LA--11224-MS

DE88 005902

*FEHM: Finite Element Heat
and Mass Transfer Code*

*George Zyvoloski
Zora Dash
Sharad Kelkar*

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

92

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency Thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

FEHM: FINITE ELEMENT HEAT AND MASS TRANSFER CODE

by

George Zyvoloski, Zora Dash, and Sharad Kelkar

ABSTRACT

The finite element heat and mass (FEHM) transfer code is a computer code developed to simulate geothermal and hot dry rock reservoirs. It is also applicable to natural-state studies of geothermal systems and ground-water flow. It solves the equations of heat and mass transfer for multiphase flow in porous and permeable media using the finite element method. The code also has provisions for a noncoupled tracer; that is, the tracer solutions do not affect the heat and mass transfer solutions. It can simulate two-dimensional, two-dimensional radial, or three-dimensional geometries. A summary of the equations in the model, the numerical solution procedure, and model verification and validation are provided in this report. A user's guide and sample problems are included in the appendices.

I. INTRODUCTION

Because geothermal systems are complex, computer codes that can simulate many of the problems encountered in geothermal reservoirs must be constructed to consider several factors; for example, the flow can be fracture dominated, and the reservoir is often made up of several geologic units. In addition, the fluid may be one phase or two. These considerations require a flexible code, and such flexibility is provided

by the finite element method. It allows the model to simulate joints, fractures, and geologic boundaries. Geothermal reservoirs and especially hot dry rock reservoirs often require a three-dimensional model. In hot dry rock reservoirs, the basic problem is fluid flow in a fracture with permeation from the fracture to the surrounding rock. If the fracture has two-dimensional flow, the problem is three dimensional. A three-dimensional computer code, to be efficient and thereby useful, must use solution techniques that require little computer storage but have very fast execution. The incomplete factorization method used in the code has both of these characteristics. In this report the flow equations are presented as well as the numerical techniques used to solve the equations. In addition, verification examples and user documentation are provided.

II. MATHEMATICAL FORMULATION

Detailed derivations of the governing equations for two-phase geothermal reservoirs have been presented by several investigators (Mercer et al. 1974, Mercer and Faust 1975, and Brownell et al. 1975, for example), and therefore only a brief development will be presented here.

Conservation of mass is expressed by the equation

$$\frac{\partial A_m}{\partial t} + \nabla \cdot \bar{f}_m + q_m = 0 \quad , \quad (1)$$

where the mass per unit volume A_m is given by

$$A = \phi(S_v \rho_v + S_l \rho_l) \quad (2)$$

and the mass flux $\bar{f}_m$ is given by

$$\bar{f}_m = \rho_v \bar{V}_v + \rho_l \bar{V}_l \quad . \quad (3)$$

Here ϕ is the porosity of the matrix, S_v and S_l are saturations, ρ_v and ρ_l are densities, and $\bar{V}_v$ and $\bar{V}_l$ are velocities with the subscripts v and l indicating quantities for the vapor phase or the liquid phase, respectively. Source or sink terms (that is, bores or reinjection wells) are represented by the term q_m .

Conservation of fluid-rock energy is expressed by the equation

$$\frac{\partial A_e}{\partial t} + \nabla \cdot \bar{f}_e + q_e = 0 \quad , \quad (4)$$

where the energy per unit volume A_e is given by

$$A_e = (1 - \phi) \rho_r u_r + \phi (S_v \rho_v u_v + S_l \rho_l u_l) \quad (5)$$

and the energy flux $\bar{f}_e$ is given by

$$\bar{f}_e = \rho_v h_v \bar{V}_v + \rho_l h_l \bar{V}_l - K \nabla T \quad . \quad (6)$$

Here the subscript r refers to the rock matrix; u_r , u_v , and u_l are specific internal energies; h_v and h_l are specific enthalpies; K is an effective thermal conductivity; T is the temperature; and q_e is the energy contributed from sources and sinks.

To complete the governing equations it is assumed that Darcy's Law applies to the movement of each phase (capillary pressure is neglected):

$$\bar{V}_v = - \frac{k R_v}{\mu_v} (\nabla p - \rho_v g) \quad (7)$$

and

$$\bar{V}_l = - \frac{k R_l}{\mu_l} (\nabla p - \rho_l g) \quad . \quad (8)$$

Here k is the permeability, R_v and R_l are the relative permeabilities, μ_v and μ_l are viscosities, p is the pressure, and g represents the acceleration due to gravity. (For simplicity, the equations are shown for an isotropic medium, though this restriction does not exist in the

computer code.) The relative permeabilities used in the code are a version of a form suggested by Corey (1954) and adopted by Mercer and Faust (1975):

$$R_v = (1 - S_l^{*2}) (1 - S_l^2) \quad (9)$$

and

$$R_l = S_l^{*4} \quad , \quad (10)$$

where $S_l^* = (S_l - S_{lr} - S_{vr}) / (S_l - S_{lr} - S_{vr})$ and S_{lr} , S_{vr} are the residual saturations at which the liquid phases respectively become immobile. The values $S_{lr} = 0.3$ and $S_{vr} = 0.05$ have been used by other authors and are used in this work.

Using Darcy's Law the basic conservation equations (1) and (4) can be rewritten

$$-\nabla \cdot (D_m \nabla p) + q_m + \frac{\partial}{\partial y} (D_{ml} \rho_l + D_{mv} \rho_v) + \frac{\partial A_m}{\partial t} = 0 \quad (11)$$

and

$$-\nabla \cdot (D_e \nabla p) + \nabla \cdot (K \nabla T) + q_e + \frac{\partial}{\partial y} (D_{el} \rho_l + D_{ev} \rho_v) + \frac{\partial A_e}{\partial t} = 0, \quad (12)$$

where Y is oriented in the direction of gravities. Here the transmissibilities D_m and D_e are given by

$$D_m = D_{ml} + D_{mv} \quad (13)$$

and

$$D_e = h_l D_{ml} + h_v D_{mv} \quad (14)$$

with

$$D_{ml} = \frac{k R_l \rho_l}{\mu_l}, \quad D_{mv} = \frac{k R_v \rho_v}{\mu_v} \quad . \quad (15)$$

The functional dependence of the thermodynamic and transport quantities in equations (11) and (12) on p and h is not given here; the reader is referred to Mercer and Faust (1975) for details.

The mixture enthalpy h is related to the other variables by the formula

$$h = (\rho_v h_v S_v + \rho_l h_l S_l) / (\rho_v S_v + \rho_l S_l) \quad . \quad (16)$$

The source and sink terms in equations (1) and (4) arise from bores, and if the total mass withdrawal q_m for each bore is specified, then the energy withdrawal q_e is determined as follows:

$$q_e = q_v h_v + q_l h_l \quad , \quad (17)$$

where

$$q = \sigma q_m, \quad q_l = (1 - \sigma) q_m$$

and

$$\sigma = 1 / (1 + \frac{\rho_l R_l \mu_v}{\rho_v R_v \mu_l}) \quad . \quad (18)$$

The form of equation (18) shows how important the relative permeabilities ratio R_l/R_v is in controlling the discharge composition. The limited field data available (Grant 1977) show that there may be significant differences between the relative permeability formula used here in equations (4) and (10) (and by other authors) and relative permeabilities in the field.

Besides simulating the flow of heat and mass, the code is also capable of modeling tracer transport. The tracer transport equation was developed in the same way as the mass and energy equations. The tracer equation is

$$\nabla \cdot (D_c \nabla p) + g \frac{\partial}{\partial y} (c_l D_{ml} \rho_l + c_v D_{mv} \rho_v) + \nabla \cdot (D' \nabla C) + \frac{\partial A_c}{\partial t} + q_c = 0, \quad (19)$$

where

$$D_c = c_l D_{ml} + c_v D_{mv} \quad . \quad (20)$$

Here c is the concentration of the chemical species and is expressed as a mass fraction of the total mass of fluid. The term D' is the dispersivity. The terms q_c and A_c are also related to quantities defined for the mass balance equation,

$$q_c = c_v q_v + c_l q_l \text{ and } A_c = \phi(c_v S_v \rho_v + c_l S_l \rho_l) \quad .$$

The tracer equation is not directly coupled with the mass and energy balance equations. After the combined mass and energy equations are solved, equation (19) is solved using the existing pressure field. Although the code allows for concentration change with temperature and pressure, present usage has been limited to nonreactive inert tracers.

III. FORMULATION OF FINITE ELEMENT EQUATIONS

The finite element equations are generated using the Galerkin formulations. For a detailed presentation of the finite element method the reader is referred to Zienkiewicz (1977). In this method the flow domain, Ω , is assumed to be divided into finite elements; and variables P and h , along with the accumulation terms A_m and A_e , are interpolated on each element:

$$P = [N]^T \{P\}, \quad h = [N]^T \{h\}, \quad A_m = [N]^T \{A_m\}, \quad \text{and} \quad A_e = [N]^T \{A_e\} \quad ,$$

where the $[N]$ are the shape functions.

