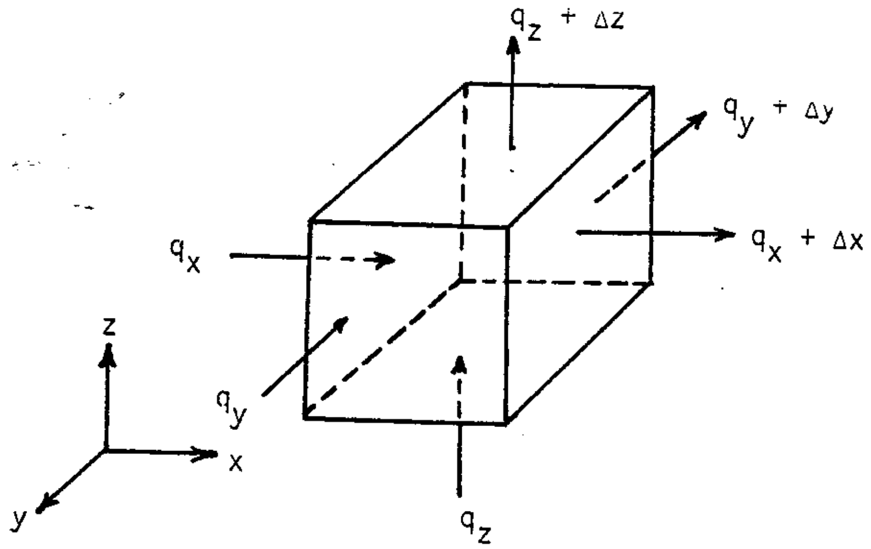
$(\Delta z \rho \theta|_{t+\Delta t} - \Delta z \rho \theta|_t) \, \Delta x \Delta y$ allowing for vertical compressibility.

where Q_{in} = mass flow rate added from a source; $M \, T^{-1}$,

Q_{out} = mass flow rate withdrawn by a sink; $M \, T^{-1}$,

q_i = volumetrix flux in "i" direction; $L \, T^{-1}$,

θ = porosity.



Substituting into the conservation of mass equation;

$$\rho q_x|_x \Delta y\Delta z\Delta t + \rho q_y|_y \Delta x\Delta z\Delta t + \rho q_z|_z \Delta y\Delta x\Delta t + Q_{in} \Delta t$$

$$- \rho q_x|_{x+\Delta x} \Delta y\Delta z\Delta t - \rho q_y|_{y+\Delta y} \Delta x\Delta z\Delta t + \rho q_z|_{z+\Delta z} \Delta y\Delta x\Delta t - Q_{out} \Delta t$$

$$= (\Delta z\rho\theta|_{t+\Delta t} - \Delta z\rho\theta|_t) \Delta x\Delta y$$

Rearranging and dividing by ΔxΔyΔzΔt results in

$$- \frac{\rho q_x|_{x+\Delta x} - \rho q_x|_x}{\Delta x} - \frac{\rho q_y|_{y+\Delta y} - \rho q_y|_y}{\Delta y} - \frac{\rho q_z|_{z+\Delta z} - \rho q_z|_z}{\Delta z}$$

$$+ \frac{Q_{in} - Q_{out}}{\Delta x\Delta y\Delta z} = \frac{\frac{1}{\Delta z} (\Delta z\rho\theta|_{t+\Delta t} - \Delta z\rho\theta|_t)}{\Delta t}$$

Taking the limit as ΔxΔyΔzΔt goes to zero,

$$- [\frac{\partial}{\partial x} (\rho q_x) + \frac{\partial}{\partial y} (\rho q_y) + \frac{\partial}{\partial z} (\rho q_z)] + Q = \frac{1}{\Delta z} \frac{\partial}{\partial t} (\Delta z\rho\theta) \qquad (A)$$

where Q = net rate of mass flow added or withdrawn per unit volume of porous media due to the operation of sources and sinks.

The piezometric pressure is defined as

$$P^* = P + \rho gh .$$

The piezometric head may be defined as

$$h' = \frac{P}{\rho g} + h$$

where h = elevation above a datum; L.

Therefore, the hydraulic head may be expressed as

$$h' = \frac{P^*}{\rho g}$$

Substituting into Darcy's Law,

$$q_x = - K_x \frac{\partial h'}{\partial x} = - \frac{k\rho g}{\mu} \partial(\frac{P^*}{\rho g})/\partial x = \frac{k_x}{\mu} \frac{\partial P^*}{\partial x} = - \frac{k_x}{\mu} [\frac{\partial P}{\partial x} + \rho g \frac{\partial h}{\partial x}] \; .$$

Similarly

$$q_y = - \frac{k_y}{\mu} [\frac{\partial P}{\partial y} + \rho g \frac{\partial h}{\partial y}]$$

$$q_z = - \frac{k_z}{\mu} [\frac{\partial P}{\partial z} + \rho g \frac{\partial h}{\partial z}]$$

Substituting into equation A,

$$\frac{\partial}{\partial x} [\frac{k_x\rho}{\mu} (\frac{\partial P}{\partial x} + \rho g \frac{\partial h}{\partial x}) + \frac{\partial}{\partial y}[\frac{k_y\rho}{\mu} (\frac{\partial P}{\partial y} + \rho g \frac{\partial h}{\partial y})] + \frac{\partial}{\partial z} [\frac{k_z\rho}{\mu} (\frac{\partial P}{\partial z} + \rho g \frac{\partial h}{\partial z})]$$

$$+ Q = \frac{1}{\Delta z} \frac{\partial}{\partial t} (\Delta z\rho\theta).$$

Expanding the right hand side,

$$\frac{1}{\Delta z} \frac{\partial}{\partial t} (\Delta z\rho\theta) = \frac{1}{\Delta z} (\Delta z\rho \frac{\partial\theta}{\partial t} + \theta\frac{\partial(\Delta z\rho)}{\partial t}$$

$$= \frac{1}{\Delta z} [\Delta z\rho \frac{\partial\theta}{\partial t} + \theta(\Delta z \frac{\partial\rho}{\partial t} + \rho\frac{\partial\Delta z}{\partial t})]$$

$$= \rho\frac{\partial\theta}{\partial t} + \theta \frac{\partial\rho}{\partial t} + \frac{1}{\Delta z} \theta \rho \frac{\partial\Delta z}{\partial t}$$

Walton (1970) defines

$$\frac{\partial\Delta z}{\Delta z} = \alpha\partial P$$

where α = vertical compressibility of the porous media; $L^2 F^{-1}$. According to a derivation by Davis and DeWiest (1967),

$$\frac{\partial \theta}{\partial t} = (1 - \theta) \frac{\partial P}{\partial t}$$

so

$$\rho\alpha(1 - \theta) \frac{\partial P}{\partial t} + \frac{\partial}{\partial t} + \alpha\rho\theta \frac{\partial P}{\partial t} = \frac{1}{\Delta z} \frac{\partial}{\partial t} (\Delta z \rho \theta).$$

To evaluate $\frac{\partial \rho}{\partial t}$, density is assumed to be a function of temperature and pressure. ρ may be expanded about a "ρ_o" in a Taylor series truncated with first order terms.

$$\rho = \rho_o + \left(\frac{\partial \rho}{\partial t}\right)_o (T - T_o) + \left(\frac{\partial \rho}{\partial P}\right)_o (P - P_o) \qquad (B)$$

where ρ_o is the average density of the fluid at a reference temperature (T_o) and pressure (P_o).

From equation (B), the following may be defined,

$$\beta_T = \frac{\left(\frac{\partial \rho}{\partial t}\right)_o}{\rho_o} \qquad \beta_P = \frac{\left(\frac{\partial \rho}{\partial P}\right)_o}{\rho_o}$$

where

β_T = coef. of fluid thermal volume expansion; T^{-1},

β_P = compressibility coef. for fluid; $L^2 F^{-1}$.

Now

$$\rho = \rho_o + \rho_o \beta_T (T - T_o) + \rho_o \beta_P (P - P_o) \quad .$$

Taking the time derivative

$$\frac{\partial \rho}{\partial t} = \frac{\partial \rho_o}{\partial t} + \rho_o \beta_T \frac{\partial T}{\partial t} + \partial_o \beta_P \frac{\partial P}{\partial t}$$

so

$$\theta \frac{\partial \rho}{\partial t} = \theta(\rho_o \beta_T \frac{\partial T}{\partial t} + \rho_o \beta_P \frac{\partial P}{\partial t}) \quad .$$

Thus

$$\frac{1}{\Delta z} \frac{\partial}{\partial t} (\Delta z \rho \theta) = \rho \alpha (1 - \theta) \frac{\partial P}{\partial t} + \theta \rho_o \beta_T \frac{\partial T}{\partial t} + \theta \rho_o \beta_P \frac{\partial P}{\partial t} + \alpha \rho \theta \frac{\partial P}{\partial t}$$

$$= (\rho \alpha (1 - \theta) + \theta \rho_o \beta_p + \alpha \rho \theta) \frac{\partial P}{\partial t} + \theta \rho_o \beta_T \frac{\partial T}{\partial t}$$

$$= (\rho \alpha + \theta \rho_o \beta_p) \frac{\partial P}{\partial t} + \theta \rho_o \beta_T \frac{\partial T}{\partial t} \quad .$$

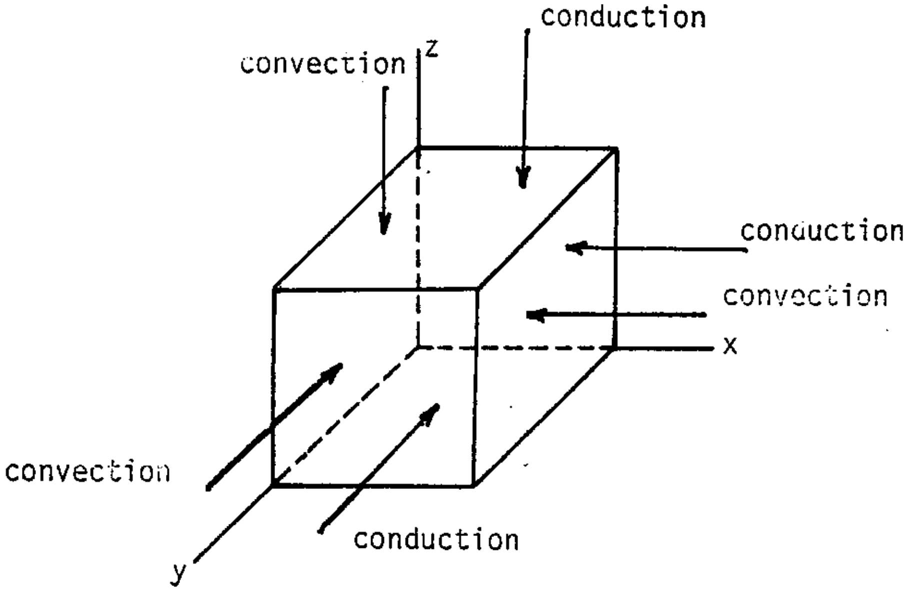
Thus, the final groundwater flow equation becomes

$$\frac{\partial}{\partial x} [\frac{k_x \rho}{\mu} (\frac{\partial P}{\partial x} + \rho g \frac{\partial h}{\partial x})] + \frac{\partial}{\partial y} [\frac{k_y \rho}{\mu} (\frac{\partial P}{\partial y} + \rho g \frac{\partial h}{\partial y})] + \frac{\partial}{\partial z} [\frac{k_z \rho}{\mu} (\frac{\partial P}{\partial z} + \rho g \frac{\partial h}{\partial z})]$$

$$+ Q = (\rho \alpha + \theta \rho_o \beta_p) \frac{\partial P}{\partial t} + \theta \rho_o \beta_T \frac{\partial T}{\partial t}$$

Equation of Energy Transport

Consider an incremental volume element.



The conservation of energy principle requires that, over a time increment Δt;

Energy in - Energy out = Energy accumulation

Referring to the above figure and defining the mass flux (v) as the products of the volumetric flux and fluid density;

$$v_x\ C_p\ (T - T_{ref})|_x\ \Delta y \Delta z \Delta t + v_y\ C_p\ (T - T_{ref})|_y \Delta x \Delta z \Delta t$$

$$+ v_z\ C_p\ (T - T_{ref})|_z\ \Delta x \Delta y \Delta t$$

$$- v_x\ C_p\ (T - T_{ref})|_{x+\Delta x}\ \Delta y \Delta z \Delta t - v_y\ C_p\ (T - T_{ref})|_{y+\Delta y}\ \Delta x \Delta z \Delta t$$

$$- v_z\ C_p\ (T - T_{ref})|_{z+\Delta z}\ \Delta x \Delta y \Delta t$$

$$+ q_x|_x\ \Delta y \Delta z \Delta t + q_y|_y\ \Delta x \Delta z \Delta t + q_z|_z\ \Delta x \Delta y \Delta t$$

$$- q_x|_{x+\Delta x}\ \Delta y \Delta z \Delta t - q_y|_{y+\Delta y}\ \Delta x \Delta z \Delta t - q_z|_{z+\Delta z}\ \Delta x \Delta y \Delta t$$

$$+ C_p\ (T_{inj} - T_{ref})\ Q_{in}\ \Delta t - C_p\ (T - T_{ref})\ \dot{Q}_{out}\ \Delta t -$$

$$- C_p\ (T_p - T)\ \dot{Q}_{out}\ \Delta t$$

$$= (\rho_r\ C_r\ \Delta z\ (T - T_{ref})|_{t+\Delta t} - \rho_r\ C_r\ \Delta z\ (T - T_{ref})|_t)\ \Delta x \Delta y$$

The first set of terms represent the energy flow due to convection, the second set of terms represent the energy flow due to conduction, the next terms represent the energy added or withdrawn due to the operation of mass sources or sinks. The last group of terms represent the energy accumulated during t. All flow terms (v, Q) are expressed as mass flows, not volumetric. "v" is a mass flux term, not seepage velocity. The elemental volume is assumed deformable in the z direction.

From Fourier's Law;

$$q_x = - K_x \frac{\partial T}{\partial x} \qquad q_y = - K_y \frac{\partial T}{\partial y} \qquad q_z = - K_z \frac{\partial T}{\partial z}$$

substituting into the above equation and dividing through by $\Delta x \Delta y \Delta z \Delta t$;

$$- \frac{[v_x\ C_p\ (T - T_{ref})|_{x+\Delta x} - v_x\ C_p\ (T - T_{ref})|_x]}{\Delta x}$$

$$- \frac{[v_y\ C_p\ (T - T_{ref})|_{y+\Delta y} - v_y\ C_p\ (T - T_{ref})|_y]}{\Delta y}$$

$$- \frac{[v_z\ C_p\ (T - T_{ref})|_{z+\Delta z} - v_z\ C_p\ (T - T_{ref})|_z]}{\Delta Z}$$

$$+ \frac{(K_x \frac{\partial T}{\partial x}|_{x+\Delta x} - K_x \frac{\partial T}{\partial x}|_x)}{\Delta x} + \frac{(K_y \frac{\partial T}{\partial y}|_{y+\Delta y} - K_y \frac{\partial T}{\partial y}|_y)}{\Delta y}$$

$$+ \frac{(K_z \frac{\partial T}{\partial z}|_{z+\Delta z} - K_z \frac{\partial T}{\partial z}|_z)}{\Delta z}$$

$$+ \frac{C_p\ (T_{inj} - T_{ref})\ \dot{Q}_{in} - C_p\ (T - T_{ref})\ Q_{out} - C_p\ (T_p - T)\ \dot{Q}_{out}}{\Delta x \Delta y \Delta z}$$

$$= \frac{\frac{1}{\Delta z}\ [\rho_r\ C_r\ \Delta z\ (T - T_{ref})|_{t+\Delta t} - \rho_r\ C_r\ \Delta z\ (T - T_{ref})|_t]}{\Delta t}$$

Taking limits as $\Delta x \Delta y \Delta z \Delta t$ approaches zero;

$$- \frac{\partial}{\partial x} [v_x\ C_p\ (T - T_{ref})] - \frac{\partial}{\partial y} [v_y\ C_p\ (T - T_{ref})]$$

$$- \frac{\partial}{\partial z} [v_z\ C_p\ (T - T_{ref})]$$

$$+ \frac{\partial}{\partial x} (K_x \frac{\partial T}{\partial x}) + \frac{\partial}{\partial y} (K_y \frac{\partial T}{\partial y}) + \frac{\partial}{\partial z} (K_z \frac{\partial T}{\partial z})$$

$$+ C_p\ Q_{in}\ [T_{inj} - T_{ref}] - C_p\ (T - T_{ref})\ Q_{out} - C_p\ (T_p - T)\ \dot{Q}_{out}$$

$$= \frac{1}{\Delta z} \frac{\partial}{\partial t} [\rho_r\ C_r\ \Delta z\ (T - T_{ref})].$$

$\dot{Q}$ is the mass flow rate per unit volume. Now, the storage term may be expanded with

$$\rho_r\ C_r\ \Delta z = \Delta z\ (\rho_f\ C_f\ (1 - \theta) + \rho_w\ C_p\ \theta)$$

so $\frac{1}{\Delta z} \frac{\partial}{\partial t} [\rho_r\ C_r\ \Delta z\ (T - T_{ref})] = \frac{1}{\Delta z} \frac{\partial}{\partial t} [\Delta z\ (\rho_f\ C_f\ (1 - \theta) + \rho_w\ C_p\ \theta)$

$(T - T_{ref})]$.

