Computers Math. Applic. Vol. 19, No. 8/9, pp. 147–161, 1990
Printed in Great Britain

0097-4943/90 $3.00 + 0.00
Pergamon Press plc

MULTIQUADRICS—A SCATTERED DATA APPROXIMATION SCHEME WITH APPLICATIONS TO COMPUTATIONAL FLUID-DYNAMICS—II

SOLUTIONS TO PARABOLIC, HYPERBOLIC AND ELLIPTIC PARTIAL DIFFERENTIAL EQUATIONS

E. J. KANSA
Lawrence Livermore National Laboratory, L-200, P.O. Box 808, Livermore, CA 94550, U.S.A.

Abstract—This paper is the second in a series of investigations into the benefits of multiquadrics (MQ). MQ is a true scattered data, multidimensional spatial approximation scheme. In the previous paper, we saw that MQ was an extremely accurate approximation scheme for interpolation and partial derivative estimates for a variety of two-dimensional functions over both gridded and scattered data. The theory of Madych and Nelson shows for the space of all conditionally positive definite functions to which MQ belongs, a semi-norm exists which is minimized by such functions.

In this paper, MQ is used as the spatial approximation scheme for parabolic, hyperbolic and the elliptic Poisson's equation. We show that MQ is not only exceptionally accurate, but is more efficient than finite difference schemes which require many more operations to achieve the same degree of accuracy.

1. INTRODUCTION

In the previous paper [1], we have shown multiquadrics (MQ) is an excellent approximation to two-dimensional surfaces and their partial derivatives. We have referred to the work of Franke [2] who tested 29 different techniques of spatial approximation using a variety of exact test functions. He concluded that of all of the methods tested, MQ developed by Hardy [3, 4] outperformed all methods tested. Likewise, Stead [5] had shown that the partial derivative estimates obtained by MQ on those surfaces with significant curvature were excellent. She recommended a quadratic approximation for relatively flat surfaces and MQ for intermediate to high curvature.

The theoretical justification for the success of MQ has been given by Micchelli [6] and Madych and Nelson [7]. Micchelli [6] proved that the MQ coefficient matrix is always invertible for distinct points. Madych and Nelson [7] showed that for the space of conditionally positive definite functions to which MQ belongs, a semi-norm exists and is minimized by such functions.

In the previous paper [1], we showed that MQ can yield approximations of exceptionally high accuracy by permitting the shape parameter, r^2, to vary with the basis functions. By numerical experimentation, we found that the best results occurred with the following expansion

$$r^2(j) = r^2 \min(r_{\max}^2/r_{\min}^2)^{(j-1)/(N-1)}, \quad j = 1, 2, \ldots, N, \tag{1}$$

and $r\ \max^2$ and $r\ \min^2$ are input parameters. Using equation (1), the MQ expansion is given as

$$f(\mathbf{x}) = \sum_{j=1}^{N} a_j g(\mathbf{x} - \mathbf{x}_j), \tag{2}$$

where

$$g(\mathbf{x} - \mathbf{x}_j) = [d^2(\mathbf{x} - \mathbf{x}_j) + r_j^2]^{1/2}, \tag{3}$$

$$d^2(\mathbf{x} - \mathbf{x}_j) = (x - x_j)^2 + (y - y_j)^2 + \cdots. \tag{4}$$

The coefficients, $\{a_j\}$ are found by solving a set of linear equations in terms of the basis functions. For example, for traditional MQ we solve

$$\sum_{j=1}^{N} a_j g(\mathbf{x}_i - \mathbf{x}_j) = F(\mathbf{x}_i), \quad i = 1, 2, \ldots, N, \tag{5}$$

and $f(\mathbf{x}_i) = F(\mathbf{x}_i)$ which are given. Exceptional accuracy using MQ is achieved by transforming the original set to appear more scattered, and by domain decomposition. These schemes greatly improve the condition number of the MQ coefficient matrix.