These approximations are introduced in equations (11), (12), and (19), and the Galerkin formulation (described by Zienkiewicz and Parekh 1973) is applied. The following equations are derived:

$$[T_m(P, h)] \{P\} + C \frac{\partial A_m}{\partial t} + C \{q_m\} - \{R_m\} = \{F_m\} \quad (21)$$

and

$$[T_e(P, h)] \{P\} + C \frac{\partial A_e}{\partial t} + C \{q_e\} - \{R_e\} = \{F_e\} \quad , \quad (22)$$

where

$$T_{mij} = \int_{\Omega} \nabla N_i \cdot D_m^{UP} \nabla N_j dV \quad , \quad (23)$$

$$T_{eij} = \int_{\Omega} \nabla N_i \cdot D_e^{UP} \nabla N_j dV \quad , \quad (24)$$

$$C_{ij} = \int_{\Omega} N_i N_j dV_j \quad , \quad (25)$$

$$R_{mi} = \int_{\Omega} \frac{\partial N_i}{\partial y} \cdot N_j D_m^{UP} \rho_m dV_j \quad ,$$

and

$$R_{ei} = \int_{\Omega} \frac{\partial N_i}{\partial y} \cdot N_j D_m^{UP} \rho_e dV_j \quad .$$

Equations (23) and (24) need some comment. The terms D_m^{UP} and D_e^{UP} indicate an upstream-weighted transmissibility (Dalen 1974). This technique has worked well in the low-order elements (3-node triangle, 4-node quadrilateral), where the schemes resemble difference techniques. (Research is ongoing to determine the applicability of using upstream-weighted transmissibilities when using 8-node quadratic elements.) The upstream weighting is determined by comparing the velocities at the nodes i and j . The shape function coefficients are generated in a unique way that requires the integrations in equations (23) and (24) to be performed only once and the nonlinear coefficients to be separated from this integration. The reader is referred to Zyvoloski (1983) for more details.

The integration scheme used in this report is similar to that described by Young (1981). His implementation differs from common methods in that it uses Lobatto instead of Gauss integration. The net effect is that, while retaining the same order of integration accuracy (at least for linear and quadratic elements), there are considerably fewer nonzero terms in the resulting matrix equations. Figure 1 shows a comparison of the nodal connections for Lobatto and Gauss integration methods. It should be noted that these results hold on an orthogonal grid only. If a nonorthogonal grid were introduced, then additional nonzero terms would appear in the Lobatto quadrature method. Note also that the linear elements yield the standard 5- or 7-point difference scheme. The reader is referred to Young's paper for more details.

LINEAR

QUADRATIC

LOBATTO INTEGRATION

● CONVENTIONAL INTEGRATION

Figure 1. Comparison of nodal connections with conventional and Lobatto integrations.

IV. THE SOLUTION METHOD

The application of the finite element method to the partial differential equations (with finite differencing of the temporal terms) yields, in general, a system of nonlinear equations. At this point the incomplete LU decomposition techniques are employed. The presentation of the theory below follows that of Behie and Vinsome (1982). The matrix equation resulting from the finite element process is given by

$$[A]\{x\} = \{r\} \quad , \quad (26)$$

where $[A]$ is the solution matrix, $\{x\}$ is the solution vector, and $\{r\}$ is the residual.

For computation, $\{x\}$ is replaced by $\{x\}^{n+1} - \{x\}^n$ and $\{r\}$ is replaced by $\{r\} - [A]\{x\}^n$ where n is the iteration number. Thus, only changes in $\{x\}$ are calculated.

The matrix $[A]$ is partially factored into upper and lower triangular form:

$$[\bar{A}] = [L][U] \quad , \quad (27)$$

where $[\bar{A}] \neq [A]$ because, in general, $[L]$ and $[U]$ will contain only some of the steps of the elimination process. The amount of factorization is limited by computer storage considerations. Once factored the solution is easy to carry out as it consists of a forward and a backward substitution:

$$\{v\} = [U]^{-1} \{[L]^{-1} \{r\}\} \quad , \quad (28)$$

where $\{v\}$ is the approximate change in $\{x\}$. The following recursion scheme is used to refine the approximation:

$$\{v\}^{n+1} = [A]^{-1} \{r\}^n = [U]^{-1} \{[L]^{-1} \{r\}^n\} \quad (29)$$

$$\{x\}^{n+1} = \{x\}^n + \{v\}^{n+1} \quad (30)$$

$$\{r\}^{n+1} = \{r\}^n - [A]\{v\}^{n+1} \quad , \quad (31)$$

where n is the iteration index and $\{x\}^0 = 0$ because the original matrix equation is written in residual form [see paragraph following equation (26)]. The solution scheme, comprising equations (27) through (31), may be accelerated by overrelaxation procedures, but it is better to use orthogonalization-minimization methods. If the solution matrix were symmetric, conjugate gradient methods could be used. For nonsymmetric matrices, the ORTHOMIN procedure (Behie and Vinsome 1982) is used, although other minimization methods are available.

For comparison purposes, the simple overrelaxation procedure is discussed first. In the recursion scheme [equations (29-30)], equation (30) is replaced by $\{x\}^{n+1} = \{x\}^n + w\{v\}^{n+1}$, and equation (31) is replaced by $\{r\}^{n+1} = \{r\}^n - w[A]\{v\}^{n+1}$. Here w is the overrelaxation parameter and satisfies the relation $1 < w < 2$. After equation (31) is calculated, the procedure returns to equation (29) and continues until $\{r\}^{n+1}$ is reduced to some predetermined value. For many reservoir problems, convergence is extremely slow, and the orthominimization procedures provide a valuable alternative.

The ORTHOMIN procedure is

$$\{v\}^{n+1} = [\bar{A}]^{-1}\{r\}^n \quad (32)$$

and

$$\{q\}^{n+1} = \{v\}^{n+1} - \sum_{i=1}^n a_i^{n+1} \{q\}^i, \quad (33)$$

where i is the number of orthogonalizations and ranges from 1 to north.

$$\{x\}^{n+1} = \{x\}^n + w^{n+1} \{q\}^{n+1} \quad (34)$$

and

$$\{r\}^{n+1} = \{r\}^n - w^{n+1} [A]\{q\}^{n+1}, \quad (35)$$

where

$$a_i^{n+1} = \frac{([A]\{q\}^i, [A]\{v\}^{n+1})}{([A]\{q\}^i, [A]\{q\}^i)} \quad (36)$$

and

$$w^{n+1} = \frac{(\{r\}^n, [A]\{q\}^{n+1})}{([A]\{q\}^{n+1}, [A]\{q\}^{n+1})} \quad (37)$$

A mathematical theory similar to that for conjugate gradient methods (for example, showing a reduction in condition number with each iteration) is not available for ORTHOMIN. In basic terms the a_i^{n+1} in equations (33) and (36) insure that the $[A]\{q\}^i$ are orthogonal and that the w^{n+1} in equations (35) and (37) are such that the square of the $\{r\}^{n+1}$ is minimized. For a mathematical discussion of conjugate gradients, the reader is referred to Meijerink and Van der Vorst (1981). The only user-specified parameter "NORTH" is the number of orthogonalizations before restarting the ORTHOMIN procedure. The ORTHOMIN procedure requires storage of all previous steps. To minimize the storage required, the procedure is restarted after every "NORTH" iteration. The more orthogonalizations before restarting, the faster the convergence, but this benefit must be weighed against the greater storage and longer running time required for the additional orthogonalizations.

What has been described above is the naturally ordered incomplete factorization method. The incidence matrix is shown in Figure 2. It is often computationally advantageous to reorder the nodes, directly eliminate a portion of them, and use incomplete factorization to eliminate those remaining. This procedure is commonly known as red-black partitioning. To generate the red-black partition, all nodes are first labeled red except the first node which is labeled black. Then label of the next red node with no black neighbors is changed to black. The process continues through the original node list. Once the solution matrix is found, the generalized red-black ordering algorithm is applied to the matrix equation $[A]\{x\} = \{r\}$. Our description follows that of Behie and Forsythe (1984). The matrix equation is of the form

$$\begin{bmatrix} A_{RR} & A_R \\ A_B & A_{BB} \end{bmatrix} \begin{Bmatrix} x_R \\ x_B \end{Bmatrix} = \begin{Bmatrix} r_R \\ r_B \end{Bmatrix}, \quad (38)$$

Figure 2. Matrix band structures showing second order fill-in.

x - original matrix positions.
o - fill-in with order 2 factorization.

where A_{RR} and A_{BB} are diagonal matrices (A_{BB} only for some element types) and A_R and A_B are banded matrices. A_{RR} is normalized, and equation (38) becomes equivalent to

$$\begin{bmatrix} I & A_R' \\ A_B & A_{BB} \end{bmatrix} \begin{bmatrix} x_R \\ x_B \end{bmatrix} = \begin{bmatrix} r_R' \\ r_B \end{bmatrix} \quad (39)$$

A_B can be eliminated, and the system is written (Figure 3)

$$\begin{bmatrix} I & A_R' \\ 0 & A_{BB} - A_B A_R' \end{bmatrix} \begin{Bmatrix} X_R \\ X_B \end{Bmatrix} = \begin{Bmatrix} r_R' \\ r_B - A_B r_B' \end{Bmatrix}. \quad (40)$$

This system is completely uncoupled; we need only to solve

$$[A_{BB} - A_B A_R'] \{X_B\} = \{r_B - A_B r_B'\} \quad (41)$$

and use back substitution to find $\{X_R\}$.

The operations described in equations (38)-(41) are performed on the nonzero matrix elements only and are directed through a pointer system. For linear elements with Lobatto quadrature, half the unknowns are removed from the factorization process. The lower right corner of Figure 3 shows the RB connectivity matrix. The approximate matrix storage for the element types is shown in Table I. The differences between the 2-dimensional and 3-dimensional results are due to the noncyclic nature of the finite-element-produced solution matrices. For more details and some programming considerations, the reader is referred to Behie and Forsythe (1984).

