WATER RESOURCES RESEARCH, VOL. 25, NO. 3, PAGES 551–558, MARCH 1989

Application of the Lanczos Algorithm to the Solution of the Groundwater Flow Equation

W. Scott Dunbar[1]

Department of Mechanical Engineering, University of Victoria, Victoria, British Columbia, Canada

Allan D. Woodbury

Department of Geological Sciences, McGill University, Montreal, Quebec, Canada

The solution to the finite element matrix-differential equations resulting from the discretization of the groundwater flow equation is normally carried out by a finite difference approximation to the time derivative. The total computational effort in solving a fluid flow problem is then directly related to the number of unknowns and the number of time steps required to obtain accurate and stable solutions. The Lanczos algorithm uses an orthogonal matrix transformation to reduce the finite element equations to a much smaller tridiagonal system of first-order differential equations. This new system can be solved by a standard tridiagonal solution algorithm with very little computational effort. A matrix-vector multiplication is then used to obtain the original solution at desired time steps. The algorithm is used to accurately simulate the drawdown of synthetic two-dimensional aquifers, including ones with substantial hydraulic conductivity contrasts. The method affords an efficient means of solving large problems, particularly when time durations are long.

Introduction

The use of numerical techniques, based on either finite elements or finite difference methods, has become the standard procedure for solving steady state and transient problems in groundwater hydrology. For geometrically irregular domains the finite element technique is the favored method of solution. It is well-known that for complex systems in two or three dimensions the demand on computing resources can be high because of the need to store large matrices within a time marching algorithm [*Frind and Pinder*, 1982]. In addition, scientists are now pushing the limits of available computer resources (even on large vector computers) for problems such as transient flow in multiple three-dimensional networks of fractures.

In this paper we develop a decomposition technique based on the Lanczos method to solve the transient groundwater flow equation. This method has the potential to render problems with large meshes into equivalent systems of much smaller size. Consequently, large savings in computer time can be realized. Our focus in this paper is to demonstrate the usefulness of the algorithm for a series of hypothetical, but realistic examples. These examples consist of areal two-dimensional aquifers with large hydraulic conductivity contrasts and anisotropic properties.

Background

Groundwater flow is described by the following initial mixed boundary value problem defined for the domain Ω with boundary Γ

$$\begin{aligned} \nabla \cdot (\kappa \cdot \nabla u) + r &= S\frac{\partial u}{\partial t} \\ u &= u_b(t) \qquad \text{on } \Gamma_u \\ -\kappa \cdot (\nabla u \cdot \mathbf{n}) &= q_b(t) \qquad \text{on } \Gamma_q \\ u(\mathbf{x}, 0) &= u^0(\mathbf{x}) \qquad \text{in } \Omega \end{aligned} \tag{1}$$

where u is hydraulic head, κ is the hydraulic conductivity or transmissivity tensor, S is the specific storage or storativity, t is time, and r is the volume flow rate of a source or sink; q_b is a specified flux term and $\mathbf{n}$ is the unit outward normal to the boundary of the domain.

Solution of (1) by finite element or finite difference methods leads to the following system of first-order differential equations (see, for example, *Huyakorn and Pinder* [1983, p. 130])

$$\mathbf{M}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{f} \tag{2}$$

where $\mathbf{u}$ is a vector (length n) of hydraulic heads at nodes of a finite element mesh, $\dot{\mathbf{u}} = \partial\mathbf{u}/\partial t$, $\mathbf{K}$ is the "conductivity" matrix, and $\mathbf{M}$ is the "capacity" matrix. The vector $\mathbf{f}$ includes the effects of the source term as well as the Neumann (flow) boundary conditions. Both $\mathbf{K}$ and $\mathbf{M}$ are symmetric and positive definite matrices.

The solution of (2) may be obtained by direct numerical time integration algorithms such as the Crank-Nicolson method (see, for example, *Huyakorn and Pinder* [1983, p. 130])

$$\left(\mathbf{M} + \frac{\Delta t}{2}\mathbf{K}\right)\mathbf{u}^{s+1} = \left(\mathbf{M} - \frac{\Delta t}{2}\mathbf{K}\right)\mathbf{u}^{s} + \Delta t f^{s+1/2} \tag{3}$$

where the superscript s denotes the quantity evaluated at time $t = (s - 1)\Delta t$. For implicit time integration methods, there are no restrictions on the time step Δt except those required to accurately track the solution given time variations in boundary conditions and/or sources.

At each time step, the solution of (3) for $\mathbf{u}^{s+1}$ involves the solution of a system of equations. The computational effort is therefore directly related to the number of unknowns and the

[1] Now at Acres International Limited, Niagara Falls, Ontario.

Paper number 88WR03748.
0043-1397/89/88WR-03748$05.00

number of time steps. For large problems and long time intervals with relatively short time steps, the computational effort and required storage can become burdensome.

One may reduce the size of the problem by means of Rayleigh-Ritz methods. These methods employ the fact that a groundwater system described by (2) can be accurately described by a few "modes" much like the dynamic behavior of a linear structural system [*Bathe*, 1982, chapter 10]. The characteristics of these methods depend on the choice of modes as described in the next section. The result of each of these methods is twofold.

1. Equation (2) is transformed to a system of equations whose size is equal to the number of modes used. The number of modes (m) is typically much smaller than the original size of the problem (n).

2. The smaller system of equations can be decoupled to form a system of linear first-order differential equations which, if the boundary conditions and sources are time independent, is amenable to an analytical solution.

The motivation for this investigation is to develop an efficient tool for the solution of large groundwater flow problems, involving large conductivity contrasts as well as general types of initial and boundary conditions.

Reduction Methods

The most general type of reduction begins by writing $\mathbf{u} = \mathbf{Z}\mathbf{w}$, where $\mathbf{Z}$ is a $n \times m$ $(m \leq n)$ matrix whose columns are linearly independent. In linear algebraic terms, this constitutes a change of basis. Substituting this expression into (2) and multiplying by $\mathbf{Z}^T$, the transpose of $\mathbf{Z}$, gives

$$\mathbf{Z}^T\mathbf{M}\mathbf{Z}\dot{\mathbf{w}} + \mathbf{Z}^T\mathbf{K}\mathbf{Z}\mathbf{w} = \mathbf{Z}^T\mathbf{f} = \mathbf{g} \qquad (4)$$

