Received by, OSTR
JUL 01 1991

FINAL REPORT

**Improved Computational Schemes for the Numerical Modeling of
Hydrothermal Resources in Wyoming**

by

Henry P. Heasler, Department of Geology and Geophysics
John H. George, Department of Mathematics
Myron B. Allen, Department of Mathematics
University of Wyoming
Laramie, Wyoming 82071

**DO NOT MICROFILM
COVER**

Prepared for
U. S. Department of Energy
Division of Geothermal Energy
Idaho Operations Office
Contract No. DE-FG07-88ID12738

May, 1990

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

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.

FINAL REPORT

Improved Computational Schemes for the Numerical Modeling of
Hydrothermal Resources in Wyoming

by

Henry P. Heasler, Department of Geology and Geophysics
John H. George, Department of Mathematics
Myron B. Allen, Department of Mathematics
University of Wyoming
Laramie, Wyoming 82071

Prepared for
U. S. Department of Energy
Division of Geothermal Energy
Idaho Operations Office
Contract No. DE-FG07-88ID12738

May, 1990

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

NOTICE

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, expressed 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. References 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.

ABSTRACT

A new method, the Conjugate Gradient Squared (CGS) solution technique, is shown to be extremely effective when applied to the finite-difference solution of conductive and convective heat transfer in geologic systems. The CGS method is compared to the Successive Over/Under Relaxation schemes, a version of the Gaussian elimination method, and the Generalized Minimum Residual (GMRES) approach. The CGS procedure converges at least ten times faster than the nearest competitor.

The model is applied to the Thermopolis hydrothermal system, located in northwestern Wyoming. Modeled results are compared with measured temperature-depth profiles and results from other studies. The temperature decrease from 72 °C to 54 °C along the crest of the Thermopolis anticline is shown to result from cooling of the geothermal fluid as it moves to the southeast. Modeled results show correct general trends, however, a time-varying three-dimensional model will be needed to fully explain the effects of mixing within the aquifers along the crest of the anticline and thermal affects of surface surface topography.

ACKNOWLEDGMENTS

The authors wish to thank the U. S. Department of Energy, Grant No. DE-GF07-88ID12738, for funding this project. The authors also wish to thank Howard P. Ross, University of Utah Research Institute, for his critical review of this report.

TABLE OF CONTENTS

	<u>Page</u>
ABSTRACT	iii
ACKNOWLEDGMENTS	iv
TABLE OF CONTENTS	v
LIST OF FIGURES	vi
LIST OF TABLES	vii
INTRODUCTION	1
Previous Research	1
Heat Transfer Equation	2
SOLUTION METHODS	4
Objectives of the Research Effort on Conjugate Gradient Methods	4
Organization	4
Recommendations	9
Newton's Law of Cooling	9
One Dimensional Case	9
Second Order Accuracy	11
Three dimensional CGS Algorithm	12
Recommendations	13
COMPUTER PROGRAM DESCRIPTION	15
Mechanics of Program	16
Program Output	18
Program Outline	19
Compiling and Running Program HTDOE	21
GEOLOGY OF THE THERMOPOLIS HYDROTHERMAL SYSTEM	24
THERMAL MODELING OF THE THERMOPOLIS HYDROTHERMAL SYSTEM	30
Improvements to the model	39
SUMMARY AND CONCLUSIONS	41
REFERENCES	43
APPENDICES	46
Appendix A, Program Listing	46
Appendix B, Input file HEATDAT	67
Appendix C, Input file CROSSDAT	69
Appendix D, Measured temperature-depth data for the Thermopolis area	76

LIST OF FIGURES

<u>Figure</u>	<u>Page</u>
1. Convergence of the CGS method	5
2. Convergence of the GMRES method	8
3. Convergence for the 3-dimensional CGS method	13
4. Location map for the Thermopolis Hydrothermal study area	26
5. Generalized geologic map for the Thermopolis hydrothermal system showing elevation of static water levels	27
6. Geologic cross-section used in the finite- difference modeling	31
7. Temperature-depth profile for well UWT-1	32
8. Temperature-depth profile for well UWT-2	33
9. Temperature-depth profile for well UWT-3	33
10. Calculated temperatures for no water flow and vertical flux side boundaries	34
11. Calculated temperature-depth profiles for Figure 10 for the left, center, and right portions of the cross-section	34
12. Calculated temperatures ($^{\circ}\text{C}$) for a Darcian velocity of $10^{-6} \text{ m sec}^{-1}$	35
13. Calculated temperatures ($^{\circ}\text{C}$) for a specified left temperature boundary and no water flow	37
14. Calculated temperatures ($^{\circ}\text{C}$) for a Darcian velocity of $10^{-8} \text{ m sec}^{-1}$	37
15. Calculated temperatures ($^{\circ}\text{C}$) for a Darcian velocity of $10^{-6} \text{ m sec}^{-1}$	38
16. Calculated temperatures ($^{\circ}\text{C}$) for a Darcian velocity of $10^{-5} \text{ m sec}^{-1}$	38
17. Calculated temperatures ($^{\circ}\text{C}$) for a Darcian velocity of $10^{-4} \text{ m sec}^{-1}$	39
18. Calculated temperature-depth profile for a Darcian velocity of $10^{-4} \text{ m sec}^{-1}$	39

LIST OF TABLES

<u>Table</u>	<u>Page</u>
1. Chemical data for springs and wells in the Thermopolis area	28
2. Thermal conductivity and thickness parameters used in the computer modeling	31

INTRODUCTION

The principal topics included in this report are: a discussion of the equations used to model the conductive and convective transport of heat; a discussion of the finite-difference solution methods used to solve the heat transfer equations; a discussion of the developed computer model; a general description of the Thermopolis hydrothermal system; new temperature-depth data; and results of computer numerical modeling of heat transfer in the Thermopolis hydrothermal system.

Previous Research

Hydrologists have long recognized the dependence between water flow and ground temperatures. Many researchers have tried to utilize the relationship between groundwater flow and temperature in an effort to determine water flow velocity and direction (Bredehoeft and Papadopoulos, 1965; Cartwright, 1970; Donaldson, 1962; Sorey, 1971; Keys and Brown, 1978; Kilty and Chapman, 1980; Smith and Chapman, 1983). The relationship between groundwater flow and temperature has also been used in an effort to determine hydrogeologic properties (Bair and Parizek, 1978).

Other researchers have considered improved computational schemes in the solution of highly convective fluid flow problems. Allen (1984) discusses upwinding techniques for fluid flow problems and Peyret and Taylor (1983) discuss the use of second order correct approximations applied to convection-diffusion equations using upwinding methods. These authors provide further

reference to the large body of literature on discrete methods appropriate for convection-dominated flows.

Heat Transfer Equation

The steady-state conductive and convective transport of heat may be expressed by the following differential equation (Hanel, et al., 1988):

$$\nabla \cdot (K \nabla T) + c\rho (\mathbf{V} \cdot \nabla T) = 0$$

where K represents thermal conductivity of the solid-fluid component, $\mathbf{V}$ represents the Darcian velocity of the fluid, T represents temperature, and c and ρ represent the specific heat and density of the fluid.

As discussed by Kilty et al. (1978), the validity of the above equation depends primarily on two conditions. First, it is valid only if the divergence of the velocity field and the viscous dissipative forces are negligible. Both of these conditions are satisfied in the macro-scale of groundwater flow considered. A second requirement for the validity of the above equation is that the solid and fluid phases must be coexisting media. This restriction is well satisfied if the pore spaces and fractures through which the flow takes place are much smaller than the distance over which there is resolvable change in temperature (Singh and Dybbs, 1974). If these restrictions are not met, the heat transfer equation will contain additional terms as discussed by Singh and Dybbs (1974) and Slattery (1972).

An interrelation exists between the mass flux velocity field and the temperature field. Thermal buoyancy can affect the

temperature which subsequently affects the buoyancy resulting in a non-linear equation. If thermal buoyancy is the primary driving force, free convection results. Forced convection results when the mass flux velocity field is solely the result of external pressures and can be determined independently of the temperature field. Thus, the mass flux may be determined from a relationship such as Darcy's law and the consequent temperature disturbances calculated.

SOLUTION METHODS

Several methods have been investigated to solve the equations of groundwater-temperature flow in a more efficient manner than the successive over relaxation (SOR) schemes (Anderson et al., 1984) currently in use. These methods include the generalized minimum residual (GMRES) (Saad and Shultz, 1986), the conjugate gradient squared (CGS) (Sonneveld, 1989), and the Gaussian elimination with partial pivoting. The Gaussian elimination was used to check the calculations in a low order prototypical problem. The conjugate gradient methods outperformed the SOR method by a large margin by reducing the number of iterations to calculate a solution. Also, the conjugate gradient method used significantly less central processing unit (cpu) time than the Gaussian elimination.

Objectives of the Research Effort on Conjugate Gradient Methods

Here, the objective is to use the conjugate gradient method to obtain a rapidly convergent solution to the large banded system produced by the difference approximation to the groundwater-temperature flow model. Since the solution by the conjugate gradient needs only the multiplication of the matrix by an approximate solution vector, the method is quite storage efficient and does not require the storage of the full matrix, only the bands. The method seems to be very effective on the problems on which we have tried it.

Organization

The application of conjugate gradient class solvers (Golub

and Van Loan, 1989) to groundwater-temperature flow problems is the main thrust of this portion of the research. The use of operator splitting (LU factorization) was not developed, as the conjugate gradient solver is essentially an exact solver with a small increase in memory over the approximate LU factorization scheme. To see how effective the LU splitting technique is, we used the L and U splitting as the preconditioner for the CGS method. Thus, one iteration of the CGS method compares to using the LU splitting. In the sample problem, it took eleven more iterations to achieve the desired precision using the CGS (see Figure 1).

Figure 1 Convergence of the CGS method.

Assume the groundwater-temperature flow model is replaced by an approximate difference model that has the following form:

$$Ax = b$$

To solve this equation, we use the CGS method with preconditioning; as is given by the following algorithm:

$$r_0 := P_L(b - Ax_0); \bar{r}_0 = r_0$$

$$q_0 := p_{-1} := 0$$

$$\rho_{-1} := 1; n := 0$$

If the error or residual is greater than a preset tolerance then begin;

$$\rho_n := \bar{r}_0^T r_n; \beta_n = \frac{\rho_n}{\rho_{n-1}}$$

$$u_n = r_n + \beta_n q_n;$$

$$p_n = u_n + \beta_n(q_n + \beta_n p_{n-1});$$

$$v_n := P_L A P_U p_n;$$

$$\sigma_n := \bar{r}_0^T v_n; \alpha_n = \frac{\rho_n}{\sigma_n};$$

$$q_{n+1} := u_n - \alpha_n v_n;$$

$$v_n := \alpha_n P_U (u_n + q_n);$$

$$x_{n+1} := x_n + v_n ;$$

$$r_{n+1} := r_n - P_L A v_n ;$$

$$n = n+1$$

Now update n , and iterate until the residual is below the tolerance. The P subscripted matrices are the lower and upper preconditioner, respectively.

As the matrices we are solving are highly nonsymmetric, the resulting oscillations in the solution method caused by complex eigenvalues can greatly slow the convergence history. The oscillations are shown in Figure 2. The conjugate gradient method systematically selects eigenvalues that will be 'filtered' out of the iteration. The generalized minimum residual idea is due to Saad and Schultz (1986). A nonlinear version of the generalized minimum residual method is discussed in (Huang et al., 1986), but it does not come with an effective algorithm.