Figure 3. Reduced block matrices with a RB ordered system.

TABLE I
APPROXIMATE NUMBER OF ELIMINATED VARIABLES
FOR DIFFERENT ELEMENT TYPES AND RED-BLACK ORDERING

Ordering	Element Type		Matrix Solution	ORTHOMIN
Natural	L-L	2D	5*NEQ	2*NEQ*(NORTH+1)
		3D	7*NEQ	2*NEQ*(NORTH+1)
	Q-L	2D	9*NEQ	2*NEQ*(NORTH+1)
		3D	13*NEQ	2*NEQ*(NORTH+1)
	L-G	2D	9*NEQ	2*NEQ*(NORTH+1)
		3D	27*NEQ	2*NEQ*(NORTH+1)
Red-Black	L-L	2D	5*(NEQ/2)	2*(NEQ/2)*(NORTH+1)
		3D	7*(NEQ/2)	2*(NEQ/2)*(NORTH+1)
	Q-L	2D	9*(5*NEQ/8)	2*(5*NEQ/8)*(NORTH+1)
		3D	13*(17*NEQ/24)	2*(17*NEQ/24)*(NORTH+1)
	L-Q	2D	9*(3*NEQ/4)	2*(3*NEQ/4)*(NORTH+1)
		3D	27*(7*NEQ/8)	2*(7*NEQ/8)*(NORTH+1)
Red-Black ²	L-L	2D	5*(NEQ/3)	2*(NEQ/3)*(NORTH+1)
		3D	7*(5*NEQ/12)	2*(5*NEQ/12)*(NORTH+1)
	Q-D	2D	9*(NEQ/2)	2*(NEQ/2)*(NORTH+1)
		3D	13*(11*NEQ/18)	2*(11*NEQ/18)*(NORTH+1)
	L-D	2D	9*(2*NEQ/3)	2*(2*NEQ/3)*(NORTH+1)
		3D	27*(5*NEQ/6)	2*(5*NEQ/6)*(NORTH+1)

NOTE: NORTH is the number of orthogonalizations.
NEQ is the number of equations.

An incomplete LU decomposition technique, presented earlier, is applied to the red-black-ordered system. The matrix $[A_{BB} - A_B A_R']$ is partially factored into upper and lower triangular form:

$$[\bar{A}] = [L] [U] \quad , \quad (42)$$

where $[\bar{A}]$ does not necessarily equal $[A_{BB} - A_B A_R']$ because, in general, $[L]$ and $[U]$ do not contain all the steps of elimination. The approximate solution is accelerated by the orthogonalization-minimization method, ORTHOMIN, described above.

V. EXAMPLE PROBLEMS

Example problems can demonstrate the capabilities of the FEHM code, and a few analytical solutions can be used to verify it. Some confidence can be gained by running problems with published solutions. A selection of the examples provided in this section will also be examined in Appendix B in which both input and output files are given.

A. Example 1

This example demonstrates code performance in a heat conduction calculation. A three-dimensional cube heated on the sides is shown in Figure 4, and the thermal properties are given in Table II. The sampling points for error computation are also shown in Figure 4. The performance of linear and quadratic elements is shown in Figure 5. The results show that the linear and quadratic elements both provide accurate solutions, but the quadratic elements are a little more efficient. The spatial discretization used to produce Figure 5 ranged from 3 to 9 nodes per dimension.

Figure 4. Solution domain for 3-D heat conduction example.

TABLE II
PARAMETERS FOR HEAT TRANSFER EXAMPLE

Parameter	Value
Thermal conductivity	2.7 W/m $\cdot$ °C
Rock density	2700 kg/m ³
Rock specific heat	1010 J/kg $\cdot$ °C
Block length	1 m
Block width	1 m
Block depth	1 m

Figure 5. Comparison of results for the 3-D heat conduction example.

B. Example 2

This problem is a model of a two-phase, highly permeable geothermal reservoir originally proposed by Toronyi and Farouq Ali (1977) and solved by a number of authors (Mercer and Faust 1975, Thomas and Pierson 1978). The model reservoir is shown in Figure 6. The input for the problem is presented in Table III. As with Thomas and Pierson, time steps of 10 days with an initial time step of 8.3 days were used. The final state corresponds to 19% of the original water mass removed. Figure 6 also shows the results obtained in this study using quadrilateral elements compared with the results obtained by Thomas and Pierson. Good agreement is evident. This is somewhat surprising considering the elements have a length/width ratio of 10.

Figure 6. Solutions domain and results for the Toronyi example.

TABLE III
PARAMETERS FOR TORONYI EXAMPLE

Parameter	Symbol	Value
Permeability	k	$9.869 \times 10^{-13} \text{ m}^2$
Thermal conductivity	κ	$1.730 \text{ W/(m}\cdot\text{°C)}$
Porosity	ϕ	0.05
Rock density	ρ_r	2563 kg/m^3
Rock specific heat	C_r	$1010 \text{ J/kg}\cdot\text{°C}$
Aquifer length	--	1828 m
Aquifer width	--	182.8 m
Initial water saturation	$s_{i,j}^o$	0.2
Initial pressure	p_{ij}^o	44.816 bar
Discharge	q_m	$0.082021 \text{ kg/(s}\cdot\text{m)}$

C. Example 3

The third problem is a model of a geothermal field in which recharge occurs across a constant pressure and enthalpy boundary. The input data are presented in Table IV. The model reservoir is depicted in Figure 7. This problem was proposed recently as a Department of Energy geothermal code test problem (Molloy 1982). The simulation is for 10 years. In this example the finite element (FE) quadrilaterals took 41 time steps to model the 10-year production, while the finite difference (FD) scheme (same grid) took 49 time steps. To get an idea of the temporal truncation error, a more accurate 284-step simulation was also performed. The pressure and temperature of the production and observation wells, as well as the outlet enthalpy, are presented in Figures 8-10. As can be seen from the figures, the FD and FE schemes plot very close to each other, but the FE scheme is closer to the presumably more accurate solution.

TABLE IV
PARAMETERS FOR DOE TEST PROBLEM

Parameter	Symbol	Value
Permeability	k	$2.5 \times 10^{-14} \text{ m}^2$
Thermal conductivity	κ	1 W/(m·°C)
Porosity	ϕ	0.35
Rock density	ρ_r	2563 kg/m ³
Rock specific heat	C_r	1010 J/kg·°C
Aquifer length	--	300 m
Aquifer width	--	200 m
Initial pressure	P^o_{ij}	36 bar
Discharge	q_m	0.05 kg/(s·m)

Initial temperature distribution
where r is the distance from the origin

$$T(x,y,o) = \begin{cases} 240^\circ\text{C} & r \leq 100 \text{ m} \\ 240 - 160 \left(\frac{r-100}{200} \right)^2 + 80 \left(\frac{r-100}{200} \right)^4 & 100 < r < 300 \text{ m} \\ 160^\circ\text{C} & r \geq 300 \text{ m} \end{cases} \text{ } ^\circ\text{C}$$

Figure 7. Solutions domain for DOE test problem.

Figure 8. Pressure histories for DOE test problem.

Figure 9. Temperature histories for DOE test problem.

Figure 10. Outlet enthalpy for DOE test problem.

APPENDIX A

AUTOMATIC MESH GENERATION GENMESH

The code GENMESH, which is designed to be used in conjunction with the finite element code FEHM, operates by dividing a user-defined solution space into a prescribed number of finite elements. In order to simplify geometric considerations, the code utilizes a "master space" where all elements are rectangles (or cubes) with unit sides. Results in the physical space, where the domain boundaries may be irregular, are obtained by using a one-to-one map between the two spaces. Isoparametric elements are used, where the shape functions define the coordinate mapping as well as provide the approximating basis functions. The mesh generation procedure allows the user to divide the global domain in the physical space into a number of subdomains termed "blocks." Each block is simply connected. Each block is mapped from its own "master space" using the desired shape functions. The mesh generator subdivides the "master space" corresponding to each block into elements. Any given block is subdivided into elements of a given type, although the elements need not be of the same size. Different blocks may use different types of elements. The reader is referred to An Introduction to Finite Element Computations by Hinton and Owen (pp. 328-346) for the details of the procedure and the algorithm. Note that material properties are specified at each node in FEHM (see Appendix B), unlike conventional finite element procedure.

A detailed description of the input to the code follows. The input must be placed in a file called MDAT. The output is found on a file called DAT7. The output consists of a list of nodal coordinates and elements in a suitable format for the code FEHM. Free format is used for the input.

Group 1

Title, up to 80 characters long

Group 2

NPOIN, NELEM, KX, KY, KZ, NDIME

NPOIN - total number of points used to define the solution space

NELEM - number of blocks used to define the solution space

KX - order of element in x-direction

KY - order of element in y-direction

KZ - order of element in z-direction

NDIME - number of dimensions of solution space (2 or 3)

Group 3 (NUMEL Block Times)

NUMEL, (LNODS(NUMEL,INOD), INODE = 1,20)

NUMEL - Block number

LNODS (NUMEL, INOD) - 20 (3D) or 8 (2D) nodes defining the block
NUMEL

Group 4

CORDX (I), I = 1, NPOIN

CORDX (I) - x-coordinates of points used to define the solution
space

Group 5

CORDY (I), I = 1, NPOIN

CORDY (I) - y-coordinates of points used to define the solution
space

Group 6 (only if NDIME = 3)

CORDZ (I), I = 1, NPOIN

CORDZ (I) - z-coordinates of points used to define the solution
space

NOTE : Groups 7-10 are repeated for each block.

