PROCEEDINGS, Seventeenth Workshop on Geothermal Reservoir Engineering
Stanford University, Stanford, California, January 29-31, 1992
SGP-TR-141

A Fast Method for Evaluating A Simplified Hot Dry Rock Heat Flow Problem

R.G. Adair
Susan Petty Consulting
703 Santa Paula
Solana Beach, CA 92075

ABSTRACT

I present optimizations to the computation of Elsworth's single zone, hot dry rock thermal recovery model. These enhancements lead to as much as a 6-fold increase in computational speed. The greatest time savings derive from an efficient evaluation of the model's thermal response due to a step in heat flux, which is required for solution of the more general problem via Duhamel's Principle. Further enhancements come from taking advantage of the special structure of the model's finite difference equation.

Reductions in execution speed were sought in order to facilitate the model's implementation on AT-class microcomputers. The PC-based application requires multiple evaluations of the model. Typical execution times on a 33 MHz 80386 microcomputer for 128 time steps were 7 seconds, as compared with 25-42 seconds for the non-optimized approach, and for 512 time steps were 28 and 100-168 seconds, respectively; the timing of the non-optimized method depended upon particulars of the dimensionless variables.

INTRODUCTION

DOE is sponsoring the development of software tools which estimate the impact of research and development on the cost of geothermal power generation [*Petty et al.*, 1988]. Although a mature tool for hydrothermal resources exists, work on geopressured and hot dry rock resources continues. A goal for this software is that it execute in a reasonable amount of time on AT-class microcomputers. To this end, computational enhancements were sought for the single zone, hot dry rock thermal recovery model of *Elsworth* [1989a], which was adopted for the hot dry rock software tool. Figure (1) shows a typical prediction of Elsworth's model for parameters assumed to be appropriate for the Fenton Hill Hot Dry Rock Project [*Robinson and Kruger*, 1988], which Table (1) lists. The results illustrate the difference in magnitude of temperature change with time for a large (800 m) single reservoir compared to a small (200 m) reservoir. Note that in all of the cases the temperature has dropped below a 150° C temperature by the end of a typical 30 year project life.

This paper describes the enhancements to the methods of *Elsworth* [1989a]. The greatest savings in time come from the efficient evaluation of the step response of the spherical reservoir, including its analytic evaluation, which Elsworth computes numerically. Further savings come from exploiting the special structure of the finite difference equations which approximate the pertinent differential equation. The resulting matrix equation was amenable to a fast inversion method which also has extremely modest memory requirements. In addition, given one solution, a solution correct to first order in small perturbations to the dimensionless variables can be computed with substantially less effort that the exact solution.

Elsworth [1989b] has since expanded his model to include multiple porous zones. The methods presented here can be applied with minor modification to *Elsworth* [1989b].

ELSWORTH'S MODEL

Statement of the Problem

The theory which *Elsworth* [1989a] presents will only be summarized here. Conceptually, heat is extracted from a porous, spherical inclusion in an otherwise uniform, infinite half space. Water of a given initial temperature circulates through the sphere and returns at a time varying temperature which the flow rate and the reservoir porosity dictate. The inclusion and half space are in thermal equilibrium with each other prior to the circulation of water.

Elsworth assumes that the circulating water immediately attains equilibrium with the reservoir upon entry, a reasonable assumption given the low thermal conductivity of most host rocks. To render the problem tractable, Elsworth further considers only the temperature averaged over the surface of the sphere.

Given this model and its assumptions, Elsworth shows how to compute the time variation of the average temperature from the step heat flow response of a sphere in an infinite space. The solution in an infinite half space follows from the method of images to satisfy either zero heat flux or constant temperature at the surface of the infinite half space. Application of Duhamel's Principle then defines the heat flux variation into the half space at the sphere's surface in terms of the temperature rate. The following differential equation describes the energy balance between the semi-infinite heat reservoir and the spherical inclusion:

$$\begin{aligned}\langle q_T\rangle &= q_F\rho_F c_F\langle T_o(t)-T_i\rangle+\frac{4\pi a^3\rho_S c_S}{3}\frac{\partial}{\partial t}\langle T_o\rangle \\ &= 4\pi a\int_0^t C(t-\tau)\frac{\partial}{\partial\tau}\langle T_R-T_o(\tau)\rangle\, d\tau \end{aligned}\qquad (1)$$

$$\langle T_o(t=0)\rangle = T_R$$

or in terms of dimensionless variables, assuming that the inlet and initial half space temperatures T_i and T_R are constant, and after some rearrangement,