Madych and Nelson [6] consider a general case of interpolants which includes a polynomial of degrees less than some fixed integer m given by

$$f(\mathbf{x}) = \sum_{n=1}^{N} a_j g(\mathbf{x} - \mathbf{x}_j) + \sum_{|\alpha|<m} k_\alpha \mathbf{x}^\alpha, \tag{6}$$

where a_j and k_α must satisfy

$$\sum_{j=1}^{N} a_j g(\mathbf{x}_i - \mathbf{x}_j) + \sum_{|\alpha|<m} k_\alpha \mathbf{x}_i^\alpha = f_i(\mathbf{x}_i) = F_i, \quad i = \{1, 2, \ldots, N\}, \tag{7a}$$

$$\sum_{j=1}^{N} a_j \mathbf{x}_j^\alpha = 0, \quad |\alpha| < m. \tag{7b}$$

In this paper, we use MQ as spatial approximate rather than the standard local polynomial approximation. We show that MQ greatly reduces the spatial truncation error eliminating the restrictions of uniform gridding, numerical diffusion and signal dispersion. We show MQ is an excellent approximation for fixed nodal Eulerian calculations, variable spaced moving nodal problems, and two-dimensional scattered data problems. We will show MQ works excellently on the parabolic, hyperbolic and the elliptic Poisson's equation. Not only are the MQ results more accurate than the finite difference results, but the MQ is computationally more efficient since far fewer nodal values are required to obtain highly accurate results.

2. APPLICATION OF MQ TO PARTIAL DIFFERENTIAL EQUATIONS

2.1. The linear advection-diffusion problem

The literature, see Adams [8], contains may schemes to solve the linear advection-diffusion equation with various degrees of success. We present an implicit time marching scheme on the Eulerian frame using MQ for the spatial approximations. We show that adding numerical diffusion for an MQ representation is not necessary. Further, our MQ scheme is quite accurate to cell Peclet or Reynold's numbers for 0.5–10.0. We compare our scheme with a finite difference (FD) scheme and find MQ is superior.

The following moving front problem which is taken from Adams [8] has an analytic solution. We solve the following advection-diffusion equation

$$\partial f/\partial t + u\,\partial f/\partial x - D\,\partial^2 f/\partial x^2 = 0, \tag{8}$$

with initial and boundary conditions

$$f(x, 0) = 0, \quad \text{for } 0 \leqslant x < \infty, \tag{9a}$$

$$f(0, t) = 1, \quad \text{for } t > 0, \tag{9b}$$

$$f(L, t) = 0, \quad \text{for } t > 0, \quad L \to \infty, \tag{9c}$$

where x and t are the space, and time coordinates, u is a constant input velocity and D is the diffusion coefficient.

The exact solution is given for two cases: $D > 0$ and $D = 0$. For $D > 0$, we have

$$f(x, t) = 0.5[\text{erfc}(w1) + \exp(ux/D)\text{erfc}(w2)], \tag{10a}$$

where

$$w1 = (x - ut)/2\sqrt{Dt}, \tag{10b}$$

$$w2 = (x + ut)/2\sqrt{Dt}, \tag{10c}$$

and erfc is the complementary error function.

For $D = 0$, we have

$$f = 1, \quad \text{for } x < ut. \tag{11a}$$

$$= 0, \quad \text{otherwise.} \tag{11b}$$

The advective-diffusion equation can be written compactly for either the FD or MQ approximation as

$$\partial f/\partial t = Wf, \tag{12}$$

where

$$W = I(D\,\partial^2/\partial x^2 - u\,\partial/\partial x), \tag{13}$$

and where I is the identity operator.

Using the standard implicit approximation to equation (12) gives

$$(f_j^{n+1} - f_j^n) = \theta\,\Delta t W f_j^{n+1} + \Delta t(1-\theta) W f_j^n, \tag{14}$$

where

$$f_j^n = f(x_j; t^n), \tag{15a}$$

$$f_j^{n+1} = f(x_j; t^{n+1}), \tag{15b}$$

and

$$0 \leqslant \theta \leqslant 1.$$

We collect like terms to obtain

$$H_+ f_j^{n+1} = H_- f_j^n, \tag{16}$$

where

$$H_+ = I - \Delta t \theta W, \tag{16a}$$

and

$$H_- = I + \Delta t(1-\theta) W. \tag{16b}$$

The advanced time solution is given by

$$f_j^{n+1} = P f_j^n = H_+^{-1} H_- f_j^n, \tag{17}$$

subject to the boundary conditions,

$$f(0, t) = 1,$$

$$f(L, t) = 0.$$

Consider first the finite difference approximation to the solution of the advection-diffusion equation. The spatial coordinate, x, is discretized uniformly