Group 7

KBLOC, NDIVX, NDIVY, NDIVZ

KBLOC - block number

NDIVX - number of divisions in x-direction

NDIVY - number of divisions in y-direction

NDIVZ - number of divisions in z-direction

NOTE : The number of nodes in a direction = number of divisions + 1.

Group 8

WEITX (I), I = 1, NDIVX

WEITX (I) - weighting of the Ith division in the x-direction in
block KBLOC

Group 9

WEITY (I), I = 1, NDIVY

WEITY (I) - weighting of the Ith division in the y-direction in
block KBLOC

Group 10

WEITZ (I), I = 1, NDIVZ (only if NDIME = 3)

WEITZ (I) - weighting of the Ith division in the z-direction in
block KBLOC

APPENDIX B

MACROCOMMAND INPUT FOR FEHM

The finite element heat and mass FEHM transfer code contains a macrocontrol structure for data input that offers added flexibility to the input process. Blocks of data can be entered in any order, and any blocks unnecessary to a particular problem can be disregarded. All entries are free format, which adds flexibility, but requires that values be entered for all input variables (no assumed null values).

Although most input data are contained in the input deck, some information must first be entered from the terminal. Most of the information required before the input deck can be read pertains to input and output files and is prompted for from within the program. All files are of the form TAPEn, where n is the tape (unit) number. The terminal input follows:

(free format) INPT, IOUT, IREAD, ISAVE, IDC

INPT - unit number of the input file

IOUT - unit number of the file on which nodal information (pressures, enthalpies, etc.) is printed

IREAD - unit number of file on which initial values of pressure and enthalpy are found. (If IREAD = 0, initial values are generated from input data. See macrocontrol statement pres.)

ISAVE - unit number of file on which final time nodal information is printed (for restart purposes)

IDC - user subroutine parameter
if IDC = 0 the code makes no USER subroutine calls
if IDC $\neq$ 0 the code calls subroutine USER (user defined) at each time step

The remainder of the data is read from unit INPT.

The macrocommand structure makes use of a set of control statements recognized by the input module of the program. When a macrocontrol statement is encountered in the input deck, a certain set of data is expected and read from the input deck. In this way, the input is divided into separate, unordered blocks of data. The input deck is therefore a collection of macrocontrol statements, each followed by its associated data block, as the example deck shows. Note that, although the input is free format, macrocontrol statements must appear in the first four columns of a line.

Table B-I lists the macrocontrol statements with brief descriptions of the data associated with each. A detailed list of the macrocontrol statements and the input variables associated with each is given in Table B-II. In Appendix D examples of input files are given.

TABLE B-I
MACROCONTROL STATEMENTS FOR FEHM

Control Statement	Description
node	node numbers for output and time histories
cont	contour plot data
sol	solution specifications
init	initial value data
rock	rock density, specific heat, and porosity data
pres	initial pressure and enthalpy data
cond	thermal conductivity data
perm	permeability and velocity data
flow	flow data
time	time step and time of simulation data
ctrl	program control parameters
coord	node coordinate data
elem	element node data
trac	tracer data
stop	signals the end of input

TABLE B-II
INPUT DESCRIPTION FOR FEHM

Control statement node

Group 1 - M

M - number of nodes for which information will be printed on IOUT (see terminal input). If $M < 0$, pressure and temperature will be written for all nodes.

Group 2 - MN (1), MN (2) ... MN (M)

MN - node numbers for which information will be printed on IOUT, M nodes (see Group 1).

Group 3 - IJKP, IJKQ, IJ1, IJ2

IJKP - first node for time history of pressure, enthalpy, and temperature

IJKQ - second node for time history of pressure, enthalpy, and temperature

IJ1 - first node for time history of flow rate

IJ2 - second node for time history of flow rate

Control statement cont

Group - NDBM, CORDD

NDBM - normal direction of the contour plane (1 = X, 2 = Y, 3 = Z). The coordinate system is defined in Figure B-1.

CORDD - coordinate value of contour plane (m)

NOTE : A contour plot will be drawn when the time step changes (see control statement time), as well as at the beginning and end of the simulation.

Control statement sol

Group 1 - IADI, NTT, INTG

IADI - parameter used to specify dual porosity solution
 if IADI = 0 single porosity solution
 if IADI = 1 dual porosity solution

NTT - parameter that defines the type of solution required
 if NTT > 0 heat and mass transfer solution
 if NTT < 0 heat transfer solution

INTG - parameter that defines element integration type
 if INTG ≤ 0 Lobatto quadrature is used
 if INTG > 0 Gauss quadrature is used

Control statement init

Group 1 - PEIN, EIN, TIN1, GRAD1, DEPTH, TIN2, GRAD2, QUAD

PEIN - initial value of pressure (MPa). If initial values are read from IREAD (see terminal input), then this value is ignored. If gravity is present, this is the value of the pressure at node 1, and the other nodal pressures are adjusted by considering the hydraulic head.

EIN - initial value of enthalpy (MJ). If $EIN \leq 0$, then the initial enthalpies are calculated using the pressure and thermal gradient.

NOTE : The initial temperatures are set according to the following formulas only if $EIN < 0$. Otherwise, the initial temperatures are determined from PEIN and EIN.

$$0 \leq Z \leq \text{DEPTH} \quad T = \text{TIN1} + \text{GRAD } 1 \times Z$$

$$Z > \text{DEPTH} \quad T = \text{TIN2} + \text{GRAD } 2 \times Z + \text{QUAD} \times Z^2$$

TIN1 - defined in formulas above (°C)

GRAD1 - defined in formulas above (°C/m)

DEPTH - defined in formulas above (m)

TIN2 - defined in formulas above (°C)

GRAD2 - defined in formulas above (°C/m)

QUAD - defined in formulas above (°C/m²)

Group 2 - SS, SLR, SVR

SS - initial liquid saturation
 if SS = 0 saturation is calculated from
 initial pressure and enthalpy
 if SS > 0 initial enthalpy is calculated from
 initial pressure and saturation

SLR - residual liquid saturation

SVR - residual vapor saturation

NOTE : SLR and SVR refer to constants in the Corey representation of relative permeabilities.

Control statement rock

Group 1 - JA, JB, JC, DENRD, CPRD, PSD, VOLF

JA - a parameter used to assign rock density, rock specific heat, and porosity. JA is the first do loop number (i.e., DO 100 I = JA, JB, JC). When JA = 0, input for this control statement is terminated.

JB - second do loop parameter used to assign rock density, rock specific heat, and porosity

JC - third do loop parameter

DENRD - rock density (kg/m^3)

CPRD - rock specific heat (MJ/kg)

PSD - porosity

VOLF - volume factor, by which the calculated nodal volumes are multiplied

NOTE : The code proceeds to the next control statement when a line is encountered with JA = 0 or JA < 0 (calls subroutine USER).

Control statement pres

Group 1 - JA, JB, JC, PHRD, EIND

JA, JB, JC - defined as in control statement rock

PHRD - initial pressure (MPa)

EIND - initial enthalpy (MJ). If EIND < 0, then ABS (EIND) is interpreted as a temperature ($^{\circ}\text{C}$) and the enthalpy calculated accordingly.

NOTE : The initial values defined in control statement pres supersede all others.

Control statement cond

Group 1 - JA, JB, JC, THXD, THYD, THZD

JA, JB, JC - defined as in control statement rock

THXD - thermal conductivity (W/mK) in the x-direction

THYD - thermal conductivity (W/mK) in the y-direction

THZD - thermal conductivity (W/mK) in the z-direction

Control statement perm

Group 1 - JA, JB, JC, PNXD, PNYD, PNZD, VXD, VYD, VZD

JA, JB, JC - defined as in control statement rock

PNXD - permeability in the x-direction

PNYD - permeability in the y-direction

PNZD - permeability in the z-direction

VXD - velocity (m/s) in the x-direction, used for heat transfer solution only

VYD - velocity (m/s) in the y-direction

VZD - velocity (m/s) in the z-direction

Control statement flow

Group 1 - JA, JB, JC, SKD, ESKD, AIPED

JA, JB, JC - defined as in control statement rock

SKD - heat and mass source strength (kg/s), heat only (MJ/s). Positive value indicates flow out of the reservoir. See AIPED for additional details.

ESKD - enthalpy (MJ/kg) of fluid injected. If the fluid is flowing from the reservoir, then the in-place enthalpy is used. If $ESKD < 0$, then ABS (ESKD) is interpreted as a temperature and the enthalpy calculated accordingly. In heat only problems with $ESKD < 0$, the node is in contact with a large heat pipe that supplies heat to the node through an impedance AIPED so as to maintain its temperature near ABS (ESKD).

Large values (approximately 10^3) of AIPED are recommended.

AIPED - impedance parameter. If AIPED is nonzero, the code interprets SKD as a flowing wellbore pressure (MPa) with an impedance ABS (AIPED). If $AIPED < 0$, flow is only allowed out of the well. For heat only, AIPED thermal resistance.