We first applied the standard preconditioned conjugate gradient (CG) method for non-symmetric problems. This technique did not work because the symmetric part of the matrix had negative eigenvalues, which allowed the phenomenon of stagnation (Saad and Shultz, 1986) to take place. That is, the residual went to zero without converging to a solution! This required developing the CGS and GMRES methods for our class of problems. After some study, the CGS method was chosen because the GMRES scheme had a variable number of vectors to store, while the CGS method had a fixed number of additional vectors to deal with.

For the sample problem we used to check the methods, the solution by the CGS iteration converges to a solution in 12 iterations. The GMRES using 6 earlier computed solution vectors, converged to the same accuracy in 186 iterations. The computation time per step is about the same. Thus, on this problem, the CGS method is about 15 times faster than the GMRES (6) method. The two methods are compared in Figures 1 (CGS) and 2 (GMRES).

Figure 2 Convergence of the GMRES method.

The CGS method is quite sensitive to the choice of preconditioner, and further research in this area should prove useful as a way of speeding convergence. However, several of the ILU, MILU type preconditioners have been tried, and the method

which uses

$$A=L+D+U \approx (L+D) (I+D^{-1}U)$$

gives the least number of iterations in our problems. This could change with more complex preconditioners.

Recommendations

In the class of nonsymmetric matrices that we are solving, the CGS algorithm performed very well. The improvement over the SOR was 15 or 16 iterations for CGS compared to several thousand for SOR. As we have only preliminary results on the time varying case, we did not use the operator splitting LU scheme, but relied on the CGS solver. This method gives an "exact" solution in the sense that the method should converge in a finite number of steps. The Gaussian elimination method was used to verify the accuracy of the solution produced the CGS procedure.

Newton's Law of Cooling

Newton's law of cooling is used to define the boundary condition at the earth-air interface. The equations are;

$$K \nabla T \cdot \mathbf{n} = H(T - T_{\infty})$$

where K is the thermal conductivity, H is the heat transfer coefficient, T is the temperature and n is the normal vector to the earth's surface.

We will use the fictitious node approach which can be illustrated as follows: the o symbols on the top are in the air. The fictitious

nodes are then solved for using Newton's law of cooling. This is accomplished by using a central difference on the space variables and obtaining the values of the fictitious nodes.

These values are then substituted into the equations at the top of the layer. The symbol x represents nodal points in the earth which are the values that are solved for by the CGS method.

Assuming that the typical nodal temperature at the earth's surface, obtained from a central differencing of the equations representing the temperature in the earth, can be written as and the boundary condition at the earth's surface which is used to solve for the fictitious nodal points is Solving for the fictitious nodes, and substituting into the equation at the top of the layer, gives

$$(a_{1,0} + e_{1,0})T_{1,1} + b_{1,0}T_{1+1,0} + (c_{1,0} - \frac{2\Delta y H e_{1,0}}{K})T_{1,0} + d_{1,0}T_{1-1,0} = -\frac{2\Delta y H e_{1,0}T_{\infty}}{K}$$

We can now solve for the temperature at the earth points by solving the complete system of equations formed by differencing the equations representing the heat flow with the first equation modified.

One Dimensional Case

Newton's law in the one space dimension is

$$-K \frac{dT}{dx} = H(T - T_{\infty})$$

Assume this is the boundary condition at the upper surface of

the earth at $x=0$, and at the lower rock surface, $x=L$,

$$-K \frac{dT}{dx} = C_1$$

Assume the second derivative of the temperature is zero, which is the equation for steady-state heat distribution in one dimension, gives

$$T = A + Bx$$

Doing the algebra,

$$B = -\frac{C_1}{K}; \quad A = -\frac{C_1}{H} + T_\infty$$

which gives a procedure for 'adjusting' the parameters to fit some experimental data.

If there are two layers, the solution is similar; the flux is considered to be constant at the inner boundary, and the same conditions at the upper

and lower surface as in the previous problem. To solve for A, B, C, and D we satisfy continuity at the inner boundary on both the temperature and its derivative.

Second Order Accuracy

Making the convection term second order accurate imposes the usual cell Reynolds or Peclet number condition that occurs in the one dimensional problem (Strikwerda, 1989). A discussion of this problem is also discussed by Peyret and Taylor (1983). To retain

the accuracy needed to avoid problems with the central difference, more grid refinement is needed. We have experimented with a third order accurate scheme on some simple examples, which indicates that methods can be devised that are higher order accurate if some sort of biased higher order derivative is used. Central difference methods will always be limited in flow problems. This problem needs much further study.

Three Dimensional CGS Algorithm

In three dimensions, the size of the arrays used to model the temperature in geologic basins are too large to run on the 640 kilobytes limitation imposed by a personal computer. Lahey Computer Corporation has recently introduced an extended memory manager and a fortran which runs in extended memory. This allows much larger arrays to be processed by a personal computer with extended memory. However, the purpose of this project was to develop fortran computer code which would solve heat transfer problems on a typical personal computer.

A three dimensional CGS solver was written and verified by changing the dimensions of the problem to several typical values that would be used to model a basin. The usual quick CGS convergence was observed as shown in Figure 3. Since the CGS solver obtains an 'exact' solution, that is to machine accuracy, without adding a large memory storage problem, we feel that using any approximate method such as LU factorization would be counterproductive. To show how the CGS solver works in 3D, we applied the CGS solution method to a typical 80x80 nonsymmetric

matrix typical of the 3D (20x5x4) simulations occurring in the temperature modeling problem. The convergence history is shown in Figure 3.

Figure 3. Convergence for the 3 dimensional CGS method.

Recommendations

In the class of nonsymmetric matrices that we are solving, the CGS algorithm performed very well. The improvement over the SOR on a 15x15 grid was 15 or 16 iterations for CGS compared to several thousand for SOR. As we have only preliminary results on the time varying case, we did not use the operator splitting LU scheme, but relied on the CGS solver. This method gives an

"exact" solution in the sense that the method should converge in a finite number of steps. The Gaussian elimination method was used to verify the accuracy of the solution produced by the CGS procedure.

COMPUTER PROGRAM DESCRIPTION

As previously discussed, the purpose of this program is to numerically solve for the temperature structure in a cross-section of the earth. Known parameters include a basal heat flux into the cross section, a mean annual air temperature, thermal conductivity values at the node points in the cross-section, and the Darcian flux (water flux) at node points. A complete listing of the fortran source code for the program is given in Appendix A.

The equation solved is:

$$\nabla \cdot (K \nabla T) - c_p (V \cdot \nabla T) = 0$$

where K is thermal conductivity, T is the temperature solution (a vector), V is the Darcian flux or water velocity (a vector), c is the specific heat of water, and ρ is the density of water.

The difference approximation of this equation uses centered differences for the conduction term. The program allows the user to specify one of two differencing methods for the convective term. The convective term may be differenced using either a backwards difference oriented upstream (upwinding) or a centered difference.

The program uses the harmonic mean to calculate internodal thermal conductivity values for the conductive difference equations. If a solution for purely conductive heat transport is desired, The Darcian flux should be set to zero.

The vertical boundaries of the cross section are defined as

no horizontal flux boundaries. The bottom boundary of the cross-section is a vertical flux boundary (specified by the input basal heat flux). The upper boundary of the cross-section is a flux boundary constrained by Newton's Law of cooling (Heat Flux = Heat Transfer Coefficient * (Temperature at ground surface - Temperature at infinity)). The temperature at infinity is taken to be the mean annual air temperature.

The depth spacing (Y direction) and horizontal spacing (X direction) are set to constant values.

Mechanics of the Program

Two data files (HEATDAT. and CROSSDAT.) will be read into the main program. HEATDAT contains the mean annual air temperature ($^{\circ}\text{C}$), basal heat flux into the model (10^{-3} watts m^{-2}), horizontal spacing (meters), vertical spacing (meters), convergence criteria (usually in the range 10^{-3} to 10^{-6}), and the maximum number of iterations allowed if the Gauss-Siedel method is used (an integer value). See Appendix B for a listing of a sample HEATDAT data file.

File CROSSDAT contains the thermal conductivity matrix and two matrices defining the Darcian flux (water velocity). The matrices are subdivided into data blocks which are NCOLS wide. Thus, if the discretization of the cross-section resulted in a matrix that was 30 columns wide by 49 rows deep and NCOLS was 15, two data blocks (each 15 columns by 49 rows) would represent the thermal conductivity structure. Each block of data is separated by an integer dummy variable (SPACER) to enhance readability of

the data matrices. The block of data values representing the Darcian velocity in the x-direction is separated from the y-direction data blocks by two input variables. These variables are the density of the fluid (RHO) in kg m^{-3} and the specific heat of the fluid (SPHEAT) in $\text{watts sec kg}^{-1} \text{K}^{-1}$. See Appendix C for a listing of a sample CROSSDAT data file.

Constants which are defined in the program include the column dimension (COLDIM), row dimension (ROWDIM), and total dimension (TOTDIM).

Matrices used in the program include:

COND --- thermal conductivity matrix
 VALUE -- coefficient matrix
 DVX ---- water velocity in the x-direction
 DVY ---- water velocity in the y-direction
 TS ----- temperature solution.

Vectors used in this program include:

CT---values of the first column of VALUE(I,J)
 CL --- values of the second column of VALUE(I,J)
 CDEN - values of the third column of VALUE(I,J)
 CR --- values of the fourth column of VALUE(I,J)
 CB --- values of the fifth column of VALUE(I,J)
 BOUND- values of the inhomogeneous terms
 (boundaries, sources, etc>)
 T ---- values of the temperature initialization
 and solution.

Program Output

The program creates a text file named RESULTS. Contained in RESULTS are the matrix size for the problem (number of columns and rows), total number of nodes, air temperature (C), surface coefficient of heat transfer ($\text{W m}^{-2} \text{K}^{-1}$), vertical heat flux (10^{-3}W m^{-2}), the convergence criteria (difference between sums of solutions), horizontal spacing (m), vertical spacing (m), thermal conductivity matrix ($\text{W m}^{-1} \text{K}^{-1}$), density of the fluid (kg m^{-3}), specific heat of the fluid ($\text{watts sec kg}^{-1} \text{K}^{-1}$), Darcian velocity in the X direction (m sec^{-1}), Darcian velocity in the Y direction (m sec^{-1}), and the calculated temperature solution matrix ($^{\circ}\text{C}$).

The program also creates text files for graphing. File CROSS.GRD is the calculated temperature solution constructed for input into Golden Software's contouring program SURFER. File SURFACE.DAT lists data pairs of distance along the top of the model and calculated surface temperature. File TEMPDEP.DAT contains data for three temperature-depth profiles at the left, center, and right of the model. The data file contains four columns of data representing depth, left boundary temperature, center temperature, and right boundary temperature. Two other data files, CGSCONV.DAT and SORCONV.DAT, contain the number of iterations verses calculated error for the conjugate gradient squared method and the Gauss-Siedel method, respectively.

Program Outline

The main program and subroutines are summarized as follows.
See Appendix A for a complete program listing.

