LBL-35011
UC-000
Preprint

Lawrence Berkeley Laboratory

UNIVERSITY OF CALIFORNIA

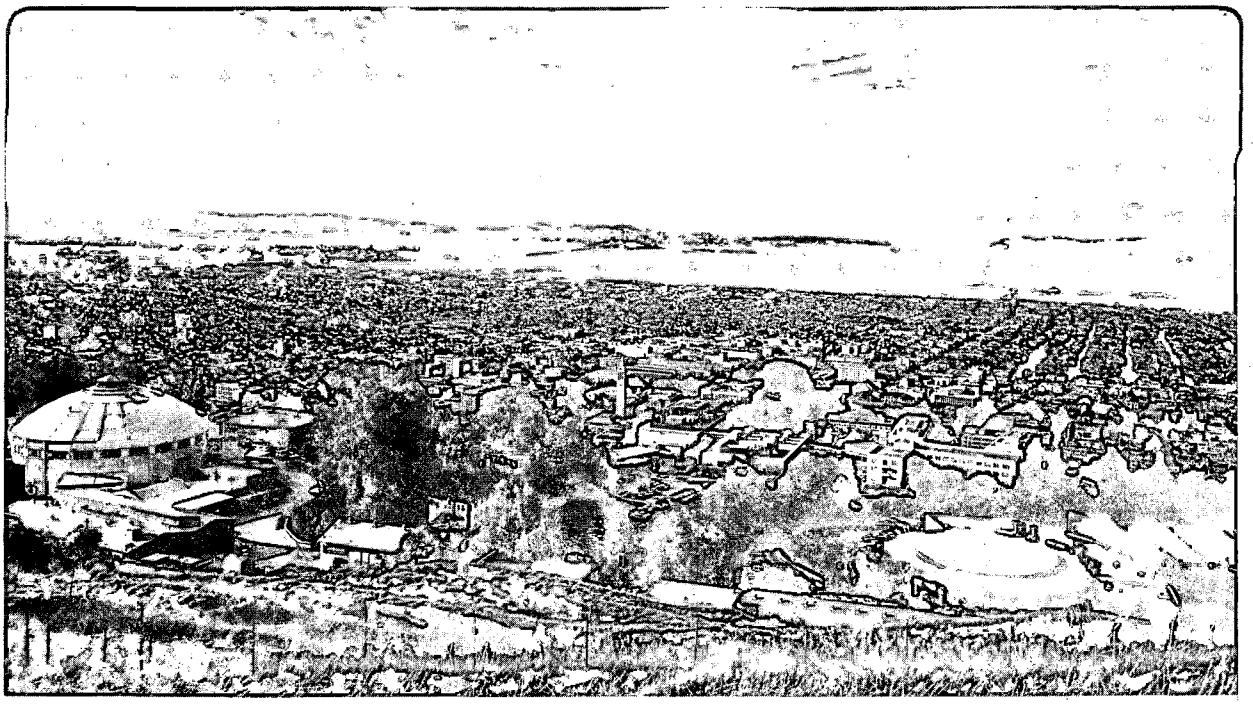
EARTH SCIENCES DIVISION

Submitted to International Journal on Scientific Computing and Modeling

The Laplace Transform Multiquadric Method: A Highly Accurate Scheme for the Numerical Solution of Partial Differential Equations

G.J. Moridis and E.J. Kansa

December 1993



LOAN COPY
Circulates
for 4 weeks
Bldg. 50 Library.
Copy 2
LBL-35011

Prepared for the U.S. Department of Energy under Contract Number DE-AC03-76SF00098

DISCLAIMER

This document was prepared as an account of work sponsored by the United States Government. Neither the United States Government nor any agency thereof, nor The Regents of the University of California, 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 its 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, or The Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or The Regents of the University of California and shall not be used for advertising or product endorsement purposes.

Lawrence Berkeley Laboratory is an equal opportunity employer.

DISCLAIMER

This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal 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 its 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, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.

LBL-35011
UC-000

The Laplace Transform Multiquadric Method: A Highly Accurate Scheme for the Numerical Solution of Partial Differential Equations

G. J. Moridis
Earth Sciences Division
Lawrence Berkeley Laboratory
University of California
Berkeley, CA 94720

and

E.J. Kansa
Physics Department
Lawrence Livermore National Laboratory
Livermore, California 94551

December 1993

This work was supported by the Director, Office of Environmental Restoration and Waste Management, Office of Technology Development, of the U.S. Department of Energy under Contract No. DE-AC03-76SF00098, and the Lawrence Livermore National Laboratory under contract No. W-7405-ENG-48 with the collaboration of the managers and technical staff of the Westinghouse Savannah River Company.

THE LAPLACE TRANSFORM MULTIQUADRIC METHOD: A HIGHLY ACCURATE SCHEME FOR THE NUMERICAL SOLUTION OF PARTIAL DIFFERENTIAL EQUATIONS

G. J. MORIDIS
Earth Sciences Division, Lawrence Berkeley Laboratory, Berkeley, CA 94720

E. J. KANSA
Physics Dept., Lawrence Livermore National Laboratory, Livermore, CA 94551-0808

Abstract– We combine the numerical inversion of Laplace Transforms to integrate partial differential equations (PDEs) in time, with an exponentially-convergent grid-free spatial approximation scheme called multiquadrics (MQ) for the spatial terms. The new method yields remarkably accurate numerical solutions, and the computational effort holds the promise of being orders of magnitude more efficient than traditional finite difference (FD) or finite element (FE) methods.

1. INTRODUCTION

We will present a summary and results of a novel technique for solving partial differential equations (PDEs). We have replaced the traditional finite difference time marching scheme with the numerical inversion of the Laplace Transforms (LT) which eliminates temporal truncaction errors and the need for many time integration steps. In addition, we have replaced the traditional finite difference (FD) and finite element (FE) spatial discretization schemes by an exponentially-convergent, grid-free, scattered data approximation scheme called multiquadrics (MQ). The spatial approximations are not only highly accurate, but also monotonic. When the LT and MQ schemes are combined, they form the LTMQ scheme which yields very accurate numerical solutions when compared to traditional methods. Moreover, the LTMQ scheme has the potential of being orders of magnitude more efficient.

There are two traditional time marching schemes, explicit and implicit, which are based on a low order Taylor series expansion of the previous time-step solutions and the new unknown time solution. Explicit schemes are basically extrapolation methods which are unstable unless the time step is restricted by some relation to the spatial mesh spacing, e.g. the Courant-Friedrichs-Lewy (CFL) time step restriction. Although implicit methods can be made unconditionally stable, large time steps can introduce large temporal truncation errors.

Both FD and FE schemes also suffer from spatial truncation errors. In PDEs which are predominantly hyperbolic, spatial truncation errors propagate along the characteristics contaminating the time-advanced solutions. Evidence of this is the need for upwind differencing for the advective parts of PDEs using either FD or FE schemes. Upwind differencing adds a stabilizing numerical diffusion which can be orders of magnitude larger than the actual physical diffusion. Thus the physics represented by the PDEs are changed

to accommodate the deficiencies of the numerical solution scheme.

The fundamental problem for either FD or FE schemes is that they are restricted to low-order polynomial spatial approximations. Higher-order polynomial schemes are notoriously oscillatory between the collocation data points, resulting in very poor derivative estimates. Hence low-order linear or quadratic schemes are preferred, and are accurate provided that the dependent function is essentially linear or quadratic within the interval under consideration. But if the dependent function is very rapidly varying in space, then the distance between collocation points must be radically decreased to ensure that the function is indeed linear or quadratic in that interval. If U represents a set of dependent variables, the effective length scale is approximately given by

$$\ell = \frac{U}{|\nabla U|} \tag{1}$$

A scheme such as adaptive local mesh refinement is a technique used to refine the spatial discretization; its performance is restricted by the memory limitations of the computer on which the PDE calculations are being performed.

We contend that the familiar problems associated with the truncation errors of traditional FD and FE schemes are due to the very low order, non-monotonic nature of such schemes, and it is more fruitful to examine schemes such as the LTMQ scheme. This paper is divided into four additional sections; Section 2 will discuss the MQ method and introduce the LT method for time integration; Section 3 will discuss two numerical inversion schemes for Laplace Transforms and the procedure for nonlinear problems; Section 4 will present some of our recent LTMQ results for PDE problems, and Section 5 will summarize our results and procedure, and will discuss future directions.

2. THE MQ SCHEME AND ITS USE IN THE PDE SOLUTION

The multiquadric (MQ) method was first introduced by Hardy [1,2] who successfully applied this method for approximating surfaces and bodies from field data; Hardy [3] has written a detailed review article summarizing its rapid growth in use since it was first introduced. In 1982, Franke [4] published a detailed comparison of 29 different scattered data schemes against analytic problems. Of all the techniques tested, he concluded that MQ performed the best in accuracy, visual appeal, and ease of implementation, even against various finite element schemes.

The basic MQ method assumes that any function can be expanded as a finite series of upper hyperboloids,

$$F(\mathbf{x}) = \sum_{j=1}^{N} \alpha_j \, g(\mathbf{x} - \mathbf{x}_j), \quad \mathbf{x} \in \mathbf{R}^d \tag{2}$$

where N is the total number of data centers under consideration,

$$g(\mathbf{x} - \mathbf{x}_j) = \sqrt{(\mathbf{x} - \mathbf{x}_j)^2 + \Delta^2} \tag{3}$$

$(\mathbf{x} - \mathbf{x}_j)^2$ is the square of the Euclidean distance in $\mathbf{R}^d$, and $\Delta^2 > 0$ is an input shape parameter. Note that the basis function g is continuously differentiable, and is a type of spline approximation.

The expansion coefficients α_j are found by solving a set of full linear equations,

$$f(\mathbf{x}_i) = \sum_{j=1}^{N} \alpha_j \; g(\mathbf{x}_i - \mathbf{x}_i) \tag{4}$$

Micchelli [5] proved that MQ belongs to a class of conditionally positive-definite radial basis functions (RBFs). He showed that equation (4) is always solvable for distinct points. Madych and Nelson [6, 7] proved that the MQ interpolation always produces a minimal semi-norm error, and that the MQ interpolant and derivative estimates converge exponentially as the density of data centers increases.

Buhmann [8,9] proved MQ converges spectrally and faster as the spatial dimension increases. Kansa [10,11] argued that the interpolation scheme is intimately connected with the numerical solution of PDEs. For example, the standard finite difference scheme is based on either a linear or quadratic Lagrangian interpolation polynomial (LIP) and the differencing scheme is based upon differentiation of this polynomial. The spatial pattern over which the LIP is defined is known as a stencil; the LIP is continuous up to the order of the polynomial. Across stencils, the function is continuous, but not the derivatives.

In contrast, the MQ interpolant is continuously differentiable over the entire domain of data centers, and the spatial derivative approximations were found to be excellent, most especially in very steep gradient regions where traditional methods fail. This excellent ability to approximate spatial derivatives is due in large part to a slight modification of the original MQ scheme that permits the shape parameter to vary with the basis function.