Consider the nth-order generalized eigenproblem $\mathbf{K}\mathbf{Z}_n = \mathbf{M}\mathbf{Z}_n\mathbf{\Lambda}_n$, where $\mathbf{Z}_n$ is a $n \times n$ matrix whose columns are eigenvectors of $\mathbf{K}\mathbf{Z}_n = \mathbf{M}\mathbf{Z}_n\mathbf{\Lambda}_n$ and $\mathbf{\Lambda}_n$ is a diagonal matrix of corresponding eigenvalues. If $\mathbf{Z}$ in (4) is a $n \times m$ matrix whose columns are eigenvectors of $\mathbf{K}\mathbf{Z}_n = \mathbf{M}\mathbf{Z}_n\mathbf{\Lambda}_n$ corresponding to the m smallest eigenvalues of $\mathbf{K}\mathbf{Z}_n = \mathbf{M}\mathbf{Z}_n\mathbf{\Lambda}_n$, then $\mathbf{Z}^T\mathbf{M}\mathbf{Z} = \mathbf{I}$, where $\mathbf{I}$ is the $m \times m$ identity matrix ($\mathbf{Z}$ is $\mathbf{M}$ orthogonal) and $\mathbf{Z}^T\mathbf{K}\mathbf{Z} = \mathbf{\Lambda}$, where $\mathbf{\Lambda}$ is a diagonal matrix of the m smallest eigenvalues of $\mathbf{K}\mathbf{Z}_n = \mathbf{M}\mathbf{Z}_n\mathbf{\Lambda}_n$. Equation (4) then reduces to an uncoupled system of order m

$$\dot{\mathbf{w}} + \mathbf{\Lambda}\mathbf{w} = \mathbf{g}$$

for which an analytical solution is possible if $\mathbf{g}$ is time-independent

$$w_i(t) = w_i^0 e^{-\lambda_i t} + \frac{g_i}{\lambda_i}(1 - e^{-\lambda_i t}) \qquad 1 \leq i \leq m \qquad (5)$$

where $\mathbf{w}^0 = \mathbf{Z}^T\mathbf{M}\mathbf{u}^0$ is a vector of the transformed initial conditions, $\mathbf{u}^0$. If $\mathbf{g}$ is time-dependent, a time stepping algorithm must be applied to each equation of the system. Note that although one could choose any subset of the n eigenvalues of $\mathbf{K}\mathbf{Z}_n = \mathbf{M}\mathbf{Z}_n\mathbf{\Lambda}_n$, the exponentials in (5) show that it is the smallest eigenvalues that essentially describe the solution.

Solution methods of this type were first proposed for the corresponding heat transfer problem by *Gallagher and Mallett* [1971]. *Shih and Skladany* [1983] also presented a similar approach and *Sahuquillo* [1983] developed the method for application to groundwater flow problems. The method has also been applied to the transport equation [*Nickell et al.*, 1979; *Hwang et al.*, 1984]. The main difficulties with this approach are as follows.

1. Although efficient eigenvalue routines to compute the m smallest eigenvalues of $\mathbf{K}\mathbf{Z}_n = \mathbf{M}\mathbf{Z}_n\mathbf{\Lambda}_n$ are available [*Smith et al.*, 1976; *Bathe*, 1982, chapter 12] each is iterative in nature and may be slow to converge, especially if the matrices $\mathbf{K}$ and $\mathbf{M}$ are large.

2. Eigenvalue routines are difficult to implement if the matrices are so large that out of core matrix operations are to be performed.

3. The number of computer operations required for an eigenvalue solution is much greater than that required for the standard Crank-Nicolson scheme.

An alternative reduction method is based on the Lanczos process. Rather than find a diagonal matrix of the m smallest eigenvalues, the Lanczos method uses orthogonal transformations to compute a tridiagonal matrix $\mathbf{T}$ of size $m \ll n$ by a recursive scheme, that is, each recursion results in one more row and column in $\mathbf{T}$. The transformations are used to reduce (2) to a small tridiagonal system.

A physically understandable derivation of the Lanczos process may be developed by noting from (5) that it is the small eigenvalues of $\mathbf{K}\mathbf{Z}_n = \mathbf{M}\mathbf{Z}_n\mathbf{\Lambda}_n$ which form the significant part of the solution. It can be shown that the sequence $\mathbf{y}_i = \mathbf{K}^{-1}\mathbf{M}\mathbf{y}_{i-1}$, $i \to \infty$ converges to a multiple of the eigenvector corresponding to the smallest eigenvalue of $\mathbf{K}\mathbf{Z}_n = \mathbf{M}\mathbf{Z}_n\mathbf{\Lambda}_n$ (see, for example, *Bathe* [1982, pp. 610–616]). Consequently, a scheme which uses $\mathbf{K}^{-1}$ as the governing matrix will yield information on the smaller eigenvalues of $\mathbf{K}\mathbf{Z}_n = \mathbf{M}\mathbf{Z}_n\mathbf{\Lambda}_n$.

Therefore one may rewrite the mth order eigenvalue problem as

$$\mathbf{K}^{-1}\mathbf{M}\mathbf{Z} = \mathbf{Z}\mathbf{\Lambda}^{-1} = \mathbf{Z}\mathbf{D}$$

where $\mathbf{D}$ is a diagonal matrix of reciprocal eigenvalues. The governing equation of the Lanczos algorithm is derived by replacing the diagonal matrix $\mathbf{D}$ with a tridiagonal matrix $\mathbf{T}$

$$\mathbf{K}^{-1}\mathbf{M}\mathbf{Q} = \mathbf{Q}\mathbf{T} \qquad (6)$$

where

$$\mathbf{T} = \begin{pmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \ddots & \ddots & \beta_m \\ & & & \beta_m & \alpha_m \end{pmatrix}$$

The vectors $\mathbf{Q} = (\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_m)$ are called Lanczos vectors. These vectors are also linearly independent and $\mathbf{M}$ orthogonal (i.e., $\mathbf{Q}^T\mathbf{M}\mathbf{Q} = \mathbf{I}$). The eigenvalues of $\mathbf{T}$ give good approximations to the smaller eigenvalues of $\mathbf{K}\mathbf{Z}_n = \mathbf{M}\mathbf{Z}_n\mathbf{\Lambda}_n$ [see *Bathe*, 1982, pp. 597–598; *Hughes*, 1987, pp. 582–598].