NOTE : If the porosity of the node is zero, then there is only an enthalpy solution, and the code forms a source proportional to the enthalpy difference $E - ESKD$, where E is the in-place enthalpy and ESKD is a specified enthalpy. The source term is given by

$$Q = AIPED (E - ESKD) \quad .$$

Control statement time

Group 1 - DAY, TIMS, NSTEP, II, YEAR, MONTH

- DAY - initial time step size (days)
- TIMS - simulation maximum time (days)
- NSTEP - maximum number of time steps allowed
- II - printout interval for nodal information (pressure, enthalpy, etc.), as set up under control statement node
- YEAR - year that simulation starts
- MONTH - month that simulation starts

Group 2 - DIT1, DIT2, DIT3, ITC

- DIT1 - time (days) for time step change
- DIT2 - new time step size (days)
- DIT3 - implicitness factor for new time step (use 1 for backward Euler, 1.5 for second-order implicit scheme).
- ITC - new printout interval

NOTE : A contour plot will be drawn at each DIT1.

Control statement ctrl

Group 1 - MAXIT, EPM, EPE, NORTH, ICKLMT, IFACT, IRENUM

- MAXIT - maximum number of iterations allowed in either the overall Newton cycle or the inner cycle to solve for the corrections at each iteration
- EPM - tolerance for Newton cycle (nonlinear equation tolerance)
- EPE - tolerance for the linear equation solution routine
- NORTH - number of orthogonalizations done in the linear equation solver
- ICKLMT- parameter that specifies the number of times the code will decrease the time step size if needed to help convergence. This will occur if PMAX, EMAX, PMIN, EMIN (see Group 3), or MAXIT is exceeded.

IFACT - parameter that specifies the factorization mode of the Jacobian matrix. IFACT = 0 means that the solution matrix will be factored. IFACT $\neq$ 0 means that the previous factors will be used in the solution routine.

IRENUM- parameter that determines whether renumbering of the equations will occur. IRENUM = 0 means that no renumbering will occur. IRENUM $\neq$ 0 means that the equations in the solution routine will be renumbered according to the Variable Partial Minimum Degree Algorithm.

Group 2 - JA, JB, JC, IGAUS

JA, JB, JC - defined as in control statement rock

IGAUS - the order of partial Gauss elimination

Group 3 - AM, AS, GRAV, UPWGT, PMAX, EMAX, PMIN, EMIN

AM - parameter that determines the form of the relative permeability expression

if AM = 1	the standard Corey relationship is used (see text)
if AM = 2	a relative liquid permeability proportional to the 4th power of the liquid saturation is used with the vapor relative permeability equal to one minus the liquid permeability

AS - implicitness factor

if AS = 1	standard pure implicit formulation will be used
if AS = 1.5	a second-order implicit method will be used

GRAV - gravity (m/s^2)

UPWGT - value of upstream weighting ($0.5 \leq \text{UPWGT} \leq 1$)

PMAX - maximum pressure (MPa) allowed by the code before the time step size is decreased

EMAX - maximum enthalpy (MJ) allowed by the code before the time step size is decreased

PMIN - minimum pressure (MPa) allowed by the code before the time step size is decreased

EMIN - minimum enthalpy (MJ) allowed by the code before the time step size is decreased

Group 4 - IAMM, AIAA, DAYMAX

IAMM - maximum number of iterations for which the code will multiply the time step size

AIAA - time step multiplier

DAYMAX- maximum time step size (days)

Group 5 - ICNL

ICNL - parameter that specifies the geometry

if ICNL = 0 three - dimensional

if ICNL = 1 X - Y slice

if ICNL = 2 X - Z slice

if ICNL = 3 Y - Z slice

if ICNL = 4 X - Y radial slice

if ICNL = 5 X - Z radial slice

if ICNL = 6 Y - Z radial slice

Control statement coor

Group 1 - N

N - number of nodes in the grid

Group 2 - MB, CORD1, CORD2, CORD3

MB - node number. If the difference between the just-read value of MB and the previously read value of MB is greater than 1, intermediate values will be interpolated by the code.

CORD1 - x-coordinate (m) of node MB

CORD2 - y-coordinate (m) of node MB

CORD3 - z-coordinate (m) of node MB

Control statement elem

Group 1 - NS, NEI, KX, KY, KZ

NS - number of nodes per element

NEI - number of elements

KX - order of polynomial + 1 in the x-direction

KY - order of polynomial + 1 in the y-direction

KZ - order of polynomial + 1 in the z-direction

Group 2 - MB, NELM (1), NELM (2)... NELM (NS)

MB - element number. If the difference between the just-read value of MB and the previously read value of MB is greater than 1, intermediate values will be interpolated by the code.

NELM (1) - first node of element MB

NELM (2) - second node of element MB

NELM (NS) - last node of element MB

Control statement trac

Group 1 - AND, ALPL, ALPV, BETL, BETV, EPC, DAYCON, UPWGTA

AND - initial tracer concentration

ALPL - the coefficient in the velocity-dependent term in the diffusion coefficient term (If ALPL = 0, no velocity dependence.)

ALPV - same as ALPL, except for vapor phase

BETL - coefficient in the velocity-dependent diffusion term (i.e., $D_L = D_0 + V \cdot \text{BETL}$)

BETV - same as BETL, except for vapor phase

EPC - equation tolerance for tracer solution

DAYCON- time (days) at which the heat and mass transfer solution is not calculated and the tracer solution is calculated with constant (in-time) velocities

UPWGTA- upstream weighting term for the tracer solution

Group 2 - JA, JB, JC, TCXD, TCYD, TCZD

JA - first do loop parameter used to define tracer dispersion coefficients

if JA < 0 dispersion coefficients are input
 from USERC, a user-defined
 subroutine

if JA = 0 input for this group is terminated

JB - second do loop parameter used for defining tracer dispersion coefficients

JC - third do loop parameter

TCXD - constant dispersion coefficient in the x-direction

TCYD - constant dispersion coefficient in the y-direction

TCZD - constant dispersion coefficient in the z-direction

Group 3 - JA, JB, JC, ANQD

JA, JB, JC - defined as in Group 2

ANQD - initial concentration of tracer, which will supersede the value given by AND in Group 1. Note that if initial values of pressure and enthalpy are read from a file, then the code will also read initial values for the tracer from the file.

Group 4 - JA, JB, JC, CNSK, T1SK, T2SK

JA, JB, JC - defined as in Group 2

CNSK - injection concentration of injection node. If it is a production node, then the in-place concentration is used

T1SK - time (days) when tracer injection begins

T2SK - time (days) when tracer injection ends

NOTE : Injection nodes should be specified in control statement flow.

Control statement stop

NOTE : No input is associated with this control statement. It signals the end of input, and as such it always appears as the last line of an input deck.

Figure B-1. FEHM coordinate system.

APPENDIX C

THE POSTPROCESSOR TESPLOT

The code TESPLOT draws history and contour plots from data contained in files PLT and CONTOUR, respectively. The variables available for a time plot are listed in the following example, and the contour plot can plot temperature, pressure, velocity vectors, and concentration. It uses calls to the DISSPLA graphics package, but it can easily be modified to use other plotting packages. It is an interactive code with user input entered from the terminal. Usage is best illustrated by the following example in which computer responses will be indicated by lower case letters and explanatory comments will be in parentheses.

The executable file for TESPLOT is called XPLOT

The code is executed by typing

XPLOT

The prompt for this program is a question mark (?). After typing XPLOT the computer will respond with

NOTE: History plot data should be in local file PLT
 contour plot data should be in local file contour
 time plot: 0, contour plot: 1, end: <0 ?0

(For this example we ask for a history plot; if the user wants to escape the program, a negative input will do so.)

choose variable for time history:

read in ichos where:

```
ichos=0(all done)
ichos=1(enthalpy)
ichos=2(flowrate 1)
ichos=3(flowrate 2)
ichos=4(temperature 1)
ichos=5(temperature 2)
ichos=6(average outlet temperature)
ichos=7(pressure 1)
ichos=8(pressure 2)
ichos=9(average outlet pressure)
ichos=10(power)
ichos=11(concentration 1)
ichos=12(concentration 2)
ichos=13(ratio: tracer mass out/tracer mass in)
```

?5

(The temperature at well 2 is desired as a function of time; if you wish to exit the program, type "0".)

number of plot points 7

(The computer has found 7 time history points in file PLT.)

read conmax (usage: c=c/conmax)

?1

(This will not appear if a tracer is not being modeled; if a tracer is being modeled, then the value for conmax must be greater than zero.)

pmin 12.35 pmax= 12.40 daymin= 0.0000 daymax= 1.000

(pmin and pmax are the minimum and maximum values of the variable in question over the time step being observed; in this case, we see minimum and maximum temperature at well 2 from 0 to 0.1 days.)

do you want to nondimensionalize p ?

form: p(nd)=a(p-b), read in a,b

?1,0

(You may choose to nondimensionalize the plot by inputting the appropriate values for a and b as shown above; otherwise, the input should be as shown. The same holds for the next statement with respect to time.)

do you want to nondimensionalize time ?

form: days (nd)=a(days-b), read in a,b

?1,0

pmin 12.35 pmax= 12.40 daymin= 0.000 daymax= 1.000

read in pmin, pmax, daymin, daymax

?12.3 12.45 0 1

(The computer again echoes the upper and lower limits of the data, and the user is asked to input the limits which need to be seen. In this case, we surround the variable limits with some free space in order to more easily read the graph.)

semi-log or log-log plot ?

read in iq where

iq.lt.0 means semi-log

iq.eq.0 means linear

iq.gt.0 means log-log

?0

(This statement asks you to choose the desired type of graph. For semi-log or log-log plots, the number of cycles on the log scale (s) are adjusted to include the limits. The limits on the log scale (s) must not be less than zero. Here we ask for a linear plot.)

read in increments for time scale

read in increments for variable scale

read in lines/increment for time scale

read in lines/increment for variable scale

?0.2 0.03 2 3

(The increments are the incremental spacing for the labels on each of the axes, and the lines/increment create a grid of lines. We will have a graph with a label every 0.2 time segments, with a line in between each label, and with labels every 0.03 degrees with two equidistant lines between them.)

read in headings

one line title, one line x, one line y

?temp vs time

?days

?degrees c

(The program will automatically capitalize every letter in your headings.)