Instead of using the expansion in equation (2), Kansa [10, 11] used the following:

$$f(\mathbf{x}) = \sum_{j=1}^{N} \alpha_j \; \sqrt{(\mathbf{x} - \mathbf{x}_j)^2 + \Delta_j^2} \tag{5}$$

where

$$\Delta_j^2 = \Delta_{min}^2 \left(\frac{\Delta_{max}^2}{\Delta_{min}^2} \right)^{(j-1)/(N-1)} \qquad \text{for} \quad j = 1, 2, \ldots, N \tag{6}$$

and Δ_{max}^2 and Δ_{min}^2 are two input parameters chosen so that the ratio

$$\Delta_{max}^2 / \Delta_{min}^2 \simeq 10 \text{ to } \; 1,000,000.$$

Baxter [12] and Madych [13] have proved that under certain circumstances very large values of the shape parameter are desirable. As Kansa [10, 11] showed, the ad hoc formula in equation (5) is a way to have at least one very large value of a shape parameter without incurring the onset of severe ill-conditioning problems.

Spatial partial derivatives of any function are formed by differentiating the spatial basis functions. Consider a two-dimensional problem, the first and second derivatives are given by simple differentiation:

$$\left.\frac{\partial f}{\partial x}\right|_{x_i} = \sum_{j=1}^{N} \alpha_j \; (x_i - x_j)/g_{ij} \tag{7}$$

$$\left.\frac{\partial f}{\partial y}\right|_{x_i} = \sum_{j=1}^{N} \alpha_j \; (y_i - y_j)/g_{ij} \tag{8}$$

$$\left.\frac{\partial^2 f}{\partial x^2}\right|_{x_i} = \sum_{j=1}^{N} \alpha_j \; \left\{1 - \left[(x_i - x_j)/g_{ij}\right]^2\right\}/g_{ij} \tag{9}$$

$$\left.\frac{\partial^2 f}{\partial y^2}\right|_{x_i} = \sum_{j=1}^{N} \alpha_j \; \left\{1 - \left[(y_i - y_j)/g_{ij}\right]^2\right\}/g_{ij} \tag{10}$$

$$\left.\frac{\partial^2 f}{\partial x \partial y}\right|_{x_i} = -\sum_{j=1}^{N} \alpha_j \; (x_i - x_j)/(y_i - y_j)/g_{ij}^3 \tag{11}$$

where

$$g_{ij} = \left[(x_i - x_j)^2 + (y_i - y_j)^2 + \Delta_j^2\right]^{1/2} \tag{12}$$

Equations (5) and (7) through (12) give us a framework to approximate PDEs using the MQ basis function expansions. In general, we treat a domain Ω over which the PDEs are defined as a set of boundary condition (BC) problems and the interior problem. Let Γ denote the boundary over which Dirichlet and/or Neumann BC are applied; let the total number of data centers defining Γ be M. In the interior region Ω/Γ, the number of data centers at which the PDEs are defined is $(N - M)$. So together with the BC and interior points, there are N equations in N unknowns. The MQ PDE problem for both elliptic and hyperbolic and parabolic PDEs can be recast into the following form:

$$\mathbf{W} \; \underline{\alpha} = \mathbf{p} \tag{13}$$

where the coefficient matrix $\mathbf{W}$ and the column vector $\mathbf{p}$ are partitioned as follows:

$$\mathbf{W} = \begin{pmatrix} \mathbf{W}(\Omega/\Gamma) \\ \mathbf{W}(\Gamma) \end{pmatrix}, \qquad \mathbf{p} = \begin{pmatrix} \mathbf{p}(\Omega/\Gamma) \\ \mathbf{p}(\Gamma). \end{pmatrix} \tag{14}$$

If W is nonsingular, then the expansion coefficients can be found as $\underline{\alpha} = \mathbf{W}^{-1} \; \mathbf{p}$. Once the expansion coefficients are determined, the dependent variable over Ω is found at any point $\mathbf{x}$ using equation (5).

The procedure for forming N equations in N unknowns is different for time-independent elliptic PDEs than it is for time-dependent hyperbolic and/or parabolic PDEs. To illustrate some of the more general procedures, let us consider first the following linear elliptic PDE problem in two dimensions,

$$a_1 \frac{\partial U}{\partial x} + a_2 \frac{\partial U}{\partial y} - a_3 \frac{\partial^2 U}{\partial x^2} - a_4 \frac{\partial^2 U}{\partial y^2} = Q(x, y). \tag{15}$$

defined over the unit square in Figure 1. Along Γ_1 there are M_1 points at which U is specified as

$$U(x, y) = w(x, y = 0) = v(x) \tag{16}$$

Along Γ_2 there are M_2 points at which U is specified as

$$U(x, y) = z(x = 0, y) = z(y) \tag{17}$$

Along Γ_3 there are M_3 points at which the gradient is specified as

$$\frac{\partial U(x, y)}{\partial x} = q(x, y = 1) = q(x) \tag{18}$$

Finally, Γ_4 there are M_4 points at which the gradient is specified as

$$\frac{\partial U(x, y)}{\partial y} = r(x = 1, y) = r(y) \tag{19}$$

A system of N equations in N unknowns can be formed with the following matrix structure:

$$\mathbf{W}\ \underline{\alpha} = \mathbf{p}$$

where

$$\mathbf{W} = \begin{pmatrix} \mathbf{W}(\Omega/\Gamma) \\ \mathbf{W}(\Gamma_1) \\ \mathbf{W}(\Gamma_2) \\ \mathbf{W}(\Gamma_3) \\ \mathbf{W}(\Gamma_4), \end{pmatrix} \qquad \mathbf{p} = \begin{pmatrix} \mathbf{p}(\Omega/\Gamma) \\ \mathbf{p}(\Gamma_1) \\ \mathbf{p}(\Gamma_2) \\ \mathbf{p}(\Gamma_3) \\ \mathbf{p}(\Gamma_4). \end{pmatrix} \tag{20}$$

The matrix $\mathbf{W}$ has the following components which change with the row index i:
On Ω/Γ we have

$$\begin{aligned} W_{ij} = \frac{1}{g_{ij}} \big[& a_1(x_i - x_j) + a_2(y_i - y_j) - (a_3 + a_4) \\ & + \frac{[a_3(x_i - x_j)^2 + a_4(y_i - y_j)^2]}{g_{ij}} \big] \qquad \forall\, j \in [1, N], \quad \text{and} \\ p_i = Q(x_i, y_i) \qquad & \text{for} \quad i = 1, 2, \ldots, (N - M). \end{aligned} \tag{21}$$

On Γ_1 we have

$$\begin{aligned} W_{ij} &= g_{ij} \quad \forall\, j \in [1, N], \quad \text{and} \\ P_i &= w(x_i) \qquad \text{for} \quad i = (N - M + 1), \ldots, (N - M + M_1). \end{aligned} \tag{22}$$

On Γ_2 we have

$$\begin{aligned} W_{ij} &= g_{ij} \quad \forall\, j \in [1, N], \quad \text{and} \\ P_i &= z(y_i) \qquad \text{for } i = (N - M + M_1 + 1), \ldots, (N - M + M_1 + M_2). \end{aligned} \tag{23}$$

On Γ_3 we have

$$\begin{aligned} W_{ij} &= \frac{x_i - x_j}{g_{ij}} \quad \forall\, j \in [1, N], \quad \text{and} \\ P_i &= q(x_i) \qquad \text{for } i = (N - M + M_1 + M_2 + 1), \ldots, (N - M + M_1 + M_2 + M_3). \end{aligned} \tag{24}$$

On Γ_4

$$\begin{aligned} W_{ij} &= \frac{y_i - y_j}{g_{ij}} \quad \forall\, j \in [1, N], \quad \text{and} \\ P_i &= r(y_i) \qquad \text{for } i = (N - M - M_1 + M_2 + M_3 + 1), \ldots, N, \end{aligned} \tag{25}$$

where

$$M = M_1 + M_2 + M_3 + M_4 \quad \text{and} \quad g_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + \Delta_j^2}. \tag{26}$$

If the constants, a_1, a_2, a_3, and a_4 are chosen so that $\mathbf{W}$ is nonsingular, then equation (17) is solvable. The function U can be found everywhere within the unit square Ω by interpolation after the coefficient α_j is determined as

$$\underline{\alpha} = \mathbf{W}^{-1}\, \mathbf{p}, \tag{27}$$

and

$$U(x, y) = \sum_{j=1}^{N} \alpha_j \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2 + \Delta_j^2} \tag{28}$$

An example of a linear 2D Poisson equation solved by the MQ method is given by Kansa [11]. In the next example, we will consider a system of linear PDEs given by

$$a_1 \frac{\partial u}{\partial x} + a_2 \frac{\partial v}{\partial y} = G(x, y) \tag{29}$$

$$a_3 \frac{\partial u}{\partial x} + a_4 \frac{\partial v}{\partial y} = H(x, y) \tag{30}$$

with BCs

$$u = (x = 0, y) = s(y) \tag{31}$$

$$v = (x, y = 0) = V(x) \tag{32}$$

Suppose there are M_1 boundary points along $x = 0$, and M_2 boundary points along $y = 0$, and $M_1 \neq M_2$. The total number of data centers is N. Since there are $2N$ unknown expansion coefficients, N for the ones corresponding to u and another N for the ones corresponding to v, we must have $2N$ equation in $2N$ unknowns. The coefficient matrix $\mathbf{W}$ will be partitioned into four $N \times N$ block matrices. and the column vector $\mathbf{p}$ will partitioned into two rank N column vectors of the form:

$$\mathbf{W} = \begin{pmatrix} \mathbf{W}_1(u, \mathrm{PDE}^1) & \mathbf{W}_2(v, \mathrm{PDE}^1) \\ \mathbf{W}_3(u, \mathrm{PDE}^2) & \mathbf{W}_4(v, \mathrm{PDE}^2) \end{pmatrix} \qquad \mathbf{p} = \begin{pmatrix} \mathbf{p}_1(\mathrm{PDE}^1) \\ \mathbf{p}_2(\mathrm{PDE}^2) \end{pmatrix}, \tag{33}$$

where $\underline{\alpha} = [\underline{\alpha}^u, \underline{\alpha}^v]^T$, and $\mathrm{PDE}^1$and PDE^2 refer to equations (29) and (30) respectively.

We shall specify the elements of the matrices and column vectors. For $i = 1, 2, \ldots, (N - M_1)$ and for all $j \notin [1, N]$,

$$w_{1ij} = a_1 \frac{x_i - x_j}{g_{ij}}, \quad \text{and} \quad p_{1i} = G(x_i, y_j) \tag{34}$$

and

$$w_{1ij} = g_{ij} \quad \text{for } j \in [1, N] \quad \text{and} \quad p_{1i} = s(y_i) \text{ for } i = (N - M_1), \ldots, N. \tag{35}$$

Similarly,

$$w_{2ij} = a_2 \frac{y_i - y_j}{g_{ij}} \text{ for } j \in [N + 1, 2N], \quad \text{and} \quad p_i = G(x_i, y_j) \text{ for } i \in [1, N - M_1], \tag{36}$$

$$w_{2ij} = 0 \text{ for } i = (N - M_1), \ldots, N. \tag{37}$$

Similar constructions are performed for the second PDE to be solved:

$$w_{3ij} = a_3 \frac{x_i - x_j}{g_{ij}} \text{ for } j \in [1, N], \quad \text{and} \quad p_{2i} = H(x_i, y_j) \text{ for } i \in [N + 1, 2N - M_2], \tag{38}$$

$$w_{3ij} = 0 \text{ for } i = (2N - M_2 + 1), \ldots, 2N. \tag{39}$$

For the last block matrix,

$$w_{4ij} = a_4 \frac{y_i - y_j}{g_{ij}} \text{ for } j \in [N+1, 2N], \quad \text{and} \quad p_{2i} = H(x_i, y_j) \text{ for } i \in [N+1, 2N-M_2], \tag{40}$$

$$w_{4ij} = g_{ij} \text{ for } j \in [1, N], \quad \text{and} \quad p_{2i} = V(x_i) \text{ for } i = (2N - M_2 + 1), \ldots, 2N. \tag{41}$$

The above procedure generalizes to any number of PDEs.