$$\frac{Q_D}{4\pi}T_D+\frac{\Phi_D}{3}\frac{\partial T_D}{\partial t_D}+\left(\frac{\partial T_D}{\partial t_D}\right)\star_D C=0,\quad t_D>0 \quad (2)$$

$$T_D=1,\quad t_D=0. \quad (3)$$

Table 2 defines the dimensionless parameters in terms of physical parameters, which are themselves defined in Table 1. The function C above is the reciprocal of the sphere's step heat flux response, which depends on dimensionless sphere radius $\hat{a}=a/z$ and dimensionless time $t_D=K_R t/\rho_R c_R a^2$. The notation $\star_D$ denotes convolution performed in dimensionless time. The notation $<\cdot>$ denotes an average of the bracketed quantity over the spherical surface.

The density and thermal capacity of the spherical reservoir here are defined in terms of the corresponding properties of the fluid and rock and its porosity ϕ:

$$\rho_S\, c_S=(1-\phi)\rho_R\, c_R+\phi\, \rho_F\, c_F. \quad (4)$$

The convolutional integral in these equation builds the general heat flux solution in terms of the temperature variation on the boundary using the basic building block provided by the function C.

Finite Difference Approximation to the Differential Equation

If time is discretized in Equation (2) over N time steps $k\ \Delta t_D$, $k=1,...,N$, and the discrete versions of derivatives and integrals are employed, then the following matrix equation in terms of dimensionless parameters results:

$$\left[\Delta t_D\frac{Q_D}{4\pi}\mathbf{I}+\frac{\Phi_D}{3}\boldsymbol{\partial}+\boldsymbol{\partial}\mathbf{L}(\vec{C})\right]\vec{\mathrm{T}}_D = \left(\vec{C}+\frac{\Phi_D}{3}\hat{\mathbf{e}}_1\right)T_0 \quad (5)$$

Here, $\mathbf{I}$ is the order N identity matrix; $\boldsymbol{\partial}$ is a matrix which is all ones along the main diagonal and all -1's along the first sub-diagonal; $\mathbf{L}(\vec{C})$ is a lower triangular matrix, which has the vector $\vec{C}$ as it first column, constant entries along the main and sub-diagonals, and all zeros above the main diagonal; $\vec{C}$ is a column vector with k'th element which is the integral of the reciprocal step response C evaluated between dimensionless times $(k-1)\Delta t_D$ and $k\Delta t_D$; $\vec{\mathrm{T}}_D$ is a column vector of the dimensionless temperature values, so that element k is the dimensionless temperature at time step k; and $\hat{\mathbf{e}}_1$ is a unit column vector of all zeros except for the first element, which is one. T_0 is the initial value of the dimensionless temperature, which by definition is just 1. In the following, this value is inserted, and T_0 is dropped.

SOLUTION OF ELSWORTH'S PROBLEM

Computing the Step Response

The step response C describes the simplest heat flux response of Elsworth's model. Its reciprocal is the temperature variation, averaged over the sphere's surface, due to a step in heat flux over that surface. This reciprocal is the sum of the self-heat C_1 due to the sphere, and the heat C_2 due to the image source, which maintains boundary conditions of either constant flux or temperature at the surface of the earth. The image source heat is expressible in polar spherical coordinates as an integral over co-latitude angle. Elsworth computes this integral numerically. However, as I show in Appendix A, it may be expressed in closed form, and therefore need not be numerically evaluated.

While the closed form evaluation of the integral C_2 provides a savings in numerical effort, as well as an assurance of accuracy, there still remains the need to integrate $1/(C_1 \pm C_2)$ over the intervals of discrete time in the finite difference equation. Consideration of the behavior of the step response at small and large times suggests approximations, described in Appendix A, which require only 10%-20% of the computational effort as the exact expression, depending upon the boundary condition. These approximations, which are sums of exponentials in $\log t_D$, provide excellent fits over the entire allowable ranges of dimensionless time and radius.