The Lanczos Algorithm

The validity of (6) is demonstrated by developing an algorithm which constructs $\mathbf{T}$ and $\mathbf{Q}$. Equating columns of both sides of (6), one obtains

$$\beta_{j+1}\mathbf{q}_{j+1} = \mathbf{r}_j = \mathbf{K}^{-1}\mathbf{M}\mathbf{q}_j - \alpha_j\mathbf{q}_j - \beta_j\mathbf{q}_{j-1} \qquad 1 \leq j \leq n-1 \qquad (7)$$

Multiplication of both sides of (7) by $\mathbf{q}_j{}^T\mathbf{M}$ and $\mathbf{q}_{j+1}{}^T\mathbf{M}$ and using the $\mathbf{M}$ orthogonality requirement for the Lanczos vectors, results in equations for α_j and β_{j+1}

$$\alpha_j = \mathbf{q}_j{}^T\mathbf{M}\mathbf{K}^{-1}\mathbf{M}\mathbf{q}_j$$

$$\beta_{j+1} = \mathbf{q}_{j+1}{}^T\mathbf{M}\mathbf{r}_j = (\mathbf{r}_j{}^T\mathbf{M}\mathbf{r}_j)^{1/2}$$

Given a Lanczos vector $\mathbf{q}_j$ and β_j, α_j, a new Lanczos vector $\mathbf{q}_{j+1}$ and β_{j+1} is computed; i.e., the algorithm is recursive in nature. The choice of a starting vector is largely arbitrary;

```
$\mathbf{r}(=\mathbf{r}_0) = \mathbf{K}^{-1}\mathbf{f}_0$
$\mathbf{q}_0 = \mathbf{0}$
$\beta_1 = (\mathbf{r}^T\mathbf{M}\mathbf{r})^{\frac{1}{2}}$
$\mathbf{q}_1 = \mathbf{r}/\beta_1$
$\mathbf{p} = \mathbf{M}\mathbf{q}_1$
For $j = 1,2,\ldots$
    $\mathbf{r} \leftarrow \mathbf{K}^{-1}\mathbf{p} - \beta_j\mathbf{q}_{j-1}(=\mathbf{r}_j + \alpha_j\mathbf{q}_j)$
    $\alpha_j = \mathbf{q}_j^T\mathbf{M}\mathbf{r} = \mathbf{p}^T\mathbf{r}$
    $\mathbf{r}(=\mathbf{r}_j) \leftarrow \mathbf{r} - \alpha_j\mathbf{q}_j$
    $\mathbf{p} \leftarrow \mathbf{M}\mathbf{r}$
    $\beta_{j+1} = (\mathbf{r}^T\mathbf{M}\mathbf{r})^{1/2} = (\mathbf{r}^T\mathbf{p})^{1/2}$
    terminate loop if desired
    $\mathbf{q}_{j+1} = \mathbf{r}/\beta_{j+1}$
    $\mathbf{p} \leftarrow \mathbf{p}/\beta_{j+1}$
End $j$
```

Fig. 1. Lanczos algorithm.

however, the process should contain basis states that are the primary components of interest. In this work, the starting vector $\mathbf{q}_1$ is chosen by solving the steady state problem $\mathbf{r}_0 = \mathbf{K}^{-1}\mathbf{f}$ and normalizing so that $\mathbf{q}_1 = \mathbf{r}_0/(\mathbf{r}_0{}^T\mathbf{M}\mathbf{r}_0)^{1/2}$. This approach has the advantage that the Lanczos vectors include the steady state solution.

In practice, the recursion would be terminated after $m \ll n$ steps. The relatively small tridiagonal matrix $\mathbf{T}$ results together with a $n \times m$ matrix $\mathbf{Q}$ such that from (6) and the M orthogonality property

$$\mathbf{Q}^T\mathbf{M}\mathbf{K}^{-1}\mathbf{M}\mathbf{Q} = \mathbf{T}$$

A reduced system of equations is derived by letting $\mathbf{u} = \mathbf{Q}\mathbf{w}$ in (2). Multiplication of the resulting equation by $\mathbf{Q}^T\mathbf{M}\mathbf{K}^{-1}$ gives

$$\begin{gathered}\mathbf{Q}^T\mathbf{M}\mathbf{K}^{-1}\mathbf{M}\mathbf{Q}\dot{\mathbf{w}} + \mathbf{Q}^T\mathbf{M}\mathbf{Q}\mathbf{w} = \mathbf{Q}^T\mathbf{M}\mathbf{K}^{-1}\mathbf{f}\\ \mathbf{T}\dot{\mathbf{w}} + \mathbf{w} = \mathbf{g}\end{gathered} \tag{8}$$

$\mathbf{w}$ is the solution vector of length m in Lanczos "space." The system of equations could be solved by a time integration technique (e.g., Crank-Nicolson and the Thomas algorithm). The solution $\mathbf{u}$ is then found by the matrix-vector multiplication $\mathbf{u} = \mathbf{Q}\mathbf{w}$ at the desired time steps. The system could also be decoupled by means of a further eigenvalue-eigenvector decomposition of $\mathbf{T}$. Depending on the relative sizes of n and m, the latter option could involve more operations than the repetitive solution of the tridiagonal system of equations resulting from the time integration of (8).

Applications of the above version of the Lanczos algorithm to structural analysis have been presented by *Nour-Omid and Clough* [1984]. *Nour-Omid* [1987] also presented a similar algorithm for heat transfer problems. *Dunbar and Woodbury* [1987] adapted the algorithm to problems involving thermal convection.

One Lanczos step requires a resolution with the coefficient matrix $\mathbf{K}$, one matrix-vector multiplication with the matrix $\mathbf{M}$, and five inner or scalar-vector products. Note that both the matrices $\mathbf{K}$ and $\mathbf{M}$ need not be in core at the same time. The matrix operations can be set up in separate subroutines which take advantage of the sparsity and symmetry of the matrices and of any available out of core data storage system. The entire algorithm for the computation of $\mathbf{T}$ and the Lanczos vectors is shown in Figure 1.

An operation count comparison between the standard Crank-Nicolson and Lanczos methods is shown in Table 1.

TABLE 1. Operation Count Comparison

Item	Operation's Count	
	Crank-Nicolson	
For every time step		
matrix-vector multiplication	nn_b	
scalar multiply and addition	$2n$	
resolution	$2nn_b$	
	$n(3n_b + 2)$	
For NT time steps		$n(3n_b + 2)NT$
	Lanczos	
For every Lanczos step (vector)		
resolution	$2nn_b$	
matrix-vector multiplication	nn_b	
inner or scalar-vector products	$5n$	
	$n(3n_b + 5)$	
For m Lanczos steps (vectors)		$mn(3n_b + 5)$
For every time step (tridiagonal system)		
matrix-vector multiplication	$5m$	
scalar multiply and addition	$2m$	
resolution	$8m$	
	$15m$	
For NT time steps		$15mNT$
Total		$mn(3n_b + 5) + 15mNT$

Here n, size of matrices $\mathbf{K}$ and $\mathbf{M}$; n_b, half bandwidth of matrices $\mathbf{K}$ and $\mathbf{M}$; and m, number of Lanczos vectors. Both methods require the matrix $\mathbf{K}$ to be factored ($\frac{1}{2}nn_b{}^2$ operations).

Note that both methods require the matrix $\mathbf{K}$ to be factored which would add another $\frac{1}{2}nn_b^2$ operations to the total operation count of each method. From this comparison it may be seen that since the number of Lanczos vectors m is usually much smaller than the size of the finite element system n, the computational savings can be quite remarkable. This increase in efficiency would be even more pronounced as the size of the problem increases.