The instructions given above will produce the plot. After the plot is generated, a return command will take you back and ask you to choose another ichos value. An input of "0" will return you to the beginning of the program and will ask what type, if any, of plot you would next like to draw. For the sake of continuing this example, we will ask to draw a contour plot next. This stage could have initially been reached after inputting XPLOT.

NOTE: History plot data should be in local file PLT; contour plot data should be in local file contour.

time plot: 0, contour plot: 1, end:<0

?1

read in nci; where

nci=0 if no values are wanted on plotted contours

and nci>0 if values are to be printed on contour lines

?3

(An input for nci of a number greater than zero will produce a contour plot of solid lines, each labeled a number of times along the line with a letter of the alphabet. On the side of the plot, a key to the letters is generated denoting the value of each contour. A greater value for nci will generate fewer letters on each line. A good value with which to start is 3. An input of zero will generate a plot using solid, dashed, and dotted lines, with a key to the line types printed on the side.)

min value x= 0.00000

max value x= 1000.0

min value y= 0.00000

max value y= 1000.0

read in minx, max, miny, maxy

?0 1000 0 1000

(The code determines and prints the upper and lower bounds for the contour plane, and you are prompted to input the region of the plane in which you are interested. Here, we ask for the entire plane.)

days	tmin	tmax	pmin	pmax
0.0000	12.35	12.35	10.00	10.00
vxmax	vymax	cmin	cmax	
0.0000	0.0000	0.0000	0.0000	

read in contour spacing for t,p,c, scale

?0 0 0 0

(Here the code presents the first contour data set. Since it is at 0 days, there are no contours, as this particular example was initiated with conditions at equilibrium. The input given tells the program to skip to the next time set.)

pmax eq pmin: no plot

tmax eq tmin: no plot

cmax eq cmin: no plot

(The code confirms that no plot can be made from this data set.)

days	tmin	tmax	pmin	pmax
0.1000	12.34	12.37	10.00	10.55
vxmax	vymax	cmin	cmax	
0.1480e-04	0.3448e-02	0.0000	0.0000	

read in contour spacing for t,p,c, scale

?0 0.5 0 0

(The next contour data set shows that at 0.1 days contours exist for every parameter except tracer concentration, which is omitted because our example problem did not use a tracer. Here we choose to look at the pressure contour lines, each separated by 0.05 pressure units. The contour plot can only plot one set of contours at a time, but it will generate plots for each parameter where the contour and spacing are not zero. For example, an input of 0.005 0.05 0 0 would have first drawn a contour plot of temperatures, separated by 0.005 temperature units, and then a contour plot of pressures like those our original input generates. Copies of the two plots (Figure C-1 and Figure C-2) created by our example input are shown on the following page.)

Figure C-1. Example of temperature history plot generated by postprocessor TESPLIT.

Figure C-2. Example of pressure contour plot generated by postprocessor TESPLIT.

APPENDIX D

FEHM EXAMPLES

In this appendix input files for examples 1 and 2 in the text are given. The input to GENTA and FEHM are given and also the first page of output. For more details the reader is referred to the text.

A. Examples 1

This is the 3-D heat conduction example. The input file to GENTA is shown in Figure D-1, the input file to FEHM in Figure D-2, and the first page of output in Figure D-3.

B. Example 2

This is the model geothermal reservoir first studied by Toronyi and Farouq Ali (1977). The input file for GENTA is shown in Figure D-4, the input file for FEHM in Figure D-5, and the first page of output in Figure D-6.

```

***** 3-d model *****
20 1 3 3 3 3 0
1 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
0 .25 .5 .5 .5 .25 0 0
0 .5 .5 0
0 .25 .5 .5 .5 .25 0 0
0 0 0 .25 .5 .5 .5 .25
0 0 .5 .5
0 0 0 .25 .5 .5 .5 .25
0 0 0 0 0 0 0 0
.25 .25 .25
.5 .5 .5 .5 .5 .5 .5 .5
1 4 4 4
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1

```

Figure D-1. Input for preprocessor GENTA for 3-D heat conduction example.

```

***** 3-d test model(1,4*4*4)*****
node
  08
  001 003 011 013 063 2 4 6
  001 003 013 063
cont
  3      0.00
sol
  0  -1 -1
init
  10.0      0.0      250. 0. 0. 250. 0. 0.
  0 0 0
rock
  1 125 1 2700. 1000. .0000 1.0
  0 0 0 0. 0. 0. 0.
cond
  1 125 1 2.7e-00 2.7e-00 2.7e-00
  0 0 0 0. 0. 0.
perm
  1 125 1 1.e-30 1.e-30 1.e-30 0. 0. 0.
  0 0 0 0. 0. 0. 0. 0.
flow
  101 125 1 10.00 -100.00 1.e03
  5 125 25 10.00 -100.00 1.e03
  10 125 25 10.00 -100.00 1.e03
  15 125 25 10.00 -100.00 1.e03
  20 125 25 10.00 -100.00 1.e03
  25 125 25 10.00 -100.00 1.e03
  21 125 25 10.00 -100.00 1.e03
  22 125 25 10.00 -100.00 1.e03
  23 125 25 10.00 -100.00 1.e03
  24 125 25 10.00 -100.00 1.e03
  0 0 0 0. 0. 0.
time
  1.00 10.0 0010 05 1983 12
  0. 0. 0. 0
ctrl
  40 1000000000. 1.e-07 08 10 0 1
  1 125 1 01
  0 0 0 0
  1. 1.0 0.0 1.00 700. 2.7 0. 0.
  18 1.00 1.000
  0

```

Figure D-2. Input file for FEHM for 3-D heat conduction example.

coord
125

1	0.00000	0.00000	0.00000
2	.12500	0.00000	0.00000
3	.25000	0.00000	0.00000
4	.37500	0.00000	0.00000
5	.50000	0.00000	0.00000
6	0.00000	.12500	0.00000
7	.12500	.12500	0.00000
8	.25000	.12500	0.00000
9	.37500	.12500	0.00000
10	.50000	.12500	0.00000
11	0.00000	.25000	0.00000
12	.12500	.25000	0.00000
13	.25000	.25000	0.00000
14	.37500	.25000	0.00000
15	.50000	.25000	0.00000
16	0.00000	.37500	0.00000
17	.12500	.37500	0.00000
18	.25000	.37500	0.00000
19	.37500	.37500	0.00000
20	.50000	.37500	0.00000
21	0.00000	.50000	0.00000
22	.12500	.50000	0.00000
23	.25000	.50000	0.00000
24	.37500	.50000	0.00000
25	.50000	.50000	0.00000
26	0.00000	0.00000	.12500
27	.12500	0.00000	.12500
28	.25000	0.00000	.12500
29	.37500	0.00000	.12500
30	.50000	0.00000	.12500
31	0.00000	.12500	.12500
32	.12500	.12500	.12500
33	.25000	.12500	.12500
34	.37500	.12500	.12500
35	.50000	.12500	.12500
36	0.00000	.25000	.12500
37	.12500	.25000	.12500
38	.25000	.25000	.12500
39	.37500	.25000	.12500
40	.50000	.25000	.12500
41	0.00000	.37500	.12500
42	.12500	.37500	.12500
43	.25000	.37500	.12500
44	.37500	.37500	.12500
45	.50000	.37500	.12500
46	0.00000	.50000	.12500
47	.12500	.50000	.12500

Figure D-2. (Continued) Note coordinate and element data were generated by preprocessor GENTA.

48	.25000	.50000	.12500
49	.37500	.50000	.12500
50	.50000	.50000	.12500
51	0.00000	0.00000	.25000
52	.12500	0.00000	.25000
53	.25000	0.00000	.25000
54	.37500	0.00000	.25000
55	.50000	0.00000	.25000
56	0.00000	.12500	.25000
57	.12500	.12500	.25000
58	.25000	.12500	.25000
59	.37500	.12500	.25000
60	.50000	.12500	.25000
61	0.00000	.25000	.25000
62	.12500	.25000	.25000
63	.25000	.25000	.25000
64	.37500	.25000	.25000
65	.50000	.25000	.25000
66	0.00000	.37500	.25000
67	.12500	.37500	.25000
68	.25000	.37500	.25000
69	.37500	.37500	.25000
70	.50000	.37500	.25000
71	0.00000	.50000	.25000
72	.12500	.50000	.25000
73	.25000	.50000	.25000
74	.37500	.50000	.25000
75	.50000	.50000	.25000
76	0.00000	0.00000	.37500
77	.12500	0.00000	.37500
78	.25000	0.00000	.37500
79	.37500	0.00000	.37500
80	.50000	0.00000	.37500
81	0.00000	.12500	.37500
82	.12500	.12500	.37500
83	.25000	.12500	.37500
84	.37500	.12500	.37500
85	.50000	.12500	.37500
86	0.00000	.25000	.37500
87	.12500	.25000	.37500
88	.25000	.25000	.37500
89	.37500	.25000	.37500
90	.50000	.25000	.37500
91	0.00000	.37500	.37500
92	.12500	.37500	.37500
93	.25000	.37500	.37500
94	.37500	.37500	.37500
95	.50000	.37500	.37500
96	0.00000	.50000	.37500

Figure D-2. (Continued).