Inverting the Finite Difference Matrix

The matrices of Equation (5), $\mathbf{I}$, $\boldsymbol{\partial}$, and $\mathbf{C}$, have special structures. First note that the entries along any diagonal are constant, which is the defining feature of a *Toeplitz* matrix. Second, all diagonals above the main diagonal are zeros, making them *lower triangular* matrices. Thus, the first column, or last row, completely specify a lower triangular Toeplitz matrix. As I show in Appendix B, such a matrix can be inverted with an effort proportional to $N \log_2 N$ floating point operations, or `flops`, where N is the number of time steps and where one `flop` may be roughly defined as a computational effort which requires one multiplication or division of two floating point numbers, plus the effort to add or subtract two floating point numbers. The time required to finish a software task is generally proportional to the number of `flops` expended. The next best method, back substitution, requires $N(N+1)/2$ `flops`. The fast method derives its speed from the fact that the matrix multiplications can be computed from convolutions of the first columns of the matrices, which in turn can be done quickly with Fast Fourier Transforms. The fastest implementation of this method requires that N be an integer power of 2.

In principal this algorithm is both fast and concise. In practice, the overhead associated with the partitioning and the various FFT-aided convolutions keeps this algorithm from surpassing back substitution for $N \leq 128$ time steps. Table 3 shows a comparison of the back substitution and FFT-based methods, averaged over 4 trials, applied to random Toeplitz matrices of various sizes to solve the matrix equation $\mathbf{A}\vec{x} = \vec{y}$ for vector $\vec{x}$. These numbers were computed with The MathWorks' matrix manipulation software AT-MATLAB^{TM} on a 33 mHz 80386-based PC using a math coprocessor. The table gives the required execution time and the computational effort, which MATLAB provides. For a length of 1024 points, the FFT-based method took 6 seconds, as compared with 40 seconds for the back substitution method, a factor of nearly 7 faster. Note also, in this particular case, that the results for $N = 256$ and 512 show that the number of `flops` required do not necessarily dictate the execution time. The FFT-based method requires 1.5 to 3 times more `flops` than the back substitution method, yet executed faster. This is because MATLAB provides optimized code for power-of-2 length FFT's.

Solution for Small Perturbations

Suppose that the reservoir parameters Q_D, Φ_D and $\vec{C}$ are subjected to a small perturbation δQ_D, $\delta\Phi_D$ and $\delta\vec{C}$. Denote the lower triangular Toeplitz matrix generated by the unperturbed values as $\mathbf{H}(Q_D, \Phi_D, \vec{C})$. Then the resulting temperature perturbation $\delta\vec{T}_D(\delta Q_D, \delta\Phi_D, \delta\vec{C})$, correct to first order in the perturbations, can be found from the solution $\vec{T}_D(Q_D, \Phi_D, \vec{C})$ and the inverse matrix $\mathbf{H}^{-1}(Q_D, \Phi_D, \vec{C})$ with much less effort than the exact solution. This perturbation can be found from the Taylor series of $\vec{T}_D(x + \delta x)$, where x is a scalar:

$$\vec{T}_D(x + \delta x) = \vec{T}_D(x) + \delta x \frac{\partial}{\partial x}\vec{T}_D(x) + \mathcal{O}((\delta x)^2) \, , \quad (6)$$

so that to first order in δx,

$$\delta\vec{T}_D \;=\; \delta x \frac{\partial}{\partial x}\vec{T}_D(x) \, . \quad (7)$$

Recalling that $\vec{T}_D = \mathbf{H}^{-1}\vec{y}$, where $\vec{y} = \vec{C} + \hat{\mathbf{e}}_1\Phi_D/3$, and using the result

$$\frac{\partial}{\partial x}\mathbf{H}^{-1}(x) = -\mathbf{H}^{-1}\,\frac{\partial}{\partial x}\mathbf{H}(x)\,\mathbf{H}^{-1} \, , \quad (8)$$

(see, for example, *Gradshteyn and Ryzhik* [1980, p 1107]) the perturbation to the temperature is, after some rearrangement,

$$\delta\vec{T}_D = \delta x \left[\mathbf{H}^{-1}\frac{\partial}{\partial x}\vec{y} - \mathbf{H}^{-1}\left(\frac{\partial}{\partial x}\mathbf{H}\right)\vec{T}_D\right] \, . \quad (9)$$

Consider first a perturbation δQ_D to the dimensionless flow rate. From the last equation,

$$\delta\vec{T}_D(\delta Q_D) = -\Delta t_D \frac{\delta Q_D}{4\pi}\mathbf{H}^{-1}\vec{T}_D \, , \quad (10)$$

where the parameter dependence is on the unperturbed values unless otherwise explicitly stated. This matrix multiplication requires only one convolution to compute.