A practical application of the eigenvalue method discussed previously would probably use a variant of the Householder QR inverse iteration algorithm. The operation count for this algorithm is $\frac{2}{3}n^3 + \frac{21}{2}n^2 + mn^2 + 9mn$, where n is the size of the eigenproblem $\mathbf{KZ}_n = \mathbf{MZ}_n\mathbf{\Lambda}_n$ and m is the number of eigenvalues and eigenvectors computed [*Bathe*, 1982, p. 655]. This operations count is much greater than either the Crank-Nicolson or the Lanczos process.

IMPLEMENTATION OF THE LANCZOS ALGORITHM

Loss of Orthogonality

The main difficulty with the Lanczos algorithm is that each Lanczos vector generated is subject to loss of $\mathbf{M}$ orthogonality with respect to earlier vectors due to roundoff error and cancellation. This condition means that $\mathbf{Q}^T\mathbf{MQ} \neq \mathbf{I}$ during the Lanczos decomposition. The loss of orthogonality can be monitored at each step of the recursion by computing the quantity $\eta_{j+1,i} = \mathbf{q}_{j+1}{}^T\mathbf{Mq}_i$ given by [*Simon*, 1984; Appendix A]

$$\beta_{j+1}\eta_{j+1,i} = \beta_{i+1}\eta_{j,i+1} + (\alpha_i - \alpha_j)\eta_{j,i} - \beta_j\eta_{j-1,i} + \beta_i\eta_{j,i-1} \qquad (9)$$

$$j \geq 2 \qquad 1 \leq i \leq j-1 \qquad \eta_{j,j} = 1 \qquad \eta_{j,j-1} = \varepsilon$$

where ε is the unit roundoff error of the computer in use. When $\eta_{j+1,i} = \sqrt{\varepsilon}$, loss of orthogonality has occurred and the newly generated Lanczos vector $\mathbf{q}_{j+1}$ is orthogonalized against all preceding vectors using the Gram-Schmidt procedure. Applications of this approach, which is known as partial reorthogonalization, appear in the works by *Nour-Omid and Clough* [1984] and *Alvaro et al.* [1987]. The implementation of partial reorthogonalization used in this work are modified versions of subroutines ORTBND and PURGE which appear in the work by *Hughes* [1987, pp. 614–616]. There are other means by which orthogonality can be maintained and it is a subject of much research [*Golub et al.*, 1972; *Paige*, 1976; *Parlett and Scott*, 1979; *Cullum and Willoughby*, 1985].

Experience to date with applications to groundwater flow problems has shown that partial reorthogonalization is usually required after two of every 10 Lanczos steps. The procedure is relatively fast and adds little time to the Lanczos decomposition process.

Initial and Boundary Conditions

Typically, initial conditions are a significant part of the solution and several Lanczos vectors may be required to capture their frequency content. They can be eliminated from the solution by writing $\mathbf{u}$ in (2) as the sum of the initial conditions $\mathbf{u}^0$, plus a transient $\mathbf{v}(t)$ with zero initial condition

$$\mathbf{u}(t) = \mathbf{u}^0 + \mathbf{v}(t) \qquad \mathbf{v}(0) = \mathbf{0}$$

Substituting the above into (2) and noting that $\mathbf{M\dot{u}}^0 = \mathbf{0}$, since $\mathbf{u}^0$ is not a function of time, one obtains a differential equation for the transient $\mathbf{v}$

$$\mathbf{M\dot{v}} + \mathbf{Kv} = \mathbf{f} - \mathbf{Ku}^0 \qquad (10)$$

The right-hand side vector $\mathbf{f}$ is time-dependent if boundary conditions are time-dependent and/or wells have a nonconstant pumping history. Equation (8) suggests that in such a case the vector $\mathbf{g} = \mathbf{Q}^T\mathbf{MK}^{-1}\mathbf{f}$ would have to be evaluated at each time step during the solution of the small tridiagonal system. This calculation would completely negate any benefits in efficiency afforded by the small system. However, it may be avoided by the following procedure.

In finite element analysis, it is common practice to eliminate equations corresponding to known boundary conditions. This is accomplished by partitioning (10) as follows:

$$\begin{pmatrix} \mathbf{M}_{rr} & \mathbf{M}_{rb} \\ \mathbf{M}_{br} & \mathbf{M}_{bb} \end{pmatrix}\begin{pmatrix} \dot{\mathbf{v}}_r \\ \dot{\mathbf{v}}_b \end{pmatrix} + \begin{pmatrix} \mathbf{K}_{rr} & \mathbf{K}_{rb} \\ \mathbf{K}_{br} & \mathbf{K}_{bb} \end{pmatrix}\begin{pmatrix} \mathbf{v}_r \\ \mathbf{v}_b \end{pmatrix}$$

$$= \begin{pmatrix} \mathbf{f}_r \\ \mathbf{f}_b \end{pmatrix} - \begin{pmatrix} \mathbf{K}_{rr} & \mathbf{K}_{rb} \\ \mathbf{K}_{br} & \mathbf{K}_{bb} \end{pmatrix}\begin{pmatrix} \mathbf{u}_r{}^0 \\ \mathbf{u}_b{}^0 \end{pmatrix}$$

where the subscript b denotes boundary nodes of the finite element mesh, the subscript r denotes the remaining nodes, and the subscript rb denotes the interaction between boundary and the remaining nodes. Rewriting the first equation for the unknown values of $\mathbf{v}_r$, one obtains

$$\mathbf{M}_{rr}\dot{\mathbf{v}}_r + \mathbf{K}_{rr}\mathbf{v}_r = \mathbf{f}_r - \mathbf{K}_{rb}\mathbf{u}_b - \mathbf{M}_{rb}\dot{\mathbf{v}}_b - \mathbf{K}_{rr}\mathbf{u}_r{}^0 \qquad (11)$$

Equation (11) is the equation solved in an actual problem. It is of order n. Each of the components of the right-hand side is potentially time-dependent which, as stated previously, presents a problem for solution by the Lanczos algorithm. However, this problem can be circumvented by writing each component as the product of a scalar function of time and a constant vector. This method is detailed below.