Expanding the convection term,

$$- v_x \frac{\partial}{\partial x} [C_p \; (T - T_{ref})] - C_p \; (T - T_{ref}) \; \frac{\partial (v_x)}{\partial x}$$

$$- v_y \frac{\partial}{\partial y} [C_p \; (T - T_{ref})] - C_p \; (T - T_{ref}) \; \frac{\partial (v_y)}{\partial y}$$

$$- v_z \frac{\partial}{\partial z} [C_p \; (T - T_{ref})] - C_p \; (T - T_{ref}) \; \frac{\partial (v_z)}{\partial z}$$

$$+ \frac{\partial}{\partial x} (K_x \frac{\partial T}{\partial x}) + \frac{\partial}{\partial y} (K_y \frac{\partial T}{\partial y}) + \frac{\partial}{\partial z} (K_z \frac{\partial T}{\partial z})$$

$$+ C_p \; Q_{in} \; [T_{inj} - T_{ref})] - C_p \; (T - T_{ref}) \; \dot{Q}_{out} - C_p \; (T_p - T) \; \dot{Q}_{out}$$

$$= \frac{1}{\Delta z} \frac{\partial}{\partial t} [\Delta z \; (\rho_f \; C_f \; (1 - \theta + \rho_w \; C_p \; \theta) \; (T - T_{ref})] \; .$$

Expressing in vector notation results in

$$v \; \nabla [C_p \; (T - T_{ref})] - C_p \; (T - T_{ref}) \; \nabla \cdot v - \nabla \cdot q$$

$$+ C_p \; Q_{in} \; [T_{inj} - T_{ref}] - C_p \; \dot{Q}_{out} \; (T - T_{ref}) - C_p \; \dot{Q}_{out} \; (T_p - T)$$

$$= \frac{1}{\Delta z} \frac{\partial}{\partial t} [\Delta z \; (\rho_f \; C_f \; (1 - \theta) + \rho_w \; C_p \; \theta) \; (T - T_{ref})] \; .$$

From the continuity equation

$$- \nabla \cdot v + (Q_{in} - Q_{out}) = \frac{1}{\Delta z} \frac{\partial}{\partial t} (\Delta z \rho_w \theta) \; .$$

Substituting

$$- v \; \nabla [C_p \; (T - T_{ref})] + C_p \; (T - T_{ref}) \; (\frac{1}{\Delta z} \frac{\partial}{\partial t} (\Delta z \; \rho_w \; \theta)$$

$$- Q_{in} + \dot{Q}_{out}) - \nabla \cdot q$$

$$+ C_p \dot{Q}_{in} [T_{inj} - T_{ref}] - C_p Q_{out} [T - T_{ref}] - C_p \dot{Q}_{out} [T_p - T]$$

$$= \frac{1}{\Delta z} \frac{\partial}{\partial t} [\Delta z \, (\rho_f C_f (1 - \theta) + \rho_w C_p \theta) (T - T_{ref})] .$$

Expanding terms

$$- v \nabla[C_p (T - T_{ref})] + C_p (T - T_{ref}) [\frac{1}{\Delta z} \partial/\partial t \, (\Delta z \, \rho_w \theta)]$$

$$- C_p (T - T_{ref}) \dot{Q}_{in} + C_p (T - T_{ref}) \dot{Q}_{out} - \nabla \cdot q$$

$$+ C_p \dot{Q}_{in} [T_{inj} - T_{ref}] - C_p \dot{Q}_{out} (T - T_{ref}) - C_p \dot{Q}_{out} (T_p - T)$$

$$= \frac{1}{\Delta z} \Delta z \, \rho_f C_f (1 - \theta) \frac{\partial}{\partial t} (T - T_{ref}) + \frac{1}{\Delta z} (T - T_{ref})$$

$$x \frac{\partial}{\partial t} (\Delta z \, \rho_w \theta)]$$

$$+ \frac{1}{\Delta z} \Delta z \, \rho_w C_p \frac{\partial}{\partial t} (T - T_{ref}) + C_p \frac{1}{\Delta z} (T - T_{ref}) \frac{\partial}{\partial t} (\Delta z \, \rho_w \theta) ,$$

Cancelling terms

$$- v \nabla[C_p (T - T_{ref})] - \nabla \cdot q - C_p (T - T_{ref}) Q_{in}$$

$$+ C_p \dot{Q}_{in} [T_{inj} - T_{ref}]$$

$$- C_p \dot{Q}_{out} (T_p - T)$$

$$= \rho_f C_f (1 - \theta) \frac{\partial T}{\partial t} + \frac{1}{\Delta z} (T - T_{ref}) \rho_f C_f \frac{\partial}{\partial t} [\Delta z \, (1 - \theta)]$$

$$+ \rho_w C_p \theta \frac{\partial T}{\partial t} .$$

Again expanding, and then collecting temperature terms,

$$- v \nabla[C_p \ (T - T_{ref})] - \nabla \cdot q + C_p \ Q_{in} \ (T_{inj} - T)$$

$$+ C_p \ \dot{Q}_{out} - C_p \ \dot{Q}_{out} \ T_p$$

$$= [\rho_f \ C_f \ (1 - \theta) + \rho_w \ C_p \ \theta] \ \frac{\partial T}{\partial t} +$$

$$+ (T - T_{ref}) \ \rho_f \ C_f \ [(- \frac{\partial \theta}{\partial t}) + (1 - \theta) \ \frac{\partial \Delta z / \Delta z}{\partial t} \quad .$$

Once again

$$\frac{\partial \theta}{\partial t} = \alpha(1 - \theta) \ \frac{\partial P}{\partial t} \ \text{ and } \ \frac{\partial \Delta z}{\partial \Delta z} = \alpha dP \ .$$

Substituting and canceling vertical compressibility terms,

$$- v \nabla[C_p \ (T - T_{ref})] - \nabla \cdot q + \dot{Q}_{in} \ C_p \ (T_{inj} - T)$$

$$+ C_p \ \dot{Q}_{out} \ T - C_p \ \dot{Q}_{out} \ T_p$$

$$= [\rho_f \ C_f \ (1 - \theta) + \rho_w \ C_p \ \theta] \ \frac{\partial T}{\partial t} \quad .$$

Expanding and rearranging results in the final heat transport equation;

$$\frac{\partial}{\partial x} [K_x \ \frac{\partial T}{\partial x}] + \frac{\partial}{\partial y} [K_y \ \frac{\partial T}{\partial y}] + \frac{\partial}{\partial z} [K_z \ \frac{\partial T}{\partial z}] - C_p \ [v_x \ \frac{\partial T}{\partial x} + v_y \ \frac{\partial T}{\partial y} + v_z \ \frac{\partial T}{\partial z}]$$

$$+ C_p \ (\dot{Q}_{out} - \dot{Q}_{in}) \ T$$

$$= - C_p \ \dot{Q}_{in} \ T_{inj} + C_p \ \dot{Q}_{out} \ T_p + [\rho_f \ C_f \ (1 - \theta) + \rho_w \ C_p \ \theta] \ \frac{\partial T}{\partial t} \ .$$

In order to insure continuity between the convective term and the heat storage term, an artificial dissipation term (Φ) is added such that the right hand side of the above equation becomes

$$- C_p \dot{Q}_{in} T_{inj} + C_p \dot{Q}_{out} T_p + [\rho_f C_f (1 - \theta) + \rho_w C_p \theta] \frac{\partial T}{\partial t} + \Phi .$$

In this case, the term is a differential mass storage. To derive Φ, consider the mass transport equation in the form

$$- [\frac{\partial}{\partial x} (q_x) + \frac{\partial}{\partial y} (q_y) + \frac{\partial}{\partial z} (q_z)] + Q = \text{R.H.S.}$$

where

$$\text{R.H.S.} = \theta \rho_o \beta_T \frac{\partial T}{\partial t} - (\rho \alpha + \theta \rho_o \beta_p) \frac{\partial P}{\partial t} \quad \text{(mass storage term)}$$

This may be rewritten as

$$\text{R.H.S.} = - \text{div } (q) + Q$$

Integrating over the solution domain and dividing by total volume for a unit volume basis results in

$$\frac{1}{V} [\int_D - \text{div } (q) + \int_D Q],$$

a differential mass storage. Expressing the energy content and expanding results in

$$\Phi = \frac{1}{V} [C_p T(\int_D (- \frac{\partial v_x}{\partial x} - \frac{\partial v_y}{\partial y} - \frac{\partial v_z}{\partial z}) + \int_D Q]$$

Hence the final energy balance equation becomes

$$\frac{\partial}{\partial x} [K_x \frac{\partial T}{\partial x}] + \frac{\partial}{\partial y} [K_y \frac{\partial T}{\partial y}] + \frac{\partial}{\partial z} [K_z \frac{\partial T}{\partial z}] - C_p [v_x \frac{\partial T}{\partial x} + v_y \frac{\partial T}{\partial y} + v_z \frac{\partial T}{\partial z}]$$

$$+ C_p (\dot{Q}_{out} - \dot{Q}_{in}) T = C_p \dot{Q}_{out} T_p - C_p \dot{Q}_{in} T_{inj}$$

$$+[\rho_f C_f (1 - \theta) + \rho_w C_p \theta] \frac{\partial T}{\partial t} + \frac{1}{V} [C_p T (\int_D (- \frac{\partial v_x}{\partial x} \frac{\partial v_y}{\partial y} - \frac{\partial v_z}{\partial z}) + \int_D Q].$$

APPENDIX C

PROGRAM LISTINGS

PARTIAL LIST OF SYMBOLS USED IN THE COMPUTER CODES

A	System compressed stiffness matrix
ALPHA	Aquifer compressibility coefficient
AMATER	Alphanumeric description of each material
B	Constant vector of equations to be solved
BETAD	Derivative of basis function with respect to local coordinates
BETAP	Fluid compressibility coefficient
BETAT	Coefficient of thermal volume expansion
BNDI	Integrated boundary area
C	Element stiffness matrix
CI	Element stiffness matrix
CIV	Temperature or mass flux for a boundary node
CMAG	Boundary area vector product
CONVEC	Convection term, energy flow equation
COS x,y,z	Direction cosines
DELT	Variable time step
DELTAT	Time step
DETAIL	Alphanumeric problem description
DPDT	Derivative of fluid pressure with respect to time
DSTOR	Artificial dissipation term
DU	Jacobian of matrix
DU	Product of Jacobian and Gaussian weighting factor
ETA	Nodal values of Y in local coordinates
EZA	Nodal values of Z in local coordinates
FIFJ	Basis function cross derivative, x direction

FLUXX	Nodal flux values
GIGJ	Basis function cross derivative, y direction
HIHJ	Basis function cross derivative, z direction
IBANDW	Number of nonzero coefficient in a given matrix row
IIFLG	Boundary condition index
INDEX	Print ption controls
ITOTAL	Total number of time steps
ISIDE	Number of boundary surfaces for a given element
MCC	Order of each side of a given element
MIXNOD	Order of each mixed element corner
MMI	Material number of element
NBAND	Bandwidth of stiffnex matrix
NBPTC	Total number of boundary conditions on pressure or flux
NELEM	Total number of elements
NMAT	Number of materials
NN	Number of nodes in a given element
NNN	Maximum bandwidth of non-zero elements in stiffness matrices
NNSTIF	Maximum expected bandwidth
NOD	Node number of each element
NOCB	Node number of given boundary condition, differenciates between mass flux and heat flux
NODB	Node number of given boundary condition, differenciates between pressure and temperature
NODE	Nodes for which pressure is to be estiamted
NODEC	Nodes for which temperature is to be estimated