I. PROGRAM HEATDOE (Main program).

A. Open Files for input and output.

1. OPEN (6,"HEATDAT")
2. OPEN (7,"CROSSDAT")
3. OPEN (8,"RESULTS")
4. OPEN (9,"CROSS.GRD")
5. OPEN (10,"CGSCONV.DAT")
6. OPEN (11,"SORCONV.DAT")
7. OPEN (12,"TEMPDEP.DAT")
8. OPEN (13,"SURFACE.DAT")

B. Change parameters to change dimensions of problem. A matrix that has I rows and J columns is used.

```
PARAMETER (COLDIM =30,
            ROWDIM = 49,
            TOTDIM = 1470,
            NCOL=15,
            NB=5,NTD1=1471,ntd=1470,ncd=30)
```

C. Initialize matrices to zero.

D. Input constants from file HEATDAT.

1. Surface air temperature ($^{\circ}\text{C}$).
2. Heat flow (10^{-6} watts meter $^{-2}$).
3. Horizontal spacing (meters).
4. Vertical spacing (meters).
5. Convergence criteria.
6. Maximum iterations (integer value).

E. Call subroutine CONDUCTIVITY (thermal conductivity matrix).

F. Call Subroutine DARCIAN (water flow fluxes).

G. Build the vector of inhomogeneous terms.

H. Build the banded matrix of coefficients (call COEFF).

I. Call Newton's Law of cooling for surface boundary.

J. Solve using successive over/under relaxation (Gauss-Siedel) or conjugate gradient squared methods.

K. Output computed temperature solution to files.

1. Output to SURFER file CROSS.GRD (computed temperatures as a function of depth and horizontal spacing ready for contouring).
2. Output to GRAPHER file SURFACE.DAT (computed temperatures along the surface of the model).
3. Output to GRAPHER file TEMPDEP.DAT (three temperature-depth profiles through the computed model).

II. Subroutine CONDUCTIVITY.

A. Input the thermal conductivity matrix from file

- CROSSDAT.
 - B. Output the conductivity matrix to file RESULTS.
 - C. Calculate the harmonic mean for finite-difference coefficients.
 - D. Initialize temperature solution.
- III. Subroutine DARCIAN.
- A. Input the x-direction Darcian flux matrix from file CROSSDAT.
 - B. Input the density (kg m^{-3}) and specific heat ($\text{watts sec kg}^{-1} \text{K}^{-1}$) of the convecting fluid from file CROSSDAT.
 - C. Input the y-direction Darcian flux matrix from file CROSSDAT.
 - D. Output Darcian flux matrices to file RESULTS.
 - E. Choose finite-difference approximation method for convective flow.
 - 1. Calculate coefficients for first order upwinding.
 - 2. Calculate coefficients for second order centered.
- IV. Subroutine COEFF (banded matrix of coefficients).
- A. Build the first column.
 - B. Build the second column.
 - C. Build the third column.
 - E. Build the fourth column.
 - F. Build the fifth column.
- V. Subroutine NEWTON (calculates surface ground temperature).
- VI. Subroutine SORITERATE (calculates temperature solution using Gauss-Siedel with successive under- or over-relaxation).
- VII. Subroutine CGS (calculates temperature solution using the conjugate gradient squared method).
- A. Call subroutine MULAXC.
 - B. Call subroutine SOLL.
 - C. Call subroutine SOLU.
- VIII. Subroutine MULAXC (multiplies the non-symmetric nested factorization matrix with a given vector).
- A - The banded matrix stored from lowest subdiagonal to highest subdiagonal. The middle column is the main diagonal.
 - XX - The vector to be multiplied.
 - AX - The vector to be returned.
- XI. Subroutine SOLL (solves $(L + D) XX = RT$ where L is the lower part of matrix A, D is the diagonal of A, RT is the right hand side).
- X. Subroutine SOLU (solves $(I + D^{-1}U)XX = RT$ where D^{-1} is the diagonal inverse and U is the upper part of matrix A).

Compiling and Running Program HTDOE

Program HTDOE is written in standard Fortran 77 language. It has been compiled and run on Lahey Personal Fortran F77L and Lahey EM32 Fortran.

The first step in running the program is to create files HEATDAT and CROSSDAT. To create these files a nodal spacing along a geologic cross-section must be chosen. Then the thermal conductivity of the rocks at each node point should be input into file CROSSDAT. This may be accomplished using a text editor. The Darcian flux must also be input into file CROSSDAT. The first blocks of Darcian velocities represent flow in the X direction (positive left to right) while the second blocks represent flow in the Y direction (positive top to bottom). An important constraint on the computer model is that the mass flux along the cross-section be conserved. This means that if the aquifer were to thin by 50%, the Darcian velocity would double. See Appendices B and C for examples of HEATDAT and CROSSDAT.

A mean annual air temperature, thermal conductivity values, a basal heat flow, a surficial heat transfer coefficient, and Darcian velocities all affect the resultant temperature solution. In few cases will all of these input variables be known. However, the computer model may be run for many cases by varying the input variables over "reasonable" ranges. Such an iterative technique is useful in defining which variables are important for the particular geologic problem being studied.

Once the nodal spacing has been chosen, then the number of

columns of nodes (COLDIM and ncd), the number of rows (ROWDIM), and the total number of nodes (TOTDIM and ntd) may be determined. These values must be input into program HTDOE in the PARAMETER statement and the program recompiled. This is necessary because Fortran does not allow dynamic allocation of memory at execution time. The total number of dimensions greatly affects program speed and size. If only 400 kilobytes to 500 kilobytes of RAM is available on the computer being used, then the maximum number of nodes will be approximately 1,300. However, if 7 to 8 megabytes of RAM are available, the number of nodes can be greater than 20,000. In order to access the 7 to 8 megabytes of extended memory, special Fortran compilers must be used.

Once compiled, the program is initiated by typing the word HTDOE. The first question the program asks is for a heat transfer coefficient. Next, a Darcian velocity is requested. If the response is a numeric value of zero, then there is no Darcian velocity. If any other value is input, then that value becomes a multiplier of the values contained in file CROSSDAT. A non-zero Darcian multiplier prompts the program to ask whether a first-order upwinding or second-order centered finite-differencing scheme should be used to approximate the Darcian flow. The program computes and displays a stability parameter (P). Theory predicts that when the stability factor is greater than 2, a centered difference solution will fail to converge. However, our experience with the conjugate gradient squared method suggests that the solution will converge up to a stability constant of

100. The final item requested by the program is the solution method. If Gauss-Siedel is selected, a relaxation constant is also asked for. For a zero Darcian flow, a relaxation constant greater than one but less than two is required. For a nonzero Darcian flow, the relaxation constant must be less than one but greater than zero. The other solution technique is the CGS (conjugate gradient squared) method. Use of this solution technique is highly recommended due to its rapid convergence.

GEOLOGY OF THE THERMOPOLIS HYDROTHERMAL SYSTEM

The Thermopolis hydrothermal system is located along the Thermopolis anticline in the Bighorn Basin, northwestern Wyoming (see Figures 4 and 5). Measured water temperatures of the hydrothermal system decrease from 72 °C (162 °F) at the northwest end of the anticline to 54 °C (130 °F) at the southeast end of the system at Hot Springs State Park in Thermopolis. Static water levels in wells also decrease slightly in the same direction (see Figure 5). However, the chemistry of the warm water remains relatively constant within the hydrothermal system (see Table 1).

The heat source for the hydrothermal system is deep circulation of water. The Permian Park City Formation, Pennsylvanian Tensleep Sandstone, Mississippian Madison Limestone, and Ordovician Bighorn Dolomite are considered major aquifers in the Bighorn Basin (see Figure 3 of Hinckley et al., 1982). Far to the west of the Thermopolis anticline, these rocks are recharged by surface waters. Immediately to the west of the anticline, these aquifers have been folded to a depth of over 1.8 km (6000 feet). Given the normal temperature increase with depth for the Bighorn Basin, the water circulating through these rocks is at a temperature near 72 °C (162 °F).

The Thermopolis anticline is an asymmetrical uplift in which these aquifers are folded near the surface. Thus, water circulating through these rocks moves rapidly to near the surface without much heat dissipation. The fault bounding the anticline

severs the aquifers thus producing a barrier to water flow from west to east across the anticline (Hinckley et al, 1982; Spencer, 1986). Heated water enters the Thermopolis hydrothermal system at the northwest termination of the Thermopolis anticline. The water then flows primarily in a southeast direction along the highly fractured crest of the anticline (Spencer, 1986). A more detailed description of the mechanism of operation, geology, and hydrology of the Thermopolis hydrothermal system is given in Hinckley et al. (1982) and Spencer (1986).

Figure 4. Location map for the Thermopolis Hydrothermal study area.

General Geologic Map of the Thermopolis Area

Figure 5. Generalized geologic map for the Thermopolis hydrothermal system showing elevation of static water levels.

Table 1

Chemical data for Springs and Wells in the Thermopolis Area¹

Sample No. ²	Latitude	Longitude	Temp. (°C)	Flow (l/m)	Date Analyzed	Description
1	43° 40.3'	108° 12.2'	53	3792	7/81	Flowing well named Sacajawea near Thermopolis
2	43° 39.2'	108° 11.6'	56	11000	7/81	Big Spring at Thermopolis
3	43° 39.2'	108° 11.6'	55	0	7/81	Black Sulphur Hot Spring at Thermopolis
4	43° 39.2'	108° 11.6'	53	760	7/81	White Sulphur Hot Springs at Thermopolis
5	43° 43.6'	108° 20.5'	73	0	6/82	Geothermal test well near Thermopolis (UWT-1)
6	43° 41.6'	108° 18.6'	64	0	6/82	Geothermal test well near Thermopolis (UWT-2)
7	43° 39.6'	108° 12.6'	55	0	6/82	Geothermal test well near Thermopolis (UWT-3)
8	43° 39.2'	108° 11.6'	56	11000	6/82	Big Spring at Thermopolis
9	43° 30.8'	108° 13.4'		0	4/81	Non-flowing well owned by Carl Spomer

¹Data from Heasler, 1984.²See Figure 5 for location of samples.

Table 1 (cont.). Chemical Data for Springs and Wells in the Thermopolis Area.

(Values are reported in ppm.)

Sample No.	Na	K	B	Ca	Mg	Si	HCO ₃	F	Cl	P	NO ₃	SO ₄	pH	TDS	Cu	Cr	Ni	Fe	Cd	Zn	Mn	Pb	Ag	As	Se	Hg	Ba
1	270	54		397	77.8			2.86	338	<.5	<.5	802			<.05	<.05	<.1	.02	<.05	<.02	.05	<.2	<.05				
2	238	47		314	63.5			3.28	326	<.5	<.5	648			<.05	<.05	<.1	.30	<.05	<.02	.07	<.2	<.05				
3	268	50		310	64.3			3.38	378	<.5	<.5	708			<.05	<.05	<.1	<.02	<.05	<.02	<.05	<.2	<.05				
4	257	50		357	70.4			2.87	360	<.5	<.5	778			<.05	<.05	<.1	.02	<.05	.08	<.05	<.2	<.05				
5	212	42.6		263	60.6			3.1	205			632	7.67		<.05	.07	<.1	1.4	<.05	.03	<.05	<.2	<.05				
6	253	42.6		280	61.3			3.0	287			646	7.64		<.05	.08	<.1	1.3	<.05	.04	<.05	<.2	<.05				
7	257	42.5		336	93.6			3.1	260			911	7.77		<.05	.06	<.1	.7	<.05	.05	<.05	<.2	<.05				
8	233	42.5		320	68.5			3.7	239			631	7.70		<.05	.05	<.1	.05	<.05	.03	<.05	<.2	<.05				
9	270	35	1.0	340	66	34	620	4.5	360		.1	760	6.6	2300	.2	<.05	<.1	1.3	<.03	.30	.06	<.2	<.05	<.005	<.005	<.001	<.1

Samples 1-8 were analyzed at the Department of Geology and Geophysics, University of Wyoming, Laramie, Wyoming by Dr. Steve Boese. Sample 9 was analyzed by the Wyoming Department of Agriculture, Division of Laboratories, Laramie, Wyoming.