	97	.12500	.50000	.37500
	98	.25000	.50000	.37500
	99	.37500	.50000	.37500
	100	.50000	.50000	.37500
	101	0.00000	0.00000	.50000
	102	.12500	0.00000	.50000
	103	.25000	0.00000	.50000
	104	.37500	0.00000	.50000
	105	.50000	0.00000	.50000
	106	0.00000	.12500	.50000
	107	.12500	.12500	.50000
	108	.25000	.12500	.50000
	109	.37500	.12500	.50000
	110	.50000	.12500	.50000
	111	0.00000	.25000	.50000
	112	.12500	.25000	.50000
	113	.25000	.25000	.50000
	114	.37500	.25000	.50000
	115	.50000	.25000	.50000
	116	0.00000	.37500	.50000
	117	.12500	.37500	.50000
	118	.25000	.37500	.50000
	119	.37500	.37500	.50000
	120	.50000	.37500	.50000
	121	0.00000	.50000	.50000
	122	.12500	.50000	.50000
	123	.25000	.50000	.50000
	124	.37500	.50000	.50000
	125	.50000	.50000	.50000
elem				
8	64	2	2	2
1	1	2	6	7
2	2	3	7	8
3	3	4	8	9
4	4	5	9	10
5	6	7	11	12
6	7	8	12	13
7	8	9	13	14
8	9	10	14	15
9	11	12	16	17
10	12	13	17	18
11	13	14	18	19
12	14	15	19	20
13	16	17	21	22
14	17	18	22	23
15	18	19	23	24
16	19	20	24	25
17	26	27	31	32
18	27	28	32	33
			26	27
			27	28
			28	29
			29	30
			31	32
			32	33
			33	34
			34	35
			35	36
			36	37
			37	38
			38	39
			39	40
			41	42
			42	43
			43	44
			44	45
			46	47
			47	48
			48	49
			49	50
			51	52
			52	53
			53	54
			54	55
			55	56
			56	57
			57	58

Figure D-2. (Continued).

19	28	29	33	34	53	54	58	59
20	29	30	34	35	54	55	59	60
21	31	32	36	37	56	57	61	62
22	32	33	37	38	57	58	62	63
23	33	34	38	39	58	59	63	64
24	34	35	39	40	59	60	64	65
25	36	37	41	42	61	62	66	67
26	37	38	42	43	62	63	67	68
27	38	39	43	44	63	64	68	69
28	39	40	44	45	64	65	69	70
29	41	42	46	47	66	67	71	72
30	42	43	47	48	67	68	72	73
31	43	44	48	49	68	69	73	74
32	44	45	49	50	69	70	74	75
33	51	52	56	57	76	77	81	82
34	52	53	57	58	77	78	82	83
35	53	54	58	59	78	79	83	84
36	54	55	59	60	79	80	84	85
37	56	57	61	62	81	82	86	87
38	57	58	62	63	82	83	87	88
39	58	59	63	64	83	84	88	89
40	59	60	64	65	84	85	89	90
41	61	62	66	67	86	87	91	92
42	62	63	67	68	87	88	92	93
43	63	64	68	69	88	89	93	94
44	64	65	69	70	89	90	94	95
45	66	67	71	72	91	92	96	97
46	67	68	72	73	92	93	97	98
47	68	69	73	74	93	94	98	99
48	69	70	74	75	94	95	99	100
49	76	77	81	82	101	102	106	107
50	77	78	82	83	102	103	107	108
51	78	79	83	84	103	104	108	109
52	79	80	84	85	104	105	109	110
53	81	82	86	87	106	107	111	112
54	82	83	87	88	107	108	112	113
55	83	84	88	89	108	109	113	114
56	84	85	89	90	109	110	114	115
57	86	87	91	92	111	112	116	117
58	87	88	92	93	112	113	117	118
59	88	89	93	94	113	114	118	119
60	89	90	94	95	114	115	119	120
61	91	92	96	97	116	117	121	122
62	92	93	97	98	117	118	122	123
63	93	94	98	99	118	119	123	124
64	94	95	99	100	119	120	124	125
stop								

Figure D-2. (Continued).

time step 5
 1 4.0

year	1983	month	12	day of m	6days	5.000e+00			
cpu sec for step	0.2947e-01				total time	0.1528e+00			
node	p(mpa)	e(mj)	l sat	temp(c)	well dis	dis ent	r eq1	r eq2	
1	10.000	0.10	0.000	100.544	0.000e+00	0.000	0.0e+00	0.4e-07	
3	10.000	0.10	0.000	100.386	0.000e+00	0.000	0.0e+00	0.5e-07	
11	10.000	0.10	0.000	100.386	0.000e+00	0.000	0.0e+00	0.5e-07	
13	10.000	0.10	0.000	100.274	0.000e+00	0.000	0.0e+00	0.8e-07	
63	10.000	0.10	0.000	100.195	0.000e+00	0.000	0.0e+00	0.1e-06	
2	10.000	0.10	0.000	100.503	0.000e+00	0.000	0.0e+00	0.7e-07	
4	10.000	0.10	0.000	100.209	0.000e+00	0.000	0.0e+00	0.3e-07	
6	10.000	0.10	0.000	100.503	0.000e+00	0.000	0.0e+00	0.7e-07	

res mass0.000000 kg vap mass-.212595e-09 kq energy 33.7959 mj

net discharge 55.022 kg net energy discharge 0.00000 mj
 conservation errors:mass 0.00000 energy 0.80741e-01

this time step discharges : ,mass,enthalpy,power
 2.187e+00 kg 0.000e+00 mj 0.000e+00 mw

cumulative discharges : ,mass,enthalpy,avg power
 5.594e+01 kg 0.000e+00 mj 0.000e+00 mw

time step 10
 1 4.0

year	1983	month	12	day of m	11days	1.000e+01			
cpu sec for step	0.2947e-01				total time	0.3110e+00			
node	p(mpa)	e(mj)	l sat	temp(c)	well dis	dis ent	r eq1	r eq2	
1	10.000	0.10	0.000	100.001	0.000e+00	0.000	0.0e+00	0.7e-10	
3	10.000	0.10	0.000	100.001	0.000e+00	0.000	0.0e+00	0.1e-09	
11	10.000	0.10	0.000	100.001	0.000e+00	0.000	0.0e+00	0.1e-09	
13	10.000	0.10	0.000	100.001	0.000e+00	0.000	0.0e+00	0.1e-09	
63	10.000	0.10	0.000	100.000	0.000e+00	0.000	0.0e+00	0.2e-09	
2	10.000	0.10	0.000	100.001	0.000e+00	0.000	0.0e+00	0.1e-09	
4	10.000	0.10	0.000	100.000	0.000e+00	0.000	0.0e+00	0.5e-10	
6	10.000	0.10	0.000	100.001	0.000e+00	0.000	0.0e+00	0.1e-09	

res mass0.000000 kg vap mass-.438426e-12 kq energy 33.7501 mj

net discharge 65.189 kg net energy discharge 0.00000 mj
 conservation errors:mass 0.00000 energy 0.22342

this time step discharges : ,mass,enthalpy,power
 2.187e+00 kg 0.000e+00 mj 0.000e+00 mw

cumulative discharges : ,mass,enthalpy,avg power
 6.688e+01 kg 0.000e+00 mj 0.000e+00 mw

simulation ended: days 1.000e+01 timesteps 10

48 Figure D-3. Computer output for two time steps of 3-D heat conduction example.

```

***** toronyi model *****
08 1 2 2 1 2 0
1 1 2 3 4 5 6 7 8
0 914. 1828. 1828. 1828. 914. 0 0
0 0 0 91.4 182.8 182.8 182.8 91.4
1 7 7 1
.5 1 1 1 1 1 1 .5
.5 1 1 1 1 1 1 .5

```

Figure D-4. Input for preprocessor GENTA for Toronyi example.

```

**** toronyi example ****
node
  10
26 27 28 29 30 31 21 13 37 45
1 37 1 37
cont
  3      0.0
sol
  0      1    -1
init
      4.4816      0.3      250. 0. 0. 250. 0. 0.
.2 .05 .05
rock
  1 64      1      2563.      1010.      0.0500      1.0000
  0 0 0 0. 0. 0. 0. 0.
cond
  1 64      01 1.73e-00 1.73e-00 0.0e-00
  0 0 0 0. 0. 0. 0.
perm
  1 64      1 9.869e-13 9.869e-13 0.e-00 0. 0. 0.
0 0 0 0. 0. 0. 0. 0. 0.
flow
  029 029      1 0.082011 -025. 0.
  0 0 0 0. 0. 0
time
  8.3 78.3 0001      01 1985      07
8.3 10. 1.0 1
0. 0. 0. 0
ctrl
  40      1.e-7      1.e-08      08      40 0 0
  1 64      1      1
0 0 0 0
  1.      1.0 0.0 0.75 700.      2.7      .1      0.0
  40      1.00      010.00
1
coord
  64
      1      0.00000      0.00000      0.00000
      2      152.33      0.00000      0.00000
      3      457.00      0.00000      0.00000
      4      761.67      0.00000      0.00000
      5      1066.3      0.00000      0.00000
      6      1371.0      0.00000      0.00000
      7      1675.7      0.00000      0.00000
      8      1828.0      0.00000      0.00000
      9      0.00000      15.233      0.00000
     10      152.33      15.233      0.00000
     11      457.00      15.233      0.00000
     12      761.67      15.233      0.00000
     13      1066.3      15.233      0.00000

```

Figure D-5. Input file for FEHM for Toronyi example.