We can also generalize using MQ to systems of nonlinear PDEs. Consider

$$u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = G(x, y), \qquad u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = H(x, y) \tag{42}$$

with BCs

$$u(x = 0, y) = s(y) \qquad \text{and} \qquad v(x, y = 0) = V(x). \tag{43}$$

The set up for this problem is quite similar to the previous problem. Instead of dealing with a set of constants a_1, a_2, a_3, a_4, we use the appropriate current iterate values of $u(x_i, y_i)$ and $v(x_i, y_i)$ for the elements of the coefficient matrix. Let us define a new column vector $U = [u, v]^T$. Let S be an operator such that, by rewriting equations (42) and (43),

$$S(U) = 0. \tag{44}$$

That is,

$$S(U) = \begin{pmatrix} u\frac{\partial}{\partial x} & v\frac{\partial}{\partial y} \\ u\frac{\partial}{\partial x} & v\frac{\partial}{\partial y} \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} - \begin{pmatrix} G \\ H \end{pmatrix} = 0. \tag{45}$$

Equation (44) represents a system of nonlinear equations in the set of expansion coefficients, $[\alpha^u, \alpha^v]^T$, similar to what was done previously in equations (34) through (41). For illustrative purposes let us assume that we choose to solve equation (44) by the Newton-Raphson (NR) method. Then the NR correction vector at the kth iteration

$$\delta\, \underline{\alpha}^k = \left(\delta\alpha_1^k, \ldots, \delta\alpha_N^k\right)^T$$

is given by

$$\delta\, \underline{\alpha}^k = \mathbf{J}^{-1}\, S(U^{k-1}) \tag{46}$$

where the Jacobian $\mathbf{J}$ is given by

$$\mathbf{J} = \frac{\partial S}{\partial U}\frac{\partial U}{\partial \alpha^k} \tag{47}$$

Dubal [14] and Cook et al. [15] used MQ to solve the three dimensional nonlinear Poisson equation representing the initial state of the collision of two black-holes with arbitrary linear and angular momentum. Dubal [14] used the method of successive substitutions, and obtained satisfactory convergence within seven iterations. He also found that as he increased the number of data centers the solution accuracy increased exponentially, as predicted by theory. However, Dubal determined that ill-conditioning became a serious problem when solving his system of equations with 1094 data centers. Singular value decomposition was then needed to treat this problem.

Based on Foley's [17] work, we recommend using domain decomposition and blending whenever the number of data centers becomes too large, say above 70-100. There are several advantages to this divide and conquer strategy. First, the number of operations to solve a system of linear equations is $O(N^3)$. Second, the condition number of a full matrix increases as N increases.

Domain decomposition avoids the problems associated with the solution of large matrices, i.e. the large numbers of operations needed to solve a system of linear equations and the accompanying ill-conditioning problem. In a recent paper, Dubal [16] used a combination of domain decomposition and local refinement to solve elliptical PDEs very accurately, economically, and efficiently. He required that the function and its normal and tangential gradients be equal at the interfaces of the subdomains, thus forcing at least C^2 continuity at these points.

Galerpin and Zhang [18] combined MQ within the context of global optimization to solve PDEs. Their approach eliminates the need to deal with matrices and the unresolved question of optimally choosing the shape parameter distribution Δ_j^2. Since any set of PDEs can be cast into the form $S(U) = 0$, they minimized a functional Φ which consists of a weighted sum of the integral over the time and space domains over which the PDEs are defined, as well as constraints representing the initial and boundary conditions.

Global optimization is only efficient if the total number of parameters to be optimized at any given time is relatively small. If the domain Ω is partitioned into several subdomains over which only a few basis functions are used, then the optimization requires relatively few operations to determine the global optimum in each subdomain. This "divide-and-conquer" strategy lends itself readily to implementation on massively parallel computers.

Let us turn our attention to the treatment of time-dependent PDEs using the MQ expansion scheme. We first make the assumption that either in the fixed Eulerian frame or else in some moving Gallilean frame, the temporal and spatial parts of any dependent variable are separable. That is, we assume

$$U(\mathbf{x}, t) = \sum_{j=1}^{N} \alpha_j(t)\ g(\mathbf{x} - \mathbf{x}_j), \quad \mathbf{x} \in \mathbf{R}^d \tag{48}$$

As a first example, consider the following simple advection equation,

$$\frac{\partial U}{\partial t} + c_1 \frac{\partial U}{\partial x} + c_2 \frac{\partial U}{\partial y} = 0 \tag{49}$$

over a unit square subject to

$$U(x, y = 0) = E(x) \tag{50}$$

$$U(x = 0, y) = F(y) \tag{51}$$

After substituting equations (31) and (32) into equations (33) through (41), we obtain the following set of ordinary differential equations,

$$\sum_{j=1}^{N} \left(\frac{d\alpha_j}{dt}\ g_{ij} + \alpha_j\ h_{ij} \right) = 0 \tag{52}$$

where

$$h_{ij} = \frac{c_1(x_i - x_j) + c_2(y_i - y_j)}{g_{ij}} \quad \text{for } i = 1, \ldots, (N - M) \text{ and } j \in [1, N] \tag{53}$$

$$\sum_{j=1}^{N} \alpha_j \; g_{ij} = E(x_i) \quad \text{for } i = (N - M + 1), \ldots, (N - M + M_1), \tag{54}$$

$$\sum_{j=1}^{N} \alpha_j \; g_{ij} = F(y_i) \quad \text{for } i = (N - M + M_1 + 1), \ldots, N. \tag{55}$$

We shall examine two methods to treat the system of time-dependent ordinary differential equations and BC.

The first way is to use classical time marching methods, such as presented in Kansa [11], where either implicit methods or explicit higher-order schemes were considered. Using either explicit or implicit time marching schemes for the interior nodes and the BC still yields as system of N equations in N unknown expansion coefficients. Once the expansion coefficients are found, the solution at any time, t, over all points $\mathbf{x}$ in the domain can be found.

Let t^n be the current time at the nth time step, and $t^{n+1} = t^n + \Delta t$. Then a simple time derivative approximation is

$$\frac{d\alpha_j}{dt} \simeq \frac{\alpha_j^{n+1} - \alpha_j^n}{\Delta t} + O(\Delta t^2) \tag{56}$$

so equation (52) can be rewritten as

$$\sum_{j=1}^{N} (\alpha_j^{n+1} - \alpha_j^n) \; g_{ij} + \sum_{j=1}^{N} \left[\theta \; \Delta t \; \alpha_j^n + (1 - \theta) \; \Delta t \; \alpha_j^{n+1}\right] \; h_{ij} = 0. \tag{57}$$

Equation (57) can be written as:

$$\sum_{j=1}^{N} \alpha_j^{n+1} \left[g_{ij} - (1 - \theta) \; \Delta t \; h_{ij}\right] = b, \tag{58}$$

where

$$b_i = \sum_{j=1}^{N} (g_{ij} - \theta \; \Delta t \; h_{ij}) \, \alpha_j^n + O(\Delta t^2) \quad \text{for } i = 1, \ldots, (N - M), \text{ and } j \in [1, N], \tag{59}$$

$$\sum_{j=1}^{N} \alpha_j^{n+1} \; g_{ij} = E(x_i) \quad \text{for } i = (N - M + 1), \ldots, (N - M + M_1), \quad j \in [1, N], \tag{60}$$

and

$$\sum_{j=1}^{N} \alpha_j^{n+1}\, g_{ij} = F(y_i) \quad \text{for } i = (N - M + M_1 + 1), \ldots, N. \;\; j \in [1, N]. \tag{61}$$

If $\theta > 1/2$, the implicit time-marching scheme is stable. However, we still have truncation errors of order $O(\Delta t^2)$. Although we have a very accurate spatial approximation, the temporal truncation errors can start to contaminate the solutions in time because error propagate along characteristics. To initialize the time marching scheme we first must solve the interpolation problem for the initial value problem to find the N expansion coefficients at $t = 0$.

We shall concentrate our efforts in replacing the standard time marching scheme by the Laplace Transform scheme. The next section will deal specifically with this issue.

3. THE LTMQ SCHEME

There are two ways to approach the problem. In the first (which is also easier and simpler), the Laplace transform of the original PDE is taken first. Then the MQ scheme is applied to the transformed PDE. In the second approach the MQ scheme is applied first, and the resulting equation is then subjected to a Laplace transform. In this case we use the fact that

$$\mathcal{L}\left\{\frac{d\alpha_j^n}{dt}\right\} = \lambda\, \hat{\alpha}_j(\lambda) - \alpha_j(0), \tag{62}$$

and

$$\mathcal{L}\{\alpha_j(t)\} = \hat{\alpha}_j(\lambda), \tag{63}$$

where $\mathcal{L}\{\}$ and $\hat{}$ denote the Laplace transform of the quantity in the brackets, and λ is the Laplace space parameter. In the Laplace space, this results in

$$\sum_{j=1}^{N} [\lambda\, \hat{\alpha}_j(\lambda)\, g_{ij} + \hat{\alpha}_j(\lambda)\, h_{ij}] = e_i \quad \text{for } i = 1, \ldots, (N - M), \text{ and } j \in [1, N], \tag{64}$$

where

$$e_i = \sum_{j=1}^{N} \alpha_j(0)\, g_{ij}, \tag{65}$$

and g_{ij} is defined by equation (12), with the BCs

$$\sum_{j=1}^{N} \hat{\alpha}_j\, g_{ij} = \hat{E}(x_i) \quad \text{for } i = (N - M + 1), \ldots, (N - M + M_1), \;\; j \in [1, N], \tag{66}$$

and

$$\sum_{j=1}^{N} \hat{\alpha}_j\, g_{ij} = \hat{F}(y_i) \quad \text{for } i = (N - M + M_1 + 1), \ldots, N, \;\; j \in [1, N]. \tag{67}$$

Equations (64) trough (67) define a system of $N \times N$ linear equations in the unknown Laplace Transformed expansion coefficients $\hat{\alpha}_j(\lambda)$ and the known coefficients at time $t = 0$. To find the expansion coefficients at time $t = t_{obs}$, we first must find the expansion coefficients at time $t = 0$, i.e. the initial value problem at all N locations. The obvious advantage of the first approach is that there is no need to find the expansion coefficients at time $t = 0$, thus reducing the execution time requirements.