THERMAL MODELING OF THE THERMOPOLIS HYDROTHERMAL SYSTEM

A one-dimensional, steady-state, conductive heat transport model for the Thermopolis hydrothermal system is discussed by Hinckley et al. (1982). The purpose of the simple model was to calculate the depths of fluid circulation necessary to explain the observed well temperatures and then to determine if water flow to such depths was geologically reasonable. This simple model indicated that the water flow direction was generally from the west near the northwest end of the Thermopolis anticline. Later hydrologic studies by Jarvis (1986) and Spencer (1986) confirmed this hypothesis.

In order to more precisely determine thermal and hydrologic properties along the Thermopolis anticline, the conductive and convective transport of heat in two dimensions was numerically modeled using our previously described algorithms.

The model was constructed to represent a cross-section line parallel to the crest of the Thermopolis anticline (line A to A' as shown in Figure 5). Figure 6 is the finite-difference representation of the geologic structure along the cross-section. Thicknesses and thermal conductivities of the rock units used in the model are presented in Table 2. The cross-section is 16.5 km long and extends to a depth of 1.4 km. The finite-difference approximation of the cross-section used a horizontal spacing of 50 m and a vertical spacing of 25 m for a total of 18,810 node points.

Figure 6. Geologic cross section used in the finite-difference modeling. See Figure 5 for location.

Table 2
Thermal Conductivity and Thickness Parameters
Used in the Computer Modeling¹

Formation	Thickness (meters)	Thermal Conductivity (W m ⁻¹ K ⁻¹)	Total Thickness (meters)	Weighted Thermal Conductivity
Chugwater	50	3.0	50	3.0
Phosphoria	80	4.0	Paleozoic Aquifers 425 meters	4.0
Tensleep	85	4.4		
Amsden	85	3.3		
Madison	125	4.0		
Bighorn	50	4.6	Cambrian Undivided 340 meters	2.9
Gallatin	120	3.1		
Gros Ventre	140	2.5		
Flathead	60	3.6		
PreCambrian Granite	---	3.2		3.2

¹Values are from Heasler (1978), and Hinckley et al. (1982).

The top boundary condition for the model was specified through Newton's Law of cooling. A heat transfer coefficient was chosen such that the ground surface temperature was approximately 2 °C greater than the mean annual air temperature. The mean annual air temperature in Thermopolis is 8.0 °C (46.4 °F) (Martner, 1986). A heat transfer coefficient of 0.1 watt m⁻² results in computed surface temperatures in the range of 8.5 to 15 °C. The approximate 2 °C temperature difference was observed in three wells which were thermally re-measured as part of this study (see Figures 7, 8, and 9). The data for Figures 7, 8, and 9 are presented in Appendix D.

Figure 7. Temperature-depth profile for well UWT-1. See Figure 5 for location.

Figure 8. Temperature-depth profile for well UWT-2. See Figure 5 for location.

Figure 9. Temperature-depth profile for well UWT-3. See Figure 5 for location.

The lower boundary condition for the model was specified as a constant heat flux boundary. The value chosen to represent a regional conductive heat flux was 67×10^{-3} watts m^{-2} (from Heasler, 1978; Heasler et al., 1985; Heasler and Hinckley, 1985).

Figure 10. Calculated temperatures ($^{\circ}\text{C}$) for no water flow and vertical flux side boundaries. Dashed lines show model stratigraphy.

Figure 10 shows the modeling results for no water flux and a heat transfer coefficient of $0.1 \text{ watt m}^{-1} \text{ sec}^{-1}$. Notice in Figure 10 that the temperature contours are nearly horizontal, and do not parallel the modeled stratigraphy.

Figure 11 represents

three temperature-depth profiles through the model. The temperature-depth profiles of Figure 11 approximate a temperature solution using Fourier's law of heat conduction. Consequently,

Figure 11. Calculated temperature-depth profiles for Figure 10 for the left, center, and right portions of the cross-section.

the basic input, output, and calculations accomplished by the program seem to be correct.

Figure 12. Calculated temperatures ($^{\circ}\text{C}$) for a Darcian velocity of $10^{-6} \text{ m sec}^{-1}$.

Figure 12 shows the modeled results for a Darcian flux of $10^{-6} \text{ m sec}^{-1}$. This flux magnitude correlates to that through a clean sand. Since the Paleozoic aquifer system is composed of both sands and limestones, this Darcian flux value is considered a reasonable maximum. Note that the temperature contours shown in Figure 12 nearly parallel the modeled stratigraphy.

The Darcian flux along the crest of the Thermopoleis anticline was calculated using Darcy's law:

$$V = -K \left(\frac{dh}{dl} \right)$$

where V is the Darcian velocity (specific discharge), K (for this equation) refers to hydraulic conductivity, and dh is the change

in potentiometric surface over length dl . From Figure 5, dh is 11 meters and dl is 16.5 km. From Freeze and Cherry (1979), the range of hydraulic conductivities for a limestone is 10^{-9} to 10^{-6} $m\ sec^{-1}$ and for clean sand is 10^{-6} to 10^{-2} $m\ sec^{-1}$. Using these values for the Thermopolis anticline results in possible Darcian flows in the range of 10^{-13} to 10^{-10} $m\ sec^{-1}$ for a limestone and 10^{-10} to 10^{-5} $m\ sec^{-1}$ for a clean sand.

The modeled portion of the Thermopolis hydrothermal system is along the crest of the Thermopolis anticline. Because warm water arrives at the northwestern portion of the anticline (endpoint A on cross-section line A-A'), the left boundary condition of the model was changed to that of a specified temperature. The temperatures for the left boundary were primarily determined from the temperatures shown in Figure 8 for well UWT-2. Temperatures for the left model boundary greater than the depth of well UWT-2 were estimated using Fourier's law of heat conduction:

$$\frac{q}{K} \Delta y = \Delta T$$

where q is heat flow (67×10^{-3} watts m^{-2}), K is thermal conductivity (4.0, 2.9, or 3.2 watts $m^{-1} K^{-1}$), Δy refers to the incremental depth increase (35 m), and ΔT is the resultant temperature increase over the y spacing. Figure 13 shows the calculated temperatures for this model with no water flow.

Figure 13. Calculated temperatures ($^{\circ}\text{C}$) for a specified left temperature boundary and no water flow.

Figures 14, 15, 16, and 17 show the calculated temperature results using Darcian velocities of 10^{-7} , 10^{-6} , 10^{-5} , and 10^{-4} m sec^{-1} , respectively. Figure 18 shows three calculated temperature-depth profiles for a Darcian velocity of 10^{-4} m sec^{-1} .

Figure 14. Calculated temperatures ($^{\circ}\text{C}$) for a Darcian velocity of 10^{-7} m sec^{-1} .

Figure 15. Calculated temperatures ($^{\circ}\text{C}$) for a Darcian velocity of $10^{-6} \text{ m sec}^{-1}$.

Figure 16. Calculated temperatures ($^{\circ}\text{C}$) for a Darcian velocity of $10^{-5} \text{ m sec}^{-1}$.

Figures 14 through 18 indicate that a Darcian velocity of the magnitude $10^{-4} \text{ m sec}^{-1}$ is required to match the temperature changes along the crest of the Thermopolis anticline as shown in Figures 5, 7, 8, and 9. However, any modeled Darcian velocity has difficulty in precisely matching the temperature-depth

Figure 17. Calculated temperatures ($^{\circ}\text{C}$) for a Darcian velocity of $10^{-4} \text{ m sec}^{-1}$.

Figure 18. Calculated temperature-depth profile for a Darcian velocity of $10^{-4} \text{ m sec}^{-1}$.

profiles and thermal gradients shown in Figures 7, 8, and 9.

Improvements to the model

The computer model of the Thermopoleis hydrothermal system predicts the general observed temperature decline along the crest of the Thermopoleis anticline. However, the model has difficulty

in matching individual temperature-depth profiles.

The following items are suggested to improve the modeling results:

1. Consider the effect of topography in the upper boundary condition (using Newton's law of cooling).
2. Consider a vector distribution in the Paleozoic aquifer system which would approximate mixing of the hydrothermal fluid. This would decrease the geothermal gradient within the aquifer portion of the model but result in an overall higher total heat loss.
3. Make the computer model time-dependent so that the upper surface air temperature could be varied and the resultant temperature change calculated.

SUMMARY AND CONCLUSIONS

The Conjugate Gradient Squared (CGS) solution technique was extremely effective when applied to the finite-difference solution of heat transfer on a geologic scale. The CGS method was computationally more efficient than relaxation schemes, Gaussian elimination methods, and the Generalized Minimum Residual (GMRES) method. The CGS procedure converged at least ten times faster than the other methods. Since the CGS solver obtains an 'exact' solution, (to machine accuracy) without adding a large memory storage penalty, the use of approximate solution methods was unnecessary.

The CGS solution method calculated rapidly converging solutions to a finite-difference model of the Thermopolis hydrothermal system. The model of the Thermopolis hydrothermal system used Newton's law of cooling at the ground surface boundary, vertical heat flux side boundaries, and a constant vertical heat flux lower boundary. Model input parameters (thermal conductivity, aquifer thickness, basal heat flow) were determined from published sources. The Darcian velocity of water flow was varied through a range of possible values and the results compared with measured temperature-depth profiles. Calculated temperatures along the crest of the Thermopolis anticline were in good agreement with measured temperatures, generally showing the correct trends and inflections. Results of the model indicate that the decrease in temperature from the northwest crest of the anticline (72 °C) to the Thermopolis hot

springs (54 °C) is primarily due to cooling of the 72 °C water as it moves southeasterly towards the hot springs.

While the finite-difference model showed correct general trends, a time-varying three-dimensional model will be needed to fully explain the potential effects of groundwater mixing along the crest of the anticline and the thermal effects of surface topography.