Consider next a perturbation $\delta\Phi_D$ to the dimensionless porosity. Proceeding as above,

$$\delta\vec{T}_D(\delta\Phi_D) = \frac{\delta\Phi_D}{3}\left[\mathbf{H}^{-1}\hat{\mathbf{e}}_1 - \boldsymbol{\partial}\mathbf{H}^{-1}\vec{T}_D\right] \, , \quad (11)$$

where the result $\mathbf{H}^{-1}\boldsymbol{\partial} = \boldsymbol{\partial}\mathbf{H}^{-1}$ was used. This perturbation can also be performed with a single convolution, as follows. First, note that $\mathbf{H}^{-1}\hat{\mathbf{e}}_1$ is just the first column of $\mathbf{H}^{-1}$, $\vec{x}$, say, so that no computations at all are required for this operation. Next, note that the matrix $\boldsymbol{\partial}\mathbf{H}^{-1}$ can be computed from the first differences of $\mathbf{H}^{-1}$. No recourse to convolution is needed to compute this. Thus, only the convolution of $\vec{T}_D$ with this vector difference is required.

Finally, suppose a perturbation $\delta\vec{C}$ to the step response. Considering the individual perturbations due to each element of $\delta\vec{C}$ in Equation (8), the aggregate first order perturbation to $\vec{T}_D$ is

$$\delta\vec{T}_D(\delta\vec{C}) = \mathbf{H}^{-1}\delta\vec{C} - \boldsymbol{\partial}\,\mathbf{H}^{-1}\,\mathbf{L}(\delta\vec{C})\,\vec{T}_D\,, \qquad (12)$$

where the commutability of the matrices were applied. This operation requires two convolutions. The first convolution yields the vector $\mathbf{H}^{-1}\delta\vec{C}$. Then, the matrix product $\boldsymbol{\partial}\,\mathbf{H}^{-1}\,\mathbf{L}(\delta\vec{C})$ is found from first differences of the vector $\mathbf{H}^{-1}\delta\vec{C}$, without recourse to convolution. A second convolution yields the product of this difference operation with $\vec{T}_D$.

If sufficiently small perturbations occur to all quantities, the net result is just the sum of these three individual results.

DISCUSSION

The ideas of this paper were implemented with Microsoft's QUICKBASIC 4.5. Recursive calls were not used to invert the Toeplitz matrix. To facilitate comparison, Elsworth's original FORTRAN code was translated to QUICKBASIC 4.5. The execution time of the original code was sensitive to the particulars of the problem, while that of the optimized code was not. Several runs of 128 time steps on a 33 MHz 80386 clone without benefit of a math coprocessor yielded average execution times of 6.7 seconds for the enhanced code. On the other hand, the original code times ranged between 24.1 and 41.8 seconds, with time increasing as with the product $Q_D t_D$. Runs of 512 points required 27 seconds for the optimized code and 107—168 seconds for the original code. These results attest the success of the streamlining efforts. Furthermore, use of the perturbation scheme yielded new solutions in less than one second, typically.

Application to Multiple Zones

Elsworth [1989b] extends the single zone problem to describe a multiple zone hot dry rock resource. The discrete problem can be cast as a lower triangular *block* Toeplitz matrix equation, where the fixed diagonal entries are matrices rather than scalars. Each of the matrix entries are square matrices of order M which describe the mutual effects of the M subzones on one another at a given time step, while the temperature vector comprises subvectors of length M which give the average temperature of the M source zones at a given time step. The mutual effects of each zone play the role of the image source in the single source model. The Toeplitz matrices in Equation (5) carry over in a straightforward manner to the multiple zone problem. For example, the 1's in the matrix $\boldsymbol{\partial}$ become identity matrices of order M. It may be shown that each of the special properties of the scalar problem carry over into the matrix problem (see *Bitmead and Anderson* [1980]). In particular, the inverse matrix is also lower triangular block Toeplitz, so that its first column of matrices entries completely describe it.

Although in general these matrix entries need have no special structure, the assumptions of *Elsworth* [1989b] result in each being a symmetric Toeplitz matrix. To render this problem tractable, Elsworth assumes that the resource comprises several proximate single zones in an infinite whole space, and that the size and thermal properties of each zone are identical. This special structure means that matrix multiplications can be performed with convolutions, either directly for small M or by FFT's for larger M.

ACKNOWLEDGEMENTS

This work described in this paper was funded in part by Sandia National Laboratory Contract 66-8860 Derek Elsworth generously provided the advice, software and articles on which this paper is based. This paper would not have been possible without the comments and encouragement of S. Petty.