NOELEM	Numerical value for naming an element
NPPTC	Total number of boundary conditions on temperature or heat flux
NPT	Number of nodes for which pressure is to be estimated
NPTC	Number of nodes for which temperature is to be estimated
NSEC	Input control for distinguishing different data input
NSIDE	I.D. number of boundary sides for a given element
RHO	Fluid density
SCA	Nodal values of x in local coordinates
SPEHET	Fluid specific heat
TEMP	Temperature
TMASS1	Mass present in an element at the beginning of a time step
TMASS2	Mass present in an element at the end of a time step
TCX,Y,Z	Effective thermal conductivity in a given direction
TXK	Thermal conductivity of rock material
THRCON	Thermal conductivity of fluid
XIJCOB	Integrated basis function
XK, YK, ZK	Specific permeability in a given direction
XIXJOB	Integrated basis function product
XMEW	Fluid dynamic viscosity
XKK, YKK, ZKK	Estimated hydraulic conductivity in a given direction
ZTEST	Lowest cut off coefficient in equation solving subroutine

```
C-----------------------------------------------------------------------
C     THREE DIMENSIONAL FINITE ELEMENT SOLUTION OF PARTIAL DIFFERENTIAL
C     EQUATIONS FOR SIMULTANEOUS HEAT AND MASS TRANSPORT THROUGH POROUS
C     MEDIA.
C-----------------------------------------------------------------------
C                         PROGRAM FOR READING INPUT DATA
C-----------------------------------------------------------------------
      IMPLICIT REAL*8(A-H,O-Z)
C-----------------------------------------------------------------------
C     LIST OF SYMBOLS USED FOR ESTIMATING THE SIZE OF THE PROBLEM
C     NTT = EST. TOTAL NUMBER OF NODES
C     NPT             = NUMBER OF NODES FOR WHICH PRESSURE IS TO BE EST.
C     NPTC            = NUMBER OF NODES FOR WHICH TEMP. IS TO BE EST.
C     NELEM           = NUMBER OF ELEMENTS
C     MATN            = NUMBER OF MATERIALS
C    NBPTC            = NUMBER OF BOUNDARY POINTS HAVING PRESCRIBED BOUNDARY
C                       CONDITIONS FOR PRESSURE OR MASS FLUX.
C     NPPTC           = NUMBER OF BOUNDARY POINTS HAVING PRESCRIBED BOUNDARY
C                       CONDITIONS FOR TEMPERATURE OR HEAT FLUX.
C     NODMAX          = MAXIMUM NUMBER OF NODE POINTS IN ANY ELEMENT
C     NNSTIF          = MAXIMUM NUMBER OF NON-ZERO NODES IN ANY ELEMENT.
C                       I.E. THE MAXIMUM EXPECTED BANDWIDTH OF THE
C                       STIFFNESS MATRIX.
C-----------------------------------------------------------------------
C**** READ DATA CARD ****
      READ, NTT,NELEM,MATN,NBPTC,NPPTC,NODMAX,NNSTIF
      REWIND1
      CALL PREP(NTT,NELEM,MATN,NBPTC,NPPTC,NODMAX,NNSTIF)
      STOP
      END
C
      SUBROUTINE PREP(NNPT,NNELEM,NMATN,NNBPTC,NNPPTC,NODMAX,NNSTIF)
C-------------------------------------------------------------------
C
C     PARAMETERS ASSOCIATED WITH THE NUMBER OF NODE POINTS AND NUMBER OF
```

```
C     ELEMENTS.
C
C-----------------------------------------------------------------------
      IMPLICIT REAL*8(A-H.O-Z)
      DIMENSION X(125).Y(125).Z(125).NODB(100).BIV(100).TEMP(125).
     1NOCB(100).NUMBER(125).PRESS(125).NODE(125).NODEC(125).CIV(100)
C-----------------------------------------------------------------------
C     PARAMETERS ASSOCIATED WITH THE MATERIAL PRCPERTIES.
C-----------------------------------------------------------------------
      DIMENSION XK(2).YK(2).ZK(2).THETA(2).ALPHA(2).PZERO(2).TXK(2).
     1TYK(2).TZK(2).HCC(2).TITLE(18).ANI(4).ANJ(4).NOD(36).MATNO(36).
     2MIXNOD(8).XQW(2).YQW(2).ZQW(2).AMATER(18).ISIDE(5)
      DATA ANI/4HPRS=.4HQ-N=.4HTMP=.4HHFX=/
      DATA ANJ/3HONE.3HTWO.4HIII .4HMXEL/
C**** READ DATA CARD ****
C     STATEMENT OF PROBLEM TITLE
      READ(5.500) (TITLE(I).I=1.18)
  500 FORMAT(18A4)
      PRINT 600.(TITLE(I).I=1.18)
  600 FORMAT(1X.8X.18A4///)
      PRINT 670.NNPT.NNELEM.NMATN
      PRINT 680.NNBPTC.NNPPTC.NODMAX
C**** READ DATA CARD ****
      READ(5.503) INDEX
C     INDEX = 1; DATA CARDS ARE PRINTED AS READ
C           = 0; DATA CARDS ARE NOT PRINTED
      PRINT 622
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C     MATERIAL PRCPERTIES
      NMAT = 0
      PRINT 621
      GO TO 1
    2 PRINT 603.NMATN.NMAT
      IX = 1
C**** READ DATA CARD ****
```

```
    1 READ(5,502) NSEC,(AMATER(L),L=1,18),I,D1,D2,D3,TTO,PC,ALP,
     1DD1,DD2,DD3,HC
C      LIST OF SYMBOLS
C      NSEC               =READ CONTROL
C      AMATER             = MATERIAL DESCRIPTION
C      I                  = MATERIAL I.D. NUMBER
C      D1,D2,D3,          = INTRINSIC PERMEABILITIES IN X,Y,Z, DIRECTIONS
C      TTO                = PORCSITY MEASURED AT A REFERENCE DENSITY
C      PO                 = REFERENCE DENSITY FOR WHICH TTO IS DEFINED
C      ALP                = COMPRESSIBILITY COEFF. OF THE MEDIUM
C      DD1,DD2,DD3,       = THERMAL CONDUCTIVITIES OF MEDIUM IN X,Y,Z, DIR.
C      HC                 = HEAT CAPACITY OF MATERIAL
       IF(NSEC.NE.0) GO TO 3
       IF(I.GT.NMAT) NMAT=I
       IF(NMAT.GT.NMATN) GO TO 2
       XK(I)=D1
       YK(I)=D2
       ZK(I)=D3
       THETA(I)=TTO
       PZERO(I)=PO
       HCC(I)=HC
       TXK(I)=DD1
       TYK(I)=DD2
       TZK(I)=DD3
       ALPHA(I)=ALP
       PRINT 604,I,(AMATER(L),L=1,18),D1,D2,D3,TTO,ALP,PO,DD1,DD2,DD3,HC
       GO TO 1
C-----------------------------------------------------------------------
C      INPUT CONTROL UNIT
C-----------------------------------------------------------------------
    3  GO TO (4,5,6,7,8),NSEC
C-----------------------------------------------------------------------
C      READ NCDAL PCINT COORDINATES
C-----------------------------------------------------------------------
    4  NTT=0
       I=1
```

```
      IF(INDEX.EQ.1) PRINT 610
C
      GO TO 9
   10 PRINT 606.NNPT.NTT
      IX=2
C**** READ DATA CARD ****
    9 READ(5.506) NSEC.N.XP.YP.ZP.PINI.TINI
C     LIST OF SYMBOLS
C     NODAL NUMBER. ASSIGNED BY USER.
C    XP.YP.ZP          = X.Y.Z COORDINATES OF NOCAL POINT
C     PINI             = INITIAL PRESSURE AT NODAL PCINT
C     TINI             = INITIAL TEMPERATURE AT NODAL POINT
      IF(NSEC.NE.0) GO TO 3
      IF(N.LT.100) PRINT 622
      IF(INDEX.EQ.1)
     1PRINT 605.I.N.XP.YP.ZP.PINI.TINI
      IF(I.GT.NNPT) GO TO 10
      X(I)=XP
      Y(I)=YP
      Z(I)=ZP
      PRESS(I)=PINI
      TEMP(I)=TINI
      NUMBER(I)=N
      I = I+1
      GO TO 9
    5 PRINT 613
      IPOINT = I-1
      DO 110 N=1.IPOINT
      I = NUMBER(N)
      IF(I.LT.100) PRINT 622
      PRINT 611.I.X(N).Y(N).Z(N).PRESS(N).TEMP(N)
  110 CONTINUE
      PRINT 612
      N = 1
      INNNN = 0
      GO TO 11
   13 PRINT 608.NNELEM.N
       IX=3
C**** READ DATA CARD ****
   11 READ(5.504) NSEC.N1.NN.NQW.MM1.NSIDE.NOELEM
C      N1                 = ORDER OF GIVEN ELEMENT;
C                         = 0 MIXED
C                         = 1 LINEAR
C                         = 2 QUADRATIC
C                         = 3 CUBIC
C      NN =               NUMBER OF NODES IN GIVEN ELEMENT
C      NQW                = NUMBER OF SOURCES OR SINKS IN ELEMENT
C      MM1                = MATERIAL NUMBER OF ELEMENT
C      NSIDE              = NO. CF ELEMENT SIDES TO BE INTEGRATED
C      NOELEM             = ELEMENT NUMBER ASSIGNED BY USER.
       IF(NSEC.NE.0) GO TO 12
       READ.(NOD(I).I=1.NN)
C      NOD(I)             = DEFINED IN THE FOLLOWING MANNER;
C                         FIRST FOUR NODES ARE THE TOP CORNER NODES OF THE ELEMENT
C                         IN THE COUNTERCLOCKWISE DIRECTION
C                         NEXT FOUR NODES ARE THE BOTTOM CORNER NODES OF THE ELEMENT
C                         IN THE COUNTERCLOCKWISE DIRECTION.
C                         9 - 20 ARE THE MID NODES
C                         21 -44 ARE THE CUBIC NODES AT DIFFERENT LOCATIONS
       IF(MM1.EQ.0) READ.(MATNO(I).I=1.NN)
C      MATNO(I)           = MATERIAL NUMBER OF EACH NODE IF THE ELEMENT IS MADE UP OF
C                          MORE THAN ONE MATERIAL.
   14 IF(NQW.NE.0)
     1READ.(XQW(I).YQW(I).ZQW(I).I=1.NQW)
C      XQW.YQW.ZQW        = X.Y.Z COORDINATES OF EACH SOURCE OR SINK TERM
C**** READ DATA CARD ****
       IF(NSIDE.NE.0) READ.(ISIDE(I).I=1.NSIDE)
C      ISIDE              = I.D. NO. OF EACH SIDE OF 'NSIDE'
       M5=N1
       IF(N1.EQ.0) M5=4
       NE=NOELEM
       PRINT 607.NE.ANJ(M5).NN.NQW.MM1.(NOD(I).I=1.8)
```

```
      IF(N1.EQ.2) PRINT 689,(NOD(I),I=9,20)
      IF(N1.EQ.3) PRINT 688,(NOD(I),I=21,44)
      IF(N1.NE.0) GO TO 20
      PRINT 675,(NOD(I),I=9,NN)
      PRINT 622
C**** READ DATA CARD ****
      READ,(MIXNOD(I),I=1,8)
C     MIXNOD(I)        = THE ORDER OF EACH CORNER OF AN ELEMENT IF THE ELEMENT
C                        IS OF MIXED ORDER
      PRINT 638,(MIXNOD(I),I=1,8)
   20 IF(NQW.NE.0) PRINT 635,(I,XQW(I),YQW(I),ZQW(I),I=1,NQW)
      IF(MM1.EQ.0) PRINT 673,(MATNO(I),I=1,NN)
      IF(NSIDE.NE.0) PRINT636,(ISIDE(I),I=1,NSIDE)
C-----------------------------------------------------------------------
C     INTERNAL NODE NUMBERING IS DONE FOR FUTURE WORKS.
C-----------------------------------------------------------------------
      DO 111 I=1,NN
      NP=NOD(I)
      IF(NP.EQ.0) GO TO 111
      DO 112 J=1,IPOINT
      IF(NP.NE.NUMBER(J)) GO TO 112
      NOD(I)=J
      GO TO 111
  112 CONTINUE
      INNNN = 1
      PRINT,'ERROR - NO COORDINATES ARE GIVEN FOR NODE NO.',NOD(I),'IN
     1ELEMENT NO.',NOELEM
  111 CONTINUE
C-----------------------------------------------------------------------
C     INFORMATION REGARDING THE ELEMENT IS WRITTEN ON TAPE FILE 1
C-----------------------------------------------------------------------
      WRITE(1) N1,NN,(NOD(I),I=1,NN),MM1,NQW,NSIDE,NOELEM
      IF(MM1.EQ.0) WRITE(1) (MATNO(I),I=1,NN)
      IF(NQW.NE.0) WRITE(1) (XQW(I),YQW(I),ZQW(I),I=1,NQW)
      IF(NSIDE.NE.0) WRITE(1) (ISIDE(I),I=1,NSIDE)
      IF(N1.EQ.0) WRITE(1) (MIXNOD(I),I=1,8)
```

```
C-----------------------------------------------------------------------
      N = N+1
      IF(N.GT.NNELEM) GO TO 13
      IF(INNNN.EQ.1) GO TO 17
      MM = 0
C-----------------------------------------------------------------------
C     AREA OF THE ELEMENT IS CALCULATED AND PRINTED TO FIND THE ERROR IN
C     SEQUENTIAL NUMBERING OF THE TOP AND BOTTOM NODES.
C-----------------------------------------------------------------------
      DO 100 I=1.2
      J=I-1
      N1 = NCD(1+J*4)
      N2 = NOD(2+J*4)
      N3 = NOD(3+J*4)
      N4 = NOD(4+J*4)
      X1 = X(N1)
      X2 = X(N2)
      X3 = X(N3)
      X4 = X(N4)
      Y1 = Y(N1)
      Y2 = Y(N2)
      Y3 = Y(N3)
      Y4 = Y(N4)
      AA1=X1*(Y4-Y2)
      AA2=X2*(Y1-Y3)
      AA3=X3*(Y2-Y4)
      AA4=X4*(Y3-Y1)
      AREA=DABS(.5D0*(AA1+AA2+AA3+AA4))
      PRINT.'AREA ****** '.AREA.'SQ. METERS'
  100 CONTINUE
   17 INNNN = 0
      PRINT 672
      GO TO 11
   12 NELEM = N-1
C
C-----------------------------------------------------------------------
```

```
C     BOUNDARY CONCITIONS ON PRESSURE OR FLUID FLUX
C-----------------------------------------------------------------------
    6 WRITE(6,614)
      ENDFILE1
      REWIND1
      I = 1
C**** READ DATA CARD ****
  180 READ(5,505) NSEC,KK,IIFLG,BIVD
C     KK              = BCUNDARY NODE NUMBER
C     IIFLG           = 1 IF FRESSURE IS PRESCRIBED
C                     = 0 IF FLUX IS PRESCRIBED
C     BIVD            = VALUE OF HEAD OR FLUX
      IF(NSEC.NE.0) GO TO 3
      BIV(I)=BIVD
C-----------------------------------------------------------------------
C     BOUNDARY NODE NUMBER IS CONVERTED TO NEW NUMBER ALREADY GIVEN.
C-----------------------------------------------------------------------
  189 DO 113 IK=1,IPOINT
      KTT=NUMBER(IK)
      IF(KTT.NE.KK) GO TO 113
      KT = IK
      GO TO 114
  113 CONTINUE
      PRINT,'NOTE: IMPORTANT; NO COORDINATES ARE SPECIFIED FOR NODE ',KK
  114 NOCB(I)=KT*10+IIFLG
      AN=ANI(2)
      IF(IIFLG.EQ.0) GO TO 187
      AN=ANI(1)
  187 CONTINUE
      WRITE(6,615) KK,AN,BIV(I)
  190 I=I+1
      GO TO 180
    7 NBPTC = I-1
C-----------------------------------------------------------------------
C     BOUNDARY CONDITIONS CN TEMPERATURE OR HEAT FLUX
C-----------------------------------------------------------------------
```

```
      WRITE(6,616)
      I=1
C**** READ DATA CARD ****
  280 READ(5,505) NSEC,KK,IIFLG,BIVD
C     KK                = BOUNDARY NODE NUMBER
C     IIFLG             = 1 IF TEMPERATURE IS PRESCRIBED
C                       = 0 IF HEAT FLUX IS PRESCRIBEC
C     BIVD              = VALUE OF TEMPERATURE OR HEAT FLUX.
      IF(NSEC.NE.0) GO TO 3
      CIV(I)=BIVD
C-----------------------------------------------------------------------
C     BOUNDARY NODE NUMBER IS CONVERTED TO NEW NODE NUMBER ALREADY GIVEN
C-----------------------------------------------------------------------
  289 DO 116 IK=1,IPOINT
      KTT=NUMBER(IK)
      IF(KTT.NE.KK) GO TO 116
      KT = IK
      GO TO 117
  116 CONTINUE
      PRINT,'NOTE: IMPORTANT; NO COORDINATES ARE SPECIFIED FOR THE NOCE
     1',KK
  117 NOCB(I)=KT*10+IIFLG
      AN=ANI(4)
      IF(IIFLG.EQ.0) GO TO 287
      AN=ANI(3)
  287 CONTINUE
      WRITE(6,615) KK,AN,CIV(I)
  290 I=I+1
      GO TO 280
    8 NPPTC=I-1
      NTT=IPCINT
      NB=NBPTC
      NP=NPPTC
      MBAND = NNSTIF
C-----------------------------------------------------------------------
C     SUBROUTINE TC COUNT THE ACTUAL NUMBER OF UNKNOWNS
```

```
C------------------------------------------------------------------------
      CALL CCUNT
     1(NTT.NPT.NPTC.NB.NP.NODB.NODE.NODEC.NELEM.NOCB.NUMBER.IPOINT.
     2MBAND)
C------------------------------------------------------------------------
      PRINT.' FOR PRESSURE SOLUTION '
      CALL BANWDH(NPT.NODE.NELEM)
      PRINT.' FOR TEMPERATURE SOLUTION '
      CALL BANWDH(NPTC.NODEC.NELEM)
C     THE SIZE OF THE PROBLEM IS CHECKED TO SEE IF IT IS TCO LARGE ANC
C     DATA ERRORS ARE SOUGHT
C------------------------------------------------------------------------
  206 IF(NTT.LE.NNPT) GO TC 207
      WRITE(6.618)
      IX=4
  207 IF(NBPTC.LE.NNBPTC) GO TO 208
      WRITE(6.619)
      IX=4
  208 IF(NMAT.LE.NMATN) GO TO 210
      WRITE(6.620)
      IX=4
  210 CONTINUE
C------------------------------------------------------------------------
C     WRITE INPUT DATA ON TAPE FILE I
C------------------------------------------------------------------------
      WRITE(2) NTT.NELEM.NMAT.NBPTC.NFPTC.NCDMAX.NNSTIF.NPT.NPTC.
     1(TITLE(I).I=1.18)
      WRITE(2) (X(N).N=1.NTT)
      WRITE(2) (Y(N).N=1.NTT)
      WRITE(2) (Z(N).N=1.NTT)
      WRITE(2) IPCINT.(NUMBER(I).I=1.IPCINT)
      WRITE(2) (NCDB(N).N=1.NBPTC)
      WRITE(2) (BIV(N).N=1.NBPTC)
      WRITE(2) (NCCB(N).N=1.NFPTC)
      WRITE(2) (CIV(N).N=1.NPPTC)
      WRITE(2)  (XK(N).YK(N).ZK(N).N=1.NMAT).(THETA(N).ALPHA(N).
     1PZERO(N),HCC(N),N=1,NMAT),(TXK(N),TYK(N),TZK(N),N=1,NMAT)
      WRITE(2) NPT,(NODE(I),I=1,NPT)
      WRITE(2) NPTC,(NODEC(I),I=1,NPTC)
      WRITE(2) (PRESS(N),N=1,NTT)
      WRITE(2) (TEMP(N),N=1,NTT)
      ENDFILE2
      REWIND2
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
C     READ FORMATS
C-----------------------------------------------------------------------
C
  502 FORMAT(I1,18A4/,I5,5X,6E10.4/,4E10.3)
  503 FORMAT(I1)
  504 FORMAT(I1,I2,2X,4I3,59X,I4)
  505 FORMAT(I1,I4,I5,E12.4)
  506 FORMAT(I1,I4,2D10.4,D6.0,D9.3,D5.0)
C
C-----------------------------------------------------------------------
C     PRINT FORMATS
C-----------------------------------------------------------------------
  670 FORMAT(1X,/1X,100('*')//,'     DIMENSION IN THE PROGRAM HAS BEEN SE
     1T WITH THE FOLLOWING MAXIMUM VALUES '//,100('*')/,
     2/,'     TOTAL NUMBER CF NODES SPECIFIEC',I10//,
     3'     TOTAL NUMBER OF ELEMENTS ',I10//,
     4'     TOTAL NUMBER OF MATERIALS ',I10//)
  680 FORMAT(1X,'     TOTAL NUMBER CF NODES HAVING BOUNDARY CONDITIONS
     1SPECIFIEC FCR PRESSURE OR FLUID FLUX',I10//,1X,
     2'     TOTAL NUMBER OF NODES WITH BOUNDARY CONDITICNS SPECIFIED FCR
     3TEMPERATURE OR HEAT FLUX',I10//,1X,
     4'     MAXIMUM NUMBER OF NODES IN ONE ELEMENT',I10//,100('*'))
  622 FORMAT(1X,//)
  621 FORMAT(1X,/)
  603 FORMAT(1X,/1X,'ERROR IN NUMBER OF MATERIALS DECLARED AS ',I5,' WHI
     1LE READ AS ',I5)
```

```
604 FORMAT(1X,//5X,'PROPERTIES FOR MATERIAL ',I4,4X,'*********'//,5X,
   118A4//,10X,'K-X =',E11.4,/10X,'K-Y =',E11.4,/9X,'K-Z =',E11.4,
   2/,10X,'THEATAO=',E11.4,/10X,'ALP=',E11.4/,
   310X,'ROZERO=',E11.4,/10X,'TK-X=',E11.4,/10X,'TK-Y=',E11.4,/,10
   4X,'TK-Z=',E11.4,/,10X,'HC = ',E11.4)
610 FORMAT(10X,'*****GEOMETRY*****',//,1X,100('-')//,1X,'  NODAL POINT
   1S',8X,'X-',9X,'Y-',11X,'Z-',26X,'P-INITIAL',9X,'T-INITIAL',//1X,