The vector $\mathbf{f}_r$ contains the effects of wells and flow boundary conditions. If the scalar function $p_j(t)$ denotes the pumping history of the jth well, $1 \leq j \leq NW$, and the scalar function $q_b(t)$ denotes the time history of the flux boundary condition, then $\mathbf{f}_r$ can be written as

$$\mathbf{f}_r = \sum_{j=1}^{NW} \mathbf{f}_j p_j(t) + q_b(t)\mathbf{b}$$

where $\mathbf{f}_j$ is a vector of length n composed of zeros with ones at the equations corresponding to the nodes defining each well, and $\mathbf{b}$ is a vector of length n composed of zeros with nonzero coefficients at equations corresponding to nodes on the boundary. Values of these coefficients depend on the type of finite elements used [see *Huyakorn and Pinder*, 1983, chapter 4]. Similarly, the vectors $\mathbf{K}_{rb}\mathbf{u}_b$ and $\mathbf{M}_{rb}\dot{\mathbf{v}}_b$ can be written as

$$\mathbf{K}_{rb}\mathbf{1}u_b(t)$$

$$\mathbf{M}_{rb}\mathbf{1}\dot{v}_b(t)$$

where $\mathbf{1}$ is a vector of length equal to the number of nodes on the boundary and is composed of ones. The scalar function $v_b(t) = u_b(t) - u^0$, where $u_b(t)$ and u^0 are defined in (1). The vector $\mathbf{g}$ in (8) is therefore given by

$$\mathbf{g} = \sum_{j=1}^{NW} p_j(t)\mathbf{s}_j - q_b(t)\mathbf{h}_1 - u_b(t)\mathbf{h}_2 - \dot{v}_b(t)\mathbf{h}_3 - \mathbf{h}_4 \qquad (12)$$

where the $\mathbf{s}_j$ and $\mathbf{h}_i$ are constant vectors given by

$$\mathbf{s}_j = \mathbf{Q}^T\mathbf{M}_{rr}\mathbf{K}_{rr}{}^{-1}\mathbf{f}_j$$

$$\mathbf{h}_1 = \mathbf{Q}^T\mathbf{M}_{rr}\mathbf{K}_{rr}{}^{-1}\mathbf{b}$$

$$\mathbf{h}_2 = \mathbf{Q}^T\mathbf{M}_{rr}\mathbf{K}_{rr}^{-1}\mathbf{K}_{rb}\mathbf{1}$$

$$\mathbf{h}_3 = \mathbf{Q}^T\mathbf{M}_{rr}\mathbf{K}_{rr}^{-1}\mathbf{M}_{rb}\mathbf{1}$$

$$\mathbf{h}_4 = \mathbf{Q}^T\mathbf{M}_{rr}\mathbf{u}_r^{\,0}$$

Thus in general, a time-dependent right-hand side requires only a few extra vectors of length m to be stored. This formulation allows a problem with a given geometry but different boundary and initial conditions and/or sources to be solved without having to repeat the Lanczos decomposition.

Stopping Criterion

It is desirable to have some means of terminating the recursion when a sufficient number of Lanczos vectors has been computed. Such a criterion may be derived by using the concept of "participation factors." The participation factor p_j of the jth Lanczos vector is the component of the right-hand side vector $\mathbf{f}$ which contributes to $\mathbf{q}_j$:

$$p_j = \mathbf{q}_j^T\mathbf{f}$$

In linear algebraic terms, if $\mathbf{f}$ is orthogonal to $\mathbf{q}_j$, $p_j = 0$. If $\mathbf{f}$ is time-dependent, then p_j will be time-dependent. However, in the previous subsection, it was shown that $\mathbf{f}$ could be written as the product of a scalar function of time and a constant vector, say, $\mathbf{f} = \phi(t)\mathbf{a}$. Since the Lanczos decomposition applies only to the spatial part of the problem, it is reasonable to redefine the participation factor as

$$p_j = \mathbf{q}_j^T\mathbf{a}$$

which is time-independent. A recursive relationship may be derived (see Appendix B) for p_{j+1}. It is

$$\begin{aligned} &p_1 = \mathbf{q}_1^T\mathbf{a} \qquad p_2 = \mathbf{q}_2^T\mathbf{a} && j \le 2 \\ &p_{j+1} = \frac{-(\alpha_j p_j + \beta_j p_{j-1})}{\beta_{j+1}} && j > 2 \end{aligned} \tag{13}$$

The quantity $|p_{j+1}/p_1|$ (or possibly another ratio between participation factors) is computed at each Lanczos step so that when it falls below a given tolerance, the Lanczos recursion is terminated.

If more than one source is present and/or the boundary conditions are time dependent, $\mathbf{f}$ becomes a sum of terms like $\phi(t)\mathbf{a}$ (see equation (12)). In this case, the definition of the participation factor becomes complicated and another termination criterion would have to be employed. An example of one such criterion is given by *Nour-Omid et al.* [1983] and *Nour-Omid* [1987]. In the examples to be shown the participation factor given by (13) is used, since only one scalar time function is present in each example.

TABLE 2. Homogeneous Theis Model Parameters

Parameter	Value
Number of nodes	441
Number of elements	400
Number of materials	1
Transmissivity T	300 m^2/day
Storativity S	0.002
Initial condition $\mathbf{u}(\mathbf{x}, 0)$	10 m
Time step Δt	0.1 day
Number of time steps	250
Well pumping rate	−2000 m^3/day

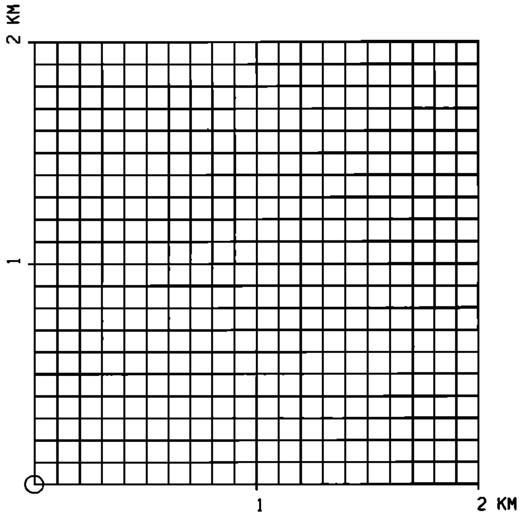

Fig. 2. Homogeneous Theis model, finite element mesh. Only one quarter of the total flow field is shown.

Computer Implementation

The algorithm is set up in two stages. The first stage computes the matrices $\mathbf{K}$ and $\mathbf{M}$ and the right-hand side vector $\mathbf{f}$. The matrix $\mathbf{K}$ is factored by the Cholesky decomposition method and the constant vectors in (12) are computed. The Lanczos decomposition is performed. The Lanczos vectors $\mathbf{Q}$, $\mathbf{g}$ vectors, and the tridiagonal matrix $\mathbf{T}$ are then stored on a disk. This data is all that is necessary to solve a problem.

The second stage solves the small tridiagonal system of first-order differential equations and computes the solution to the original problem. Separate function subroutines define the pumping histories of wells and the boundary and initial conditions. In this way, quite generalized problems can be solved.