The next step is to solve the resulting set of linear equations

$$\widehat{\mathbf{W}}(\lambda)\ \underline{\hat{\alpha}}(\lambda) = \hat{\mathbf{p}}(\lambda) \rightarrow \underline{\hat{\alpha}}(\lambda) = \widehat{\mathbf{W}}^{-1}(\lambda)\ \hat{\mathbf{p}}(\lambda) \tag{68}$$

in the LT space, then numerically invert the LT solution at the desired time, t. Note that the LT inversion needs no intermediate solutions to obtain the desired solution at $t = t_{obs}$. Not only have we made the time integration process very efficient, we have also eliminated time truncation errors.

The solution in the Laplace space is obtained from

$$\widehat{\mathbf{W}}^{-1}(\lambda)\ \hat{\mathbf{p}}(\lambda) \tag{69}$$

in a manner entirely analogous to that of equation (27). The computation of the coefficient matrix $\widehat{\mathbf{W}}^{-1}(\lambda)$ and the right-hand side vector $\hat{\mathbf{p}}(\lambda)$ in equation (69) necessitates values for the λ parameter of the Laplace space. These are provided by the two schemes: the Stehfest [19] algorithm, and the DeHoog [20] method. For a desired observation time t, the λ in the Stehfest algorithm is real and given by

$$\lambda_\nu = \frac{\ln 2}{t} \cdot \nu, \qquad \nu = 1, \ldots, N_S, \tag{70}$$

where N_S is the number of summation terms in the algorithm, and must be an even number.

In the DeHoog method, λ is a complex number given by Crump [21] and Sudicky [22] as

$$\lambda_\nu = \lambda_0 + \frac{\nu \pi j}{\Omega_p}, \qquad \lambda_0 = \mu - \frac{\ln(E_R)}{2T}, \qquad \nu = 1, \ldots, N_H \tag{71}$$

where $2\Omega_p$ is the period of the Fourier series approximating the inverse function in the interval $[0, 2\Omega_p]$, $j = \sqrt{-1}$, and $N_H = 2M_H + 1$ is an odd number. A thorough discussion of the terms μ and E_R and their significance can be found in Sudicky [21]. As is later discussed, excellent results are obtained when $\mu = 0$, $10^{-10} \leq E_R \leq 10^{-8}$, and $0.85\ t_{max} \leq \Omega_p \leq 1.15\ t_{max}$, where t_{max} is the maximum simulation time.

The solution of equation (69) returns a set of N_S or N_H vectors of the unknown $\underline{\hat{\alpha}}(\lambda)$s as

$$\underline{\hat{\alpha}}_\nu(\lambda) = \widehat{\mathbf{W}}_\nu^{-1}(\lambda_\nu)\ \hat{\mathbf{p}}_\nu(\lambda_\nu)\ , \quad \nu = 1, \ldots, N_S \text{ or } N_H. \tag{72}$$

To obtain a solution at a time t, all vectors $\underline{\hat{\alpha}}_\nu$, $\nu = 1, \ldots, N_S$ or N_H are needed, i.e. the system of simultaneous equations has to be solved N_S or N_H times.

When using the DeHoog method, the matrix equation (72) involves complex numbers. The simplest way to solve equation (72) is to declare all the quantities involved as complex variables, and use the ability of FORTRAN to perform complex arithmetic operations. However, this is a very inefficient approach because operations on complex numbers are very slow. A much faster execution is achieved by splitting each one of the discretized equations into two equations containing the real and the imaginary parts of equation (72) respectively. This approach uses only real computer arithmetic, yields 2 equations per node, and requires significantly less execution time than using complex arithmetic.

The unknown vector $\mathbf{v} = \mathbf{v}(t)$ at any time t is obtained by by numerically inverting the Laplace space solutions $\hat{\mathbf{v}}(\lambda)$. When using the Stehfest algorithm, the unknown $\mathbf{v}(t)$ is obtained from

$$\mathbf{v}(t) = \frac{\ln 2}{t} \sum_{\nu=1}^{N_S} W_\nu \, \hat{\mathbf{v}}(\lambda_\nu) \ , \tag{73}$$

where

$$W_\nu = (-1)^{\frac{N_S}{2}+\nu} \sum_{k=\frac{1}{2}(\nu+1)}^{\min\{\nu,\frac{N_S}{2}\}} \frac{k^{\frac{N_S}{2}}(2k!)}{(\frac{N_S}{2}-k)!k!(k-1)!(\nu-k)!(2k-\nu)!} \ . \tag{74}$$

Although on theoretical grounds the accuracy of the method is expected to improve with increasing N_S, Stehfest [20,23] showed that with increasing N_S the number of correct significant figures increases linearly at first and then, due to roundoff errors, decreases linearly. He determined that the optimum N_S was 10 for single precision variables (8 significant figures) and 18 for double precision variables (16 significant figures). However, Moridis and Reddell [24,25,26] reported that the solution seems to be insensitive to N_S for $6 \leq N_S \leq 20$ in diffusion-type PDEs.

The inversion of the Laplace space solution obtained with the DeHoog method is more complicated. The solution $V(t)$ is given by

$$V(t) = \frac{1}{\Omega} \exp(\lambda_0 t) \ \mathrm{Re}\left\{\frac{A_{2M}}{B_{2M}}\right\}, \tag{75}$$

where

$$A_n = A_{n-1} + d_n \ z \ A_{n-2}, \quad B_n = B_{n-1} + d_n \ z \ B_{n-2}, \quad n = 1, \ldots, 2M, \tag{76}$$

and $A_{-1} = 0$, $A_0 = d_0$, $B_{-1} = B_0 = 1$;

$$d_0 = a_0, \quad d_{2m-1} = -q_m^{(0)}, \quad d_{2m} = -e_m^{(0)}, \qquad m = 1, \ldots, M, \tag{77}$$

$$\begin{aligned} \ell = 1, \ldots, M, \qquad & e_\ell^{(\kappa)} = q_\ell^{(\kappa+1)} - q_\ell^{(\kappa)} + e_{\ell-1}^{(\kappa+1)}, \qquad \kappa = 0, \ldots, 2M-2\ell \\ \text{for } \ell = 2, \ldots, M, \qquad & q_\ell^{(\kappa)} = q_{\ell-1}^{(\kappa+1)} \ e_{\ell-1}^{(\kappa+1)} / e_{\ell-1}^{(\kappa)}, \qquad \kappa = 0, \ldots, 2M-2\ell-1, \end{aligned} \tag{78}$$

$$e_0^{(\kappa)} = 0 \text{ for } \kappa = 0, \ldots, 2M \quad \text{and} \quad q_1^{(\kappa)} = a_{\kappa+1}/a_\kappa \text{ for } \kappa = 0, \ldots, 2M-1, \tag{79}$$

and

$$a_0 = \frac{1}{2}V(\lambda_0), \quad a_\kappa = \underline{\hat{V}}(\lambda_\kappa) \quad \text{and} \quad z = \exp(j\pi t/\Omega). \tag{80}$$

A further acceleration is obtained if on the last evaluation of the recurrence relation zd_{2M} is repaced by $R_{2M}(z)$,

$$R_{2M}(z) = -h_{2M}\left[1 - \sqrt{(1 + d_{2M}\ z/h_{2M})}\right], \quad h_{2M} = \frac{1}{2}[1 + z\ (d_{2M-1} - d_{2M})], \tag{81}$$

giving

$$\tilde{A}_{2M} = A_{2M-1} + R_{2M}\ A_{2M-2}, \qquad \tilde{B}_{2M} = B_{2M-1} + R_{2M}\ B_{2M-2}, \tag{82}$$

in which case the accelerated solution at a time t is given by

$$V(t) = \frac{1}{\Omega}\ \exp(\lambda_0 t)\ \mathrm{Re}\left\{\frac{\tilde{A}_{2M}}{\tilde{B}_{2M}}\right\}. \tag{83}$$

It should be kept in mind that all the operations in equations (76) through (82) involve complex variables, and computationally intensive complex computer arithmetic must be used. As was shown by Moridis [27], the minimum M_H for an acceptable accuracy is 5, resulting in a $N_H = 11$, which indicates that the matrix equation (72) has to be solved a minimum of 11 times. For an accuracy comparable to that of the Stehfest method $M_H \geq 6$ and $N_H \geq 13$.

The DeHoog formulation offers two advantages. The first advantage is that a whole range of solutions at times t in the range $[0, T]$ can be obtained from a single set of solutions s, i.e. equation (72) needs not be solved for each t of interest. However, Moridis [27] showed that accurate solutions are obtained for $0.15T \leq t \leq T$, but significant errors are possible for $t < 0.15T$. The second advantage is the ability of the DeHoog algorithm to invert very steep solution surfaces, such as spikes and step functions. The main disantvantage of the DeHoog algorithm is the larger storage requirements (four times that of the Stehfest algorithm) and the larger (by an order of magnitude) execution time requirements [27].

The solution in the Laplace space eliminates stability and accuracy problems caused by the treatment of the time derivative in standard FD simulators, thus allowing an unlimited time-step size. The truncation error of the method is limited to that caused by the space discretization because time is not discretized, and provides a solution inherently more accurate than the standard FD method for the same grid system. The ability to use an unlimited time-step size bounds the accumulation of roundoff error by an upper limit defined as the roundoff error accumulated after the N_S or N_H solutions. Thus, LTFD offers a stable, non-increasing roundoff error irrespective of the time of observation t_{obs}, because calculations are performed at one time only by letting $\Delta t = t_{obs}$. Calculations in the standard FD method have to be performed at all the intermediate times of the discretized time domain, continuously accumulating roundoff error in the process.

4. VERIFICATION AND EVALUATION

The LTMQ method was tested in three problems of groundwater flow for which analytical solutions exist. The LTMQ solution was verified through comparison to the analytical solutions, as well as the solutions obtained from a standard implicit FD simulator. Direct solvers were used to solve simultaneous equations in the LTMQ and FD methods. Double precision variables with 20 significant figures were used in all simulations. All the following examples are from the area of flow and transport through porous media, in which the two authors specialize. However, the LTMQ method is applicable to any class of diffusion and advection-diffusion PDEs.

4.1. Verification and Test Case 1

Test case 1 investigated the one-dimensional radial flow problem towards a well of radius $r_w \to 0$ in a homogeneous circular aquifer with infinite boundaries, the analytical solution to which was given by Theis [28]. The geometry, properties, and a FD solution of this problem can be found in Moridis and Reddell [23]. A single observation was made at $t_{obs} = 10$ *days*. We used a $N = 7$ points, and a $N_S = 8$ and $N_H = 13$ for the Stehfest and DeHoog versions of the LTMQ solution.

The Stehfest and DeHoog versions of the LTMQ solution return virtually identical results, and will be referred to by the inclusive term 'LTMQ solution' hereafter. The drawdown $s = H - H(0)$ for both the analytical and the numerical solutions (LTMQ and FD) appear in Figure 2. For a more representative comparison, the LTMQ curve (in this and the remaining test cases) is computed from the MQ interpolation at the locations of the FD grid centers. The Theis solution and the LTMQ solution practically coincided, while the FD solution tended towards the LTMQ and the Theis solutions with an increasing number of time-steps (corresponding to smaller Δt's).