   2100('-'),//)
606 FORMAT(1X,'ERROR, MAXIMUM # DESCRIBED AS ',I5,'WHILE IT WAS REAC
   1 AS',I5)
605 FORMAT(1X,2I6,5X,3E12.3,15X,2E17.4)
613 FORMAT(10X,'*****GEOMETRY*****',//1X,100('-'),//1X,'   NODAL POINT
   1S',5X,'X-',8X,'Y-',11X,'Z-',17X,'PINITIAL',7X,'TINITIAL',//1X,100
   2('-'),//)
611 FORMAT(1X,I9,3(2X,E10.3),8X,2E15.4)
612 FORMAT(1X,//////10X,'*****ELEMENT INFORMATION*****'///)
608 FORMAT(1X,'ERROR; NUMBER OF ELEMENTS SPECIFIED AS',I5,'  READ AS '
   1,I5/)
636 FORMAT(1X,' BOUNDARY SURFACES ',5I4)
607 FORMAT(1X,/1X,'ELEM. # ',I3,10X,' ORDER    ',A4,10X,'NODES',I4//1X
   1,'# OF SOURCES/SINKS  ',I10,'   MATERIAL #',I10,//1X,'CORNER NODES
   2',7X,8I8/)
675 FORMAT(1X,//1X,'MID QUADRATIC NODES',12I8//1X,' MID CUBIC NODES
   1',12I8,//20X,12I8)
638 FORMAT(1X,' MIXED ELEMENT CORNER ORDER',8I10)
688 FORMAT(1X,/1X,' CUBIC NODES',8X,12I8/20X,12I8/)
689 FORMAT(1X,/1X,' MID NODES',10X,12I8/)
635 FORMAT(1X,//1X,'  DETAILS REGARDING SOURCE/SINKS IN THE ELEMENT'/,
   11X,20('  S.',I3,' X= ',
   2E10.3,' Y= ',E12.3,' Z= ',E12.3/))
673 FORMAT(1X,///1X,' MATERIAL # FOR EACH NODE  ********* ',//5X,
   18I5,40X,'CORNER NODES',//,5X,12I5,20X,'MID-QUADRATIC NODE'//,1X,
   225X,12I5,'MID CUBIC NODES',//,5X,12I5)
672 FORMAT(1X,/1X,100('-')/)
615 FORMAT(1X,I7,6X,A4,E12.4)
616 FORMAT(1X,//////10X,'*****BOUNDARY CONDITION WITH RESPECT TO
     1TEMPERATURE*****',////,1X,'BOUNDARY NODE'/)
 614 FORMAT(1X,//////10X,'*****BOUNDARY CONDITICN WITH RESPECT TO
     1PRESSURE ANC FLUX*****',////1X,' BOUNCARY NODE',/)
 618 FORMAT(1X,//,1X,'ERRCR- TOO MANY NODE PCINTS')
 619 FORMAT(1X,//,1X,'ERROR-TOO MANY BCUNDARY CONCITICNS')
 620 FORMAT(1X,//,1X,'ERROR-TOO MANY MATERIALS')
     RETURN
     END
C
C-----------------------------------------------------------------------
C     THIS SUBROUTINE DETERMINES THE NODES WHICH HAVE PRESSURE AND
C     TEMPERATURE AS UNKNOWNS.
C-----------------------------------------------------------------------
      SUBROUTINE CCUNT
     1(NTT,NPT,NPTC,NB,NP,NODB,NODE,NODEC,NELEM,NCCB,NUMBER,IPCINT,
     2MBAND)
      DIMENSION NUMBER(125),NODB(100),NOD(36),NODE(125),NODEC(125),
     1NOCB(100)
      M = 0
      DO 700 II = 1,NTT
      DO 750 J = 1,NB
      KK = NODB(J)
      IF(KK/10.NE.II) GO TO 750
      IFLAG = MOD(KK,10)
      IF(IFLAG.EQ.1) GO TO 700
      GO TO 760
 750 CONTINUE
 760 M = M+1
      NODE(M) = II
 700 CONTINUE
      NPT = M
      M = 0
      DO 770 II = 1,NTT
      DO 780 J = 1,NP
      KK = NOCB(J)
      IF(KK/10.NE.II) GO TO 780
      IFLAG = MOD(KK,10)
      IF(IFLAG.EQ.1) GO TO 770
      GO TO 790
  780 CONTINUE
  790 M = M+1
      NODEC(M) = II
  770 CONTINUE
      NPTC = M
C
C-----------------------------------------------------------------------
C     NPT = NUMBEF OF POINTS HAVING UNDETERMINEC COEFFICIENTS OF PRESS.
C     NPTC= NUMBER OF POINTS HAVING UNDETERMINED COEFFICIENTS OF TEMP..
C     AND NTT IS NUMBER OF TOTAL NODES IN THE FRCBLEM
C-----------------------------------------------------------------------
C
      PRINT 620
  620 FORMAT(1X,///1X,'    THIS IS THE INFORMATICN WITH REGARDS TO NUMEER
     1 OF UNKNCWNS  ',//1X,'     NPT = NUMEER CF NCDES FCR WHICH
     2 PRESSURE IS UNKNOWN AND',//1X,'         NPTC = NUMBER CF NODES FOR
     3 WHICH TEMPERATURE IS TO BE ESTIMATED '//)
      PRINT 622
    9 PRINT,'NPT AS ESTIMATED ',NPT
      PRINT 622
      PRINT,'  NUMBER OF UNKNOWN AS ESTIMATED ARE = '
      PRINT 622
      PRINT,(NUMBER(NODE(I)),I=1,NPT)
      PRINT 622
      PRINT,'   NUMBER OF NODES FOR WHICH TEMPERATURE IS TC BE DETERMINE
     1D = ',NPTC
      PRINT 622
      PRINT,(NUMBER(NODEC(I)),I=1,NPTC)
      PRINT 622
  622 FORMAT(1X,//)
C-----------------------------------------------------------------------
      RETURN
      END
```

```
C-----------------------------------------------------------------------
C
C-----------------------------------------------------------------------
C-------------------------------------------------------------------
      SUBROUTINE BANWDH(NPT,NODE,NELEM)
C-------------------------------------------------------------------
C     THIS SUBROUTINE ESTIMATES THE BANDWIDTH, ALCNG WITH THE UPPER
C     AND LOWER LIMITS OF THE DIAGONALS OF A SPARSE SYSTEM MATRIX.
C     THE MAXIMUM BANDWIDTH FOR SUBROUTINE EQSOLV (NNN) IS ALSO ESTIMATED
C-------------------------------------------------------------------
      DIMENSION NCD(36),NODE(125),NA(125,125),NAA(125)
      DO 300 I=1,NPT
      NAA(I)=0.0
      DO 300 J=1,NPT
  300 NA(I,J)=0.0
      DO 100 II=1,NELEM
      READ(1)  N1,NN,(NOD(I),I=1,NN),MM1,NOW,NSIDE,NCELEM
      IF(MM1.EC.0) READ(1)
      IF(NOW.NE.0) READ(1)
      IF(NSIDE.NE.0) READ(1)
      IF(N1.EQ.0) READ(1)
      DO 105 I=1,NN
      L=NOD(I)
      IF(L.EC.0) GO TO 105
      DO 106 J=1,NPT
      IF(L.NE.NODE(J)) GO TO 106
      KK=J
      GO TO 10
  106 CONTINUE
      GO TO 105
   10 DO 107 J=1,NN
      K=NOD(J)
      IF(K.EC.0) GO TO 107
      DO 108 JJ=1,NPT
      IF(NODE(JJ).NE.K) GO TO 108
      LL=JJ
      GO TO 12
  108 CONTINUE
      GO TO 107
   12 NA(KK,LL)=LL
  107 CONTINUE
  105 CONTINUE
  100 CONTINUE
      REWIND1
      NBAND=0
      MLD=0
      MUD=0
      NNN=0
      DO 110 I=1,NPT
      JJ=0
      DO 120 J=1,NPT
      IF(NA(I,J).EQ.0) GO TO 120
      JJ = JJ + 1
      NAA(JJ)=NA(I,J)
  120 CONTINUE
      IF(NNN.LT.JJ) NNN=JJ
      IF(JJ.EQ.0) GO TO 110
      NNBAND=NAA(JJ)-NAA(1)+1
      MLDD=I-NAA(1)
      MUDD=NAA(JJ)-I
      IF(NBAND.LT.NNBAND) NBAND=NNBAND
      IF(MUD.LT.MLDD) MUD=MUDD
      IF(MLD.LT.MLDD) MLD=MLDD
  110 CONTINUE
      WRITE(6,800) NBAND,MLD,MUD
  800 FORMAT(1X,/1X,100('*'),/1X,' THE BANDWIDTH IS ',I8,/5X,'MAXIMUM
     1 NUMBER OF LCWER DIAGONALS IS ',I8,/5X,' MAXIMUM NUMEER CF UPPEF
     2 DIAGONALS IS ',I8,/1X,100('*'))
      WRITE(6,801) NNN
  801 FORMAT(2X/2X,'****** MAXIMUM BANDWICTH FCR EQSOLV ',I8,' ******')
      RETURN
      END
```

```
C-----------------------------------------------------------------------
C     SUBPROGRAM FOR ESTIMATING THE GAUSSIAN INTEGERATICN FARAMETERS.
C-----------------------------------------------------------------------
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(125),Y(125),Z(125),PRESS(125),TEMP(125),NUMBER(125),
     1XK(2),YK(2),ZK(2),XQ(30),YQ(30),ZQ(30),F(30),G(30),H(30),XN(30),
     2AA1(30),AA2(30),AA3(30)
      DIMENSION FIFJ(256),GIGJ(256),HIHJ(256),XIXJOB(256),XIJCOB(30),
     1XQX(256),YQY(256),ZQZ(256),NODE(30),MCC(3),SCA(44),ETA(44),EZA(44)
     2,GQWT(3),GQPO(3),DD(3),SS(3),BETAD(3),AA(3),MIXNCD(8),NOD(36)
      DIMENSION XQW(2),YQW(2),ZQW(2),WQW(2,20),ISIDE(5)
      DIMENSION ISC(4),BNDI(8),NBA(8)
C
      REWIND1
      READ(2) NTT,NELEM,NMAT,NBPTC,NPPTC,NNSTIF,NPT,NPTC,TITLE
C
C-----------------------------------------------------------------------
C     THE VALUE OF THE ABOVE PARAMETERS HAS BEEN SET FCR A MAXIMUM OF
C     22 NON-ZERO NODES IN ANY ELEMENT.
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C     GAUSSIAN CUADRATURE INFCRMATION
C-----------------------------------------------------------------------
      DATA (SCA(I),I=1,20)/
     1-1.D0, 1.D0, 1.D0,-1.D0,-1.D0, 1.D0, 1.D0,-1.D0, 0.D0, 0.D0,
     2 0.D0, 0.D0,-1.D0, 1.D0,-1.D0, 1.D0,-1.D0, 1.D0, 1.D0,-1.D0/
      DATA(SCA(I),I=21,28)/
     1-.333333333333333, .333333333333333,-.333333333333333,
     2 .333333333333333,-.333333333333333, .333333333333333,
     3-.333333333333333, .333333333333333/
      DATA (SCA(I),I=29,44)/
     1-1.D0,-1.D0, 1.D0, 1.D0,-1.D0,-1.D0, 1.D0, 1.D0,-1.D0,-1.D0,
     2 1.D0, 1.D0, 1.D0, 1.D0,-1.D0,-1.D0/
C-----------------------------------------------------------------------
      DATA (ETA(I),I=1,28)/
     1-1.D0,-1.D0, 1.D0, 1.D0,-1.D0,-1.D0, 1.D0, 1.D0,-1.D0, 1.D0,
     2-1.D0, 1.D0, 0.D0, 0.D0, 0.D0, 0.D0,-1.C0,-1.D0, 1.D0, 1.D0,
     3-1.D0,-1.D0, 1.D0, 1.D0,-1.D0,-1.D0, 1.D0, 1.D0/
      DATA (ETA(I),I=29,36)/
     1-.333333333333333, .333333333333333,-.333333333333333,
     2 .333333333333333,-.333333333333333, .333333333333333,
     3-.333333333333333, .333333333333333/
      DATA (ETA(I),I=37,44)/
     1-1.D0,-1.D0,-1.D0,-1.D0, 1.D0, 1.D0, 1.D0, 1.D0/
C-----------------------------------------------------------------------
      DATA (EZA(I),I=1,36)/
     1 1.D0, 1.D0, 1.D0, 1.D0,-1.D0,-1.D0,-1.D0,-1.D0, 1.D0, 1.D0,
     2-1.D0,-1.D0, 1.D0, 1.D0,-1.D0,-1.D0, 0.D0, 0.D0, 0.D0, 0.D0,
     3 1.D0, 1.D0, 1.D0, 1.D0,-1.D0,-1.D0,-1.D0,-1.D0, 1.D0, 1.D0,
     4 1.D0, 1.D0,-1.D0,-1.D0,-1.D0,-1.D0/
      DATA (EZA(I),I=37,44)/
     1 .333333333333333,-.333333333333333, .333333333333333,
     2-.333333333333333, .333333333333333,-.333333333333333,
     3 .333333333333333,-.333333333333333/
C-----------------------------------------------------------------------
C
      READ(2) (X(N),N=1,NTT)
      READ(2) (Y(N),N=1,NTT)
      READ(2) (Z(N),N=1,NTT)
C
      REWIND2
      THIRD = 1.D0/3.D0
      THIRD2= 2.D0/3.D0
      R1924 = 19.D0/24.D0
      R964 = 9.D0/64.D0
C-----------------------------------------------------------------------
C     TRANSFER ELEMENT DATA TO TEMPORARY CISK STORAGE
C-----------------------------------------------------------------------
      DO 354 IX=1,NELEM
      READ(1) N1,NN,(NOD(I),I=1,NN),MM1,NQW,NSIDE,NE
      WRITE(3) N1,NN,(NOD(I),I=1,NN),MM1,NQW,NSIDE,NE
      IF(NSIDE.NE.0) READ(1) (ISIDE(I),I=1,NSIDE)
      IF(NSIDE.NE.0) WRITE(3) (ISIDE(I),I=1,NSIDE)
      IF(NQW.NE.0) READ(1)
     1(XQW(I),YQW(I),ZQW(I),I=1,NQW)
      IF(NQW.NE.0) WRITE(3)
     1(XQW(I),YQW(I),ZQW(I),I=1,NQW)
      IF(N1.NE.0) GO TO 354
      READ(1) (MIXNOD(I),I=1,8)
      WRITE(3) (MIXNOD(I),I=1,8)
  354 CONTINUE
      REWIND1
      ENDFILE3
      REWIND3
C-----------------------------------------------------------------------
C     GALSSIAN INTEGRATION PARAMETERS ARE ESTIMATED ELEMENT BY ELEMEKT.
C-----------------------------------------------------------------------
      DO 355 IX=1,NELEM
C
      WRITE(6,406)
  406 FORMAT(1X,///50X,'NEW ELEMENT',///)
      READ(3) N1,NN,(NOD(I),I=1,NN),MM1,NQW,NSIDE,NE
C
      IF(NQW.NE.0) READ(3)
     1(XQW(I),YQW(I),ZQW(I),I=1,NCW)
      NODMAX = NN
      IF(NSIDE.NE.0) READ(3) (ISIDE(I),I=1,NSICE)
      IF(N1.NE.0) GO TO 20
      READ(3) (MIXNOD(I),I=1,8)
      WRITE(6,630) (MIXNOD(I),I=1,8)
  630 FORMAT(10X,8I6)
   20 XX=0.D0
      YY=0.D0
      ZZ=0.D0
C-----------------------------------------------------------------------
C     FOR ELEMENTS OF ORDER 1 OR 2 TWC GAUSSIAN POINTS ARE USEC WHILE
C     FOR ELEMENTS OF ORDER 3 OR MIXED, 3 GAUSSIAN POINTS ARE USED.
```

```
C-----------------------------------------------------------------------
      NO = 2
      IF(NN.GT.20) NO=3
      MODEL = 3
      WRITE(6.701) NE.NO.MODEL.NN
  701 FORMAT(5X.'ELEM. # '.I4.' GAUSSIAN POINTS USED = '.I2.' MCDEL'.I4.
     1' MAX. NCDES '.I4)
      GQPO(1) = 0.D0
      GQPO(2) = 0.774596669241483
      GQPO(3) = -GOPO(2)
      GQWT(1) = 0.888888888888889
      GQWT(2) = 0.555555555555556
      GQWT(3) = GCWT(2)
      IF(NO-3) 92.95.95
   92 GQWT(1) = 1.D0
      GQWT(2) = 1.D0
      GQPO(1) = 0.577350269189626
      GQPO(2) = -0.577350269189626
   95 CONTINUE
C-----------------------------------------------------------------------
C     THE COORDINATES OF EACH NODE IN THE CURRENT ELEMENT ARE FOUND.
C-----------------------------------------------------------------------
      JJ =0
      DO 100 J=1.NN
      K = NOD(J)
      IF(K.EQ.0) GO TO 100
      ADD=0.0
      IF(J.EC.1) GO TO 21
      L = J-1
C-----------------------------------------------------------------------
C     ALLCW A VERTICAL DISTANCE OF .25 WHERE UPPER AND LOWER NODES ARE
C     THE SAME. I.E. BOTH MERGE AT ONE NCDE.
C-----------------------------------------------------------------------
      DO 103 I=1.L
      KK = NOD(I)
      IF(KK.EQ.0) GO TO 103
      IF(KK.NE.K) GO TO 103
      ADD=-0.25
      GO TO 21
  103 CONTINUE
   21 JJ = JJ+1
      XQ(J)=X(K)
      YQ(J)=Y(K)
      ZQ(J)=Z(K)+ADD
      NODE(JJ) = K
  100 CONTINUE
C-----------------------------------------------------------------------
C     THIS 'NODE' IS THE NEW SEQUENTIAL ORDER CF NCN-ZERO NCDES USEC IN
C     ALL FURTHER CALCULATIONS. AFTER THE GAUSSIAN PARAMETERS ARE
C     CALCULATED, THERE IS NO NEED TO KEEP TRACK CF THE LOCATICN OF
C     INDIVIDUAL NODES BECAUSE ALL DATA IS RECCRDED ON SCRATCH DISK SFACE
C     IN THE SAME SEQUENTIAL CRDER AS THE NODES ARE RECORDED.
C-----------------------------------------------------------------------
      NNN = JJ
      WRITE(4) NNN,(NODE(J),J=1,NNN),MM1,NOW,NE
      WRITE(6,400) NNN,(NODE(J),J=1,NNN),MM1,NCW,NE
  400 FORMAT(1X,27I4)
C     IF(MM1.EC.0) WRITE(4) (MNO(I),I=1,NNN)
      DO 60 I=1,NNN
   60 XIJCOB(I)=0.D0
      NXX = NNN*NNN
      DO 62 I=1,NXX
      FIFJ(I)=0.D0
      GIGJ(I)=0.D0
      HIHJ(I)=0.D0
      XIXJOB(I)=0.D0
      XQX(I)=0.D0
      YQY(I)=0.D0
      ZQZ(I)=0.D0
   62 CONTINLE
C-----------------------------------------------------------------------
C     INTEGRATION BY GAUSSIAN QUADRATURE
```

```
C-----------------------------------------------------------------------
      DO 49 I=1,3
   49 AA(I)=1.D0
      DO 240 L=1,NO
      EZ=GQPO(L)
      WW=GQWT(L)
      DO 240 N=1,NO
      SC=GQPO(N)
      WSC=GQWT(N)*WW
      DO 240 M=1,NC
      ET=GQPO(M)
      W=WSC*GQWT(M)
      D11=0.D0
      D12=0.C0
      D13=0.C0
      D21=0.D0
      D22=0.D0
      D23=0.D0
      D31=0.D0
      D32=0.D0
      D33=0.D0
      SS(1)=SC
      SS(2)=ET
      SS(3)=EZ
      DO 101 I=1,NN
      IF(NOD(I).EQ.0) GO TO 101
      DD(1)=SCA(I)
      DD(2)=ETA(I)
      DD(3)=EZA(I)
      DO 102 LT=1,3
  102 AA(LT)=1.D0+DD(LT)*SS(LT)
      IF(I.GT.8) GO TO 28
C-----------------------------------------------------------------------
C     BASIS FUNCTIONS FOR CORNER NODES
C-----------------------------------------------------------------------
      ALP=.125D0
      BETA=0.D0
      ALPHA=ALP*AA(1)*AA(2)*AA(3)
      MMM=N1
      IF(N1.NE.0) GO TO 1
      MCODE=MIXNOD(I)
      MCC(1)=MCODE/100
      MCC(2)=MCODE/10-MCC(1)*10
      MCC(3)=MCODE-MCC(1)*100-MCC(2)*10
    1 DO 6 LT=1.MCDEL
      D1=DD(LT)
      S1=SS(LT)
      IF(N1.NE.0) GO TO 2
      MMM=MCC(LT)
    2 GO TO (3.4.5).MMM
    3 BETA=BETA+THIRD
      BETAD(LT)=0.D0
      GO TO 6
    4 BETA=BETA+D1*S1-THIRD2
      BETAD(LT)=D1
      GO TO 6
    5 BETA=BETA+1.125D0*S1*S1-R1924
      BETAD(LT)=2.25D0*S1
    6 CONTINLE
      XN(I)=ALPHA*BETA
      DVSC=ALP*DD(1)*AA(2)*AA(3)*BETA+ALPHA*BETAD(1)
      DVET=ALP*DD(2)*AA(1)*AA(3)*BETA+ALPHA*BETAD(2)
      DVEZ=ALP*DD(3)*AA(1)*AA(2)*BETA+ALPHA*BETAD(3)
      GO TO 39
   28 IF(I.GT.20) GO TO 55
C-----------------------------------------------------------------------
C     MID-POINT QUADRATIC NODES
C-----------------------------------------------------------------------
      IF(I.GT.12) GO TO 56
      AAA=(1.D0-SC*SC)*.25D0
      XN(I)=AAA*AA(2)*AA(3)
      DVSC=-.5D0*SC*AA(2)*AA(3)
      DVET=DD(2)*AAA*AA(3)
      DVEZ=DC(3)*AAA*AA(2)
      GO TO 39
   56 IF(I.GT.16) GO TO 57
      AAA=(1.D0-ET*ET)*.25D0
      XN(I)=AA(1)*AAA*AA(3)
      DVSC=DC(1)*AAA*AA(3)
      DVET=AA(1)*(-.5D0*ET)*AA(3)
      DVEZ=AA(1)*AAA*DD(3)
      GO TO 39
   57 AAA=(1.D0-EZ*EZ)*.25D0
      XN(I)=AA(1)*AA(2)*AAA
      DVSC=DD(1)*AA(2)*AAA
      DVET=AA(1)*DD(2)*AAA
      DVEZ=AA(1)*AA(2)*(-.5D0*EZ)
      GO TO 39
C-----------------------------------------------------------------------
C     THIRD POINT CUBIC SIDE NODES
C-----------------------------------------------------------------------
   55 IF(I.GT.36) GO TO 58
      IF(I.GT.28) GO TO 59
      AAA=R964*(1.D0-SC*SC)*(1.D0+9.D0*DD(1)*SC)
      XN(I)=AAA*AA(2)*AA(3)
      DVSC=R964*(-2.D0*SC+9.D0*DD(1)*(1.D0-3.D0*SC*SC))*AA(2)*AA(3)
      DVET=AAA*DD(2)*AA(3)
      DVEZ=AAA*AA(2)*DD(3)
      GO TO 39
   59 AAA=R964*(1.D0-ET*ET)*(1.D0+9.D0*DD(2)*ET)
      XN(I)=AA(1)*AAA*AA(3)
      DVSC=DD(1)*AAA*AA(3)
      DVET=AA(1)*R964*(-2.D0*ET+9.D0*DD(2)*(1.C0-3.D0*ET*ET))*AA(3)
      DVEZ=AA(1)*AAA*DD(3)
      GO TO 39
   58 AAA=R964*(1.D0-EZ*EZ)*(1.D0+.9D0*DD(3)*EZ)
      XN(I)=AA(1)*AA(2)*AAA
      DVSC=DD(1)*AA(2)*AAA
      DVET=AA(1)*DD(2)*AAA
      DVEZ=AA(1)*AA(2)*R964*(-2.D0*EZ+9.D0*DD(3)*(1.D0-3.D0*EZ*EZ))
C
   39 AA1(I)=DVSC
      AA2(I)=DVET
      AA3(I)=DVEZ
      XQQ=XQ(I)
      YQQ=YQ(I)
      ZQQ=ZQ(I)
C
C-----------------------------------------------------------------------
C     D11 = CX/CSC ; D12 = DY/DSC ; D13 = DZ/CSC
C     D21 = CX/DET ; D22 = DY/DET ; D23 = DZ/DET
C     D31 = CX/DEZ ; D32 = DY/DEZ ; D33 = DZ/DEZ
C-----------------------------------------------------------------------