14	1371.0	15.233	0.00000
15	1675.7	15.233	0.00000
16	1828.0	15.233	0.00000
17	0.00000	45.700	0.00000
18	152.33	45.700	0.00000
19	457.00	45.700	0.00000
20	761.67	45.700	0.00000
21	1066.3	45.700	0.00000
22	1371.0	45.700	0.00000
23	1675.7	45.700	0.00000
24	1828.0	45.700	0.00000
25	0.00000	76.167	0.00000
26	152.33	76.167	0.00000
27	457.00	76.167	0.00000
28	761.67	76.167	0.00000
29	1066.3	76.167	0.00000
30	1371.0	76.167	0.00000
31	1675.7	76.167	0.00000
32	1828.0	76.167	0.00000
33	0.00000	106.63	0.00000
34	152.33	106.63	0.00000
35	457.00	106.63	0.00000
36	761.67	106.63	0.00000
37	1066.3	106.63	0.00000
38	1371.0	106.63	0.00000
39	1675.7	106.63	0.00000
40	1828.0	106.63	0.00000
41	0.00000	137.10	0.00000
42	152.33	137.10	0.00000
43	457.00	137.10	0.00000
44	761.67	137.10	0.00000
45	1066.3	137.10	0.00000
46	1371.0	137.10	0.00000
47	1675.7	137.10	0.00000
48	1828.0	137.10	0.00000
49	0.00000	167.57	0.00000
50	152.33	167.57	0.00000
51	457.00	167.57	0.00000
52	761.67	167.57	0.00000
53	1066.3	167.57	0.00000
54	1371.0	167.57	0.00000
55	1675.7	167.57	0.00000
56	1828.0	167.57	0.00000
57	0.00000	182.80	0.00000
58	152.33	182.80	0.00000
59	457.00	182.80	0.00000
60	761.67	182.80	0.00000
61	1066.3	182.80	0.00000
62	1371.0	182.80	0.00000
63	1675.7	182.80	0.00000
64	1828.0	182.80	0.00000

Figure D-5. (Continued) Note coordinate and element data were generated by preprocessor GENTA.

```

elem
  4, 49, 2, 2, 1
  1 1 2 9 10
  2 2 3 10 11
  3 3 4 11 12
  4 4 5 12 13
  5 5 6 13 14
  6 6 7 14 15
  7 7 8 15 16
  8 9 10 17 18
  9 10 11 18 19
 10 11 12 19 20
 11 12 13 20 21
 12 13 14 21 22
 13 14 15 22 23
 14 15 16 23 24
 15 17 18 25 26
 16 18 19 26 27
 17 19 20 27 28
 18 20 21 28 29
 19 21 22 29 30
 20 22 23 30 31
 21 23 24 31 32
 22 25 26 33 34
 23 26 27 34 35
 24 27 28 35 36
 25 28 29 36 37
 26 29 30 37 38
 27 30 31 38 39
 28 31 32 39 40
 29 33 34 41 42
 30 34 35 42 43
 31 35 36 43 44
 32 36 37 44 45
 33 37 38 45 46
 34 38 39 46 47
 35 39 40 47 48
 36 41 42 49 50
 37 42 43 50 51
 38 43 44 51 52
 39 44 45 52 53
 40 45 46 53 54
 41 46 47 54 55
 42 47 48 55 56
 43 49 50 57 58
 44 50 51 58 59
 45 51 52 59 60
 46 52 53 60 61
 47 53 54 61 62
 48 54 55 62 63
 49 55 56 63 64
stop

```

Figure D-5. (Continued).

```

initial m      0.293262e+07 s mass      304775.      energy      0.218785e+09

time for reading input,forming coefficients  1.03

time step      1

      4      10.7

year  1985 month      7 day of m      9days      8.300e+00
cpu sec for step      0.1903e+00 total time      0.1907e+00
node   p(mpa)      e(mj) l sat      temp(c)      well dis dis ent      r eq1      r eq2
 26      4.481      1.30 0.200      261.698      0.000e+00      0.000      0.2e-12      0.3e-12
 27      4.480      1.30 0.199      261.684      0.000e+00      0.000      -0.1e-12      -0.6e-12
 28      4.475      1.30 0.196      261.606      0.000e+00      0.000      -0.2e-11      -0.6e-11
 29      4.445      1.32 0.179      261.144      0.820e-01      2.791      0.5e-09      0.8e-10
 30      4.475      1.30 0.196      261.606      0.000e+00      0.000      -0.1e-11      -0.6e-11
 31      4.480      1.30 0.199      261.679      0.000e+00      0.000      -0.3e-12      -0.9e-12
 21      4.449      1.31 0.181      261.192      0.000e+00      0.000      0.2e-09      0.4e-09
 13      4.450      1.31 0.181      261.216      0.000e+00      0.000      0.1e-09      0.1e-09
 37      4.450      1.31 0.181      261.210      0.000e+00      0.000      0.2e-09      0.3e-09
 45      4.452      1.31 0.183      261.252      0.000e+00      0.000      0.8e-10      0.1e-09

res mass0.287381e+07 kg vap mass 305970.      kq energy0.218621e+09 mj

net discharge      58812.      kg net energy discharge      0.16414e+06 mj
conservation errors:mass      -0.45475e-11 energy      -0.10186e-09

this time step discharges : ,mass,enthalpy,power
      5.881e+04 kg      1.641e+05 mj      2.289e-01 mw

cumulative discharges : ,mass,enthalpy,avg power
      5.881e+04 kg      1.641e+05 mj      2.289e-01 mw

simulation ended: days  8.300e+00 timesteps      1

```

Figure D-6. Computer output for first time step of Toronyi example.

REFERENCES

- A. Behie and P.A. Forsythe, 1984, Incomplete factorization methods for fully implicit simulation for enhanced oil recovery, SIAM, J. Sci. Stat. Comput. 5, 543-561.
- A. Behie and P.K.W. Vinsome, 1982, Block iterative methods for fully implicit reservoir simulations, Soc. Pet. Eng. J., 658-668.
- D.H. Brownell, Jr., S.K. Garg, and J.W. Pritchett, 1975, Computer simulation of geothermal reservoirs, Paper SPE 5381, 45th California Regional Meeting of the Soc. Pet. Eng. of AIME, Ventura, California.
- A.T. Corey, 1954, The interrelation between gas and oil relative permeabilities, Prod. Mon. 19, 38-41.
- V. Dalen, 1974, Simplified finite-element models for reservoir flow problems, Soc. Pet. Eng. J. 14, 333-343.
- M.A. Grant, 1977, Permeability reduction factors at Wairakei, Paper 77-HT-52, AIChE-AIME Heat Transfer Conf., Salt Lake City, Utah.
- E. Hinton and D.R.J. Owen, 1979, An Introduction to Finite Element Computations, Pineridge Press, Swansea, Wales.
- J.A. Meijerink and H.A. Van der Vorst, 1981, An iterative solution method for linear systems of which the coefficients matrix is a symmetric M-matrix, Mathematics of Computation 31, 148-162.
- J.W. Mercer and C.R. Faust, 1975, Simulation of water- and vapor-dominated hydrothermal reservoirs, Paper SPE 5520, 50th Annual Fall Meeting of the Soc. Pet. Eng. of AIME, Dallas, Texas.
- J.W. Mercer, Jr., C.R. Faust, and G.F. Pinder, 1974, Geothermal reservoir simulation, Proc. Conf. Research for the Development of Geothermal Energy Resources, Pasadena, California.
- M.W. Molloy, 1982, Geothermal reservoir engineering code comparison project, Sixth Workshop on Geothermal Reservoir Engineering, Stanford University.
- L.K. Thomas and R.G. Pierson, 1978, Three dimensional reservoir simulation, Soc. Pet. Eng. J. 18, 151-161.
- R.M. Toronyi and S.M. Farouq Ali, 1977, Two-phase, two-dimensional simulation of a geothermal reservoir and wellbore system, Soc. Pet. Eng. J. 17, 171-183.
- L.C. Young, 1981, A finite element method for reservoir simulation, Soc. Pet. Eng. J. 21, 115-128.
- O.C. Zienkiewicz, 1977, The Finite Element Method, McGraw-Hill, London.

- O.C. Zienkiewicz and C.J. Parekh, 1973, Transient field problems - two and three dimensional analysis by isoparametric finite elements, Int. J. Numer. Methods Eng. 2, 61-70.
- G. Zyvoloski, 1983, Finite element methods for geothermal reservoir simulation, Int. J. Numer. Anal. Methods. Geomech. 7, 75-86.