REFERENCES

Bitmead, Robert R., and Anderson, Brian D.O. Asymptotically fast solution of Toeplitz and related systems of linear equations. *Linear Algebra and its Applications, 34*(1), 103—116, 1980.

Elsworth, Derek. Theory of thermal recovery from a spherically stimulated hot dry rock reservoir. *J. Geophys. Res., 94*(B2), 1927—1934, 1989a.

——— Thermal recovery from a multiple stimulated HDR reservoir. *Geothermics, 18*(5/6), 761—774, 1989b.

Gradshteyn, I.S., and Ryzhik, I.M. Tables of Integrals, Series, and Products. 1160 pp. (New York: Academic Press) 1980.

Petty, S., Entingh, D., Livesay, B. Impact Of R&D On Cost of Geothermal Power. Documentation of Model Version 2.09. Sandia Report SAND87-7018, February, 1988.

Press, W.H., Flannery, B.P., Teukolsky, S.A., and Vetterling, W.T. Numerical Recipes. The Art of Scientific Computing. 818 pp. 3rd ed. (Cambridge: Cambridge University Press) 1986.

Robinson, B.A., and Kruger, P. A comparison of two heat transfer models for estimating thermal drawdown in hot dry rock reservoirs. Proceedings, 13th Workshop on Geothermal Reservoir Engineering, Stanford University, January 19—21, 1988.

APPENDIX A
COMPUTING THE STEP RESPONSE

Analytic Solution

Elsworth [1989a] characterizes the thermal recovery of the hot dry rock resource by the average temperature variation $\langle T_R - T_o \rangle$ over the spherical reservoir surface:

$$\langle T_R - T_o \rangle = \frac{1}{4\pi a K_R}(C_1 \pm C_2), \qquad \text{(A1)}$$

where

$$C_1(t_D) = 1 - e^{t_D}\,\text{erfc}(\sqrt{t_D}), \qquad \text{(A2)}$$

and

$$C_2 = \frac{1}{2}\int_0^\pi \frac{\sin\theta d\theta}{\hat{r}(\theta)}\left\{\text{erfc}\left(\frac{\hat{r}-1}{2\sqrt{t_D}}\right) - e^{(\hat{r}-1+t_D)}\,\text{erfc}\left(\frac{\hat{r}-1}{2\sqrt{t_D}} + \sqrt{t_D}\right)\right\} \qquad \text{(A3)}$$

C_1 describes the thermal flux due to the sphere, while C_2 describes that due to a fictitious image source at distance $2z$ from the sphere, where z is the depth of the sphere in the half space. The image source maintains either constant flux (C_2 is added) or constant temperature (C_2 is subtracted) at the surface of the infinite half space. In these equations, t_D is dimensionless time, and $\hat{r} = r/a$, where r is the distance from the image source to a particular spot on the sphere's surface. The colatitude angle θ completely specifies r in a spherical coordinate system centered on the sphere with the vertical axis along the line to the image source,

$$r(\theta) = a\sqrt{1 + 4/\hat{a}^2 - (4\cos\theta)/\hat{a}}, \qquad \text{(A4)}$$

where $\hat{a} = a/z$.

Elsworth evaluates C_2 numerically; however, it has a closed form evaluation. Note that, from the above equation, $d\theta = d\hat{r}\ \hat{a}\hat{r}/2\sin\theta$, so that for any function $f(\hat{r}(\theta))$,

$$\int_0^\pi f(\hat{r}(\theta))\,\frac{\sin\theta d\theta}{\hat{r}(\theta)} = \frac{\hat{a}}{2}\int_{\hat{r}(0)}^{\hat{r}(\pi)} f(\hat{r})\,d\hat{r}. \qquad \text{(A5)}$$

Thus, the angular integral for C_2 is equivalent to one in $\hat{r}$,

$$C_2 = \frac{\hat{a}}{4}\int_{\hat{r}(0)}^{\hat{r}(\pi)}\left[\text{erfc}\left(\frac{\hat{r}-1}{2\sqrt{t_D}}\right) - e^{(\hat{r}-1+t_D)}\left(\text{erfc}\frac{\hat{r}-1}{2\sqrt{t_D}} + \sqrt{t_D}\right)\right] d\hat{r}. \qquad \text{(A6)}$$