Examples

To investigate the robustness of the proposed approach the Lanczos technique is applied to three areal confined aquifer problems in which the computed hydraulic head field is compared to the results computed from the Crank-Nicolson scheme. Spatial discretization in each example is done by means of bilinear isoparametric finite elements [*Bathe*, 1982, chapter 5]. Note that triangular elements are four noded elements with one node repeated.

The first example is a homogeneous and isotropic aquifer with a well at the center (the Theis problem). The problem is geometrically symmetric, so only one quarter of the flow field is presented. The model parameters are given in Table 2. Figure 2 shows the grid used. Note that the grid spacing near the well is not optimal in terms of solution accuracy. However, since the Crank-Nicolson scheme also suffers from the same restrictions, the comparison is meaningful.

For comparison purposes we define a percentage error term:

$$e = \max_i \left| \frac{(u_L - u_{CN})}{u_{CN}} \right| \times 100 \qquad 1 \le i \le n$$

where u_{CN} is the hydraulic head computed from the Crank-Nicolson model, u_L is the hydraulic head computed from the Lanczos method, and n is the number of nodes in the system, excluding Dirichlet nodes (i.e., the number of equations in equation (11)).

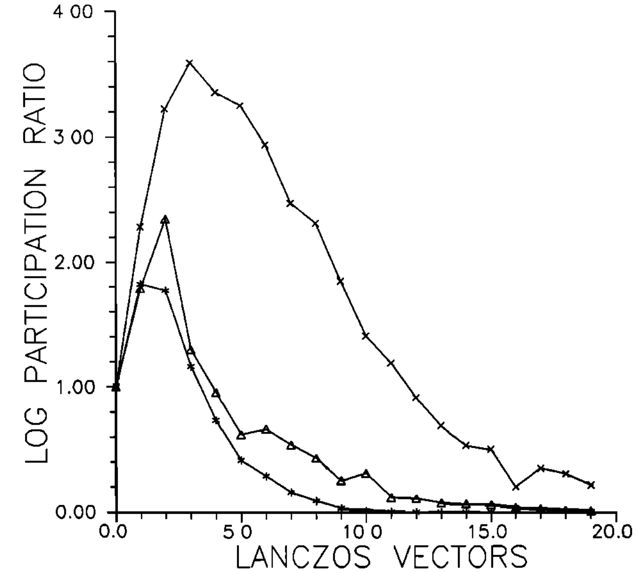

Fig. 3. Plot of log $|p_{j+1}/p_1|$ versus number of Lanczos vectors for example problems. Triangles represent the homogeneous Theis problem; asterisks, the inhomogeneous problem; crosses, the synthetic aquifer problem.

The results of the comparison are very encouraging. For a maximum percentage error of 2.78×10^{-3} at $t = 2.5$ days, only 10 Lanczos vectors are required. The maximum error occurs at the well. The plot of participation factors versus the number of Lanczos vectors is shown in Figure 3 (triangles). The ratio log $|p_{j+1}/p_1|$ does not appear to give a unique point at which to terminate the Lanczos recursion. No attempt was made to relate this ratio to a particular accuracy.

An execution time comparison that does not include input and output operations is presented in Table 3. As shown, the Lanczos algorithm is approximately 8 times faster than the Crank-Nicolson scheme for this case. For a larger problem with many time steps the solution time comparison between the two methods is a more important statistic. According to Table 1 for the homogeneous Theis model with $n = 400$ (boundary conditions are eliminated from the system), $n_b = 22$ and $m = 10$ Lanczos vectors used, the operation count ratio for the solution step is

$$\frac{n(3n_b + 2)}{15m} = \frac{400(3 \times 22 + 2)}{15 \times 10} = \frac{27,200}{150} \approx 181$$

TABLE 3. Time Comparison

Item	Time(s)
Crank-Nicolson	
Formation of matrices (n = 400)	11.97
Matrix factorization	6.70
Solution (Δt=0.1 days, 250 steps)	403.31
Total	421.98
Lanczos	
Formation of matrices (n = 400)	11.97
Matrix factorization	6.70
Lanczos decomposition (m = 10)	31.39
Solution (Δt = 0.1 days, 250 steps)	2.91
Total	52.97

Homogeneous Theis model, COMPAQ 386/387 (16 MHz), Lahey FORTRAN compiler.

TABLE 4. Inhomogeneous Theis Model Parameters

Parameter	Values
Number of nodes	441
Number of elements	400
Number of materials	2
Transmissivity T_1	300 m^2/day
Storativity S_1	0.002
Transmissivity T_2	3 m^2/day
Storativitiy S_2	0.002
Initial condition u(x, 0)	10 m
Time step Δt	0.1 day
Number of time steps	250
Well pumping rate	−2000 m^3/day

From Table 3 the time required for solution by the Crank-Nicolson method is 403.3 s, and the time required for solution by the Lanczos method is 2.9 s, giving a ratio of approximately 138, in good agreement with the ratio given by the operation count. The discrepancy is likely due to overheads caused by subroutine calls and vector copying.

Another series of computations are carried out on a heterogeneous problem. The model parameters are given in Table 4 and the grid used is shown in Figure 4. T_2 is the transmissivity adjacent to the well. Note the 100fold ratio in hydraulic conductivity contrast. The resulting drawdown field is smooth, except for steep gradients near the well. Ten Lanczos vectors are required to model these gradients giving a maximum percentage error at $t = 2.5$ days of 4.38×10^{-3} (at the well), an error comparable to that of the homogeneous model.

The plot of participation factors versus the number of Lanczos vectors is shown in Figure 3 (asterisks). The interesting aspect about this plot is that it exhibits a more rapid decline than the corresponding plot for the homogeneous model. This indicates that the steep gradients near the well can be described by relatively few Lanczos vectors.

A final example is shown on Figure 5 and is similar to a synthetic aquifer example given in *Townley and Wilson* [1980]. This example is chosen to include several of the more commonly occurring complexities in groundwater analysis. The example includes Dirichlet boundary conditions and a time-dependent source. The aquifer is heterogeneous and an-

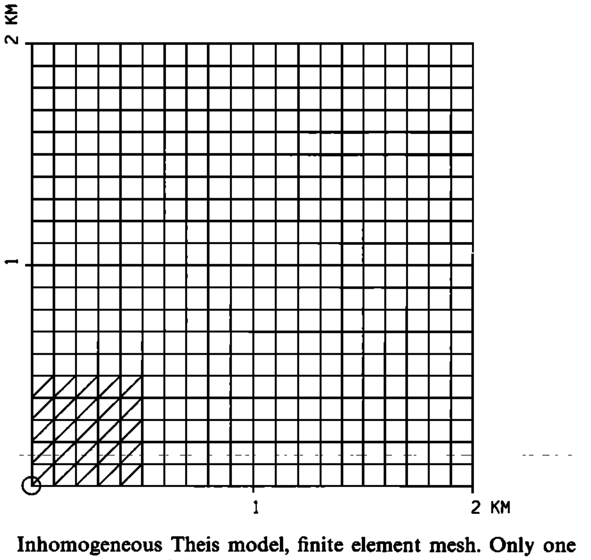

Fig. 4. Inhomogeneous Theis model, finite element mesh. Only one quarter of the total flow field is shown.