Figure 3 shows the effect of N_S on the accuracy of the Stehfest LTMQ scheme. The difference between the Theis and the LTMQ solutions is negligible for $6 \leq N_S \leq 20$. This implies that the accuracy of the Stehfest LTMQ method for this one-dimensional problem is practically insensitive to the value of N_S, and that a $N_S = 6$ suffices for an accurate solution. However, for robust results $N_S = 8$ or $N_S = 10$ is a better choice. This drastically reduces the execution time and makes the LTMQ method even more efficient than theoretically predicted [19].

Figure 4 shows the effect of N_H on the accuracy of the DeHoog LTMQ scheme. The DeHoog LTMQ solution approximates very accurately the Theis solution for $N_H \geq 11$, and for $N_H \geq 13$ the LTMQ solutions coincide. The LTMQ solution for this one-dimensional problem is practically insensitive to the value of N_H, and a $N_H = 11$ suffices for an accurate solution. However, for more robust solutions $N_H = 13$ is preferable.

4.2. Verification and Test Case 2

The second test case involved one-dimensional groundwater flow from a point of high head (at the left boundary) towards a low head at the right hand boundary. The initial head is $H_1 = H(x, 0)$; at $t = 0$ the head at the right boundary is lowered to H_2. As long

as the head at $x = 0$ remains unchanged, the piezometric head $H(x,t)$ is given by

$$H(x,t) = H_1 + (H_2 - H_1)\ \mathrm{erfc}\left\{(L-x)\ (S_0\ Z)^{1/2}\ (4\ K_x\ Z\ t)^{-1/2}\right\}, \tag{84}$$

where L is the length of the system, and Z is its thickness. The size and properties of this system appear in Figure 5, which compares the LTMQ solution to the (a) analytical solution and (b) the FD solution (obtained with a domain discretization in 52 equally-spaced gridblocks). For the LTMQ solution, $N = 10$, $N_S = 8$, and $N_H = 11$.

The two LTMQ solutions are virtually identical. The accuracy of the LTMQ solution is demonstrated by its virtual coincidence with (a) the analytical solution, and (b) the FD solution obtained with a large number of small Δt's. For larger Δt's, the FD solution shows insufficient accuracy caused by larger truncation errors in the approximation of the time derivative.

4.3. Verification and Test Case 3

Test case 3 represented transient flow into a homogeneous and anisotropic aquifer with a fully penetrating well and constant discharge conditions. The origin of this two-dimensional, infinite-acting system is placed at the well. Assuming that the axes of the cartesian system coincide with the principal axes of the permeability tensor, the piezometric head distribution at $t = 20$ *days* is predicted along the $x = y$ axis, i.e. at an angle of 45° from the x-axis. Only one quarter of the infinite domain (i.e. x in $[0,\infty)$, y in $[0,\infty)$) needs to be simulated in LTMQ and FD. For the LTMQ solution, $N = 35$, $N_S = 8$, and $N_H = 11$. A total of 625 gridblocks were used in the FD simulation.

Figure 6 presents (a) the analytical solution of Papadopulos [29], (b) the LTMQ solution, (c) the FD solutions, as well as (d) relevant information on the parameters used in this simulation. The same pattern observed in the two previous test cases is obvious: LTMQ produces an accurate solution, a fact indicated by its virtual coincidence with both the analytical solution and the FD solution for a large number of small Δt's.

4.4. Verification and Test Case 4

In test case 4 the problem of solute transport through porous media is solved. The one-dimensional advection-diffusion PDE describing this problem is

$$D\frac{\partial^2 C}{\partial x^2} - V\frac{\partial C}{\partial x} = \frac{\partial C}{\partial t,} \tag{85}$$

with initial and boundary conditions

$$C(x, t = 0) = 0, \qquad C(x = 0, t) = 1. \tag{86}$$

An analytical solution to this problem derived by Ogata and Banks [30] is given by

$$C(x,t) = \frac{1}{2}\left\{\mathrm{erfc}\left[\frac{x - Vt}{2\sqrt{Dt}}\right] + \exp\left(\frac{Vx}{D}\right)\mathrm{erfc}\left[\frac{x + Vt}{2\sqrt{Dt}}\right]\right\}. \tag{87}$$

In a traditional Finite Difference (FD) model, the maximum advisable cell Peclet number $N_{Pe} = V\ \Delta x/D$ for an accurate solution is 2. For $N_{Pe} > 2$ spatial truncation errors severely contaminate the solution. In this example, $D = 0.01$ and $V = 0.1$. In the LTMQ scheme, $N = 21$ and the maximum Peclet number was $N_{Pe} = 250$. The Stehfest LTMQ solution with a $N_S = 20$ was shown unable to provide a good solution for $N_{Pe} > 40$. The DeHoog LTMQ solution is much more powerful, and yielded a very accurate solution as shown by the comparison to the analytical solution in Figure 7. To achieve this, a $N_H = 21$ was used. The power of the LTMQ scheme is evident in this very difficult problem, as it can accurately produce a very steep solution surface with only 11 nodes. If the $N_{Pe} \leq 2$ limit was observed in a traditional FD model, then a minimum of 400 nodes would be needed.

5. SUMMARY, DISCUSSION, AND FUTURE DIRECTIONS

A new numerical method, the Laplace Transform MultiQuadrics (LTMQ) method, has been developed for the solution of (a) the diffusion-type PDE for transient, near-incompressible fluid flow, and (b) the advection-diffusion PDE of solute transport through porous media. LTMQ combines a MultiQuadrics (MQ) approximation scheme for the solution of the PDE with a Laplace transform formulation for the elimination of the need for time discretization. The use of MQ in the spatial approximations allows the accurate description of problems in complex porous media with a very limited number of gridded or scattered nodes. The Laplace transform formulation eliminates the time dependency of the problem, and consequently the need for time discretization. An unlimited time step size is thus possible without any loss of accuracy. LTMQ is currently applicable to linear or linearized problems, and is recommended when information at a limited number of observation times is sought. Problems such as the flow of groundwater or the long-term transport of solute contaminants through porous media in a steady-state water velocity field (at concentrations not significantly affecting the groundwater density) seem especially well suited to the LTMQ method. We are currently studying the use of LTMQ in the heterogeneous systems of groundwater flow and transport using a MQ interpolation scheme (to describe the porous media propertiy and/or the velocity spatial distributions) on top of the LTMQ method. Extension of the LTMQ method to the solution of non-linear problems is still the subject of research.

LTMQ proceeds in five steps: (1) a Laplace transform is performed on the PDE, (2) the transformed PDE is approximated using MQ, (3) the resulting system of simultaneous equations of the expansion coefficients is solved and the transformed vector of unknowns is determined in the Laplace space, (4) the transformed dependent variable at any point in the domain of interest is obtained by substitution in the MQ equation, and (5) the solution/prediction vector obtained in steps 3 and 4 is inverted numerically using the Stehfest algorithm [13]. The solution in the Laplace space renders the effects of the time step size on stability and accuracy irrelevant because time is no longer a consideration. LTMQ was tested against results from three one- and two-dimensional test problems obtained from a standard Finite Difference (FD) model, as well as from analytical solutions. An excellent agreement between the LTMQ, the FD and analytical solutions was observed.

Due to its formulation, LTMQ requires solution of the simultaneous equations N_S or N_H times and a combination (linear and non-linear respectively) of the resulting solutions. Although N_S theoretically ranges between 8 and 18, $N_S = 8$ is sufficient to provide an extremely accurate solution in flow problems. For the DeHoog LTMQ, the same level of accuracy is obtained for a $N_H = 13$. In transport problems, the DeHoog LTMQ is capable of obtaining very steep solution surfaces at N_{Pe} as high as 250, but requires a higher N_H.

Compared to a standard FD method, LTMQ requires drastically fewer (at least one order of magnitude) gridded or scattered nodes for the same level of accuracy but produces fully populated matrices (as opposed to sparse banded matrices in FD). Execution times may be reduced by orders of magnitude because solutions in the LTMQ scheme are necessary only at the desired observation times, while in standard numerical and MQ schemes solutions are needed at all the intermediate times of the discretized time domain (with the resulting accumulation of roundoff error). The disadvantages of having to obtain N_S or N_H solutions for a single time step are outweighed by greater accuracy and an unlimited time step size.

We have demonstrated that the LTMQ method is both more accurate and efficient than standard FD and FE schemes. The open question, however, remains how would LTMQ perform, or even be applied, to heterogeneous, multi-dimensional, highly-non-linear systems.

At present, there are precedents of using MQ to solve highly non-linear PDEs and integral equations. Dubal [14], and Cook et al. [15] have demonstrated the ability of MQ to solve the highly non-linear initial state problem of the collision of two black-holes by solving a three-dimensional, non-linear eliiptic PDE. They used a Ficard iteration scheme to iterate upon the MQ expansion coefficients, and achieved convergence fairly rapidly. Makroglou [31] solved non-linear Volterra integral equations with MQ, using a Newtonian iteration scheme to find the converged MQ expansion coefficients. She showed that the MQ expansion is very competitive compared to polynomial spline methods.

In this preliminary paper, we have solved parabolic and mixed parabolic-hyperbolic PDEs making the assumption that the solution is separable in time and space. There are limitations to this assumption, especially in cases of moving fronts or highly non-linear source terms. One approach which we propose to overcome the limitation of variable separation is to transform the original PDE in the Euler frame to a moving frame in which the PDEs are simpler (see Kansa [32]).

For example, consider the strong conservative form of the Euler equations of the form:

$$\frac{\partial U}{\partial t} + \nabla \cdot \mathbf{F} = S, \tag{88}$$

where

$$U = \begin{pmatrix} \rho \\ \mathbf{m} \\ E \end{pmatrix}, \qquad \mathbf{F} = \begin{pmatrix} \mathbf{m} \\ \mathbf{m}\mathbf{m}/\rho + p\mathbf{I} \\ (\mathbf{m}/\rho)(E+p) \end{pmatrix},$$

$p = E/(\gamma - 1) + \mathbf{m} \cdot \mathbf{m}/\rho$, $\mathbf{m}$ is the momentum density vector, E is the total energy density, $\mathbf{I}$ is the identity dyadic, γ is the ratio of specific heats at constant pressure to constant

volume, and S is the corresponding array of source terms. In a moving frame, the Euler equations have the form

$$\frac{\partial U}{\partial t} + \nabla \cdot \mathbf{F} - \mathbf{V} \cdot \nabla U = S, \tag{89}$$

where $\mathbf{V}$, the moving frame velocity, is arbitrary. Kansa [32] chose the frame velocity to have the form

$$\mathbf{V} = \begin{cases} (\nabla \cdot \mathbf{F})\nabla U / |\nabla U|^2 & \text{if } |\nabla U| > 0 \\ \mathbf{0} & \text{otherwise.} \end{cases} \tag{90}$$

In a frame moving at such a velocity $\mathbf{V}$, the PDEs have the form of the ODEs

$$\frac{dU}{dt} = S, \qquad \text{and} \qquad \frac{d\mathbf{x}}{dt} = \mathbf{V} \tag{91}$$

in the interior regions Ω/Γ. Depending upon the form of the source terms, we can, in many cases, achieve separation of time and space very simply.