REFERENCES

- Allen, M.B., 1984, Why upwinding is reasonable, in Laible, J.P., (ed.), Proceedings of the Fifth International Conference on Finite Elements in Water Resources, June 18-22, 1984, Burlington, Vermont: Springer-Verlag, Berlin, p. 13-23.
- Anderson, D.A., J.C. Tannehill, and R.H. Pletcher, 1984, Computational Fluid Mechanics and Heat Transfer: Hemisphere Publishing Corporation, New York, 599 p.
- Bair, E.S., and R.R. Parizek, 1978, Detection of permeability variations by a shallow geothermal technique: Ground Water, vol. 16, p.254-263.
- Bredehoft, J.D., and I.S. Papadopoulos, 1965, Rate of vertical groundwater movement estimated from the earth's thermal profile: Water Resources Research, vol. 1, p. 325-328.
- Cartwright, K., 1970, Groundwater discharge in the Illinois Basin as suggested by temperature anomalies: Water Resources Research, vol. 6, p. 912-918.
- Donaldson, J.G., 1962, Temperature gradients in the upper layers of the earth's crust due to convective water flows: Journal of Geophysical Research, vol. 67, p. 3449-3459.
- Freeze, R.A., and J.A. Cherry, 1979, Groundwater: Prentice-Hall, Englewood Cliffs, New Jersey, 604 p.
- Golub, G.H., and C.F. Van Loan, 1989, Matrix Computations: 2nd ed. John Hopkins University Press, Baltimore Maryland, 642 p.
- Haenel R., L. Rybach, and L. Stegna, 1988, Fundamentals of geothermics; in Handbook of Terrestrial Heat-Flow Density Determination: Haenel, R., L. Rybach, and L. Stegna (eds.), Kluwer Academic Publishers, Dordrecht, Boston, London, 486 p.
- Heasler, H.P., 1978, Heat flow in the Elk Basin oilfield, northwestern Wyoming: unpubl. M.S. thesis, University of Wyoming, Laramie, 168 p.
- Heasler, H.P., 1984, Chemical analyses of selected thermal springs and wells in Wyoming: Prepared for the U.S. Department of Energy under contract DE-FC07-79ID12026, available from the National Technical Information Service, U.S. Dept. of Commerce, Springfield, Virginia, 4 p.
- Heasler, H.P. and B.S. Hinckley, 1985, Geothermal resources of the Bighorn Basin, Wyoming: Geological Survey of Wyoming, Report of Investigations, No. 29, 28 p.

- Heasler H.P., B.S. Hinckley, K.L. Buelow, S.A. Spencer, and E.R. Decker, 1985, Geothermal resource map of Wyoming: National Oceanic and Atmospheric Administration (publishers), available from the Geological Survey of Wyoming, P.O. Box 3008, Laramie, Wyoming, scale 1:500,000.
- Hinckley, B.S., H.P. Heasler, and J.K. King, 1982, The Thermopolis hydrothermal system with an analysis of Hot Springs State Park: Geological Survey of Wyoming, Preliminary Report No. 20, 42 p.
- Huang, C-Y., S.R. Kennon, and G.S. Dulikravich, 1986, Generalized nonlinear minimum residual (GNLMR) method for iterative algorithms: J. Comp. and Appl. Math., p. 215-232.
- Jarvis, T.W., 1986, Regional hydrogeology of the Paleozoic aquifer system, southeastern Bighorn Basin, Wyoming, with an user impact analysis on Hot Springs State Park: unpubl. M.S. thesis, University of Wyoming, Laramie, 227 p.
- Keys W.F., and R.F. Brown, 1978, The use of temperature logs to trace the movement of injected water: Ground Water, vol. 16, p. 32-48.
- Kilty, K., D.S. Chapman, and C. Mase, 1978, Aspects of forced convective heat transfer in geothermal systems: Topical report for work performed under contract number EG-78-C-07-1701 between the U.S. Department of Energy and the Department of Geology and Geophysics, University of Utah, Salt Lake City, Utah, 62 p.
- Kilty, K., and D.S. Chapman, 1980, Convective heat transfer in selected geologic situations: Ground Water, vol. 18, p. 386-394.
- Peyret, R. and T.D. Taylor, 1983, Computational Methods for Fluid Flow: Springer Series in Computational Physics, Springer Verlag, New York, 358 p.
- Martner, B.E., 1986, Wyoming Climate Atlas: University of Nebraska Press, Lincoln and London, 432 p.
- Saad, Y., and M.H. Shultz, 1986, GMRES: A generalized minimum residual algorithm for solving nonsymmetric linear systems: SIAM J. SCI. STAT. COMPUT., p. 856-869.
- Singh, B.S., and A. Dybbs, 1974, Heat transfer characteristics of porous media: Proceedings of the fifth International Heat Transfer Conference, vol. 5, CT3.5, p. 98-102.

- Smith, L., and D.S. Chapman, 1983, On the thermal effect of groundwater flow 1. regional systems: Journal of Geophysical Research, vol. 88, p. 593-608.
- Sonneveld, P., 1989, CGS, a fast Lanczos-type solver for nonsymmetric linear systems: SIAM Journal SCI. STAT. COMPUT., p. 36-52.
- Slattery, J.C., 1972, Momentum, Energy, and Mass Transfer in Continua: McGraw-Hill, New York, 679 p.
- Sorey, M.L., 1971, Measurement of vertical groundwater velocity from temperature profiles in wells: Water Resources Research, vol. 7, p. 963-970.
- Spencer, S.A., 1986, Groundwater movement in the Paleozoic rocks and impact of petroleum production on water levels in the southwestern Bighorn Basin, Wyoming: unpubl. M.S. thesis, University of Wyoming, Laramie, 165 p.
- Strikwerda, J.C., 1989, Finite Difference Schemes and Differential Equations: Wadsworth and Brooks/Cole Mathematics series, Pacific Grove California, 386 p.

Appendix A
Program listing.

C*****C

Steady-State Transfer of Heat in Geologic Conditions Due to Conduction and Convection (Water Flow).

February 25, 1991

H. P. Heasler (Dept. of Geology) & J. H. George (Dept. of
Mathematics), University of Wyoming, Laramie, Wyoming 82071

1. Theory and Strategy.

The purpose of this program is to numerically solve the temperature structure in a cross-section of the earth. Known parameters include a basal heat flux into the cross section, a mean annual air temperature, thermal conductivity values at the node points in the cross-section, and the Darcian flux at node points.

The equation solved is:

$$\text{Del Dot } (K \text{ Del } T) - ((C * \text{Rho} * V) \text{ Dot Del } T) = 0$$
where Del is the Del operator, Dot is the dot product between two vectors, K is thermal conductivity, T is the temperature solution (a vector), V is the Darcian flux or water velocity (a vector), C is the specific heat of water and Rho is the density of water.

The difference approximation of this equation uses centered differences for the conduction term. The program allows the user to specify one of two differencing methods for the convective term. The convective term may be differenced using either a backwards difference oriented upstream (upwinding) or a centered difference.

The program uses the harmonic mean to calculate internodal thermal conductivity values for the conductive difference equations. If a solution for purely conductive heat transport is desired, The Darcian flux (water velocity) should be set to zero.

The vertical boundaries of the cross section are defined as no horizontal flux boundaries. The bottom boundary of the cross-section is a vertical flux boundary (specified by the input basal flux). The upper boundary of the cross-section is a flux boundary constrained by Newton's Law of cooling (Heat Flux = Heat Transfer Coefficient * (Temp at ground surface - Temp at infinity)). The temperature at infinity is taken to be the mean annual air temperature.

The depth spacing (Y direction) and horizontal spacing (X direction) are set to constant values.

2. Mechanics of the Program.

Two data files (HEATDAT. and CROSSDAT.) will be read into the main program. HEATDAT contains the mean annual air temperature (C), basal heat flux into the model (milliwatts/square meter), horizontal spacing (meters), vertical spacing (meters), convergence criterea

(usually in the range .0001 to .0000000001), and the maximum number of iterations allowed (an integer value).

File CROSSDAT contains the thermal conductivity matrix and two matrices defining the Darcian flux (water velocity). The matrices are subdivided into data blocks which are NCOLS wide. Thus, if the discretization of the cross-section resulted in a matrix that was 30 columns wide by 19 rows deep and NCOLS was 15, two data blocks (each 15 columns by 19 rows) would represent the thermal conductivity structure. Each block of data is separated by an integer dummy variable (SPACER) to enhance readability of the data matrices. The block of data values representing the Darcian velocity in the x-direction is separated from the y-direction data blocks by two input variables. These variables are the density of the fluid (RHO) in kilograms per cubic meter and the specific heat of the fluid (SPHEAT) in watts sec per kilogram Kelvin.

Constants which are defined in the program include the column dimension (COLDIM), row dimension (ROWDIM), and total dimension (TOTDIM).

Matrices used in the program include:

- COND --- thermal conductivity matrix
- VALUE -- coefficient matrix
- DVX ---- water velocity in the x-direction
- DVY ---- water velocity in the y-direction
- TS ----- temperature solution.

Vectors used in this program include:

- CT---values of the first column of VALUE(I,J)
- CL --- values of the second column of VALUE(I,J)
- CDEN - values of the third column of VALUE(I,J)
- CR --- values of the fourth column of VALUE(I,J)
- CB --- values of the fifth column of VALUE(I,J)
- BOUND- values of the inhomogeneous terms
(boundaries, sources, etc>)
- T ---- values of the temperature initialization
and solution.

3. Program Output

The program creates a text file named RESULTS. Contained in RESULTS are the matrix size for the problem (number of columns and rows), total number of nodes, air temperature (C), surface coefficient of heat transfer ($W/m^2 K$), vertical heat flux ($10^{-6} W/m^2$), the convergence criteria (difference between sums of solutions), horizontal spacing (m), vertical spacing (m), thermal conductivity matrix ($W/m K$), density of the fluid (kg/m^3), specific heat of the fluid (watts sec/kg Kelvin), Darcian velocity in the X direction (m/sec), Darcian velocity in the Y direction (m/sec), and the calculated temperature solution matrix (C).

The program also creates text files for graphing. File CROSS.GRD is the calculated temperature solution constructed for input into Golden Software's contouring program Surfer. File

C SURFACE.DAT lists data pairs of distance along the top of the
 C model and calculated surface temperature. File TEMPDEP.DAT
 C contains data for three temperature-depth profiles at the left,
 C center, and right of the model. The data file contains four
 C columns of data representing depth, left boundary temperature,
 C center temperature, and right boundary temperature. Two other
 C data files, CGSCONV.DAT and SORCONV.DAT, contain the number of
 C iterations verses calculated error for the conjugate gradient
 C squared method and the Gauss-Siedel method, respectively.
 C

C*****C

PROGRAM HEATDOE

OPEN (6,"HEATDAT")

OPEN (7,"CROSSDAT")

OPEN (8,"RESULTS")

OPEN (9,"CROSS.GRD")

OPEN (10,"CGSCONV.DAT")

OPEN (11,"SORCONV.DAT")

OPEN (12,"TEMPDEP.DAT")

OPEN (13,"SURFACE.DAT")

INTEGER I, J, M, MAXITS

INTEGER COLDIM, ROWDIM, TOTDIM, PSTART, PSTART1, PSTART2

CHARACTER WIDTH3

REAL SURFKOND, TSURF, SURFCOEFF, FLUX, DELX, DELY

REAL XHI, XLO, YHI, YLO, ZHI, ZLO

REAL*8 CONVCRIT

C-----C

C A matrix that has I rows and J columns is used. C

C ***** CHANGE PARAMETERS TO CHANGE DIMENSIONS ***** C

C-----C

PARAMETER (COLDIM =45,

\$ ROWDIM = 29,

\$ TOTDIM = 1305,

\$ NCOL=15,

& NB=5,NTD1=1306,ntd=1305,ncd=45)

REAL*8 VALUE(TOTDIM,5), BOUND(TOTDIM), T(TOTDIM)