$$x_j = \Delta x(j-1), \quad \text{for } 1 \leqslant j \leqslant N. \tag{18}$$

We use a central difference approximation for the diffusion term and upwind differencing for the convective term to give

$$Wf_j = (D/\Delta x^2) f_{j+1} - (2D/\Delta x^2 + u/\Delta x) f_j + (u/\Delta x) f_{j-1} \quad \text{for } 1 < j < N, \tag{19}$$

and for the boundaries,

$$Wf_1 = Wf_N = 0. \tag{20}$$

The matrix representation of the operators H_+ and H_- are:

$$H_+ f_j^{n+1} = (I - \Delta t \theta W) f_j^{n+1}, \tag{21a}$$

$$H_- f_j^n = (I + \Delta t(1-\theta) W) f_j^n, \tag{21b}$$

where $H\_f_j^n$ is a column vector given in terms of the known values of f at time t^n. f^{n+1} is found by solving a triadiagonal system of linear equations.

Because the inverse of a tridiagonal matrix is generally full, we find that there are fewer operations involved by solving for the unknowns using the tridiagonal algorithm which typically requires $8N$ operations. Consider next the MQ representation and solution of the advection-diffusion equation.

Following Madych and Nelson [7], we expand any continuous function, f, in terms of MQ basis functions and appended polynomial. One can show that the following expansion with a linear polynomial and MQ expansion is given by

$$f(x) = a_1 + a_2 x + \sum_{j=3}^{N} a_j \tilde{g}(x - x_j), \tag{22}$$

where

$$\tilde{g}(x - x_j) = g(x - x_j) - [(x_2 - x_j)g(x - x_1) + (x_j - x_1)g(x - x_2)]/(x_2 - x_1), \tag{23}$$

and

$$g(x - x_j) = [(x - x_j)^2 + r_j^2]^{1/2}. \tag{24}$$

The set of linear equations transforming the expansion coefficient $\{a_i\}$ to the set of discretized values of f, $\{f_i\}$, $1 \leqslant i \leqslant N$ is given by

$$f_i = \sum_{j=1}^{N} G_{ij} a_j, \tag{25}$$

where the ith row of the coefficient matrix is given by

$$G_{i,1} = 1, \tag{26}$$

$$G_{i,2} = x_i, \tag{27}$$

$$G_{i,j} = \tilde{g}(x_i - x_j), \quad \text{for } 3 \leqslant j \leqslant N. \tag{28}$$

Since G is non-singular for distinct points, see Micchelli [6], then

$$\mathbf{a} = \mathbf{G}^{-1}\mathbf{f}. \tag{29}$$

To construct the solution of the advection-diffusion equation, we obtain the first and second partial derivatives of f_i with respect to x:

$$(\partial f/\partial x)_i = a_2 + \sum_{j=3}^{N} \{\partial \tilde{g}_{ij}/\partial x\} a_j, \tag{30}$$

$$(\partial^2 f/\partial x^2)_i = \sum_{n=3}^{N} \{\partial^2 \tilde{g}_{ij}/\partial x^2\} a_j, \tag{31}$$

where

$$\partial \tilde{g}_{ij}/\partial x = \partial g_{ij}\partial x - [(x_2 - x_j)\,\partial g_{i1}/\partial x + (x_j - x_1)\,\partial g_{i2}\partial x]/(x_2 - x_1), \tag{32}$$

$$\partial^2 g_{ij}/\partial x^2 = \partial^2 g_{ij}/\partial^2 x - [(x_2 - x_j)\,\partial^2 g_{i1}/\partial x^2 + (x_j - x_1)\,\partial^2 g_{i2}/\partial^2 x]/(x_2 - x_1), \tag{33}$$

and

$$\partial g_{ij}/\partial x = (x_i - x_j) \cdot [(x_i - x_j)^2 + r_j^2]^{-1/2}, \tag{34}$$

$$\partial^2 g_{ij}/\partial x^2 = [1 - (x_i - x_j)^2/[(x_i - x_j)^2 + r_j^2]] \cdot [(x_i - x_j)^2 + r_j^2]^{-1/2}. \tag{35}$$