The solution of this integral is now straightforward, e.g. using integration by parts:

$$C_2 = \frac{1}{4}\left\{(\hat{a}+2)\text{erfc}\left(\frac{1}{\hat{a}\sqrt{t_D}}\right) + (\hat{a}-2)\text{erfc}\left(\frac{1-\hat{a}}{\hat{a}\sqrt{t_D}}\right) + \frac{2\hat{a}\sqrt{t_D}}{\sqrt{\pi}}\left(e^{-(1-\hat{a})^2/\hat{a}^2 t_D} - e^{-1/\hat{a}^2 t_D}\right) + \hat{a}\left[e^{(2(1-\hat{a})/\hat{a}+t_D)}\,\text{erfc}\left(\frac{1-\hat{a}}{\hat{a}\sqrt{t_D}} + \sqrt{t_D}\right) - e^{(2/\hat{a}+t_D)}\,\text{erfc}\left(\frac{1}{\hat{a}\sqrt{t_D}} + \sqrt{t_D}\right)\right]\right\}. \qquad \text{(A7)}$$

Approximating the Analytic Solution

Approximations to the reciprocal step response C are presented here which facilitate the rapid evaluation of both C and its integral over intervals of discrete time. At small dimensionless times $t_D \leq 10^{-3}$, the self heat term dominates with a value of approximately $2\sqrt{t_D/\pi}$, so that the integral of the reciprocal presents an integrable singularity. At larger dimensionless times $1/(C_1 \pm C_2)$ is well approximated by sums of exponential terms with exponents which are quadratic in the logarithm of time. The forms of the curve fits are constrained by the values of the step responses at large times. The integral of such a fit can generally be expressed as the sum of a linear term and erfc functions, but simple trapezoidal rule integration of the erfc terms was found to be sufficiently accurate for typical values of $\Delta t_D \leq 1$.

In the case of a constant flux boundary condition, where the step response is $C_+ = C_1 + C_2$, the functional fit for $t_D > 10^{-3}$ is

$$Q_+(x) = ax^2 + bx + c = \log\left(\frac{\alpha_+}{C_+} - 1\right), \qquad \text{(A8)}$$

where $x = \log t_D$ and $\alpha_+ = 1 + \hat{a}/2$, so that

$$\int_{(k-1)\Delta t_D}^{k\Delta t_D}\frac{dt_D}{C_+} \approx \frac{\Delta t_D}{\alpha_+} + \frac{1}{2\alpha_+}\left(e^{Q_+(x_{k-1})} + e^{Q_+(x_k)}\right). \qquad \text{(A9)}$$

The situation for the constant temperature boundary condition, with step response $C_- = C_1 - C_2$, is more complicated. In this case, the fit is to

$$Q_-(x) = \log\left[1 - \left(\frac{1}{C_-} - \frac{1}{C_+}\right)\frac{1}{\beta}\right], \qquad \text{(A10)}$$

where $\beta = 1/\alpha_- - 1/\alpha_+$, with $\alpha_- = 1 - \hat{a}/2$. Then,

$$\int_{(k-1)\Delta t_D}^{k\Delta t_D}\frac{dt_D}{C_-} \approx \frac{\Delta t_D}{\alpha_-} + \frac{1}{2\alpha_+}\left(e^{Q_+(x_{k-1})} + e^{Q_+(x_k)}\right) - \frac{\beta}{2}\left(e^{Q_-(x_{k-1})} + e^{Q_-(x_k)}\right). \qquad \text{(A11)}$$

Thus, a strategy would be to sample the exact step response over, say, half decade intervals in dimensionless time (12 functional evaluations in the 6 decades spanning $10^{-3} \leq t_D \leq 10^3$, for example), and to curve fit according to the proposed models.

APPENDIX B INVERTING A LOWER TRIANGULAR TOEPLITZ MATRIX

The bracketed matrix expression in Equation (5) is a lower triangular Toeplitz. The general solution of any lower triangular matrix system, say $\mathbf{A}\ \vec{x} = \vec{y}$, can be computed with *back substitution*, in which element k is

$$x_k = \frac{1}{A_{kk}}(y_k - \sum_{j=1}^{k-1} A_{kj}\, x_j)\ , \qquad \text{(B1)}$$

where A_{kj} is the element of $\mathbf{A}$ in the k'th row and j'th column. Back substitution uses previously computed values of the vector $\vec{x}$ to define the next value. Thus, evaluation of this equation must proceed from $k = 1$ to N in order, where N is the number of time steps. This scheme requires $\sum_{k=1}^{N} k = N(N+1)/2$ `flops`.