C
      D11=DVSC*XQC+D11
      D21=DVET*XQQ+D21
      D31=DVEZ*XQQ+D31
      D12=DVSC*YQC+D12
      D22=DVET*YQC+D22
      D32=DVEZ*YQC+D32
      D13=DVSC*ZQC+D13
      D23=DVET*ZQC+D23
      D33=DVEZ*ZQQ+D33
  101 CONTINUE
C
C
C
      DU=D11*D22*C33+D12*D23*D31+D13*C21*D32-D13*D22*D31-D21*D12*D33
     1-D11*D32*D23
      IF(DU.EQ.0.C0) GO TO 15
      DU=1.D0/CU
C-----------------------------------------------------------------------
C     THIS VALUE CF DU IS THE INVERSE OF THE CETERMINANT CF THE
C     JACOBIAN MATRIX
```

```
C-----------------------------------------------------------------------
      GO TO 16
   15 WRITE (6,602) NE,SC,ET,EZ,(NOD(I),ZQ(I),YQ(I),ZQ(I),I=1,NN)
  602 FORMAT(1X,'  DU=0.0 FOR ELEM # ',I5,' SC=',F10.5,' ET=',F10.5,
     1' EZ=',F10.5,/1X,32(' NODE',I5,' X = ',E13.6,' Y = ',E13.6,
     2' Z = ',E13.6/))
      GO TO 240
C-----------------------------------------------------------------------
C     F(I) IS THE DERIVATIVE OF THE SHAPE FACTOR 'NI' WITH RESPECT TC
C     'X', THAT IS DNI/DX. SIMILARLY, G(I) = DNI/DY WHILE H(I) = DNI/CZ.
C-----------------------------------------------------------------------
   16 DO 150 I=1,NN
      IF(NOD(I).EQ.0) GO TO 150
      F(I)=DU*(AA1(I)*D22*D33+AA3(I)*D12*D23+AA2(I)*D32*D13-AA3(I)
     1*D22*D13-AA2(I)*D33*D12-AA1(I)*D32*D23)
      G(I)=DU*(AA2(I)*D11*D33+AA1(I)*D23*D31+AA3(I)*D21*D13-AA2(I)
     1*D31*D13-AA1(I)*D21*D33-AA3(I)*D11*D23)
      H(I)=DU*(AA3(I)*D11*D22+AA2(I)*D12*D31+AA1(I)*D21*D32-AA1(I)
     1*D22*D31-AA3(I)*D12*D21-AA2(I)*D11*D32)
  150 CONTINUE
C
      XJCOB=1.D0/DU
      DU=XJCOB*W
C-----------------------------------------------------------------------
C     THIS VALUE CF DU IS THE JACOBIAN*GAUSSIAN WEIGHT
C-----------------------------------------------------------------------
C
      WRITE(6,618) DU,XJCOB
  618 FORMAT(1X,' DU= ',E13.6,' XJCOB= ',E13.6)
      II=0
      DO 220 I=1,NN
      IF(NOD(I).EC.0) GO TO 220
      II = II + 1
      FI=F(I)*DU
      GI=G(I)*DU
      HI=H(I)*DU
      XNI=XN(I)
      XNIDU=XNI*DU
      XNI=XNI*W
      JJ = 0
      DO 210 J=1,NN
      IF(NOD(J).EC.0) GC TO 210
      JJ = JJ + 1
      FJ=F(J)
      GJ=G(J)
      HJ=H(J)
      XNJ=XN(J)
C-----------------------------------------------------------------------
C     THE GAUSSIAN PARAMETERS USED IN INTEGRATICN ARE  INTEGRATED BY
C     SUMMING THE VALUES*JACOBIAN AT ALL GAUSSIAN POINTS.
C-----------------------------------------------------------------------
C
      XIXJ=XNJ*XNIDU
      IIJJ=(II-1)*NNN + JJ
      XIXJOB(IIJJ) = XIXJOB(IIJJ)+XIXJ
      FIFJ(IIJJ)=FIFJ(IIJJ)+FI*FJ
      GIGJ(IIJJ)=GIGJ(IIJJ)+GI*GJ
      HIHJ(IIJJ)=HIHJ(IIJJ)+HI*HJ
C
      XQX(IIJJ)=XCX(IIJJ)+XNIDU*FJ
      YQY(IIJJ)=YCY(IIJJ)+XNIDU*GJ
      ZQZ(IIJJ)=ZCZ(IIJJ)+XNIDU*HJ
  210 CONTINUE
      XIJCOB(II)=XIJCOB(II)+XNIDU
      IF(L.NE.1) GC TO 220
      IF(N.NE.1) GO TO 220
      IF(M.NE.1) GC TO 220
      XNI=XN(I)
      XX=XX+XNI*XC(I)
      YY=YY+XNI*YC(I)
      ZZ=ZZ+XNI*ZC(I)
  220 CONTINUE
  240 CONTINUE
C
      IF(NQW.EQ.0) GO TO 22
C-----------------------------------------------------------------------
C     THE WEIGHT TO BE GIVEN FOR EACH SOURCE OR SINK TERM TO EACH
C     NODE OF THE ELEMENT IS ESTIMATED
C-----------------------------------------------------------------------
      DO 906 I=1,NQW
      YYQW=YQW(I)
      XXQW=XQW(I)
      ZZQW=ZQW(I)
      J = 0
      WQQ=0.0
      IF(I.GT.1) GO TO 23
C-----------------------------------------------------------------------
C     FOR CASES WHERE WITHDRAWL OR INJECTION IS SPECIFIED AT THE CENTER
C     OF THE ELEMENT WITHOUT GIVING THE COORDINATES OF THE CENTER, XX,YY
C     AND ZZ ARE THE ESTIMATED COORDINATES OF THE CENTER OF THE ELEMENT.
C-----------------------------------------------------------------------
      IF(XXQW.EQ.0.0) XXQW=XX
      IF(YYQW.EQ.0.0) YYQW=YY
      IF(ZZQW.EQ.0.0) ZZQW=ZZ
   23 DO 106 JJ=1,NN
      IF(NOD(JJ).EQ.0) GO TO 106
      J = J + 1
      WQW(I,J)=1.D0/DSQRT((XXQW-XQ(JJ))**2+(YYQW-YQ(JJ))**2+(ZZQW-ZQ(JJ
     1))**2)
      WQQ=WQQ+WQW(I,J)
  106 CONTINUE
      DO 906 JJ=1,J
  906 WQW(I,JJ)=WQW(I,JJ)/WQQ
      NNN=J
      WRITE(4) ((WQW(I,J),J=1,NNN),I=1,NQW)
      WRITE(6,405) ((WQW(I,J),J=1,NNN),I=1,NQW)
  405 FORMAT(1X,8E15.6)
   22 CONTINUE
```

```
      WRITE(6,703)
  703 FORMAT(10X,' INTEGRATION OF THE ELEMENT HAS BEEN DONE ')
C-----------------------------------------------------------------------
C     ONCE THE ABOVE PARAMETERS ARE ESTIMATED AND STORED FOR EACH
C     ELEMENT, THE SOLUTION OF ANY DIFFERENTIAL EQUATION CAN BE OBTAINED
C     BY MULTIPLYING THESE PARAMETERS WITH THEIR RESPECTIVE CONSTANTS
C     TO FORM THE STIFFNESS MATRICES.
C-----------------------------------------------------------------------
C----------------------------------------------------------------------
      WRITE(4) (XIJCOB(I),I=1,NNN)
      IJ = 1
      IJJ = 0
      DO 104 I=1,NNN
      IJJ = IJJ + NNN
      WRITE(4) (FIFJ(II),GIGJ(II),HIHJ(II),II=IJ,IJJ)
      WRITE(4) (XIXJOB(II),II=IJ,IJJ)
      WRITE(4) (XGX(II),YQY(II),ZQZ(II),II=IJ,IJJ)
      IJ = IJ + NNN
  104 CONTINUE
C----------------------------------------------------------------------
C     GO TO 7
      WRITE(8) NE,NSIDE
      IF (NSIDE.EQ.0) GO TO 7
      WRITE(8) (ISIDE(I),I=1,NSIDE)
      DO 705 I=1,NSIDE
      IB=ISIDE(I)
      GO TO (71,72,73,74,75,76),IB
   71 ISC(1)=1
      ISC(2)=2
      ISC(3)=3
      ISC(4)=4
      DSF=1.D0
      GO TO 77
   72 ISC(1)=5
      ISC(2)=6
      ISC(3)=7
      ISC(4)=8
      DSF=-1.D0
      GO TO 77
   73 ISC(1)=3
      ISC(2)=4
      ISC(3)=7
      ISC(4)=8
      DSF=1.C0
      GO TO 77
   74 ISC(1)=1
      ISC(2)=2
      ISC(3)=5
      ISC(4)=6
      DSF=-1.D0
      GO TO 77
   75 ISC(1)=2
      ISC(2)=3
      ISC(3)=6
      ISC(4)=7
      DSF=1.D0
      GO TO 77
   76 ISC(1)=1
      ISC(2)=4
      ISC(3)=5
      ISC(4)=8
      DSF=-1.D0
   77 CONTINUE
      DO 78 J=1,8
      BNDI(J)=0.D0
   78 NBA(J)=0
      CC1=0.D0
      CC2=0.D0
      CC3=0.D0
      DO 706 J=1,4
      K=ISC(J)
  706 NBA(K)=NOD(K)
```

```
      DO 708 L=1,NO
      A=GQPO(L)
      WW=GQWT(L)
      DO 708 N=1,NO
      B=GQPO(N)
      W=GQWT(N)*WW
      D11=0.D0
      D12=0.D0
      D13=0.D0
      D21=0.D0
      D22=0.D0
      D23=0.D0
      SS(1)=A
      SS(2)=B
      DO 710 J=1,8
      LX=NBA(J)
      IF(NBA(J).EC.0) GO TO 710
      IF(ISIDE(I).GT.2) GO TO 725
      DD(1)=SCA(J)
      DD(2)=ETA(J)
      GO TO 727
  725 IF(ISICE(I).GT.4) GO TO 726
      DD(1)=SCA(J)
      DD(2)=EZA(J)
      GO TO 727
  726 DD(1)=ETA(J)
      DD(2)=EZA(J)
  727 CONTINUE
      DO 715 LT=1,2
  715 AA(LT)=1.D0+DD(LT)*SS(LT)
      ALP=.25D0
      ALPHA=ALP*AA(1)*AA(2)
      XN(J)=ALPHA
      DVA=ALP*CD(1)*AA(2)
      DVB=ALP*DD(2)*AA(1)
      XQ0=XQ(J)
      YQQ=YQ(J)
      ZQQ=ZQ(J)
      D11=D11+DVA*XQQ
      D12=D12+DVA*YQQ
      D13=D13+DVA*ZQQ
      D21=D21+DVB*XQQ
      D22=D22+DVB*YQQ
      D23=D23+DVB*ZQQ
  710 CONTINUE
      C1=D12*D23-D13*D22
      C2=D13*D21-D11*D23
      C3=D11*D22-D12*D21
      CMAG=DSQRT(C1**2+C2**2+C3**2)
      CC1=CC1+C1
      CC2=CC2+C2
      CC3=CC3+C3
      DO 720 J=1,8
      IF(NBA(J).EQ.0) GO TO 720
      XNI=XN(J)*W
      BNDI(J)=BNDI(J)+XNI*CMAG*DSF
  720 CONTINUE
  708 CONTINUE
      IF(ISIDE(I).GT.2) GO TO 10
      ONF=DSF
      GO TO 25
   10 IF(ISIDE(I).GT.4) GO TO 30
      ONF=-DSF
      GO TO 25
   30 ONF=DSF
   25 CONTINUE
      CCMAG=DSQRT(CC1**2+CC2**2+CC3**2)
      COSX=ONF*CC1/CCMAG
      COSY=ONF*CC2/CCMAG
      COSZ=ONF*CC3/CCMAG
      WRITE(8) COSX,COSY,COSZ,(BNDI(J),J=1,8),(NBA(J),J=1,8)
      WRITE(6,204) ISIDE(I),COSX,COSY,COSZ
```

```
204 FORMAT(1X,I4,3D20.10)
    WRITE(6,205) (NBA(J),BNDI(J),J=1,8)
205 FORMAT(1X,I4,D20.10)
    WRITE(6,206)
206 FORMAT(1X,//)
705 CONTINUE
  7 CONTINUE
355 CONTINUE
    ENDFILE4
    REWIND3
    REWIND4
    ENDFILE8
    REWIND8
    DO 360 I=1,NELEM
    READ(8) NE,NSIDE
    WRITE(9) NE,NSIDE
    IF(NSIDE.EQ.0) GO TO 360
    READ(8) (ISIDE(J),J=1,NSIDE)
    WRITE(9) (ISIDE(J),J=1,NSIDE)
    DO 361 IB=1,NSIDE
    READ(8) COSX,COSY,COSZ,(BNDI(K),K=1,8),(NBA(K),K=1,8)
    WRITE(9) COSX,COSY,COSZ,(BNDI(J),J=1,8),(NBA(J),J=1,8)
361 CONTINLE
360 CONTINUE
    ENDFILE9
    REWIND9
    REWIND8
    STOP
    END
```

```
C-----------------------------------------------------------------------
C      PROGRAM FOR THE DETERMINATION OF NODAL PRESSURES
C-----------------------------------------------------------------------
       IMPLICIT REAL*8(A-H,O-Z)
       DIMENSION NUMBER(125),NODB(100),BIV(100),DPDT(125),
      1DTDT(125),NOD(16),XK(16),YK(16),ZK(16),THETA(2),ALPHA(2),PZERO(2),
      2NODE(125),TITLE(18),DETAIL(54),NDB(125),DELT(50),
      3CI(16,16),MATNO(16),WQW(1,1),XKK(16),YKK(16),ZKK(16),
      4XIJCOB(16),FIFJ(16),GIGJ(16),HIHJ(16)
       DIMENSION NOCB(100),CIV(100),NODEC(125),X(125),Y(125),Z(125),
      1XIXJOB(16),XQX(16),YQY(16),ZQZ(16),HCC(2),TXK(2),TYK(2),TZK(2),
      2ZY(125),FLUXX(3,125),NEA(20),QW(20),TINJ(20),NWN(20),NEW(4,5),
      3NEB(4),QWT(4),TINJT(4),DMT(16)
       COMMON TEMPD(11),PRESSD(6),VISCOS(11,6)
       COMMON FLSPH(11,6),FLTHC(11,6),RHO(125),XMEW(125),SPEHET(125)
       COMMON HC,BETAT,BETAP,TEMREF,PREREF,THRCON(125)
       COMMON PRESS(125),TEMP(125),S(125,100),NCOL(125,100),BA(125)
       REWIND2
C-----------------------------------------------------------------------
C      READ PROBLEM DIMENSIONS FROM TAPE
C-----------------------------------------------------------------------
       READ(2) NTT,NELEM,NMAT,NBPTC,NPPTC,NODMAX,NNSTIF,NPT,NPTC,
      1(TITLE(I),I=1,18)
       WRITE(6,701)
  701  FORMAT('1',' DETAILS REGARDING SIZE OF PROBLEM AS READ FROM TAPE')
       WRITE(6,702) NTT,NELEM,NMAT,NBPTC,NODMAX,NNSTIF,NPT,NPTC
  702  FORMAT(1X,' NTT ',I5,' NELEM ',I5,' NMAT ',I5,' NBPTC ',I5,
      1' NODMAX ',I5,' NNSTIF ',I5,' NPT ',I5,' NPTC ',I5)
C-----------------------------------------------------------------------
C      READ DATA CARD FOR INDEX,NSTEDY,ITOTAL,ZTEST,NNN
C
       READ(5,751) INDEX,NSTEDY,ITOTAL,NNNP,NNNT,NINIT,NBOUND,IMB,ICCNT,
      1NQ,NQQ,ZTEST
  751  FORMAT(11I3,E10.3)
C-----------------------------------------------------------------------
```

```
C     LIST OF SYMBOLS
C-----------------------------------------------------------------------
C      INDEX  = 0 ,NORMAL USE
C             = 1 ,PRINT INTERMEDIATE STEPS
C     NSTEDY = 0 FOR STEADY STATE
C             = 1 FOR TRANSIENT CASE
C     ITOTAL = TOTAL NUMBER OF TIME STEPS
C             = 0 FOR STEADY STATE
C     ZTEST  = LOWEST 'CUT-OFF' COEFFICIENT OF 'A'-MATRIX IN 'EQSOLV'
C     NNNP   = MAXIMUM BANDWIDTH FOR 'EQSOLV', PRESSURE SOLUTION
C     NNNT   = MAXIMUM BANDWIDTH FOR 'EQSOLV', TEMPERATURE SCLUTION
C     NINIT  = NO OF NEW I.C. TO BE READ
C     NBOUND = NO OF NEW BOUNDARY CONDITIONS TO BE READ
C             = NO. OF NEW B.C. & I.C. TO BE READ
C     IMB    = 0. MASS BALANCE IS NOT PERFORMED
C             = 1. MASS BALANCE CALCULATED
C     ICONT  = 0. NORMAL RUN
C             = 1. IF PROGRAM IS CONTINUED FROM A PREVIOS RUN
C     NQ     = NUMBER OF WELLS
C     NQQ    = NUMBER OF WELL NODES (TOTAL)
C-----------------------------------------------------------------------
      READ(5,752) (DETAIL(I),I=1,54)
  752 FORMAT(18A4)
C     DETAIL = ALPHANUMERIC DESCRIPTION OF THE PROBLEM (3 CARDS)
      WRITE(6,703) (DETAIL(I),I=1,54)
  703 FORMAT(1X,100('*'),//1X,3(20X,18A4/),/1X,100('*')/)
      WRITE(6,704) NSTEDY,ITOTAL
  704 FORMAT(2X,' NSTEDY ',I4,' ITOTAL ',I4)
C-----------------------------------------------------------------------
C     READ PHYSIOGRAPHIC DATA FROM TAPE FILE 2
C-----------------------------------------------------------------------
      READ(2) (X(N),N=1,NTT)
      READ(2) (Y(N),N=1,NTT)
      READ(2) (Z(N),N=1,NTT)
      READ(2) IPOINT,(NUMBER(I),I=1,IPOINT)
      READ(2) (NOCB(N),N=1,NBPTC)
      READ(2) (BIV(N),N=1,NBPTC)
      READ(2) (NOCB(N),N=1,NPPTC)
      READ(2) (CIV(N),N=1,NPPTC)
      READ(2) (XK(N),YK(N),ZK(N),N=1,NMAT),(THETA(N),ALPHA(N),
     1PZERO(N),HCC(N),N=1,NMAT),(TXK(N),TYK(N),TZK(N),N=1,NMAT)
      READ(2) NPT,(NODE(I),I=1,NPT)
      READ(2) NPTC,(NODEC(I),I=1,NPTC)
      IF(ICONT.EQ.1) GO TO 820
      READ(2) (PRESS(N),N=1,NTT)
      READ(2) (TEMP(N),N=1,NTT)
  820 REWIND2
      IF(ICONT.EQ.0) GO TO 810
      READ(10) (PRESS(I),TEMP(I),RHO(I),XMEW(I),SPEHET(I),THRCON(I),
     1I=1,NTT)
  810 CONTINUE
      CALL RCPROP
C-----------------------------------------------------------------------
      READ(5,113) HO,PREREF,TEMREF,BETAT,BETAP
  113 FORMAT(6E12.5)
C     LIST OF SYMEOLS
C     HO        = REFERENCE DENSITY
C     PREREF    = REFERENCE PRESSURE
C     TEMREF    = REFERENCE TEMPERATURE
C     BETAT     = COEF. OF FLUID THERMAL VOLUME EXPANSICN
C     BETAP     = COEF. OF FLUID COMPRESSIBILTY
C-----------------------------------------------------------------------
C
      IF(NBOUNC.EC.0) GO TO 809
      DO 804 M=1,NBOUND
      READ(5,808) NBB,BIVD,CIVD
      DO 804 I=1,NBPTC
      IF(NUMBER(NCDB(I)/10).NE.NBB) GO TO 804
      BIV(I)=BIVD
      CIV(I)=CIVD
  804 CONTINUE
  808 FORMAT(I4,2E12.4)
```

```
809 IF(NINIT.EQ.0) GO TO 811
    DO 807 M=1,NINIT
    READ(5,808) NEWP,PRESN,TEMN
    DO 807 I=1,NTT
    IF(NUMBER(I).NE.NEWP) GO TO 807
    PRESS(I)=PRESN
    TEMP(I)=TEMN
807 CONTINUE
811 CALL UPDATE(IPOINT)
C
    DO 775 I=1,NTT
    DPDT(I)=0.D0
775 DTDT(I)=0.D0
    IF(INDEX.NE.2) GO TO 246
    WRITE(6,705)
705 FORMAT(1X,' THESE ARE THE INITIAL PRESSURES FOR EACH NODE')
    DO 258 I=1,NTT
258 WRITE(6,706) NUMBER(I),PRESS(I)
706 FORMAT(1X,' NODE ',I5,' PRESSURE ',E13.6)
    WRITE(6,707)
707 FORMAT(1X,' BOUNDARY NODES ')
    DO 198 I=1,NBPTC
    KK=NODB(I)/10
    IF(KK.EQ.0) GO TO 198
    NDB(I)=NUMBER(KK)
198 CONTINLE
    DO 101 N=1,NBPTC
101 WRITE(6,708) NDB(N),BIV(N)
708 FORMAT(1X,' NODE ',I5,' PRESSURE ',E13.6)
246 WRITE(6,709)
709 FORMAT(2X,/1X,' MATERIAL PROPERTIES AS READ FROM TAPE ')
    DO 247 N=1,NMAT
247 WRITE(6,710) N,XK(N),YK(N),ZK(N),THETA(N),ALPHA(N),HCC(N),
   1PZERO(N),TXK(N),TYK(N),TZK(N)
710 FORMAT(1X,' MAT # ',I3,' XK ',E13.4,' YK ',E13.4,' ZK ',E13.4,/
   11X,' THETA ',E13.4,' ALPHA ',E13.4,'  HCC  ',E13.4,/1X,' PZERO ',
     2E13.4,' TXK ',E13.4,' TYK ',E13.4,' TZK ',E13.4)
C-----------------------------------------------------------------------
C     READ TIME STEPS IF NECESSARY.
      IF(ITOTAL.NE.0) READ(5,753) (DELT(I),I=1,ITOTAL)
  753 FORMAT(6E12.4)
C-----------------------------------------------------------------------
      IF(NQ.EQ.0) GO TO 55
      READ(5,714) (NEB(I),I=1,NQ)
C     NEB(I) = NUMBER OF NODES ALONG WELL 'I'
  714 FORMAT(20I3)
      DO 58 I=1,NQ
      N=NEB(I)
   58 READ(5,714) (NEW(I,J),J=1,N)
C     NEW(I,J)= LIST OF 'J' NCDES ALONG WELL 'I'.
      READ(5,714) (NEA(I),I=1,NQQ)
C     NEA(I) = NO. OF ELEMENTS WITH WELL NODE 'I', ORDER DEFINED BY
C              'NEW' ABOVE, IN COMMON
   55 CONTINUE
C
C-----------------------------------------------------------------------
C     ITERATION FCR TRANSIENT CASE
C-----------------------------------------------------------------------
      NNN=NNNP
      DO 199 II=1,ITOTAL
      DELTAT=DELT(II)
      IF(DELTAT.EQ.0.D0) DELTAT=1.D0
C
      IF(NQ.EQ.0) GO TO 56
      READ(5,760) (QWT(I),TINJT(I),I=1,NQ)
  760 FORMAT(8E10.3)
C     QWT    = FLOW RATE OF WELL 'I'
C     TINJT  = INJECTION TEMPERATURE CF WELL 'I'
      IZ=1
      DO 812 I=1,NQ
      IW=NEB(I)
      DO 813 J=1,IW
      QW(IZ)=(1.D0/IW)*QWT(I)
      TINJ(IZ)=TINJT(I)
      NWN(IZ)=NEW(I,J)
C
C     NWN(I) = NODAL LOCATION FOR WELL 'I'
C     TINJ(I)= INJECTION TEMPERATURE OF WELL 'I'
C     QW(I)  = FLOW RATE OF WELL 'I'
C
  813 IZ=IZ+1
  812 CONTINUE
      DO 815 I=1,NQQ
  815 WRITE(6,814) NWN(I),QW(I),TINJ(I)
  814 FORMAT(1X,' SOURCE/SINK NODE ',I5,'  QW = ',E10.3,'  TINJ = ',E10
     1.3)
   56 CONTINUE
C
      NZX=NNNP+48
      ZTEST=0.D0
      DO 20 I=1,NPT
      BA(I)=0.D0
      DO 20 J=1,NZX
      S(I,J)=0.D0
   20 NCOL(I,J)=0
      IF(II.EQ.1) GO TO 45
      DO 19 I=1,NBPTC
   19 NODB(I)=IABS(NODB(I))
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C     START STIFFNESS MATRIX FORMULATION ELEMENT BY ELEMENT
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
   45 GRAV=9.78049D0
      TMASS1=0.D0
      TMASS2=0.D0
      TWFLOW=0.D0
      DO 100 IX=1.NELEM
C-----------------------------------------------------------------------
C     THIS 'NOD' IS THE SEQUENTIAL. NCN-ZERO CRDER OF NODES
C-----------------------------------------------------------------------
      READ(3) NN.(NOD(J).J=1.NN).MM1.NQW.NE