We may also use a local moving velocity frame to treat Navier-Stokes type equations similar to the Euler equation. Written in strict conservation-law form, the Navier-Stokes type PDEs have two components in the flux term: one representing transport by convection and denoted by $\mathbf{F}$, and the other representing transport by dissipative processes and denoted by $\mathbf{J}$. The Navier-Stokes type PDEs have the form

$$\frac{\partial U}{\partial t} + \nabla \cdot (\mathbf{F} + \mathbf{J}) = S, \tag{92}$$

and in the frame moving with local velocity $\mathbf{V}$, these PDEs have the form

$$\frac{\partial U}{\partial t} + \nabla \cdot (\mathbf{F} + \mathbf{J}) - \mathbf{V} \cdot \nabla U = S. \tag{93}$$

If we choose $\mathbf{V}$ as

$$\mathbf{V} = \nabla \cdot (\mathbf{F} + \mathbf{J}) \frac{\nabla U}{|\nabla U|^2}, \tag{94}$$

the Navier-Stokes equations appear as

$$\frac{dU}{dt} = S, \qquad \text{and} \qquad \frac{d\mathbf{x}}{dt} = \mathbf{V}. \tag{95}$$

In most instances, choosing a proper local moving frame can ensure that the coupled governing PDEs are indeed separable in time and space for either strictly hyperbolic Euler-type equations, or for parabolic Navier-Stokes PDEs. Nodes will tend to cluster near very high gradient regions. In two or three dimensions, a spatial approximation scheme which requires any sort of mesh becomes increasingly difficult. This is not the case with a completely mesh-free scheme such as multiquadric radial basis functions.

We can interpolate to a different point distribution without violating conservation of mass, momentum, total energy, or chemical species simply by constraining the total integral

of each conserved variable to be the same on both the old and the new distributions. By permitting each node location $\mathbf{x}_i$ in Ω/Γ to move at its local velocity $\mathbf{V}_i$, the Rankine-Hugoniot jump conditions are immediately satisfied at both shock and contact surfaces. Interactions of shocks with other shocks, contact surfaces, or boundaries can be handled by a Riemann solver to obtain the correct outgoing set of waves. Kansa [11] showed that very accurate, highly cost-effective solutions to the von-Neuman blast wave problem can be obtained by combining MQ with such a moving node scheme.

The problem of using LTMQ in highly non-linear problems is still a matter of research. Certain non-linear phenomena are known to undergo bifurcations, trifurcations, etc., depending upon small perturbations, such as in the Rayleigh instability. The problem with such an instability is to predict when solution branch points occur. However, there are other non-linear problems, such as ignition, in which the final state trajectory is expected to depend solely upon the initial conditions.

Without having done this research, our best recommendation to handle non-linear time-dependent problems would be to use an adaptive, hybrid approach which would employ LTMQ to integrate the PDEs up to a certain time prior to the onset of a criterion at which bi-furcation may be expected to occur; then switch over to a stiff time-marching solver (such as MQ) which would be used to integrate beyond the potential for bifurcation, and then switch back to the LTMQ method until bifurcation is expected again.

Acknowledgements – Work supported by the Director, Office of Environmental Restoration and Waste Management, Office of Technology Development of the U.S. Department of Energy, and performed by the Lawrence Berkeley Laboratory under contract No.DE-AC03-76SF00098, and the Lawrence Livermore National Laboratory under contract No. W-7405-ENG-48 with the collaboration of the managers and technical staff of the Westinghouse Savannah River Company. Drs. R. Zimmerman, C. Oldenburg, and S. Doss are thanked for their valuable review comments.

REFERENCES

1. R.L. Hardy, "Multiquadric Equations of Topography and Other Irregular Surfaces", *J. Geophys. Res.*, **76**, 1905-1915, (1971).
2. R.L. Hardy, "Geodetic Applications of Multiquadric Analysis", *AVN Allg. Vermess Nachr.*, **79**, 389-406, (1972).
3. R.L. Hardy, "Theory and Applications of the Multiquadric-Biharmonic Method. 20 Years of Discovery 1968-1988", *Comput. Math. Appl.*,**19**(8/9), 163-208 (1990).
4. R. Franke, "Scattered Data Interpolation: Tests of Some Methods", *Math. Comp.* , **38**, 181-200 (1982).
5. C.A. Micchelli, "Interpolation of Scattered Data: Distance Matrices and Conditionally Positive Definite Functions", *Const. Approx.*, **2**, 11-22 (1986).
6. W.R. Madych and S.A. Nelson, "Mulivariate Interpolation and Conditionally Positive Definite Functions", *Approx. Theo. Appl.*, **4**, 77-89 (1989).
7. W.R. Madych and S.A. Nelson, "Mulivariate Interpolation and Conditionally Positive Definite Functions, II", *Math. Comput.*, **54**, 211-230 (1990).
8. M.D. Buhmann, "Multivariate Interpolation in Odd-Dimensional Euclidean Space Using Multiquadrics", *Constr. Approx.*, **6**, 21-34 (1990).

9. M.D. Buhmann, "Multivariate Cardinal Interpolation with Radial Basis Functions", *Constr. Approx.*, **6**, 225-255 (1990).
10. E.J. Kansa, "Multiquadrics - A Scattered Data Approximation Scheme With Applications to Computational Fluid Dynamics I: Surface Approximations and Partial Derivative Estimates", *Comput. Math. Applic.*, **19**, 127-154 (1990).
11. E.J. Kansa, "Multiquadrics - A Scattered Data Approximation Scheme With Applications to Computational Fluid Dynamics II: Solutions to Parabolic, Hyperbolic, and Elliptic Partial differential Equations", *Comput. Math. Applic.*, **19**, 146-161 (1990).
12. B.J.C. Baxter, "The Asymptotic Cardinal Function of the Multiquadric $\phi(r) = (r^2 + c^2)^{1/2}$ as $c \to \infty$", *Comput. Math. Applic.*, **24**(12), 1-6 (1992).
13. W.R. Madych, "Miscellaneous Error Bounds for Multiquadic and Related Interpolators", *Comput. Math. Applic.*, **24**(12), 121-138 (1992).
14. M.R. Dubal, "Construction of Three-Dimensional Black-Hole Initial Data Via Multiquadrics", *Phys. Rev. D.*, **45**(5), 1178-1187 (1992).
15. G. B. Cook, M.W. Choptuik, M.R. Dubal, S. Klasky, R.A. Matzner, and S.R. Oliveira, *Three-Dimensional Initial Data for Collision of Two Black-Holes: Approximations to Numerical Relativity*, Cambridge University Press, (to appear).
16. M.R. Dubal, "Domain Decomposition and Local Refinement for Multiquadrics Approximations: Second-Order Equations in One Dimension", to appear in *J. Sci. Comput. & Modeling.*
17. T.A. Foley, "The Map and Blend Scattered Data Interpolant on a Sphere", *Comput. Math. Applic.*, **24**(12), 61-80 (1992).
18. E.A. Galerpin and Q. Zheng, "Solution and Control of PDE via Global Optimization", *Comput. Math. Applic.*, **25**, 103-118 (1993).
19. H. Stehfest, "Algorithm 368, Numerical Inversion of Laplace Transforms", *Comm. ACM*, **13**(1), 47-49, 1970.
20. F.R. DeHoog, J.H. Knight, and A.N. Stokes, "An Improved Method for Numerical Inversion of Laplace Transforms", *SIAM J. Sci. Stat. Comput.*, **3**(3), 357-366 (1982).
21. K.S., Crump, "Numerical Inversion of Laplace Transforms Using a Fourier Series Approximation", *J. Assoc. Comput. Mach.*, **23**(1), 89-96 (1976).
22. E.A. Sudicky, "The Laplace Tranform Galerkin Technique: A Time-Continuous Finite Element Theory and Application to Mass Transport in Groundwater", *Water Resour. Res.*, **25**(8), 1833-1846 (1989).
23. H. Stehfest, "Remark on Algorithm 368, Numerical Inversion of Laplace Transforms", *Comm. ACM*, **13**(1), 56, 1970.
24. G.J. Moridis and D.L. Reddell, "The Laplace Transform Finite Difference (LTFD) Method for Simulation of Flow Through Porous Media", *Water Resour. Res.*, **27**(8), 1873-1884 (1991).
25. G.J. Moridis and D.L. Reddell, "The Laplace Transform Finite Element (LTFE) Numerical Method for the Solution of the Groundwater Equation", presented at the 1991 AGU Spring Meeting, Baltimore, May 28-June 1, 1991, *EOS Trans. of the AGU*, **72**(17), 223, (1991).
26. G.J. Moridis and D.L. Reddell, "The Laplace Transform Boundary Element (LTBE) Numerical Method for the Solution of Diffusion-Type Equations", *Boundary Elements*

XIII, 83-97, Computational Mechanics Publications and Elsevier Applied Sciences, London (1991).

27. G.J. Moridis,"Evaluation of Numerical Inversion Algorithms in Laplace-Transform Based Numerical Methods for the Solution of the Equations of Flow and Transport Through Porous Media", presented at the 1992 AGU Spring Meeting, Montreal, May 12-16, 1992, *EOS Trans. of the AGU*, **73**(14), 123, (1992).
28. C.V. Theis, "The Relationship Between Lowering the Piezometric Surface and the Rate and the Duration of Discharge of a Well Using Groundwater Storage", *Trans. AGU*, 16th Ann. Meeting, Pt. 2, 519-524 (1935).
29. I.S. Papadopulos, "Nonsteady Flow to a Well in an Infinite Anisotropic Aquifer", presented at the 6th Symp. Intern. Assoc. Sci. Hydrology, Dubrovinik (1965).
30. A. Ogata and R.B. Banks,"A Solution of the Differential Equation of Longitudinal Dispersion in Porous Media", *US Geological Survey*, Prof. Parer No.411-A (1961).
31. A. Makroglou, "Radial Basis Functions in the Numerical Solution of Nonlinear Volterra Integral Equations", to appear in *Int. J. Sci. Comput. & Modeling* (1994).
32. E. Kansa, "Shock Computations with Adaptive Mesh Refinement and Moving Grids", *Comput. Math. Applic.*, **15**(6/8), 623-634, (1988).

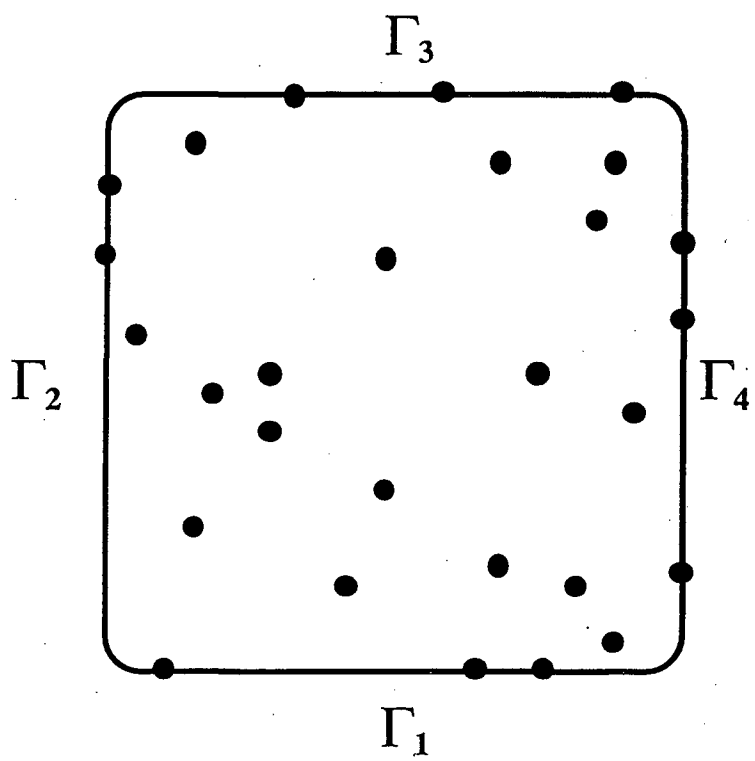


Fig. 1. The domain and boundaries of the linear elliptic PDE problem in two dimensions.