While this strategy is a vast improvement over general matrix inversion, which requires on the order of N^3 `flops` to evaluate, even greater improvement is possible by exploiting the Toeplitz structure of the matrix and its equivalence to convolution. Below, I develop an algorithm which requires order $N \log_2 N$ `flops` following the work of *Bitmead and Anderson* [1980], who show how to find the inverse of a general, full Toeplitz matrix.

The basis of the algorithm is the structure of inverse matrix. This inverse is itself lower triangular Toeplitz. If $\mathbf{L}(\vec{x})$ denotes a lower triangular Toeplitz matrix with first column $\vec{x}$, the inverse $\mathbf{L}^{-1}(\vec{x})$ of an order N, lower triangular Toeplitz matrix $\mathbf{L}(\vec{x})$ has the partitioned structure

$$\mathbf{S} \equiv \mathbf{L}^{-1}(\vec{x}) = \begin{pmatrix} \mathbf{S}_{11} & \mathbf{S}_{12} \\ \mathbf{S}_{21} & \mathbf{S}_{22} \end{pmatrix} \qquad \text{(B2)}$$

where $\mathbf{S}_{11} = \mathbf{L}^{-1}(\vec{x}[\mathtt{1:k}])$ is a lower triangular, square Toeplitz matrix of order k; $\mathbf{S}_{12}$ is a $k \times j$ matrix of all zeros, with $j = N - k$; $\mathbf{S}_{22} = \mathbf{L}^{-1}(\vec{x}[\mathtt{1:j}])$ is a lower triangular, square Toeplitz matrix of order j; and $\mathbf{S}_{21} = -\mathbf{S}_{22}\,\mathbf{L}(\vec{x})_{21}\,\mathbf{S}_{11}$ is a rectangular Toeplitz matrix of size $j \times k$, where $\mathbf{L}(\vec{x})_{21}$ is the corresponding $j \times k$ partitioning of the original matrix. The notation $\vec{x}[\mathtt{j:k}]$ denotes a column vector formed from elements j through k of $\vec{x}$.

The algorithm proceeds by recognizing that the inverse matrix is completely defined by its first column; that the various matrix multiplications involved in defining the $\mathbf{S}$ partitions are essentially convolutions, which can be rapidly computed using FFT's *Press et al.* [1986, Ch. 12]; and that the solution can be found by recursively calling the algorithm to solve successively smaller partitions of the original matrix. Note, however, that the algorithm need not be formulated recursively. In general, the algorithm must call itself twice per recursion: once to find $\mathbf{S}_{11}$, and again to find $\mathbf{S}_{22}$. Only the first column of $\mathbf{S}_{21}$ is required to complete the computation of the inverse of the lower triangular Toeplitz matrix which was passed to the routine. Moreover, the memory requirements are of order N since only single column vectors are needed to specify the matrices, in contrast to the N^2 values needed for a full matrix.

In particular, if the lower triangular Toeplitz matrix has order N which is a power of 2, and the partitioning is done by halving the matrix size, so that the partition matrices are all the same size, the algorithm is at its fastest. This is because, first, $\mathbf{S}_{11} = \mathbf{S}_{22}$, which obviates the need for computing one matrix inversion per recursion, and, second, power-of-two FFT's are computed most quickly.

Once the inverse of the Toeplitz equation is found, the solution to the finite difference equation, Equation (5), is given by the convolution of the vector on the right hand side of this equation with the vector which defines the inverse matrix, again a task which can be done quickly with FFT's. Explicitly, if $\mathbf{L}(\vec{x})$ defines the inverse matrix, then the solution is given by the first half of

$$\vec{x} \star \left(\vec{C} + \frac{\Phi_D}{3}\hat{\mathbf{e}}_1\right) \qquad \text{(B3)}$$

Implementing the Inversion Algorithm

Figure 2 describes the implementation of the inversion algorithm. Only the recursive implementation is outlined, but the extension to non-recursive inversion is straightforward. Various notations are adopted to facilitate concise pseudo-code. $\vec{x}[\mathtt{j:k}]$ denotes a vector comprising elements j through k of vector $\vec{x}$. $\mathbf{L}(\vec{x})$ denotes a lower triangular Toeplitz matrix which has $\vec{x}$ as its first column. $[\vec{x};\ \vec{y}]$ denotes an augmented column vector comprising $\vec{x}$ atop $\vec{y}$. Also, in performing FFT-aided convolutions, vectors of length N must be augmented, or padded, by N zeros. Unless otherwise stated, this augmentation occurs at the end of the vector. The latter half of this convolution is generally discarded, while the other half represents the operation of the matrix multiplication.