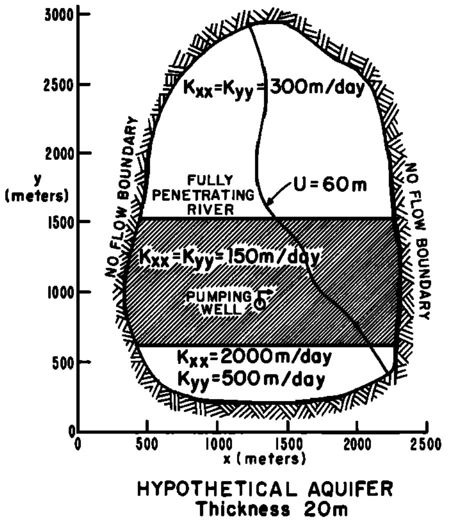

Fig. 5. Synthetic aquifer example [after *Townley and Wilson*, 1980]. Boundary conditions and hydraulic parameters shown.

isotropic with respect to hydraulic conductivity. The model parameters are given in Table 5. Figure 6 shows the finite element grid used.

For a maximum percentage error of 1.34 × 10^{-3} at $t = 140$ days, only 10 Lanczos vectors are required. Figure 3 (crosses) shows the plot of participation factors versus the number of Lanczos vectors. This simulation shows that the proposed numerical scheme is efficient and accurate and could be applied to virtually any groundwater flow model.

Conclusions

The Lanczos algorithm reduces the finite element matrix differential equations of size n to a much smaller tridiagonal system of size m by means of orthogonal matrix transformations. The algorithm is recursive in nature, each step of the recursion giving an additional Lanczos vector and an additional row and column of the tridiagonal matrix. The resulting small system can be solved by a standard tridiagonal solution algorithm. A matrix-vector multiplication then gives the desired solution.

TABLE 5. Synthetic Aquifer Model Parameters

Parameter	Value
Number of nodes	93
Number of elements	81
Number of materials	3
Conductivity κ_{xx1}	300 m/day
Conductivity κ_{yy1}	300 m/day
Conductivity κ_{xx2}	150 m/day
Conductivity κ_{yy2}	150 m/day
Conductivity κ_{xx3}	2000 m/day
Conductivity κ_{yy3}	500 m/day
Aquifer thickness	20 m
Storativity of all materials	0.005
Initial condition $\mathbf{u}(\mathbf{x}, 0)$	60 m
Time step Δt	1 day
Number of time steps	140
Well pumping rate, $p(t)$	
$p(t)$[m³/day] = −20,000 $0 \leq t \leq 50$ days	
$p(t)$[m³/day] = −5,000 $50 \leq t \leq 100$ days	
$p(t)$[m³/day] = 0 $t > 100$ days	

Constant head conditions of 60 m are shown in Figure 5.

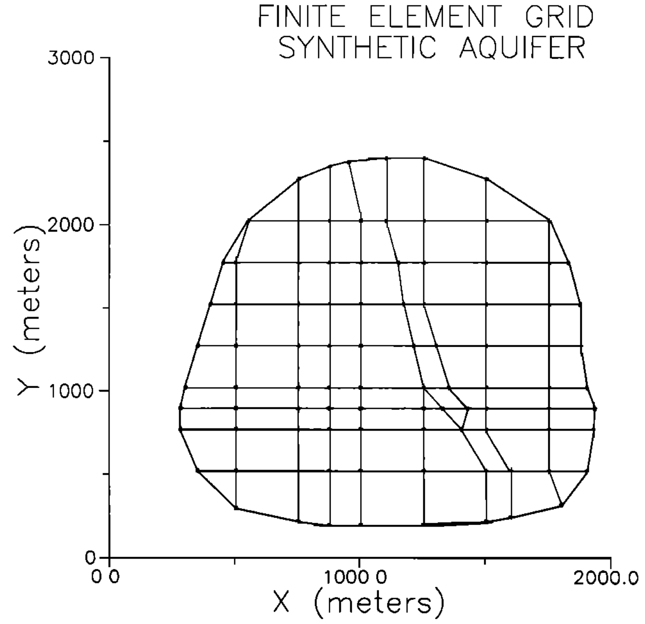

Fig. 6. Synthetic aquifer example, finite element mesh.

The main advantage of the method stems from the fact that whereas resolution time for the Crank-Nicolson method is proportional to $n \times n_b$, where n_b is the half bandwidth of the matrices **K** and **M**, resolution of the tridiagonal system is only proportional to m. Consequently, the decrease in solution effort is pronounced, particularly on large problems.

The algorithm was used to accurately simulate the drawdown of synthetic two-dimensional aquifers, including ones with substantial hydraulic conductivity contrasts. It was found that only a small number of Lanczos vectors was required to accurately match the drawdown computed by a typical Crank-Nicolson scheme applied to the original system of equations. Thus the method affords an efficient means of solving large problems, particularly when time durations are long.

The concept of participation factors can be used to determine when to terminate the Lanczos recursion, although it may be difficult to detect the termination point automatically. In addition, in the case of multiple wells or time-dependent boundary conditions, the participation factor becomes difficult to define. Further tests are required to evaluate participation factors or other criteria for determining when to terminate the recursion.

Although other complexities such as multiple wells and time-dependent flux boundary conditions were not included in the examples, these should not pose a problem using the formulation described herein. It appears that the method could be applied to any realistic groundwater flow model. Future work is intended to verify this.

Appendix A: Derivation of Equation (9)

For the Lanczos steps i and j, (7) is

$$\beta_{j+1}\mathbf{q}_{j+1} = \mathbf{K}^{-1}\mathbf{M}\mathbf{q}_j - \alpha_j\mathbf{q}_j - \beta_j\mathbf{q}_{j-1} \tag{A1}$$

$$\beta_{i+1}\mathbf{q}_{i+1} = \mathbf{K}^{-1}\mathbf{M}\mathbf{q}_i - \alpha_i\mathbf{q}_i - \beta_i\mathbf{q}_{i-1} \tag{A2}$$

Multiplying (A1) by $\mathbf{q}_i^T M$ and (A2) by $\mathbf{q}_j^T M$, subtracting and letting $\eta_{k,l} = \mathbf{q}_k^T\mathbf{M}\mathbf{q}_l$ gives (9):

$$\beta_{j+1}\eta_{j+1,i} = \beta_{i+1}\eta_{j,i+1} + (\alpha_i - \alpha_j)\eta_{j,i} - \beta_j\eta_{j-1,i} + \beta_i\eta_{j,i-1} \tag{A3}$$

$$j \geq 2 \qquad 1 \leq i \leq j-1 \qquad \eta_{j,j} = 1 \qquad \eta_{j,j-1} = \varepsilon$$

where ε is the unit roundoff error of the computer in use. Note that $\eta_{k,l} = \eta_{l,k}$ and $\eta_{i,0} = 0$.