REAL TS(ROWDIM,COLDIM), DVX(ROWDIM,COLDIM), DVY(ROWDIM,COLDIM)

REAL COND(ROWDIM,COLDIM)

REAL*8 Q(NTD), Z(NTD), W(NTD), R(NTD), Y(NTD), U(NTD), V(NTD), P(NTD)

C-----C

C INITIALIZE VALUES TO ZERO. C

C-----C

DO 105, I=1,TOTDIM

BOUND(I) = 0.0

T(I) = 0.0

DO 105, J=1,5

VALUE(I, J) = 0.0

```

      DVX(I,J) = 0.0
      DVY(I,J) = 0.0
      TS(I,J) = 0.0

```

```

105  CONTINUE

```

```

C-----C
C      INPUT CONSTANTS (HEAT FLUX (MILLIWATTS/METER**2), HORIZONTAL
C      SPACING (METERS), DELX IS THE HORIZONTAL GRID SPACING, DELY
C      IS THE VERTICAL SPACING (METERS), AND COND(I,J) IS THE
C      THERMAL CONDUCTIVITY MATRIX (WATTS/METER K).
C-----C
      WRITE(8,3) COLDIM, ROWDIM, TOTDIM
3      FORMAT(/'The matrix size for this problem is', I3, ' columns and',
      1I3, ' rows.'/ 'The total number of nodes is', I5, '.')
      READ (6,4) TSURF, FLUX, DELX, DELY
4      FORMAT (F10.2)
44     FORMAT (E12.4)
11     FORMAT (I5)
      READ (6,44) CONVCRIT
      READ (6,11) MAXITS
      PRINT *, 'Input the surface coefficient of heat transfer (.05 to 5
      1W/m^2K) '
      READ *, SURFCOEFF
      WRITE (8,5) TSURF
      WRITE (8,55) SURFCOEFF
      WRITE (8,6) FLUX
      WRITE (8,7) DELX
      WRITE (8,8) DELY
      DY2=DELY**2
5      FORMAT(/'The air temperature is ', F5.1, ' degrees C.')
55     FORMAT('The surface coef. of heat transfer is ', E8.2, ' W/ m^2 K.')
6      FORMAT(/'The vertical heat flux is ', F6.1, ' 10^-6 W/m**2.')
7      FORMAT(/'The horizontal spacing is ', F8.1, ' meters.')
8      FORMAT('The vertical spacing is ', F7.1, ' meters.')
      WRITE (8,43) CONVCRIT
43     FORMAT(/'The convergence criterea is: ', E12.4)
C-----C
C      INPUT THE THERMAL CONDUCTIVITY MATRIX
C-----C
      CALL CONDUCTIVITY (DELX, DELY, COLDIM, ROWDIM, TOTDIM,
      &                    NCOL, VALUE, COND, T, TSURF, FLUX,
      &                    PSTART, PSTART1, PSTART2, WIDTH3)
C-----C
C      INPUT THE DARCIAN VELOCITY MATRICES.
C-----C
      CALL DARCIAN (DELX, DELY, RHO, SPHEAT, VEL, COLDIM,
      &              ROWDIM, TOTDIM, NCOL, VALUE, DVX, DVY)
C-----C

```

```

C      BUILD THE VECTOR OF INHOMOGENEOUS TERMS.
C-----C
C
C      LOWER BOUNDARY, CONSTANT VERTICAL FLUX
C-----C
C*** FOR LEFT CONSTANT BOUNDARY, CHANGED FROM +1 TO +2
C-----C
      DO 250 I=TOTDIM-COLDIM+1,TOTDIM
          BOUND(I) = (((FLUX * .001) * DELY) /
&                (VALUE(I, 1) * DY2))
250  CONTINUE

C      WRITE (8,35)
C35   FORMAT(///'THE INHOMOGENEOUS TERMS (A VECTOR) ARE: '/')
C      WRITE (8,30) (BOUND(I),I=1,TOTDIM)
C30   FORMAT (5(1PD14.4))
C-----C
C      BUILD THE MATRIX OF COEFFICIENTS
C-----C
C      CALL COEFF(TOTDIM,ROWDIM,COLDIM,VALUE)
C-----C
C      Call Newton's Law of cooling
C-----C
      DO 240 I=2,COLDIM-1
          CALL NEWTON (BOUND, VALUE, TOTDIM, DELY, TSURF,
&                SURFCOEFF, SURFKOND, I)
240  CONTINUE

C-----C
C      Solve using Successive Over/Under Relaxation
C      or conjugate gradient
C-----C
100  PRINT *, ' '
      PRINT *, 'Input a number for desired solution technique: '
      PRINT *, ' 1-Gauss Siedel with relaxation. '
      PRINT *, ' 2-Conjugate gradient with preconditioning. '
      READ *, SOLUTION

      IF (SOLUTION.EQ.1) THEN
          CALL SORITERATE (VALUE, BOUND, T, COLDIM, TOTDIM,
1          MAXITS, CONVCRIT)
      ELSEIF (SOLUTION.EQ.2) THEN
          CALL CGS(VALUE, BOUND, NCD, NTD, NB, T, R, Q, U,
&                P, V, Z, W, Y, MAXITS, CONVCRIT)
      ELSE
          PRINT *, 'Improper choice . . . . Try again'
          GOTO 100
      ENDIF

```

```

C-----C
C      Output Temperature solution to files.      C
C-----C
      WRITE (8,45)
45  FORMAT(' The solution (degrees Celsius) is: ')
      M = 1
      ZHI = 0.0
      ZLO=1000.0
      DO 510 I=1,ROWDIM
        DO 510 J=1,COLDIM
          TS(I, J) = T(M)
          IF(T(M).GT.ZHI) ZHI=T(M)
          IF(T(M).LT.ZLO) ZLO=T(M)
          M = M + 1
510  CONTINUE
      DO 550 JRI=1,PSTART1,15
      JRJ = JRI + 14
      WRITE (8,42) ((TS(I,J),J=JRI,JRJ),I=1,ROWDIM)
C     WRITE (*,42) ((TS(I,J),J=JRI,JRJ),I=1,ROWDIM)
      42  FORMAT (15F5.1)
      WRITE (8,46)
      46  FORMAT (/)
550  CONTINUE
      IF (PSTART2.GT.COLDIM) GOTO 555
      WRITE (8,WIDTH3) ((TS(I,J), J=PSTART2,COLDIM),I=1,ROWDIM)
555  CONTINUE

C----- OUTPUT TO SURFER FILE CROSS.GRD-----C
      WRITE (9,61)
61  FORMAT ('DSAA')
      WRITE (9,62) COLDIM, ROWDIM
62  FORMAT (I5,I5)
      XLO = 0.0
      XHI = ((COLDIM-1) * DELX)
      YLO = (-(ROWDIM-1) * DELY)
      YHI = 0.0
      WRITE (9,63) XLO, XHI
      WRITE (9,63) YLO, YHI
      WRITE (9,63) ZLO, ZHI
63  FORMAT (F9.1, F9.1)
      WRITE (9,66) ((TS(I,J), J=1,COLDIM,1),I=ROWDIM,1,-1)
66  FORMAT (10F7.1)

C----- OUTPUT TO GRAPHER FILE SURFACE.DAT-----C
71  FORMAT (2F10.2)
      J = 0.0
      DO 610 I = 1, COLDIM
      DISTANCE = J*DELX
      J = J+1
      TEMPERATURE = TS(1,I)
      WRITE (13,71) DISTANCE, TEMPERATURE

```



```

DO 90 JRI=1,RSTART1,NCOL
JRJ = JRI + (NCOL - 1)
READ (7,14) SPACER
READ (7,WIDTH) ((COND(I,J), J=JRI,JRJ), I=1,ROWDIM)
90  CONTINUE
    WRITE (CHARCL2,'(I2)') PWID
    WIDTH2= '('//CHARCL2//F5.1'
    IF (RSTART2.GT.COLDIM) GOTO 85
    READ (7,14) SPACER
    SPACER = SPACER + 1
    READ(7,WIDTH2) ((COND(I,J), J=RSTART2,COLDIM),I=1,ROWDIM)

C-----C
C      OUTPUT THE THERMAL CONDUCTIVITY MATRIX      C
C-----C
85  CONTINUE
    PSTART = COLDIM / 15
    PSTART1 = PSTART * 15
    PSTART2 = PSTART1 + 1
    PWID2 = COLDIM - PSTART1
    WRITE (8,10)
10  FORMAT('The thermal conductivity matrix (W/m K) is:')
    DO 115 JRI=1,PSTART1,15
        JRJ = JRI + 14
        WRITE (8,12) ((COND(I,J), J=JRI,JRJ),I=1,ROWDIM)
12  FORMAT(15F5.1)
        WRITE (8,19)
19  FORMAT (/)
115 CONTINUE
    IF (PSTART2.GT.COLDIM) GOTO 95
    WRITE (CHARCL3,'(I2)') PWID2
    WIDTH3='('//CHARCL3//F5.1'
    WRITE (8,WIDTH3) ((COND(I,J), J=PSTART2,COLDIM),I=1,ROWDIM)
95  CONTINUE

C-----C
C      COMPUTE THE THERMAL CONDUCTIVITY USING THE HARMONIC MEAN.      C
C-----C
    N = 0
    DO 100 I=1,ROWDIM
    DO 110 J=1,COLDIM
        JL = J - 1
        JR = J + 1
        IT = I - 1
        IB = I + 1
        IF(J.EQ.1) JL=JR
        IF(J.EQ.COLDIM) JR=JL
        IF(I.EQ.1) IT=IB
        IF(I.EQ.ROWDIM) IB=IT
        N = N + 1
        VALUE(N,4)=(2.0*COND(I,J)*COND(I,JR))/(COND(I,J)
&                + COND(I,JR))/(DX2)

```



```

      VALUE(N,2)=(2.*COND(I,J)*COND(I,JL))/(COND(I,J)
&      + COND(I,JL))/(DX2)
      VALUE(N, 5) = (2.0 * COND(I, J) * COND(IB, J)) /
&      (COND(I, J) + COND(IB, J))/(DY2)
      VALUE(N, 1) = (2.0 * COND(I, J) * COND(IT, J)) /
&      (COND(I, J) + COND(IT, J))/(DY2)
      VALUE(N, 3) = (VALUE(N, 4) + VALUE(N, 2))
&      + (VALUE(N, 5) + VALUE(N, 1))

```

```

110      CONTINUE
100      CONTINUE

```

```

C-----C
C      INITIALIZE TEMPERATURES AND SET CONSTANT TEMPERATURES      C
C-----C

```

```

      TEMPCON = (DELY * (FLUX * .001)) / COND(1, COLDIM)

```

```

      K = 0

```

```

      DO 310 L=0,TOTDIM-COLDIM,COLDIM

```

```

      DO 300 I=0,COLDIM-1

```

```

          T((I + 1) + L) = (TSURF) + (K * TEMPCON)