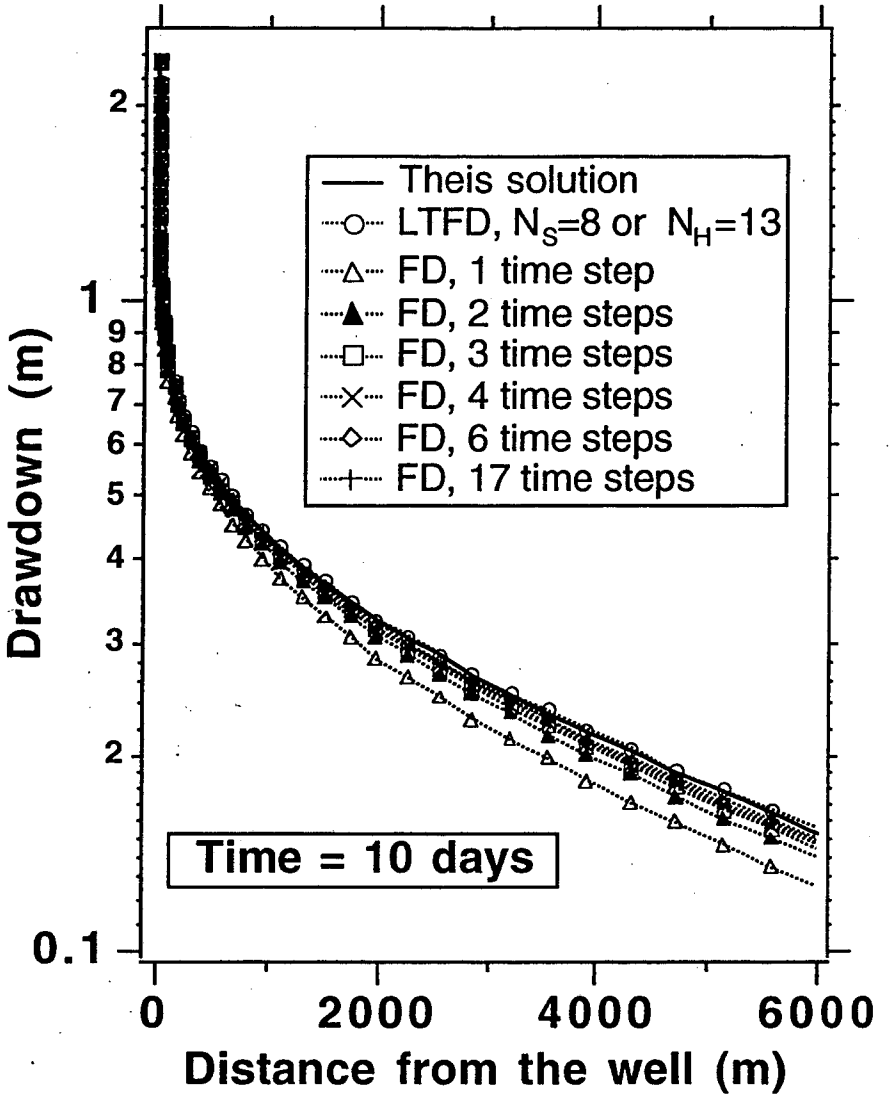


Fig. 2. Comparison of the LTMQ solutions to the Theis and FD solutions in Test Case 1.

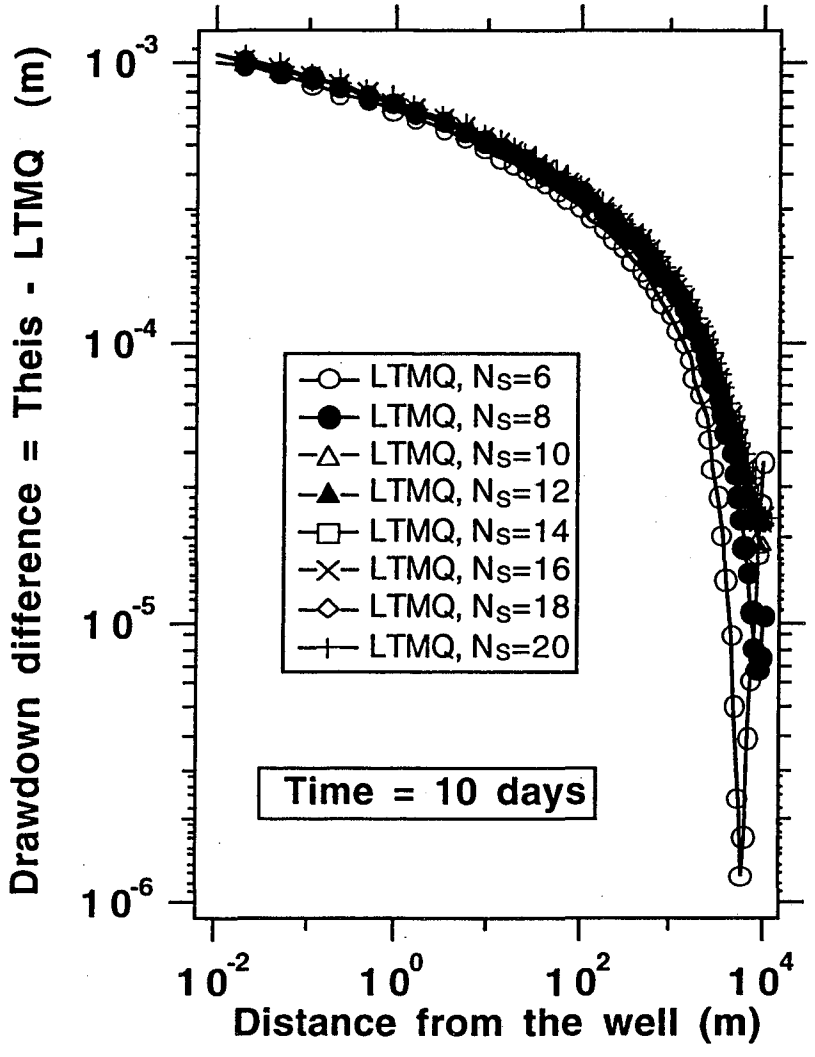


Fig. 3. Effect of N_S on the accuracy of the Stehfest LTMQ method in Test Case 1.

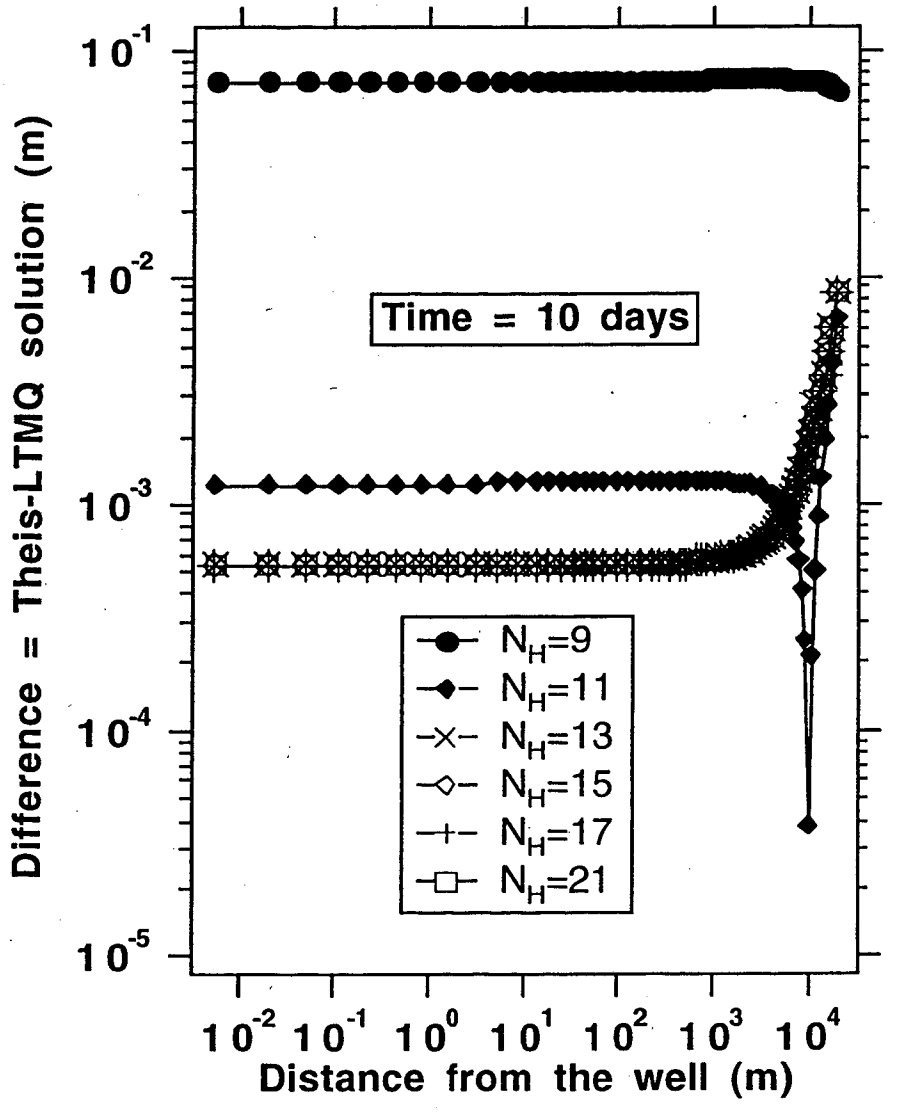


Fig. 4. Effect of N_H on the accuracy of the DeHoog LTMQ method in Test Case 1.