APPENDIX B: DERIVATION OF EQUATION (13)

Multiplication of (7) by the constant vector **a** gives

$$\mathbf{a}^T\mathbf{r}_j = \mathbf{a}^T\mathbf{K}^{-1}\mathbf{M}\mathbf{q}_j - \alpha_j\mathbf{a}^T\mathbf{q}_j - \beta_j\mathbf{a}^T\mathbf{q}_{j-1} \tag{B1}$$

Since $\mathbf{r}_0 = \mathbf{K}^{-1}\mathbf{a} = \beta_1\mathbf{q}_1$, $\mathbf{a} = \beta_1\mathbf{K}\mathbf{q}_1$ so that

$$\mathbf{a}^T\mathbf{K}^{-1}\mathbf{M}\mathbf{q}_j = \beta_1\mathbf{q}_1{}^T\mathbf{K}\mathbf{K}^{-1}\mathbf{M}\mathbf{q}_j = 0 \qquad j > 1$$

From (8), $\mathbf{r}_j = \beta_{j+1}\mathbf{q}_{j+1}$. Therefore (B1) becomes

$$\beta_{j+1}\mathbf{a}^T\mathbf{q}_{j+1} = -\alpha_j\mathbf{a}^T\mathbf{q}_j - \beta_j\mathbf{a}^T\mathbf{q}_{j-1}$$

or in terms of participation factors $p_j = \mathbf{q}_j{}^T\mathbf{a}$

$$\beta_{j+1}p_{j+1} = -\alpha_j p_j - \beta_j p_{j-1}$$

from which (13) is obtained.

Acknowledgments. The authors wish to express their sincere appreciation for the encouragement and assistance provided by Bahram Nour-Omid of the Lockheed Palo Alto Research Laboratory and by Leslie Smith of the Department of Geological Sciences at the University of British Columbia. This research was supported by grants from the Natural Sciences and Engineering Research Council of Canada. In addition, the authors would like to thank the reviewers of the original manuscript for their many useful suggestions.

REFERENCES

Alvaro, L. G., L. L. Coutinho, E. C. P. Lima, and N. F. F. Ebecken, The application of the Lanczos mode superposition method in dynamic analysis of offshore structures, *Comput. Struct.*, *25*, 615–625, 1987.

Bathe, K. J., *Finite Element Procedures In Engineering Analysis*, Prentice-Hall, Englewood Cliffs, N. J., 1982.

Cullum, J. K., and R. A. Willoughby, *Lanczos Algorithms for Large Symmetric Eigenvalue Computations*, vol. 1, *Theory*, Springer-Verlag, New York, 1985.

Dunbar, W. S., and A. D. Woodbury, Semi-analytic solution to the heat convection equation, in *Numerical Methods in Thermal Problems*, vol. 5, edited by R. W. Lewis, K. Morgan, and W. G. Habashi, Pineridge, Swansea, United Kingdom, 1987.

Frind, E. O., and G. F. Pinder, The principal direction technique for the solution of the advection-dispersion equation, in *Scientific Computing*, pp. 305–313, North-Holland, Amsterdam, 1982.

Gallagher, R. H., and R. H. Mallett, Efficient solution processes for finite element analysis of transient heat conduction, *Trans. ASME*, *93*, 257–263, 1971.

Golub, G. H., R. Underwood, and J. H. Wilkinson, The Lanczos algorithm for the symmetric $Ax = \lambda Bx$ problem, *Rep. STAN-CS-72-270*, Dep. of Comput. Sci., Stanford Univ., Stanford, Calif., 1972.

Hughes, T. J. R., *The Finite Element Method*, Prentice-Hall, Englewood Cliffs, N. J., 1987.

Huyakorn, P. S., and G. F. Pinder, *Computational Methods In Subsurface Flow*, Academic, San Diego, Calif., 1983.

Hwang, J. C., W. C. Cho, and G. T. Yeh, An eigenvalue solution continuous in time to the spatially discretized solute transport equation in steady groundwater flow, *Water Resour. Res.*, *20*, 1725–1732, 1984.

Nickell, R. E., D. K. Gartling, and G. Strang, Spectral decomposition in advection-diffusion analysis by finite element methods, *Comput. Methods Appl. Mech. Eng.*, *17/18*, 561–580, 1979.

Nour-Omid, B., Lanczos method for heat conduction analysis, *Int. J. Numer. Methods Eng.*, *24*, 251–262, 1987.

Nour-Omid, B., and R. W. Clough, Dynamic analysis of structures using Lanczos coordinates, *Earthquake Eng. Struct. Dyn.*, *12*, 565–577, 1984.

Paige, C. C., Practical use of the symmetric Lanczos process with re-orthogonalization, *BIT*, *10*, 183–195, 1976.

Parlett, B. N., and D. S. Scott, The Lanczos algorithm with selective re-orthogonalization, *Math. Comput.*, *33*, 217–238, 1979.

Sahuquillo, A., An eigenvalue numerical technique for solving unsteady linear groundwater models continuously in time, *Water Resour. Res.*, *19*, 87–93, 1983.

Shih, T. M., and J. T. Skladany, An eigenvalue method for solving transient heat conduction problems, *Numer. Heat Transfer*, *6*, 409–422, 1983.

Simon, H. D., The Lanczos algorithm with partial re-orthogonalization, *Math. Comput.*, *42*, 115–142, 1984.

Smith, B. T., J. M. Boyle, J. J. Dongarra, and C. B. Moler, *Matrix Eigensystem Routines—EISPACK Guide, Lecture Notes in Computer Science 6*, 2nd ed., Springer-Verlag, New York, 1976.

Townley, L. R., and J. L. Wilson, Description of and user's manual for a finite element aquifer flow model, AQUIFEM-1, *Rep. 252*, Ralph M. Parsons Lab. for Water Resour. and Hydrodyn., Mass. Inst. of Technol., Cambridge, 1980.

W. S. Dunbar, Acres International Limited, Niagara Falls, Ontario, Canada L2E 6W1.

A. D. Woodbury, Department of Geological Sciences, McGill University, 3450 University Street, Montreal, Quebec, Canada H3A 2A7.

(Received February 16, 1988;
revised September 13, 1988;
accepted September 26, 1988.)