Table 1. Physical Parameter Definition

Parameter	Units	Definition	Fig. 1 Values
t	s	Time	$\leq$ 30 years
$q_T(r,t)$	W	Thermal flux	
q_F	m^3/s	Fluid flow rate	150 & 300 gal/min
c_F	J/kg-deg C	Fluid heat capacity	2095
ρ_F	kg/m^3	Fluid density	895
T_i	deg C	Fluid inlet temperature	50
$T_o(t)$	deg C	Fluid outlet temperature	
K_R	W/m-deg C	Rock thermal conductivity	2.5
c_R	J/kg-deg C	Rock heat capacity	1020
ρ_R	kg/m^3	Rock density	2600
T_R	deg C	Rock initial temperature	275
a	m	Sphere radius	200 & 800
z	m	Depth of sphere's center	4000
c_S	J/kg-deg C	Sphere heat capacity	
ρ_S	kg/m^3	Sphere density	
ϕ	(fraction)	Sphere porosity	0.05
$C(t)$	(none)	Heat flux step response	

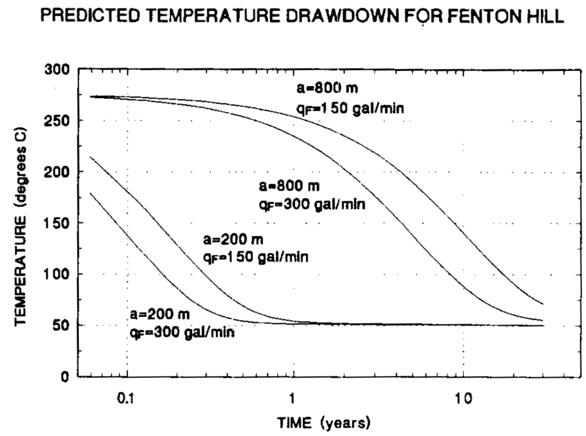

Figure 1

Table 2. Dimensionless Parameter Definition

Parameter	Definition	Name
$\hat{a}$	$\frac{a}{z}$	Radius
Φ_D	$\frac{\rho_S c_S}{\rho_R c_R}$	Porosity or heat capacity
Q_D	$\frac{q_F \rho_F c_F}{K_R a}$	Flow rate
T_D	$\frac{< T_o - T_i >}{T_R - T_i}$	Temperature
t_D	$\frac{K_R t}{\rho_R c_R a^2}$	Time

Recursive Inversion of Lower Triangular Toeplitz Matrix

1. Given: First column $\vec{x}$ of order N lower triangular Toeplitz matrix. N must be a power of 2.
2. If length of vector $N = 1$
   (a) Return first column of inverse $= 1/x[1]$.
3. Else
   (a) Use algorithm (recurse) to compute first column $\vec{L}_{11}$ of $\mathbf{L}^{-1}(\vec{x}[1:N/2])$.
   (b) Compute convolution $\vec{L}_{11} \star \vec{x}[N/2+1:N]$ using FFTs. Save first half as $\vec{L}_{21}$.
   (c) Compute convolution $\vec{L}_{11} \star [\vec{0}; \vec{x}[2:N/2]]$ using FFTs. Add first half to $\vec{L}_{21}$. Note that $\vec{x}[2:N/2]$ is pre-padded with $N/2+1$ zeros.
   (d) Compute convolution $\vec{L}_{11} \star \vec{L}_{21}$ using FFTs. Save first half as $\vec{L}_{21}$
   (e) Return first column of inverse of $\mathbf{L}(\vec{x})$ as augmented vector $[\vec{L}_{11}; -\vec{L}_{21}]$.

Figure 2.

Table 3. Comparison of Toeplitz Inversion Methods

	FFT-Based		Back Substitution	
Time Steps	Compute Time (s)	Effort (flops)	Compute Time (s)	Effort (flops)
4	0.11	907	0.03	22
8	0.29	2,713	0.06	78
16	0.49	6,992	0.08	286
32	0.74	16,499	0.18	1,086
64	1.10	37,401	0.41	4,222
128	1.61	83,089	1.13	16,638
256	2.43	182,522	3.34	66,046
512	3.91	390,208	11.08	263,166
1024	5.78	824,717	39.64	1,050,622