```

```

          IF (L.EQ.0) T((I+1) + L) = TSURF

```

```

300      CONTINUE

```

```

310      K = K + 1

```

```

      RETURN
      END

```

```

C!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!C
      SUBROUTINE DARCIA(DXL, DELY, RHO, SPHEAT, VEL,
&      COLDIM, ROWDIM, TOTDIM, NCOL, VALUE, DVX, DVY)
      INTEGER COLDIM, ROWDIM, TOTDIM, CONVECAN
      INTEGER NCOL, RSTART1, RSTART2, PSTART1, PSTART2, I, J
      REAL RHO, SPHEAT, WCON, WCONX, WCONY, VEL, SPACER
      REAL*8 VALUE(TOTDIM,5), VLENGTH
      REAL DXL, DELY
      CHARACTER WIDTH*8,CHARCOL*2,CHARCL2*2,WIDTH2*8,CHARCL3*2,WIDTH3*8
      REAL DVX(ROWDIM, COLDIM), DVY(ROWDIM, COLDIM)

```

```

C-----C
C      INPUT THE DARCIA VELOCITY MATRICES. DVX IS THE DARCIA
C      VELOCITY IN THE POSITIVE X DIRECTION (LEFT TO RIGHT) AND      C
C      DVY IS THE VELOCITY FROM THE BOTTOM TO THE TOP OF THE      C
C      CROSS SECTION.      C
C      ALSO INPUT THE DENSITY OF WATER IN KILOGRAMS PER      C
C      CUBIC METER (RHO) AND THE SPECIFIC HEAT OF WATER IN      C
C      WATTS SEC PER KILOGRAM KELVIN (SPHEAT), AND THE      C
C      DARCIA VELOCITY MULTIPLIER (VEL).      C
C-----C

```

```

      WRITE (CHARCOL,'(I2)') NCOL

```

```

WIDTH='('//CHARCOL//'F5.1)'
RSTART = COLDIM / NCOL
RSTART1 = RSTART * NCOL
RSTART2 = RSTART1 + 1
PWID = COLDIM - RSTART1
WRITE (CHARCL2,'(I2)') PWID
WIDTH2='('//CHARCL2//'F5.1)'
PSTART = COLDIM / 15
PSTART1 = PSTART * 15
PSTART2 = PSTART1 + 1
PWID2 = COLDIM - PSTART1
WRITE (CHARCL3,'(I2)') PWID2
WIDTH3='('//CHARCL3//'F5.1)'

4      FORMAT (F10.2)
19     FORMAT (/)

DO 111 JRI=1,RSTART1,NCOL
      JRJ = JRI + (NCOL - 1)
      READ (7,4) SPACER
      SPACER = SPACER + 1
      READ (7,WIDTH) ((DVX(I,J),J=JRI,JRJ),I=1,ROWDIM)
111    CONTINUE
      IF (RSTART2.GT.COLDIM) GOTO 112
      READ (7,4) SPACER
      READ (7,WIDTH2) ((DVX(I,J),J=RSTART2,COLDIM),I=1,ROWDIM)
112    CONTINUE
      READ (7,4) RHO,SPHEAT
      PRINT *, 'Input the Darcian velocity '
      READ *,VEL

223    FORMAT (E8.1)
DO 113 JRI=1,RSTART1,NCOL
      JRJ = JRI + (NCOL - 1)
      READ(7,4) SPACER
      SPACER = SPACER + 1
      READ(7,WIDTH) ((DVY(I,J),J=JRI,JRJ),I=1,ROWDIM)
113    CONTINUE
      IF (RSTART2.GT.COLDIM) GOTO 114
      READ (7,4) SPACER
      READ (7,WIDTH2) ((DVY(I,J),J=RSTART2,COLDIM),I=1,ROWDIM)
114    CONTINUE

      ZERO = 0.0
      IF (VEL.EQ.ZERO) GOTO 150
      WRITE(8,17) RHO,SPHEAT
17     FORMAT('//The density used is ',F7.1,' kg / cubic meters,//,
      & 'the specific heat is ',F7.1,' watts sec per kg Kelvin.'//)
      WRITE(8,15) VEL
15     FORMAT('/The Darcian velocity (' ,E8.1,') in the X direction is:
      & '/')
      DO 116 JRI=1,PSTART1,15

```



```

      DO 410 J=1,COLDIM
        N = N + 1
        WCONY = WCON * DVY(I,J)
        VALUE(N,1) = VALUE(N,1) + ((WCONY+ABS(WCONY))) / (2.0*DELY)
        VALUE(N,5) = VALUE(N,5) - ((WCONY-ABS(WCONY))) / (2.0*DELY)
        WCONX = WCON * DVX(I,J)
        VALUE(N,2) = VALUE(N,2) + ((WCONX+ABS(WCONX))) / (2.0*DELX)
        VALUE(N,4) = VALUE(N,4) - ((WCONX-ABS(WCONX))) / (2.0*DELX)
        VALUE(N,3) = VALUE(N,3)
&          + ((WCONY + ABS(WCONY))) / (2.0 * DELY)
&          - ((WCONY - ABS(WCONY))) / (2.0 * DELY)
&          - ((WCONX - ABS(WCONX))) / (2.0 * DELX)
&          + ((WCONX + ABS(WCONX))) / (2.0 * DELX)
410    CONTINUE
400    CONTINUE
      GOTO 150

```

```

C-----C
C  CALCULATE COEFFICIENTS FOR SECOND ORDER CENTERED
C
C  ADD THE EFFECT OF WATER FLOW INTO THE COEFFICIENT MATRIX      C
C  CONSTANTS.  COMPUTE A CONSTANT THAT WILL BE USED IN A LOOP.  C
C-----C
130    WCON = RHO * SPHEAT * VEL
      N = 0
      DO 420 I=1,ROWDIM
        DO 430 J=1,COLDIM
          N = N + 1
          WCONY = WCON * DVY(I,J)
ccc    print *, 'wcony = ',wcony
          VALUE(N, 1) = VALUE(N, 1) + (WCONY) / (2.0 * DELY)
          VALUE(N, 5) = VALUE(N, 5) - (WCONY) / (2.0 * DELY)
          WCONX = WCON * DVX(I,J)
          VALUE(N, 2) = VALUE(N, 2) + (WCONX) / (2.0 * DELX)
          VALUE(N, 4) = VALUE(N, 4) - (WCONX) / (2.0 * DELX)
430    CONTINUE
420    CONTINUE
      GOTO 150

150    RETURN
      END

```

```

C!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!C
  SUBROUTINE COEFF(TOTDIM, ROWDIM, COLDIM, VALUE)
  INTEGER I,J,JN,N,TOTDIM,COLDIM,ROWDIM
  REAL*8 VALUE(TOTDIM,5)
C-----C

```

```

C      BUILD THE MATRIX OF COEFFICIENTS
C-----C
C
C      BUILD THE FIRST COLUMN
C
DO 120 I=1,TOTDIM
  J = 1
  VALUE(I, J) = VALUE(I, 1)
  IF (I.LT.COLDIM) VALUE(I,J) = 0.0
  DO 130 N=1,ROWDIM -1
    JN = N * COLDIM
    IF ((I.EQ.JN).OR.(I.EQ.JN+1)) VALUE(I,J) = 0.0
130  CONTINUE
C  CHANGE FROM +1 TO +2 FOR CONSTANT LEFT BOUNDARY
    IF(I.GE.TOTDIM-COLDIM+1) VALUE(I,J) = -1.0
120  CONTINUE

C-----C
C      BUILD THE SECOND COLUMN
C-----C
DO 140 I=1,TOTDIM
  J = 2
  VALUE(I, J) = VALUE(I, 2)
  DO 150 N=1,ROWDIM
    JN = N * COLDIM
    IF (I.EQ.JN+1) VALUE(I,J)=0.0
    IF (I.EQ.JN) VALUE(I,J)=-VALUE(I,3)*.25
150  CONTINUE
    IF (I.GE.TOTDIM-COLDIM+1) VALUE(I,J)=0.0
    VALUE(1,2)=0.0
140  CONTINUE

C-----C
C      BUILD THE THIRD COLUMN      I rows J columns
C
C-----C
C This takes care of upper left corner by setting central = right coeff
  VALUE(1,3) = VALUE(1,3)*.25
  DO 160 I=2,TOTDIM
    J = 3
    VALUE(I,J) = -VALUE(I,3)
    DO 170 N=1,ROWDIM-1
      JN = N * COLDIM
      IF (I.EQ.JN) VALUE(I,J) = -VALUE(I,3)*.25
C      IF ((I.EQ.JN).AND.(I.EQ.COLDIM))
C 1      VALUE(I,J) = -VALUE(I,3)*4.0
      IF (I.EQ.JN+1) VALUE(I,J) = -VALUE(I,3)*.25
      IF ((I.EQ.JN+1).AND.(I.EQ.(TOTDIM-COLDIM+1)))
C 1      VALUE(I,J) = -VALUE(I,3)*4.0
170  CONTINUE
C      IF (I.LE.COLDIM) VALUE(I,J) = -VALUE(I,3)*.25
      IF (I.GE.TOTDIM-COLDIM+1) VALUE(I,J) = 1.0

```



```

C      Air thermal cond. = .000139 W / meter K      C
C      Surface coefficient of heat transfer (.05 to 25 W / m^2 K)      C

C      Do first row (surface)      C
C      Define conductive coeff. for fictitious upper node      C
C      VALUE (I,1) SET EQUAL TO 0.0 FOR < COLDIM BY COEFF
SURFKOND = -( VALUE (I,3) *.25) * DY2
VALUE(I,1) = (SURFKOND / DY2)

C      This adds lower and upper coeffs for one pt below surface      C
VALUE(I,5) = VALUE(I,5) + VALUE(I,1)

C      This modifies central coeff.      C
C      NEXT TWO LINES USE SURFKOND
      VALUE(I,3) = (VALUE(I,3)) -
&                ((2.0 * DELY * SURFCOEFF)/SURFKOND)*VALUE(I,1)

C      Modify inhomogeneous vector      C
      BOUND(I) = ((-2.0 * DELY * SURFCOEFF)/ SURFKOND)
&                * VALUE(I,1) * (TSURF)

      RETURN
      END

```

```

C!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!C
      SUBROUTINE SORITERATE(VALUE, BOUND, T, COLDIM,
&                          TOTDIM, MAXITS, CONVCRIT)
C-----C
C      GAUSS SIEDEL WITH SOR      C
C-----C
      INTEGER I,K,N, TOTDIM, COLDIM, MAXITS
      REAL*8 VALUE(TOTDIM, 5), T(TOTDIM), BOUND(TOTDIM)
      REAL*8 ERR, TOLD, TSUM, SUM, SOR, CONVCRIT
      PRINT *, 'Input the relaxation constant (0<x<2). '
      READ *,SOR
      WRITE (8,57) SOR
57  FORMAT ('The relaxation constant is ',F5.2)
      WRITE (8,41) CONVCRIT
41  FORMAT (// 'The convergence criteria is ',1PD14.4)

      K = 1
320  IF (K.GT.MAXITS) GOTO 900
      IF (MOD(K,100) .EQ. 0) THEN
          PRINT *, '    K = ',K, '      err = ',err
          ENDIF
      IF (MOD(K,25) .EQ. 0) THEN
          WRITE (11,42) K,ERR
          ENDIF
42  FORMAT (I6,E12.2)