C     IF(MM1.EQ.0) READ(3) (MATNO(I).I=1.NN)
      IF(NQW.NE.0) READ(3) ((WQW(I.J).J=1.NN).I=1.NQW)
      IF(INDEX.NE.4) GO TO 23
      WRITE(6.711) NE.NN
  711 FORMAT(2X./1X.' FOR ELEMENT # '.I4.' TOTAL NODES ARE '.I4.' IN THE
     1 FOLLOWING SEQUENCE; FIRST TOP. THEN BOTTCM CORNERS. MID. AND
     2 CUBIC ')
      WRITE(6.712) (NUMBER(NOD(I)).I=1.NN)
  712 FORMAT(1X.8I10)
   23 CONTINUE
      NODMAX=NN
      MN=MM1
      DO 107 I=1.NN
      K=NOD(I)
C-----------------------------------------------------------------------
C     ESTIMATION OF HYDRAULIC CONDUCTIVITY AS A FUNCTICN CF DENSITY AND
C     VISCOSITY
      XKK(I)=XK(MN)*RHO(K)/XMEW(K)
      YKK(I)=YK(MN)*RHO(K)/XMEW(K)
      ZKK(I)=ZK(MN)*RHO(K)/XMEW(K)
      IF(INDEX.EQ.1) WRITE(6.715) K.XKK(I).YKK(I).ZKK(I)
  715 FORMAT(1X.' NODE '.I4.' KX '.E13.4.' KY '.E13.4.' KZ '.E13.4)
  107 CONTINUE
      READ(3) (XIJCOB(J).J=1.NN)
      AP=ALPHA(MN)
      DO 901 I=1.NODMAX
      L=NOD(I)
      PIJ=0.D0
      CCCJ=0.D0
      READ(3) (FIFJ(J).GIGJ(J).HIHJ(J).J=1.NN)
      READ(3) (XIXJOB(J),J=1,NN)
      READ(3) (XQX(J),YQY(J),ZQZ(J),J=1,NN)
      DO 102 J=1,NODMAX
      K=NOD(J)
      PPIJ=0.D0
      CIJ=FIFJ(J)*XKK(J)+GIGJ(J)*YKK(J)+HIHJ(J)*ZKK(J)
      HKX=XK(MN)*RHO(K)*RHO(K)*GRAV/XNEW(K)
      HKY=YK(MN)*RHO(K)*RHO(K)*GRAV/XNEW(K)
      HKZ=ZK(MN)*RHO(K)*RHO(K)*GRAV/XNEW(K)
      CCIJ=HKX*FIFJ(J)+HKY*GIGJ(J)+HKZ*HIHJ(J)
      CCCJ=CCCJ+CCIJ*Z(K)
      IF(NSTEDY.EQ.0) GO TO 802
      TTO=THETA(MN)
      COF1=(RHC(K)*AP+TTO*HO*BETAP)
      PPIJ=XIXJOB(J)*COF1/DELTAT
  802 CI(I,J)=CIJ+PPIJ
      PIJ=PIJ+PPIJ*PRESS(K)
  102 CONTINUE
      PROPM=THETA(MN)*HO*BETAT*DTDT(L)*XIJCOB(I)
      QQW=0.D0
      ZY(I)=QQW-PROPM+PIJ-CCCJ
      TMASS1=TMASS1+RHO(L)*THETA(MN)*XIJCOB(I)
      TMASS2=TMASS2+RHO(L)*(THETA(MN)+AP*DPDT(L)*DELTAT)*XIJCOB(I)
      IF(INDEX.NE.1) GO TO 901
      L=NUMBER(L)
      WRITE(6,716) L,ZY(I),(CI(I,J),J=1,NN)
  716 FORMAT(1X,' NODE ',I4,' ZY= ',E13.4,/1X,'  C=   ',8E13.4)
  901 CONTINUE
C-----------------------------------------------------------------------
C     THE BOUNDARY CONDITIONS ARE CONSIDERED
C-----------------------------------------------------------------------
C
      DO 334 J=1,NODMAX
      NRO=NOD(J)
      DO 331 K=1,NBPTC
      KK=IABS(NODB(K))
      IF(KK.EQ.0) GO TO 331
      K1=KK/10
      IF(NRQ.NE.K1) GO TO 331
      XXX=1.D0
      IF(NODB(K).LT.0) XXX=0.D0
      NODB(K)=-KK
      IIFLG=MOD(KK,10)
      BIVD=BIV(K)
      ZY(J)=ZY(J)+XXX*BIVD
      IF(IIFLG.EQ.0) GO TO 330
C-----------------------------------------------------------------------
C     PRESSURE IS EQUATED TO B.C. OF NODE IN CASE:
C     1) BOUNDARY VALUE FOR NODE IS DIFFERENT THAN I.C.
C     2) IF B.C. CHANGES (USER SUPPLIED OPTION)
C-----------------------------------------------------------------------
      PRESS(NRQ)=EIVD
      ZY(J)=BIVD*XXX
      CI(J,J)=0.D0
      DO 327 M=1,NODMAX
      ZY(M)=ZY(M)-CI(M,J)*BIVD
      CI(M,J)=0.D0
      CI(J,M)=0.D0
  327 CONTINUE
      GO TO 330
  331 CONTINUE
  330 CONTINUE
  334 CONTINUE
      IF(INDEX.NE.1) GO TO 175
      DO 170 J=1,NN
      NRC=NOC(J)
      NRQ=NUMBER(NRQ)
      WRITE(6,717) NRQ,ZY(J),(CI(J,I),I=1,NN)
  717 FORMAT(1X,' NODE= ',I4,' ZY= ',E15.6,/1X,' C= ',8E15.6/1X,8E15.6/
     11X,8E15.6/1X,8E15.6)
  170 CONTINUE
C-----------------------------------------------------------------------
```

```
C     THE ELEMENT MATRIX IS NOW ADDED INTO THE SYSTEM MATRIX
C-----------------------------------------------------------------------
  175 DO 350 K=1,NODMAX
      IF(CI(K,K).EQ.0.D0) GO TO 350
      KM=NOD(K)
      DO 108 I=1,NPT
      IF(KM.EQ.NODE(I)) GO TO 8
  108 CONTINUE
      GO TO 350
    8 KK=I
      DO 345 L=1,NODMAX
      IF(CI(K,L).EQ.0.D0) GO TO 345
      JL=NOD(L)
      DO 109 I=1,NPT
      IF(JL.EQ.NODE(I)) GO TO 9
  109 CONTINUE
      GO TO 345
    9 JJ=I
      IF(NCOL(KK,1).NE.0) GO TO 128
      NCOL(KK,1)=JJ
      S(KK,1)=CI(K,L)
      GO TO 345
  128 DO 129 LL=1,NNNP
      IF(NCOL(KK,LL).EQ.0) GO TO 131
      IF(NCOL(KK,LL).LT.JJ) GC TO 129
      IF(NCOL(KK,LL).NE.JJ) GO TO 132
      S(KK,LL)=S(KK,LL)+CI(K,L)
      GO TO 345
  129 CONTINUE
  132 LAST=NCOL(KK,LL)
      ALAST=S(KK,LL)
      N1N=NNNP-1
      DO 133 LLL=LL,N1N
      L1=LLL+1
      NEXT=NCOL(KK,L1)
      ANEXT=S(KK,L1)
      S(KK.L1)=ALAST
      NCOL(KK.L1)=LAST
      IF(NEXT.EQ.0) GO TO 131
      LAST=NEXT
  133 ALAST=ANEXT
  131 S(KK.LL)=CI(K.L)
      NCOL(KK,LL)=JJ
  345 CONTINUE
      BA(KK)=BA(KK)+ZY(K)
  350 CONTINUE
  100 CONTINUE
      REWIND3
C
      IF(NQ.EQ.0) GO TO 906
      DO 50 I=1.NGQ
      K=NWN(I)
      DO 904 J=1.NPT
      N=NUMBER(NODE(J))
      IF(N.EQ.K) GO TO 905
  904 CONTINUE
  905 BA(J)=BA(J)+QW(I)
   50 CONTINUE
  906 CONTINUE
C
      CALL ECSOLV(NPT.NNNP.ZTEST)
      DO 110 I=1.NPT
      K=NODE(I)
      SI1=BA(I)
      IF(NSTEDY.EC.0) GO TO 6
      DPDT(K)=(SI1-PRESS(K))/DELTAT
    6 PRESS(K)=SI1
  110 CONTINUE
      IF(IMB.EQ.0) GO TO 823
      IF(II.EQ.1) GO TO 824
      DMODTT=DELT(II-1)
      CALL WATEAL
```

```
     1(FLUXX,TWFOLD.TMASS2.TMOLD.DMODTT.NELEM)
  824 TMOLD=TMASS1
      TWFOLD=TWFLCW
  823 CONTINUE
            CALL FLUX(NTT.NELEM.PRESS.NNN.FLUXX.Z.RHO.XMEW.XK.YK.ZK.
     1NODB.NEPTC)
      NZX=NNNT+48
      DO 640 I=1.NPTC
      BA(I)=0.D0
      DO 640 J=1.NZX
      S(I.J)=0.D0
  640 NCOL(I.J)=0
C-----------------------------------------------------------------------
C     INITALIZE BCUNDARY NODES THAT ARE MADE NEGATIVE ONCE THE HEAT
C     FLUX OR TEMPERATURE IS CONSIDERED.
C-----------------------------------------------------------------------
      DO 600 I=1.NPPTC
  600 NOCB(I)=IABS(NOCB(I))
C-----------------------------------------------------------------------
C     ELEMENT BY ELEMENT FORMULATION OF THE STIFFNESS MATRIX FOR
C     TEMPERATURE SOLUTION.
C-----------------------------------------------------------------------
      DO 601 IX=1.NELEM
      READ(3) NN.(NOD(J).J=1.NN).MM1.NOW.NE
C     IF(MM1.EQ.0) READ(3) (MATNO(I).I=1.NN)
      IF(NOW.NE.0) READ(3) ((WOW(I.J).J=1.NN).I=1.NOW)
      IF(NOW.NE.0) IB=IB+1
      TTO=THETA(MM1)
      MN=MM1
      HCCMN=HCC(MN)
C     IF(MN.NE.0) GO TO 602
C     MN=MATNO(I)
C 602 IF(MN.EQ.0) MN=1
      READ(3) (XIJCOB(J).J=1.NN)
      SIJ=0.D0
      DIJ=0.D0
      DO 603 I=1,NN
      L=NOD(I)
      JB=NUMBER(L)
C
      MCHECK=0
      IF(NQ.EQ.0) GO TO 948
      DO 51 IP=1,NQQ
      K=NWN(IP)
      IF(JB.EQ.K) GO TO 54
   51 CONTINUE
      GO TO 948
   54 IU=NEA(IP)
      MCHECK=1
  948 CONTINUE
C
      READ(3) (FIFJ(J),GIGJ(J),HIHJ(J),J=1,NN)
      READ(3) (XIXJOB(J),J=1,NN)
      READ(3) (XQX(J),YQY(J),ZQZ(J),J=1,NN)
      STOR=0.D0
      DO 1141 L=1,NN
      LL=NOD(L)
 1141 STOR=STOR+(FLUXX(1,LL)*XQX(L)+FLUXX(2,LL)*YQY(L)+FLUXX(3,LL)*ZQZ(L
     1))/XIJCOB(I)
      DO 604 J=1,NN
      K=NOD(J)
      PPIJ=0.D0
      QWELL=0.C0
      CONVEC=(XQX(J)*FLUXX(1,K)+YQY(J)*FLUXX(2,K)+ZQZ(J)*FLUXX(3,K))
     1*SPEHET(K)
      TCX=TXK(MN)*((THRCON(K)/TXK(MN))**TTO)
      TCY=TYK(MN)*((THRCON(K)/TYK(MN))**TTO)
      TCZ=TZK(MN)*((THRCON(K)/TZK(MN))**TTO)
      CIJ=TCX*FIFJ(J)+TCY*GIGJ(J)+TCZ*HIHJ(J)
      IF(MCHECK.EC.0) GO TO 608
      AU=DFLOAT(IU)
      QWELL=(1.D0/AU)*QW(IP)*SPEHET(I)*XIXJCB(J)/XIJCOB(I)
  608 IF(NSTEDY.EQ.0) GO TO 943
      PPIJ=(PZERO(MN)*HCCMN*(1.DO-TTO)+TTO*RHO(K)*SPEHET(K))*XIXJOB(J)/
     1DELTAT
      DSTOR=STOR*XIXJOB(J)*SPEHET(K)
  943 CI(I,J)=CONVEC+CIJ+QWELL-DSTOR
      SIJ=SIJ+PPIJ
      IF(I.EQ.J) DIJ=DIJ+PPIJ
      IF(I.EQ.J) CMT(I)=PPIJ
  604 CONTINUE
C     PROGRAM IS CODED FOR A LUMPED TIME MATRIX FOR THE TEMPERATURE SCLUTION.
  603 CONTINUE
      DO 6060 I=1,NN
      L=NOD(I)
      ZY(I)=DMT(I)*SIJ*TEMP(L)/DIJ
 6060 CI(I,I)=CI(I,I)+DMT(I)*SIJ/DIJ
      IF(INDEX.NE.1) GO TO 609
      WRITE(6,610) NE
      DO 611 I=1,NN
      L=NOD(I)
      L=NUMBER(L)
      WRITE(L,612) L,ZY(I),(CI(I,J),J=1,NN)
  610 FORMAT(1X,' STIFFNESS MATRIX BEFORE BCUNDARY CONCITICNS CONSIDERED
     1 ',' ELEMENT # ',I5)
  612 FORMAT(1X,'NODE= ',I3,' ZY= ',E15.6,' C= ',8E10.3/1X,8E10.3/
     11X,8E10.3/1X,8E10.3)
  611 CONTINUE
C-----------------------------------------------------------------------
C     BOUNDARY CCNDITIONS ARE CONSIDERED.
C-----------------------------------------------------------------------
  609 DO 613 J=1,NN
      NRQ=NOD(J)
      DO 614 K=1,NPPTC
      KK=IABS(NOCB(K))
      IF(KK.EQ.0) GO TO 614
      K1=KK/10
      IF(NRQ.NE.K1) GO TO 614
      XXX=1.D0
      IF(NOCE(K).LT.0) XXX=0.D0
      NOCB(K)=-KK
      IIFLG=MOD(KK,10)
      BIVD=CIV(K)
C-----------------------------------------------------------------------
C     FLUX ADDED IN.
C-----------------------------------------------------------------------
      ZY(J)=ZY(J)+XXX*BIVD
      IF(IIFLG.EQ.0) GO TO 613
C-----------------------------------------------------------------------
C     TEMPERATURE SPECIFIED.
C-----------------------------------------------------------------------
      TEMP(NFQ)=EIVD
      ZY(J)=BIVD*XXX
      CI(J,J)=0.D0
      DO 615 M=1,NN
      ZY(M)=ZY(M)-CI(M,J)*BIVD
      CI(M,J)=0.D0
      CI(J,M)=0.D0
  615 CONTINUE
      GO TO 613
  614 CONTINUE
  613 CONTINUE
C-----------------------------------------------------------------------
C     THE ELEMENT MATRIX IS ADDED INTC THE SYSTEM MATRIX.
C-----------------------------------------------------------------------
      DO 616 K=1,NN
      IF(CI(K,K).EC.0.D0) GO TO 616
      KM=NOD(K)
      DO 617 I=1,NPTC
      IF(KM.EQ.NOCEC(I)) GO TO 618
  617 CONTINUE
      GO TO 616
  618 KK=I
      DO 620 L=1,NN
      IF(CI(K,L).EC.0.D0) GO TO 620
      JL=NOD(L)
      DO 621 I=1,NPTC
      IF(JL.EQ.NCCEC(I)) GO TO 622
  621 CONTINUE
      GO TO 620
  622 JJ=I
      IF(NCOL(KK,1).NE.0) GO TO 623
      NCOL(KK,1)=JJ
      S(KK,1)=CI(K,L)
      GO TO 620
  623 DO 624 LL=1,NNNT
      IF(NCOL(KK,LL).EQ.0) GO TO 625
      IF(NCOL(KK,LL).LT.JJ) GO TO 624
      IF(NCOL(KK,LL).NE.JJ) GO TO 626
      S(KK,LL)=S(KK,LL)+CI(K,L)
      GO TO 620
  624 CONTINUE
  626 LAST=NCOL(KK,LL)
      ALAST=S(KK,LL)
      LNX=NNNT-1
      DO 627 LLL=LL,LNX
      L1=LLL+1
      NEXT=NCOL(KK,L1)
      ANEXT=S(KK,L1)
      S(KK,L1)=ALAST
      NCOL(KK,L1)=LAST
      IF(NEXT.EQ.0) GO TO 625
      LAST=NEXT
  627 ALAST=ANEXT
  625 S(KK,LL)=CI(K,L)
      NCOL(KK,LL)=JJ
  620 CONTINUE
      BA(KK)=BA(KK)+ZY(K)
  616 CONTINUE
  601 CONTINUE
```

```
      REWIND3
      REWIND8
C
      IF(NQ.EQ.0) GO TO 930
      DO 931 I=1,NOQ
      K=NWN(I)
      DO 932 J=1,NPTC
      N=NUMBER(NODEC(J))
      IF(N.EQ.K) GO TO 933
  932 CONTINUE
  933 IF(QW(I).LT.0.D0) TINJ(I)=TEMP(J)+.05D0*(2.D1-TEMP(J))
      BA(J)=BA(J)+QW(I)*SPEHET(J)*TINJ(I)
  931 CONTINUE
  930 CONTINUE
C
      CALL EQSOLV(NPTC,NNNT,ZTEST)
      DO 629 I=1,NPTC
      K=NODEC(I)
      SI1=BA(I)
      IF(NSTEDY.EQ.0) GO TO 629
      DTDT(K)=(SI1-TEMP(K))/DELTAT
  629 TEMP(K)=SI1
      CALL UPDATE(IPOINT)
      WRITE(6,770) (TITLE(I),I=1,18)
  770 FORMAT(1X,/5X,18A4)
      WRITE(6,769) II,DELTAT
  769 FORMAT(1X,/1X,' RESULTS OF PRESSURE ESTIMATION AFTER TIME STEP ',
     1I4,'  DELTAT= ',E15.6,'  SECONDS')
      WRITE(6,641)
  641 FORMAT(1X,'NODE',9X,'PRESSURE',8X,'TEMPERATURE',10X,'RHO',14X,
     1'XMEW',13X,'SPEHET',9X,'THRCON',11X,'COMPRS')
      DO 630 I=1,NTT
      N=NUMBER(I)
      IF(N.LT.100) WRITE(6,631)
  631 FORMAT(1X,//)
      WRITE(6,632) N,PRESS(I),TEMP(I),RHO(I),XMEW(I),SPEHET(I),THRCON(I)
```

```
   1,BETAP
 632 FORMAT(1X,I4,5X,7E17.8)
 630 CONTINUE
 199 CONTINLE
     WRITE(10) (PRESS(I),TEMP(I),RHO(I),XMEW(I),SPEHET(I),THRCON(I),
    1I=1,NTT)
     ENDFILE10
     REWIND10
     STOP
     END
C
C
     SUBROUTINE WATBAL
    1(FLUXX,TWFOLD,TMASS,TMOLD,DELTAT,NELEM)
     IMPLICIT REAL*8(A-H,O-Z)
     DIMENSION ISIDE(5),BNDI(8),NBA(8),FLUXX(3,125)
     TBFLOW=0.D0
     DO 100 IX=1,NELEM
     READ(4) NE,NSIDE
     IF(NSIDE.EQ.0) GO TO 100
     READ(4) (ISIDE(I),I=1,NSIDE)
     DO 101 IB=1,NSIDE
     READ(4) COSX,COSY,COSZ,(BNDI(J),J=1,8),(NBA(J),J=1,8)
     DO 110 II=1,8
     IF(NBA(II).EQ.0) GO TO 110
     QXA=FLLXX(1,NBA(II))
     QYA=FLUXX(2,NBA(II))
     QZA=FLUXX(3,NBA(II))
     FNORM=CXA*CCSX+QYA*COSY+QZA*COSZ
     TBFLOW=TBFLCW-FNORM*DABS(BNDI(II))
 110 CONTINUE
 101 CONTINUE
 100 CONTINUE
     REWIND4
     WATBL=(TMASS-TMOLD)+((TBFLOW-TWFOLD)*DELTAT)
     RATIO=(TBFLCW-TWFOLD)*DELTAT/(TMASS-TMOLC)
```

```
      WRITE(6,120) TBFLOW,TWFOLD,TMOLC,TMASS
  120 FORMAT(1X,/5X,'TOTAL BOUNDARY FLOW = ',E13.6,' KG',/5X,'TOTAL WELL
     1 FLOW = ',E13.6,'KG',/5X,'TOTAL MASS AT BEG. OF TIME STEP = ',
     2E13.6,'KG',/5X,'TOTAL MASS AT END CF TIME STEP = ',E13.6,'KG')
      WRITE(6,121) WATBL,RATIC
  121 FORMAT(1X,'MASS BALANCE = ',E13.6,'  RATIO = ',E13.6)
      RETURN
      END
C
C-----------------------------------------------------------------------
      SUBROUTINE RDPROP
C-----------------------------------------------------------------------
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON TEMPD(11),PRESSD(6),VISCOS(11,6)
      COMMON FLSPH(11,6),FLTHC(11,6),RHO(125),XMEW(125),SPEHET(125)
      COMMON HO,BETAT,BETAP,TEMREF,PREREF,THRCON(125)
      COMMON PRESS(125),TEMP(125),S(125,100),NCCL(125,100),EA(125)
      DO 10 J=1,6
      DO 10 I=1,11
   10 READ(5,11) TEMPD(I),PRESSD(J),VISCOS(I,J),FLSPH(I,J),FLTHC(I,J)
   11 FORMAT(3X,2E11.4,11X,3E11.4)
    9 FORMAT(1X,' PRESSURE, PASCAL',6E18.8)
   30 FORMAT(1X,' TEMPERATURE, DEGREES CELCIUS')
   12 FORMAT(1X,F10.0,7X,6E18.8)
      PRINT 13
   13 FORMAT(1X,' FLUID DYNAMIC VISCOSITY, PASCAL SECONDS')
      PRINT 9,(PRESSD(J),J=1,6)
      PRINT 30
      DO 14 I=1,11
   14 PRINT 12,TEMPD(I),(VISCCS(I,J),J=1,6)
      PRINT 15
   15 FORMAT(1X,' FLUID SPECIFIC HEAT, JOULES PER KILOGRAM PER DEGREE
     1 CELCILS')
      PRINT 9,(PRESSD(J),J=1,6)
      PRINT 30
      DO 16 I=1,11
   16 PRINT 12,TEMPD(I),(FLSPH(I,J),J=1,6)
      PRINT 17
   17 FORMAT(1X,' FLUID THERMAL CCNDUCTIVITY, WATTS PER METER PER DEGREE
     1 CELCIUS')
      PRINT 9,(PRESSD(J),J=1,6)
      PRINT 30
      DO 18 I=1,11
   18 PRINT 12,TEMPD(I),(FLTHC(I,J),J=1,6)
      RETURN
      END
C-------------------------------------------------------------------------
C
      SUBROUTINE UPDATE(IPCINT)
      IMPLICIT REAL*8(A-H,O-Z)
      COMMON TEMPD(11),PRESSD(6),VISCOS(11,6)
      COMMON FLSPH(11,6),FLTHC(11,6),RHO(125),XMEW(125),SPEHET(125)
      COMMON HO,BETAT,BETAP,TEMREF,PREREF,THRCON(125)
      COMMON PRESS(125),TEMP(125),S(125,100),NCOL(125,100),BA(125)
      DO 1 K=1,IPCINT
      I = TEMP(K)/10+1
      I1 = I+1
      J = PRESS(K)/2E6+1
      J1 = J+1
      PF=(PRESS(K)-PRESSD(J))/(PRESSD(J1)-PRESSC(J))
      TF=(TEMP(K)-TEMPD(I))/(TEMPD(I1)-TEMPD(I))
      TFPF = TF*PF
      TPD = TF-TFPF
      RHO(K)=HO+HC*BETAT*(TEMP(K)-TEMREF)+HO*BETAP*(PRESS(K)-PREREF)
      XMEW(K)=VISCOS(I,J)+PF*(VISCOS(I,J1)-VISCOS(I,J))
     1+TFPF*(VISCCS(I1,J1)-VISCOS(I,J1))
     2+TPD*(VISCOS(I1,J)-VISCOS(I,J))
      SPEHET(K)=FLSPH(I,J)+PF*(FLSPH(I,J1)-FLSFH(I,J))
     1+TFPF*(FLSPH(I1,J1)-FLSPH(I,J1))
     2+TPD*(FLSPH(I1,J)-FLSPH(I,J))
      THRCON(K)=FLTHC(I,J)+PF*(FLTHC(I,J1)-FLTFC(I,J))
     1+TFPF*(FLTHC(I1,J1)-FLTHC(I,J1))
     2+TPD*(FLTHC(I1,J)-FLTHC(I.J))
    1 CONTINUE
      RETURN
      END
C
C-------------------------------------------------------------------------------
      SUBROUTINE FLUX(NTT.NELEM.PRESS.NNN.FLUXX.Z.RHO.XMEW.XK.YK.ZK.
     1NODB.NEPTC)
C-----------------------------------------------------------------------
C     SUBROUTINE FOR THE ESTIMATION OF FLUX AS A FUNCTIONAL REPRESENTATION
C-----------------------------------------------------------------------
      IMPLICIT REAL*8(A-H.O-Z)
      DIMENSION XIXJOB(16).NOD(16).C(16.16).ZY(3.125).SL(3.125).
     1XKF(16).YKF(16).ZKF(16).XQX(16).YQY(16).ZQZ(16).FLUXX(3.125)
      DIMENSION XK(16).YK(16).ZK(16).Z(125)
      DIMENSION PRESS(125).RHO(125).XMEW(125).S(56.104).ICCL(56.104).
     1B(56)
      DIMENSION NCDB(100).IZY(20).ZZY(3.20)
C-----------------------------------------------------------------------
C     ASSEMBLE THE STIFFNESS MATRIX ELEMENT BY ELEMENT
C-----------------------------------------------------------------------
      GRAV=9.7804SD0
      NZX=NNN+48
      DO 160 I=1.NTT
      DO 160 J=1.NZX
      S(I.J)=0.D0
  160 ICOL(I.J)=0
      DO 161 I=1.3
      DO 161 J=1.NTT
  161 SL(I.J)=0.D0
      DO 105 IX=1.NELEM
      READ(3) NN.(NOD(J).J=1.NN).MM1.NQW.NE
C     IF(MM1.EQ.0) READ(3) (MATNO(I).I=1.NN)
      IF(NQW.NE.0) READ(3)
      READ(3)
      DO 162 J=1.NN
      K=NOD(J)
      XKF(J)=XK(MM1)*RHO(K)/XMEW(K)
      YKF(J)=YK(MM1)*RHO(K)/XMEW(K)