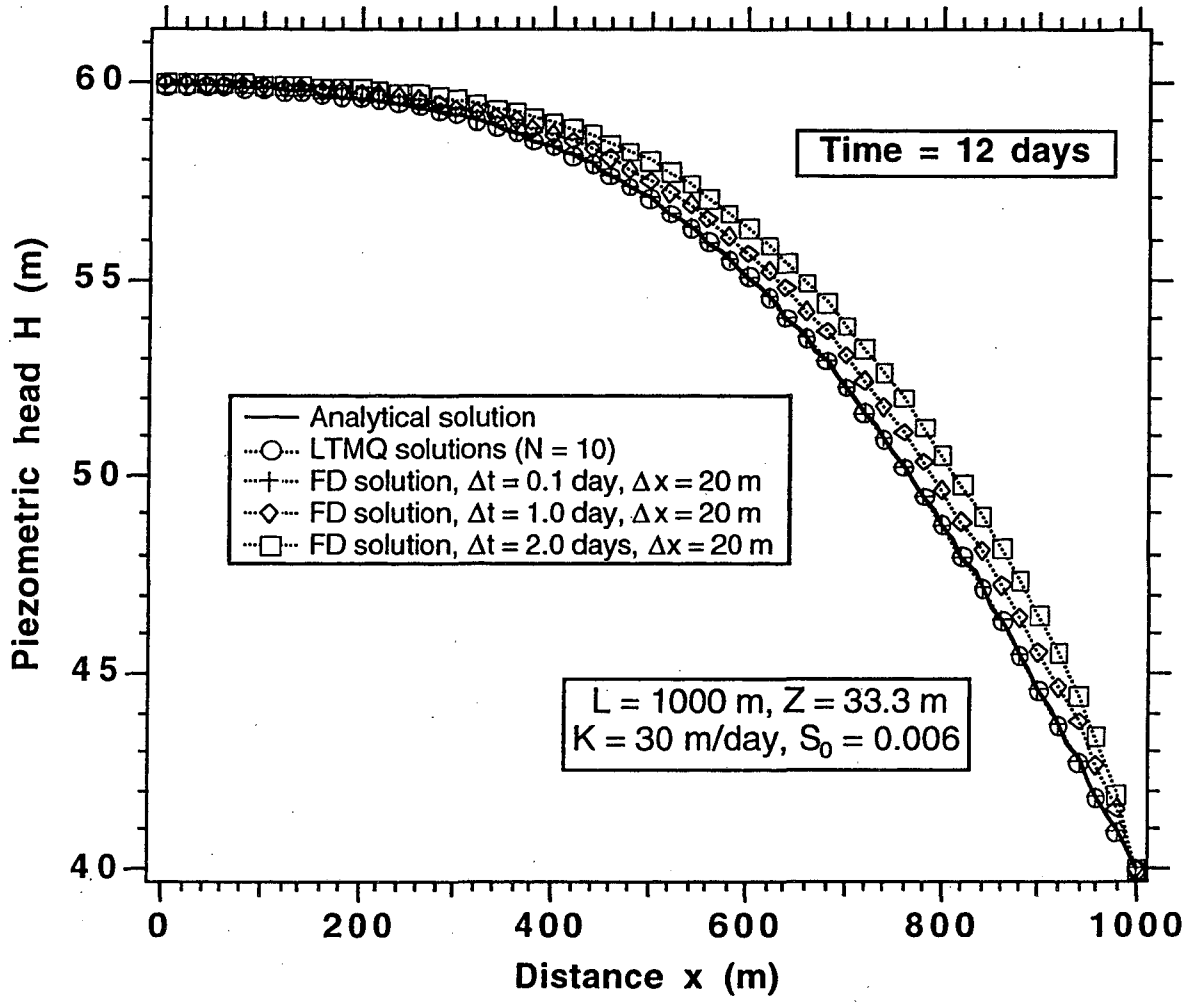


Fig. 5. Comparison of the LTMQ solutions to the analytical and the FD solutions in Test Case 2.

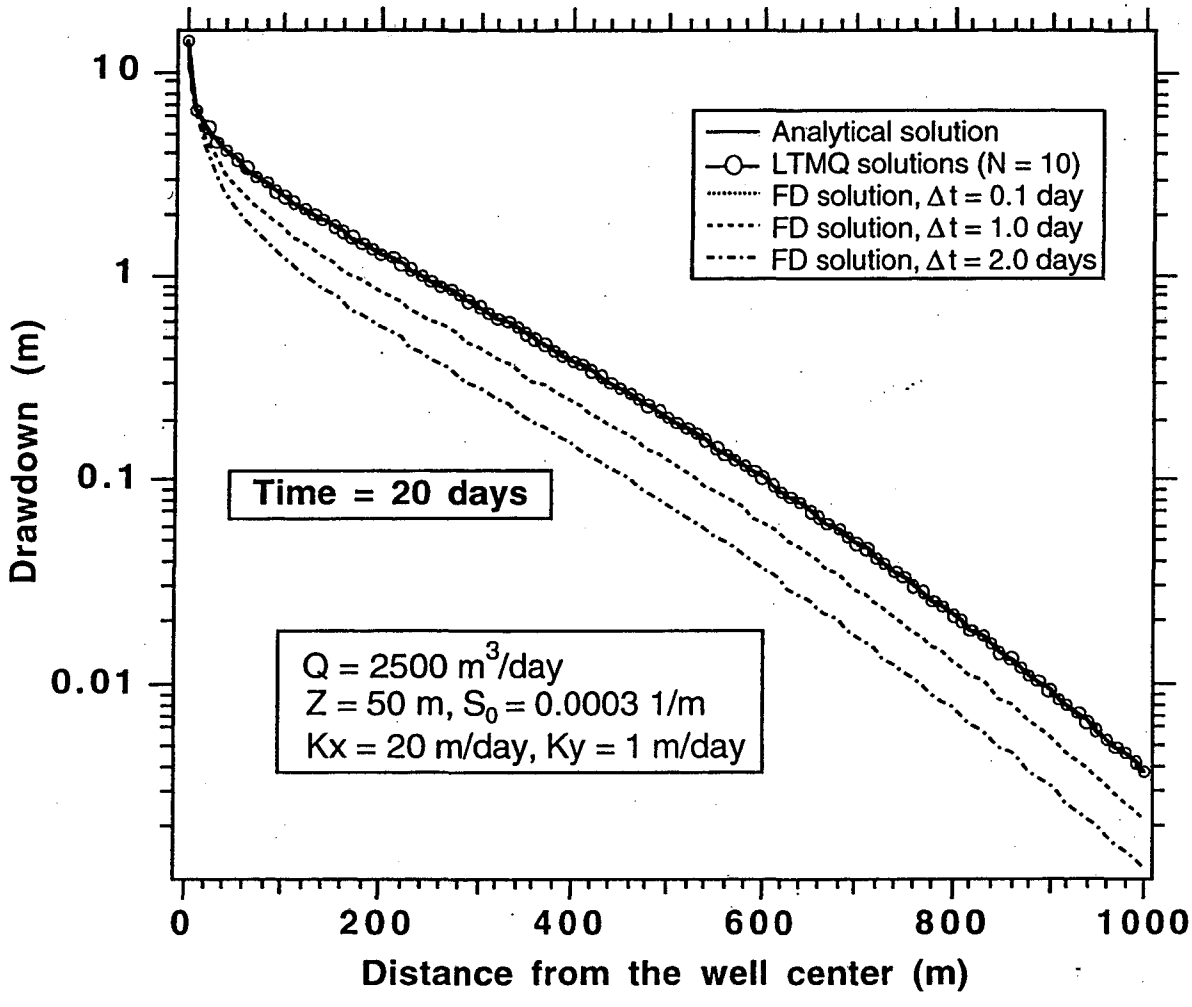


Fig. 6. Comparison of the LTMQ solutions to the analytical and the FD solutions in Test Case 3.

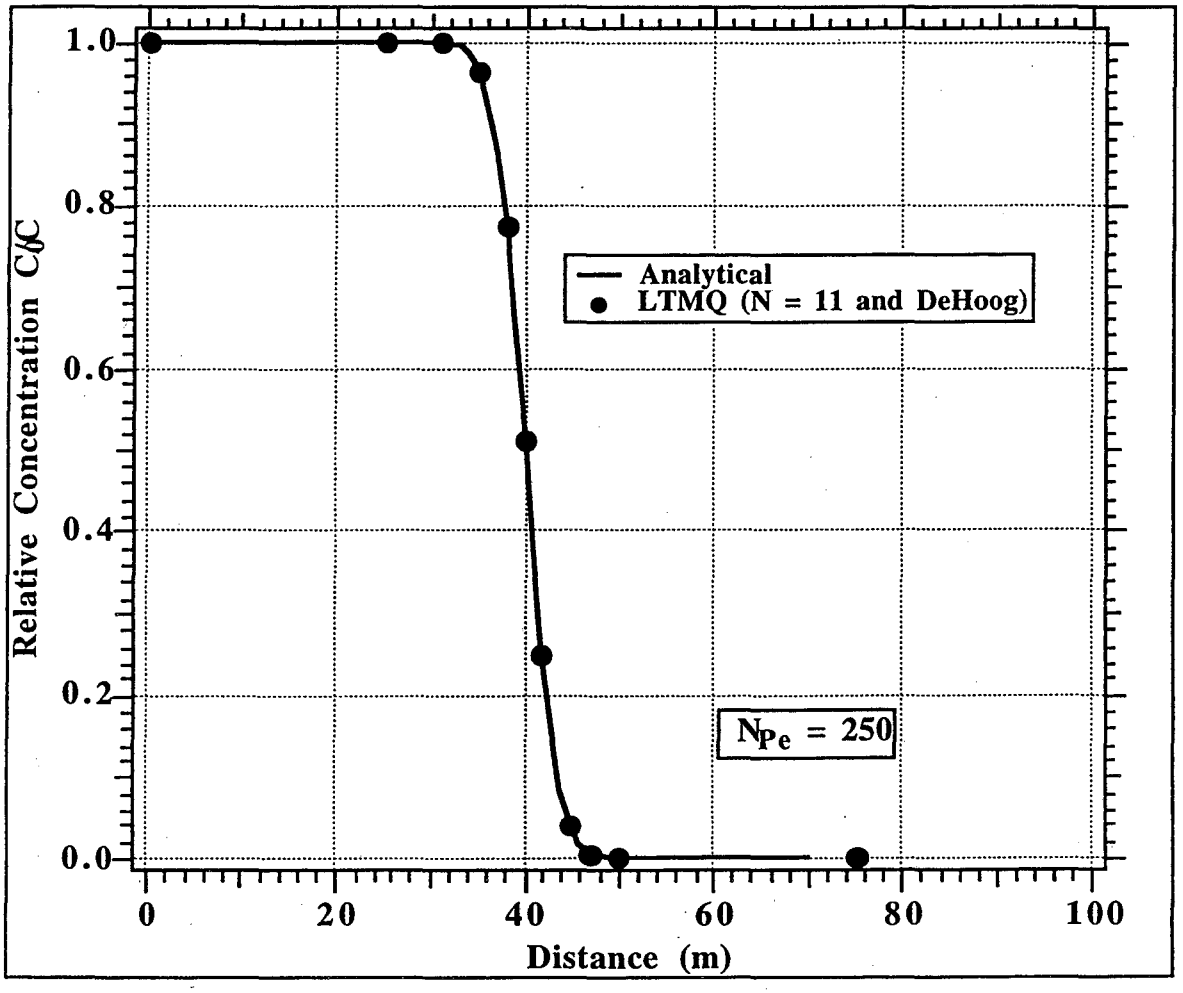


Fig. 7. Comparison of the DeHoog LTMQ and the analytical solutions in the solute transport problem of Test Case 4.

LAWRENCE BERKELEY LABORATORY
UNIVERSITY OF CALIFORNIA
TECHNICAL INFORMATION DEPARTMENT
BERKELEY, CALIFORNIA 94720

AAS316
LBL Libraries