```

```

ERR=0.0
N = COLDIM
DO 330 I=1,TOTDIM
  IF (I .EQ. 1) THEN
    SUM = VALUE(I,4) * T(I+1) + VALUE(I,5) * T(I+N)
  ELSEIF ((I .LE. COLDIM) .AND. (I .GT. 1)) THEN
    SUM = VALUE(I,2) * T(I-1) + VALUE(I,4) * T(I+1)
    &      + VALUE(I,5) * T(I+N)
  ELSEIF ((I .GT. (TOTDIM - COLDIM)) .AND. (I .LT. TOTDIM)) THEN
    SUM = VALUE(I,1) * T(I-N) + VALUE(I,2) * T(I-1)
    &      + VALUE(I,4) * T(I+1)
  ELSEIF (I .EQ. TOTDIM) THEN
    SUM = VALUE(I,1) * T(I-N) + VALUE(I,2) * T(I-1)
  ELSE
    SUM = VALUE(I,1) * T(I-N) + VALUE(I,2) * T(I-1)
    &      + VALUE(I,4) * T(I+1) + VALUE(I,5) * T(I+N)
  END IF

  TSUM = (BOUND(I) - SUM) / VALUE(I, 3)
  TOLD = T(I)
  T(I) = (1.0D0 - SOR) * T(I) + (SOR * TSUM)
  ERR=DABS(T(I)-TOLD)+ERR
330  CONTINUE
  K = K + 1
  IF (ERR .GT. (CONVCRIT)) GOTO 320
  PRINT *, 'Number of iterations = ', K
800  WRITE (8,40) K
  40  FORMAT(/'The number of iterations is: ',I4)
  GOTO 910
900  WRITE (8,50)
  50  FORMAT(/'MAXIMUM NUMBER OF ITERATIONS EXCEEDED. ',I3)
  GOTO 800
910  CONTINUE

  RETURN
  END

```

```

C!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!C
  SUBROUTINE CGS(A,B,NCD,NTD,NB,X,R,Q,U,P,V,Z,W,Y,
  &              MAXITS,CONVCRIT)

```

```

C-----C
C
C THIS ROUTINE PERFORMS A MULTIPLICATION OF AN ARBITRARY
C VECTOR ( RHS ) WITH THE APPROXIMATE INVERSE FOUND
C-----C
C

```

```

  REAL*8 A(NTD,NB),B(NTD),X(NTD),R(NTD),Q(NTD),U(NTD),P(NTD),

```



```

&    V(NTD),Z(NTD),W(NTD),Y(NTD)
REAL*8 RES,SUM,SUM1,SUM2,A1,B1,CONVCRT
INTEGER ITER

```

```

C
C      A SOLVER....
C-----C
    ITER=0
    CALL MULAXC(A,NCD,NTD,NB,X,W)
    DO 3 I=1,NTD
        W(I) = B(I)-W(I)
    3    CONTINUE

    CALL SOLL(A,NCD,NTD,NB,R,W)
    DO 1 I=1,NTD
        Z(I) = R(I)
        P(I) = 0.0D0
        Q(I) = 0.0D0
    1    CONTINUE
    SUM2 = 1.0D0
12    SUM = 0.0D0
    SUM1 = 0.0D0
    ITER = 1+ITER
    DO 4 I=1,NTD
        SUM = SUM + Z(I)*R(I)
    4    CONTINUE
    A1 = SUM/SUM2
    SUM2 = SUM
    DO 5 I=1,NTD
        U(I) = R(I)+A1*Q(I)
        P(I) = U(I)+A1*(Q(I)+A1*P(I))
    5    CONTINUE
    CALL SOLU(A,NCD,NTD,NB,V,P)
    CALL MULAXC(A,NCD,NTD,NB,V,W)
    CALL SOLL(A,NCD,NTD,NB,V,W)
    SUM1 = 0.0D0
    DO 6 I=1,NTD
        SUM1 = SUM1 + Z(I)*V(I)
    6    CONTINUE
    B1 = SUM/SUM1
    DO 7 I=1,NTD
        Q(I) = U(I)-B1*V(I)
        W(I) = Q(I)+U(I)
    7    CONTINUE
    CALL SOLU(A,NCD,NTD,NB,V,W)
    CALL MULAXC(A,NCD,NTD,NB,V,Y)
    CALL SOLL(A,NCD,NTD,NB,W,Y)
    RES = 0.0D0
    RESx=0.0D0
    DO 17 I=1,NTD
        X(I) = X(I)+V(I)*B1
        R(I) = R(I)-W(I)*B1
        RES = RES+DABS(R(I))

```

```

      RESx = RESx+DABS(V(I)*B1)
17  CONTINUE
      PRINT*, '      Residual,  Iteration      ', RESx, ITER
800  FORMAT (I5,E12.2)
      WRITE (10,800) ITER, RESx
      IF (RESx .LT. CONVCRIT) GOTO 8
      IF (ITER .GT. MAXITS) GOTO 8
C      ELSE
C          GO TO 12
C      ENDIF
      8 RETURN
      END

```

```

C!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!C
      SUBROUTINE MULAXC(A,NCD,NTD,NB,XX,AX)
C
C-----C
C
C  THIS ROUTINE MULTIPLIES THE NON-SYMMETRIC NESTED
C
C  FACTORIZATION MATRIX WITH A GIVEN VECTOR
C
C      A    - THE MATRIX OF BANDS...STORED FROM LOWEST SUBDIAGONAL
C             TO HIGHEST SUBDIAGONAL  THE MIDDLE COLUMN OF A BEING
C             THE MAIN DIAGONAL
C      XX    - THE VECTOR TO BE MULTIPLIED
C      AX    - THE VECTOR TO BE RETURNED
C
C-----C
C
C      REAL*8      XX(NTD) , AX(NTD) , A(NTD,NB)
C
C-----C
C
C      WE MULTIPLY THE MAIN DIAGONAL FIRST...
C
C-----C
C
C      DO 2 I = 1,NTD
C          AX(I) = A(I,3) * XX(I)
C          print*,c(i),i,'in mulax 2'
C      2 CONTINUE
C
C-----C
C
C      THEN DO THE FIRST LOWER AND UPPER SUBDIAGONALS
C
C-----C
C

```

```

DO 3 I = 2,NTD
  AX(I-1) = AX(I-1) + A(I-1,4) * XX(I)
  AX(I) = AX(I) + A(I,2) * XX(I-1)
  print*,c(i-1),c(i),i,i-1,'in mulax 3'
3 CONTINUE

```

```

C-----C
C
C AND THEN THE OUTER MOST SUBDIAGONALS
C-----C
C

```

```

DO 4 I = 1,NTD-NCD
  AX(I) = AX(I) + A(I,5) * XX(I+NCD)
  AX(I+NCD) = AX(I+NCD) + A(I+NCD,1) * XX(I)
  print*,c(i+ncd),x(i),c(i), i,i+ncd,'in mulax 4'
4 CONTINUE

```

```

C
  RETURN
  END

```

```

C!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!C
  SUBROUTINE SOLL(A,NCD,NTD,NB,XX,RT)

```

```

C-----C
C
C SOLVES (L + D) XX = RT
C   L LOWER PART OF A, D= DIAGONAL OF A, RT = RIGHT HAND SIDE
C-----C

```

```

  REAL*8 A(NTD,NB) , RT(NTD) , XX(NTD)
C-----C

```

```

C
C THE FIRST BLOCK:
C-----C

```

```

C
C   A FORWARD SOLVER....
C-----C

```

```

  XX(1) = RT(1) / A(1,3)
  DO 2 I = 2,NCD
    XX(I) = ( RT(I) - A(I,2) * XX(I-1) ) / A(I,3)
  2 CONTINUE

```

```

C-----C
C
  DO 3 I = NCD+1,NTD

```

```

      XX(I) = ( RT(I) - A(I,2) * XX(I-1)
&      - A(I,1) * XX(I-NCD) ) / A(I,3)
3 CONTINUE
  RETURN
  END

```

```

C!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!C
  SUBROUTINE SOLU(A,NCD,NTD,NB,XX,RT)
C-----C
C
C      SOLVES (I+D-1*U)XX = RT
C      D-1 = DIADIONAL INVERSE U = UPPER PART OF A
C-----C
C
C      REAL*8 A(NTD,NB) , RT(NTD) , XX(NTD)
C
C      A BACKWARD SOLVER....
C-----C
C
C      XX(NTD) = RT(NTD)
C      DO 2 I = NTD-1,NTD-NCD+1,-1
C        XX(I) = RT(I) -( A(I,4) * XX(I+1) ) / A(I,3)
C      2 CONTINUE
C-----C
C
C      DO 3 I = NTD-NCD,1,-1
C        XX(I) = RT(I)+( - A(I,4) * XX(I+1)
&        - A(I,5) * XX(I+NCD) ) / A(I,3)
C      3 CONTINUE
C      RETURN
C      END

```

Appendix B

Sample input file HEATDAT.

8.0
67.00
50.0
50.0
50.0
0
25.0
.0001
2000

Surface Temperature (C)
Heat Flow (milliwatts/meter²)
Horizontal Spacing (meters)
Horizontal spacing
horizontal spacing
Spacing Change (X-direction)
Vertical Spacing (meters)
Convergence Criteria
Maximum Iterations (Integer value)

Appendix C

Sample input file CROSSDAT.

1

[illegible]

Appendix D

Measured temperature-depth data for the Thermopolis area.

Well: UWT1

Measured: June, 1982

Depth (m)	Temp. (°C)
9.3	33.580
19.3	40.280
29.3	45.670
39.3	50.740
49.3	54.920
59.3	58.346
69.3	60.167
79.3	63.156
89.3	63.963
97.8	64.315

Measured: December, 1990

Depth (m)	Temp. (°C)
6.1	29.036
12.2	43.999
21.3	50.425
30.5	51.151
39.6	52.754
48.8	55.519
57.9	56.674
67.1	58.603
76.2	61.929
79.2	63.114
82.0	63.321

Well: UWT2

Measured: June, 1982

Depth (m)	Temp. (°C)
9.1	22.110
19.1	23.040
29.1	25.060
39.1	27.830
49.1	31.160
59.1	34.830
69.1	38.530
79.1	42.910
89.1	46.090
99.1	50.130
109.1	55.160
119.1	60.630
129.1	67.700
131.1	71.650
139.1	72.137
159.1	72.257
169.1	72.347
189.1	72.478
198.1	72.497

Measured: December, 1990

Depth (m)	Temp. (°C)
3.0	16.910
6.1	18.999
9.1	19.990
12.2	22.058
15.2	23.612
18.3	24.801
21.3	25.831
27.4	27.992
36.6	31.154
45.7	34.837
54.9	39.011
64.0	42.651
73.1	46.400
76.2	47.550
82.3	49.834
91.4	53.637
101.2	57.718
131.4	71.454
137.2	71.780
143.2	71.827
149.3	71.854
155.4	71.873
161.5	71.896
167.6	71.922
173.7	71.943
179.8	71.967
185.9	72.014
192.0	72.027
198.1	72.136
200.2	72.158

Well: UWT3

Measured: June, 1982

Depth (m)	Temp. (°C)
9.2	35.930
19.20	42.590
29.20	45.562
39.20	51.804
49.20	52.914
59.20	54.502

Measured: December, 1990

Depth (m)	Temp. (°C)
3.0	20.583
9.1	26.149
18.3	32.946
24.4	40.637
30.5	43.746
35.7	49.304
39.6	50.176
45.7	50.790
51.8	50.880
57.9	51.167