      ZKF(J)=ZK(MM1)*RHO(K)/XMEW(K)
  162 CONTINUE
      DO 110 I=1,NN
      READ(3)
      READ(3) (XIXJOB(J),J=1,NN)
      READ(3) (XQX(J),YQY(J),ZQZ(J),J=1,NN)
      ZY(1,I)=0.D0
      ZY(2,I)=0.D0
      ZY(3,I)=0.D0
      DO 110 J=1,NN
      K=NOD(J)
      C(I,J)=XIXJCB(J)
      XXX=-XKF(J)*(PRESS(K)+RHO(K)*GRAV*Z(K))*XQX(J)
      XXY=-YKF(J)*(PRESS(K)+RHO(K)*GRAV*Z(K))*YCY(J)
      XXZ=-ZKF(J)*(PRESS(K)+RHO(K)*GRAV*Z(K))*ZQZ(J)
      ZY(1,I)=ZY(1,I)+XXX
      ZY(2,I)=ZY(2,I)+XXY
      ZY(3,I)=ZY(3,I)+XXZ
  110 CONTINUE
      DO 334 J=1,NN
      NRQ=NOD(J)
      DO 331 K=1,NBPTC
      KK=IABS(NODB(K))
      IF(KK.EQ.0) GO TO 331
      K1=KK/10
      IF(NRQ.NE.K1) GO TO 331
      IIFLG=MOD(KK,10)
      IF(IIFLG.EQ.1) GO TO 330
      DO 327 M=1,NN
      ZY(1,M)=ZY(1,M)-C(M,J)*.5D-6
      ZY(2,M)=ZY(2,M)-C(M,J)*.1D-6
      ZY(3,M)=ZY(3,M)-C(M,J)*.711D-4
      C(M,J)=0.D0
      C(J.M)=0.D0
  327 CONTINUE
      C(J.J)=1.D0
      ZY(1.J)=.5D-6
      ZY(2.J)=.1D-6
      ZY(3.J)=.711D-4
      GO TO 330
  331 CONTINUE
  330 CONTINUE
  334 CONTINUE
      DO 118 K=1.NN
      KK=NOD(K)
      DO 120 L=1.NN
      IF(C(K.L).EC.0.D0) GO TO 120
      JJ=NOD(L)
      IF(ICOL(KK.1).NE.0) GO TO 346
      ICOL(KK.1)=JJ
      S(KK.1)=C(K.L)
      GO TO 120
  346 DO 347 LL=1.NNN
      IF(ICOL(KK.LL).EQ.0) GO TO 352
      IF(ICOL(KK.LL).LT.JJ) GO TO 347
      IF(ICOL(KK.LL).NE.JJ) GC TO 349
      S(KK.LL)=S(KK.LL)+C(K.L)
      GO TO 120
  347 CONTINUE
  349 LAST=ICOL(KK.LL)
      ALAST=S(KK.LL)
      LX=NNN-1
      DO 351 LLL=LL.LX
      L1=LLL+1
      NEXT=ICOL(KK.L1)
      ANEXT=S(KK.L1)
      S(KK.L1)=ALAST
      ICOL(KK.L1)=LAST
      IF(NEXT.EQ.0) GO TO 352
      LAST=NEXT
  351 ALAST=ANEXT
  352 S(KK,LL)=C(K,L)
      ICOL(KK,LL)=JJ
  120 CONTINUE
      DO 121 J=1,3
  121 SL(J,KK)=SL(J,KK)+ZY(J,K)
  118 CONTINUE
  105 CONTINUE
      REWIND3
      DO 130 I=1,NTT
  130 WRITE(8) (ICOL(I,J),J=1,NZX),(S(I,J),J=1,NZX),(SL(J,I),J=1,3)
      ENDFILE8
      REWIND8
      DO 150 N=1,3
      CALL FSOLV(NTT,NNN,N,B)
      NPT=NTT
      DO 151 I=1,NTT
  151 FLUXX(N,I)=B(I)
  150 CONTINUE
      DO 937 J=1,NTT
  937 WRITE(6,936) (FLUXX(I,J),I=1,3)
  936 FORMAT(1X,3D20.10)
      WRITE(9) (FLUXX(1,J),J=1,NTT)
      WRITE(9) (FLUXX(2,J),J=1,NTT)
      WRITE(9) (FLUXX(3,J),J=1,NTT)
      ENDFILE9
      REWIND9
      RETURN
      END
C
      SUBROUTINE FSOLV(NPT,NNN,NTIAL,B)
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION B(56),AA(56),NPIV(56),IBANDW(56),NNCOL(56),SS(3)
      DIMENSION A(56,104),NCOL(56,104)
C---------------------------------------------------------------------
```

```
C      INITALIZE THE BANDWIDTH COUNTER
C-----------------------------------------------------------------------
       NZX=NNN+48
       N=NTIAL
       NTT=NPT
       DO 110 I=1,NPT
   92 READ(8,END=93) (NCOL(I,J),J=1,NZX),(A(I,J),J=1,NZX),(SS(J),J=1,3)
       IF(NCOL(I,1).EQ.0) GO TO 92
  110 B(I)=SS(N)
   93 NPT=I-1
       REWIND8
       NDIF=NTT-NPT
       IF(NDIF.NE.0) WRITE(6,90) NDIF,NPT
   90 FORMAT(1X,I4,' ROWS HAVE ZERO CCEFICIENTS AND ARE NCT CCNSIDEREC.
     1 NEW # OF ECATIONS ARE ',I4)
       MAXCOL=0
       DO 100 I=1,NPT
       IBANDW(I)=0
  100 CONTINUE
       NBANDW=NNN
       DO 1 I=1,NPT
       DO 2 J=1,NNN
       NC=NCOL(I,J)
       IF(NC.EQ.0) GO TO 1
       IBANDW(I)=J
    2 CONTINUE
    1 CONTINUE
       ZTEST=0.D0
       LX=NPT-1
       DO 3 LL=1,LX
C-----------------------------------------------------------------------
C      FIND THE ROW WITH MINIMUM BANDWIDTH
C-----------------------------------------------------------------------
       KK=100000
       DO 4 I=LL,NPT
       IC=IBANDW(I)
```

```
      IF(IC.LE.0) GO TO 4
      IF(IC.GE.KK) GO TO 4
      MINROW=I
      KK=IC
    4 CONTINUE
C-----------------------------------------------------------------------
C      INTERCHANGE ROWS. -- MINROW WITH LL
C-----------------------------------------------------------------------
      LM=IBANDW(LL)
      M=MINROW
      DO 135 I=1.LM
      NNCOL(I)=NCCL(M.I)
      NCOL(M.I)=NCOL(LL.I)
      AA(I)=A(M,I)
      A(M.I)=A(LL.I)
  135 CONTINUE
      SAVE=B(LL)
      B(LL)=B(M)
      B(M)=SAVE
      IBANDW(LL)=IBANDW(M)
      IBANDW(M)=LM
C-----------------------------------------------------------------------
C     FIND BIG 'A' IN MINRCW
C-----------------------------------------------------------------------
      NC=IBANDW(LL)
      NNCOL(NC+1)=0
      MINROW=LL
      A1=0.D0
      I=MINRCW
      DO 30 J=1.NC
      AAA=DABS(AA(J))
      IF(AAA.LT.A1) GO TC 30
      A1=AAA
      IY=J
   30 CONTINUE
      MINCOL=NNCOL(IY)
      NPIV(LL)=MINCOL
C-----------------------------------------------------------------------
C     NORMALIZE THE MINROW
C-----------------------------------------------------------------------
      X=AA(IY)
      DO 6 J=1,NC
      AA(J)=AA(J)/X
      NCOL(MINROW,J)=NNCOL(J)
    6 A(MINROW,J)=AA(J)
      B(MINRCW)=B(MINROW)/X
      A(MINROW,IY)=1.DO
      AA(IY)=1.DO
C-----------------------------------------------------------------------
C     FIND THE ROWS WHICH CCNTAIN THE MINCOL
C-----------------------------------------------------------------------
      LY=LL+1
      DO 7 I=LY,NPT
      IF(IBANDW(I).EQ.0) GO TO 7
      NC=IBANDW(I)
      DO 9 J=1,NC
      IF(NCOL(I,J)-MINCCL) 9,10,7
C-----------------------------------------------------------------------
C     IF NCOL(I,J) IS EQUAL TO MINCOL THEN THE RCW CONTAINS THE
C     VARIABLE.  NOPROW = THE ROW BEING OPERATED ON.
C-----------------------------------------------------------------------
   10 NOPROW=I
      JKOP=1
      JKPI=1
      C=-A(NOPROW,J)
      B(NOPROW)=B(MINROW)*C+B(NOPROW)
   11 CONTINUE
      IF(NNCOL(JKPI).EQ.0) GO TO 7
      IF(NCOL(NOPROW,JKOP).EQ.0) GO TO 12
      IF(NNCOL(JKPI)-NCOL(NOPROW,JKOP)) 12,13,14
   12 IBANDW(I)=IBANDW(I)+1
      IF(MAXCOL.LT.IBANDW(I)) MAXCOL=IBANDW(I)
      II=IBANDW(I)
      JKL=JKCP+1
   17 IX=II-1
      A(NOPRCW,II)=A(NOPROW,IX)
      NCOL(NOPROW,II)=NCOL(NOPROW,IX)
      II = IX
      IF(IX.GE.JKL) GO TO 17
      A(NOPROW,JKOP)=AA(JKPI)*C
      NCOL(NOPROW,JKOP)=NNCOL(JKPI)
      IX=NCOL(NOPROW,JKOP)
      GO TO 16
C-----------------------------------------------------------------------
C     MINROW AND THE ROW BEING CONSIDERED CONTAIN THIS ELEMENT.
C     SHIFTING OF BOTH ROWS IS DONE AND NOPROW IS CPERATED.
C-----------------------------------------------------------------------
   13 IX=NCOL(NOPROW,JKOP)
      IF(IX.EQ.MINCOL) GO TO 18
      X=AA(JKPI)*C+A(NOPROW,JKOP)
      A(NOPRCW,JKOP)=X
C-----------------------------------------------------------------------
C     TEST TO SEE IF ANY OTHER ELEMENTS WERE ELIMINATED OTHER THAN
C     MINCOL IN THE NOPROW
C-----------------------------------------------------------------------
       ATEST=DABS(X)-ZTEST
      IF(ATEST.GT.0.D0) GO TO 16
   18 IBANDW(NOPROW)=IBANDW(NOPROW)-1
      IF(IBANDW(NCPROW)) 20,20,21
   20 WRITE(6,601) MINROW,MINCOL,NOPROW
  601 FORMAT(1X,' MATRIX IS SINGULAR ',/2X,'  MINROW= ',I5,'  MINCOL= ',
     1I5,' RCW OPERATED= ',I5)
   21 IX=IBANDW(NCPROW)
      DO 22 NK=JKCP,IX
      A(I,NK)=A(I,NK+1)
   22 NCOL(I,NK)=NCOL(I,NK+1)
      IX = IX + 1
      NCOL(I,IX)=0
      A(I.IX)=0.D0
      JKPI=JKPI+1
      GO TO 11
   16 JKPI=JKPI+1
C-----------------------------------------------------------------------
C     MINROW DOES NOT CONTAIN THIS ELEMENT. SHIFT NOPROW AND CONTINUE.
C-----------------------------------------------------------------------
   14 JKOP=JKOP+1
      GO TO 11
    9 CONTINUE
    7 CONTINUE
    8 CONTINUE
C-----------------------------------------------------------------------
C     ELIMINATE MINROW AND MINCOL FROM BEING CONSIDERED AGAIN
C-----------------------------------------------------------------------
    3 CONTINUE
      NPIV(NPT)=NCOL(NPT.1)
C-----------------------------------------------------------------------
C     BACK SUBSTITUTION
C-----------------------------------------------------------------------
      DO 150 I=1.NTT
  150 AA(I)=0.D0
      DO 138 I=1.NPT
      II=NPT-I+1
      LM=IBANDW(II)
      NP=NPIV(II)
      IF(NP.EQ.0) GO TO 138
      DO 139 J=1.LM
      NN=NCOL(II.J)
      IF(NN.EQ.0) GO TO 139
      IF(NN.EQ.NP) GO TO 140
      B(II)=B(II)-AA(NN)*A(II.J)
      GO TO 139
  140 IJ=J
  139 CONTINUE
      AA(NP)=B(II)/A(II.IJ)
  138 CONTINUE
C----------------------------------------------------------------------
C     NPT IS AGAIN MADE EQUAL TO ORIGINAL NPT TO AVOID ERROR IN DATA OF
C     MAIN PROGRAM
C----------------------------------------------------------------------
      NPT=NTT
C----------------------------------------------------------------------
C     STORE THE SOLUTION IN THE 'B' VECTOR
C----------------------------------------------------------------------
      DO 23 I=1,NPT
      B(I)=AA(I)
   23 CONTINUE
      RETURN
      END
C
C----------------------------------------------------------------------
C     SUBROUTINE EQSOLV, AN EQUATION SOLVER WHICH USES COMPRESSED
C     MATRICES (STORING NON-ZERO COEFFICIENTS IN ONE MATRIX AND
C     THEIR CORRESPONDING COLUMN IDENTIFICATIONS IN ANOTHER MATRIX).
C----------------------------------------------------------------------
C
      SUBROUTINE EQSOLV(NPT,NNN,ZTEST)
C
C----------------------------------------------------------------------
C      NPT    = MAXIMUM NUMBER OF ROWS IN SYSTEM OF EQUATIONS
C     NNN     = MAXIMUM NUMBER OF NON-ZERO COEFF. IN ANY ROW
C     A       = MATRIX CONTANING NON-ZERO COEFF. OF SYSTEM OF EQUATIONS
C     B       = CONSTANT VECTOR OF EQUATIONS TO BE SOLVED
C             = ALSO USED FOR RETURING THE SOLUTION
C     IBANDW  = NUMBER OF NON-ZERO COEFF IN A GIVEN ROW
C     AA      = SINGLE DIMENSIONED ARRAY USED FOR PIVOTAL ROW
C     NNCOL   = SINGLE DIMENSIONED ARRAY USED FOR COLUMN INDEX OF
C               PIVOTAL ROW
C     NPIV    = SINGLE DIMENSIONED ARRAY TO STORE THE PIVOTAL
C               COLUMN INDEX
C     NCOL    = MATRIX CONTAINING INDICES OF NON-ZERO COEFF. OF A
```

```
C-----------------------------------------------------------------------
C
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION AA(125),NPIV(125),IBANDW(125),NNCOL(125)
      COMMON TEMPD(11),PRESSD(6),VISCOS(11,6)
      COMMON FLSPH(11,6),FLTHC(11,6),RHO(125),XMEW(125),SPEHET(125)
      COMMON HO,BETAT,BETAP,TEMREF,PREREF,THRCCN(125)
      COMMON PFESS(125),TEMP(125),A(125,100),NCCL(125,100),B(125)
C-----------------------------------------------------------------------
C     INITALIZE THE BANDWIDTH COUNTER
C-----------------------------------------------------------------------
      MAXCOL=0
      DO 100 I=1,NPT
      IBANDW(I)=0
  100 CONTINUE
      NBANDW=NNN
      DO 1 I=1,NPT
      DO 2 J=1,NNN
      NC=NCOL(I,J)
      IF(NC.EQ.0) GO TO 719
      IBANDW(I)=J
    2 CONTINUE
C-----------------------------------------------------------------------
C     CHECK FOR RCWS WHICH HAVE ALL ZERO COEFFICIENTS
C-----------------------------------------------------------------------
  719 IF(J.NE.1) GO TO 1
      WRITE(6,600) I
  600 FORMAT(5X,'ALL THE COEFICIENTS CF ROW ',I5,' ARE ZERCS. TO AVCIC
     1 SINGULARITY AND AN INVALID INDEX. NCOL(I,J) IS MADE EQUAL TO I
     2 AND A(I,J) IS MADE EQUAL TO 1')
      NCOL(I,J) = I
      A(I,J)=1.D0
    1 CONTINUE
      LX=NPT-1
      DO 3 LL=1,LX
C-----------------------------------------------------------------------
```

```
C      FIND THE ROW WITH MINIMUM BANDWIDTH
C-----------------------------------------------------------------------
       KK=100000
       DO 4 I=LL.NPT
       IC=IBANDW(I)
       IF(IC.LE.0) GO TO 4
       IF(IC.GE.KK) GO TO 4
       MINROW=I
       KK=IC
     4 CONTINUE
C       INTERCHANGE ROWS. -- MINROW WITH LL
C-----------------------------------------------------------------------
C-----------------------------------------------------------------------
       LM=IBANDW(LL)
       M=MINRCW
       DO 135 I=1.LM
       NNCOL(I)=NCCL(M.I)
       NCOL(M.I)=NCOL(LL.I)
       AA(I)=A(M.I)
       A(M.I)=A(LL.I)
   135 CONTINUE
       SAVE=B(LL)
       B(LL)=B(M)
       B(M)=SAVE
       IBANDW(LL)=IBANDW(M)
       IBANDW(M)=LM
C-----------------------------------------------------------------------
C      FIND BIG 'A' IN MINRCW
C-----------------------------------------------------------------------
       NC=IBANDW(LL)
       NNCOL(NC+1)=0
       MINROW=LL
       A1=0.D0
       I=MINRCW
       DO 30 J=1.NC
       AAA=DABS(AA(J))
```

```
      IF(AAA.LT.A1) GO TO 30
      A1=AAA
      IY=J
   30 CONTINUE
      MINCOL=NNCOL(IY)
      NPIV(LL)=MINCOL
C-----------------------------------------------------------------------
C     NORMALIZE THE MINROW
C-----------------------------------------------------------------------
      X=AA(IY)
      DO 6 J=1,NC
      AA(J)=AA(J)/X
      NCOL(MINROW,J)=NNCOL(J)
    6 A(MINRCW,J)=AA(J)
      B(MINRCW)=B(MINROW)/X
      A(MINROW,IY)=1.D0
      AA(IY)=1.D0
C-----------------------------------------------------------------------
C     FIND THE ROWS WHICH CONTAIN THE MINCCL
C-----------------------------------------------------------------------
      LY=LL+1
      DO 7 I=LY,NPT
      IF(IBANDW(I).EQ.0) GO TO 7
      NC=IBANDW(I)
      DO 9 J=1,NC
      IF(NCOL(I,J)-MINCCL) 9,10,7
C-----------------------------------------------------------------------
C     IF NCOL(I,J) IS EQUAL TO MINCOL THEN THE ROW CONTAINS THE
C     VARIABLE.  NOPROW = THE ROW BEING OPERATED ON.
C-----------------------------------------------------------------------
   10 NOPROW=I
      JKOP=1
      JKPI=1
      C=-A(NCPROW,J)
      B(NOPRCW)=B(MINROW)*C+B(NOPROW)
   11 CONTINUE
```

```
      IF(NNCOL(JKPI).EQ.0) GO TO 7
      IF(NCOL(NOPROW,JKOP).EQ.0) GO TO 12
      IF(NNCOL(JKPI)-NCOL(NOPROW,JKOP)) 12,13,14
   12 IBANDW(I)=IBANDW(I)+1
      IF(MAXCOL.LT.IBANDW(I)) MAXCCL=IBANDW(I)
      II=IBANDW(I)
      JKL=JKCP+1
   17 IX=II-1
      A(NOPROW,II)=A(NOPROW,IX)
      NCOL(NOPROW,II)=NCOL(NOPROW,IX)
      II = IX
      IF(IX.GE.JKL) GO TO 17
      A(NOPRCW,JKCP)=AA(JKPI)*C
      NCCL(NCPROW,JKOP)=NNCOL(JKPI)
      IX=NCOL(NOPROW,JKOP)
      GO TO 16
C-----------------------------------------------------------------------
C     MINROW AND THE ROW BEING CONSIDERED CCNTAIN THIS ELEMENT.
C     SHIFTING OF BOTH ROWS IS DONE AND NOPROW IS OPERATED.
C-----------------------------------------------------------------------
   13 IX=NCOL(NOPROW,JKOP)
      IF(IX.EQ.MINCOL) GO TO 18
      X=AA(JKPI)*C+A(NOPROW,JKOP)
      A(NOPROW,JKOP)=X
C-----------------------------------------------------------------------
C     TEST TO SEE IF ANY OTHER ELEMENTS WERE ELIMINATED CTHER THAN
C     MINCOL IN THE NOPROW
C-----------------------------------------------------------------------
       ATEST=DABS(X)-ZTEST
      IF(ATEST.GT.0.D0) GO TO 16
   18 IBANDW(NCPRCW)=IBANDW(NOPROW)-1
      IF(IBANDW(NCPROW)) 20,20,21
   20 WRITE(6,601) MINROW,MINCCL,NOPRCW
  601 FORMAT(1X,' MATRIX IS SINGULAR ',/2X,'  MINRCW= ',I5,'  MINCOL= ',
     1I5,' RCW OPERATED= ',I5)
   21 IX=IBANDW(NCPROW)
```

```
      DO 22 NK=JKCP.IX
      A(I.NK)=A(I.NK+1)
   22 NCOL(I.NK)=NCOL(I.NK+1)
      IX = IX + 1
      NCOL(I.IX)=0
      A(I.IX)=0.D0
      JKPI=JKPI+1
      GO TO 11
   16 JKPI=JKPI+1
C-----------------------------------------------------------------------
C     MINROW DOES NOT CONTAIN THIS ELEMENT. SHIFT NOPROW ANC CONTINUE.
C-----------------------------------------------------------------------
   14 JKCP=JKOP+1
      GO TO 11
    9 CONTINUE
    7 CONTINUE
    8 CONTINUE
C-----------------------------------------------------------------------
C     ELIMINATE MINROW AND MINCOL FROM BEING CCNSIDERED AGAIN
C-----------------------------------------------------------------------
    3 CONTINUE
      NPIV(NPT)=NCOL(NPT.1)
C-----------------------------------------------------------------------
C     BACK SUBSTITUTION
C-----------------------------------------------------------------------
      DO 137 I=1.NPT
  137 AA(I)=0.D0
      DO 138 I=1.NPT
      II=NPT-I+1
      LM=IBANDW(II)
      NP=NPIV(II)
      IF(NP.EQ.0) GO TO 138
      DO 139 J=1.LM
      NN=NCOL(II.J)
      IF(NN.EC.0) GO TO 139
      IF(NN.EQ.NP) GO TO 140
      B(II)=B(II)-AA(NN)*A(II,J)
      GO TO 139
  140 IJ=J
  139 CONTINUE
      AA(NP)=B(II)/A(II,IJ)
  138 CONTINUE
C-----------------------------------------------------------------------
C     STORE THE SCLUTION IN THE 'B' VECTOR
C-----------------------------------------------------------------------
      DO 23 I=1,NFT
      B(I)=AA(I)
   23 CONTINUE
      RETURN
      END
```

APPENDIX D

SOME VECTOR OPERATIONS

SOME VECTOR OPERATIONS

Given the vectors

$$\overline{A} = a_x\overline{i} + a_y\overline{j} + a_z\overline{k} = \begin{Bmatrix} a_x \\ a_y \\ a_z \end{Bmatrix}$$

and

$$\overline{B} = b_x\overline{i} + b_y\overline{j} + b_z\overline{k} = \begin{Bmatrix} b_x \\ b_y \\ b_z \end{Bmatrix}$$

a vector C perpendicular to both A and B may be defined by the vector cross product as

$$\overline{C} = \overline{A} \times \overline{B} = \begin{Bmatrix} a_y b_z - a_z b_y \\ a_z b_x - a_x b_z \\ a_x b_y - a_y b_x \end{Bmatrix}$$

If the vector $A = a_x\overline{i} + a_y\overline{j} + a_z\overline{k}$ makes angles α, β, and γ, respectively, with the positive x, y, and z axes, then cos α, cos β, cos γ, its direction cosines, are given by

$$\cos\alpha = a_x[a_x{}^2 + a_y{}^2 + a_z{}^2]^{-1/2}$$
$$\cos\beta = a_y[a_x{}^2 + a_y{}^2 + a_z{}^2]^{-1/2}$$
$$\cos\gamma = a_z[a_x{}^2 + a_y{}^2 + a_z{}^2]^{-1/2}$$

LIST OF SYMBOLS

Symbol	Definition	Units
A	Area	L^2
$\overline{A}$	Vector	
a	Vector coordinate	
$\overline{B}$	Vector	
b	Vector coordinate	
b	Flow layer thickness	L
$\overline{C}$	Vector	
C_p	Fluid specific heat	$F\ L\ M^{-1}\ deg^{-1}$
C	Unknown Galerkin solution parameters	
C_r	Rock specific heat	$F\ L\ M^{-1}\ deg^{-1}$
D	Domain	
det	Matrix determinant	
e	Elemental domain	
F	Function	
f	Known function of independent variables	
g	Acceleration of gravity	$L\ T^{-2}$
h	Elevation above a datum	L
h_d	Depth below top of flow layer	L
h'	Hydraulic head	L
I	Differential operator	
i	Node	
J	Jacobian matrix	
K	Effective thermal conductivity	$F\ T^{-1}\ deg^{-1}$

Symbol	Definition	Units
K_t	Horizontal thermal conductivity in flow layer	$F\ T^{-1}\ deg^{-1}$
K_c	Hydraulic conductivity	$L\ T^{-1}$
k	Specific permeability	L^2
k_1	Thermal conductivity of rock solids	$F\ T^{-1}\ deg^{-1}$
k_2	Thermal conductivity of saturating fluid	$F\ T^{-1}\ deg^{-1}$
L	Number of nodes in an element	
l_x, l_y, l_z	Direction cosines	
N_i	Basis function for node "i"	
$\overline{N}_i$	Alternate basis function for node "i"	
P	Fluid pressure	$F\ L^{-2}$
P_x	Pressure at distance "x"	$F\ L^{-2}$
P_o	Reference pressure	$F\ L^{-2}$
$P*$	Piezometric pressure	$F\ L^{-2}$
ΔP	Pressure change	$F\ L^{-2}$
Q	Strength of a mass source or sink	$M\ T^{-1}\ L^{-3}$
Q_w	Well flow rate	$M\ T^{-1}\ L^{-3}$
Q_{in}	Rate of mass inflow from source terms	$M\ T^{-1}\ L^{-3}$
Q_{out}	Rate of mass outflow from sink terms	$M\ T^{-1}\ L^{-3}$
q_x, q_y, q_z	Volumetric fluid fluxes (mass flow equation)	$L\ T^{-1}$
q_x, q_y, q_z	Heat fluxes (energy transport equation)	$F\ L^{-2}\ T^{-1}$

Symbol	Definition	Units
R	Residual error	
R	Radius	L
r	Rock subscript	
S	Surface boundary	
S	An integration parameter	
T	Temperature	deg
T_m	Transmissivity	$L^2 T^{-1}$
T_{inj}	Injection temperature	deg
T_p	Pumping temperature	deg
Tref	Aquifer reference temperature	deg
t	Time	
U	Dimensionless temperature $\frac{T - T_o}{T_i - T_o}$	
u	$\frac{R^2 S}{4T_m t}$	
V	Volume	L^3
v_i	Mass fluid flux in the "i" direction	$M L^{-2} T^{-1}$
W	Weighting function	
w	Water subscript	
X	$\frac{2x}{b}$	
x	Cartesian coordinate	L
y	Cartesian coordinate	L
z	Cartesian coordinate	L
α	Vertical compressibility of porous media	$L^2 F^{-1}$
α	$(\frac{K_r C_r P_r}{K_t C_t P_t})^{0.5}$	

Symbol	Definition	Units
β_p	Coefficient of fluid compressibil- compressibility	$L^2\ F^{-1}$
β_T	Coefficient of fluid thermal volume expansion	deg^{-1}
γ	$\frac{Q\ C_w\ \rho_w}{4\ K_t}$	
δ	Dirac Delta function	
ζ	Transformed local coordinate	
η	Transformed local coordinate	
θ	Porosity	
ϕ_i	Nodal value of a field variable	
$\overline{\phi}_i$	Approximation of a field variable	
μ	Fluid dynamic viscosity	$F\ T\ L^{-2}$
ξ	Transformed local coordinate	
ρ	Fluid density	$M\ L^{-3}$
ρ_o	Reference fluid density	$M\ L^{-3}$
ρ_t	Density of media material	$M\ L^{-3}$
ρ_w	Fluid density	$M\ L^{-3}$
ρ_r	Effective aquifer density	$M\ L^{-3}$
τ	$\frac{4\ K_t\ t}{C_t\ {}_t b^2}$	
Φ	Mass energy content	$F\ L^{-2}\ T^{-1}$
∇	del operator	
Δ	Differential	
ν	$\frac{Q\ C_{w\ w}}{4\ \ bK_t}$	
ω	$\frac{2R}{b}$	