Using the operator notation for the ith value of f, f_i,

$$Wf_i = (D\,\partial^2/\partial x^2 - u\,\partial/\partial x)f_i = \sum_{j=1}^{N} w_{ij} a_j, \tag{36}$$

where

$$w_{i1} = 0, \tag{36a}$$

$$w_{i2} = -u, \tag{36b}$$

$$w_{ij} = \left(D \frac{\partial^2 \tilde{g}_{ij}}{\partial x^2} - u \frac{\partial \tilde{g}_{ij}}{\partial x} \right), \quad \text{for } 3 \leqslant j \leqslant N. \tag{36c}$$

In the numerical examples to be presented, we solve the advection-diffusion equations in the Eulerian frame with the spatial nodes fixed for all time. As with finite element approximations, we assume that the spatial basis functions are fixed, and that the expansion coefficients vary in time, i.e.

$$a_j = a_j(t). \tag{37}$$

With this approximation, the advection-diffusion equation is written as

$$\partial f / \partial t = Wf, \quad \text{or equivalently}$$

$$G \, \mathrm{d}a/\mathrm{d}t = wa. \tag{38}$$

We choose to solve equation (38) implicitly using the following approximation

$$\mathrm{d}a/\mathrm{d}t = (a^{n+1} - a^n)/\Delta t, \tag{39}$$

and

$$wa = \theta w a^{n+1} + (1 - \theta) w a^n,$$

to obtain

$$(G - \Delta t \theta w) a^{n+1} = [G + \Delta t (1 - \theta) w] a^n. \tag{40}$$

Writing

$$H_+ = G - \Delta t \theta w, \tag{40a}$$

$$H_- = G + \Delta t (1 - \theta) w, \tag{40b}$$

we have

$$a^{n+1} = H_+^{-1} H_- a^n, \tag{41}$$

provided

$$H_+ \text{ is non-singular.}$$

Recalling

$$a = G^{-1} f, \tag{42}$$

we obtain

$$f^{n+1} = P f^n, \tag{43}$$

where

$$P = G H_+^{-1} H_- G^{-1}, \tag{44}$$

for all interior nodes $1 < j < N$. However, the advection-diffusion equation has the boundary conditions

$$f(0, t) = 1, \tag{45}$$

$$f(L, t) = 0, \quad \text{for } t > 0. \tag{46}$$

Table 1. Variation of diffusion coefficient, cell Peclet number and effective diffusion coefficient

D	Pe	D_{eff}
0.1	0.5	0.125
0.01	5.0	0.035
0.001	50.0	0.026

These boundary conditions are readily implemented by expanding the matrix P to have such boundary conditions by requiring

$$P_{1,1} = 1, \tag{47}$$

$$P_{1,j} = 0, \quad \text{for } 2 \leqslant j \leqslant N, \tag{48}$$

and

$$P_{N,j} = 0, \quad \text{for } 1 \leqslant j < N, \tag{49}$$

$$P_{N,N} = 1. \tag{50}$$

Note that we may readily impose boundary conditions on the partial derivatives with respect to x at either $x = 0$ or $x = L$ by using the partial derivatives of the basis functions.

As in previous examples in Ref. [1], we used domain decomposition to reduce the full matrix into a set of seven overlapping segments, each containing nine nodal values. Thus each segment is a full block matrix with 81 elements. There are 567 elements over all segments. The solution from the right and left are blended using the method of weighted averages.

We performed a series of calculations using both MQ and FD schemes in which the following parameters were held constant: $u = 1.0$, $\Delta x = 0.05$, $N = 51$, $\Delta t = 0.002$, $t_{\text{final}} = 1.000$. We varied only the diffusion coefficients, D: 0.1, 0.01, 0 and 0.001. We can show that for an upwind FD scheme, the effective diffusion coefficient is given by

$$D_{\text{eff}} = D(1 + \tfrac{1}{2}\text{Pe}_{\text{cell}}), \tag{51}$$

where

$$\text{Pe}_{\text{cell}} = u\,\Delta x/D.$$

Table 1 shows variation of diffusion coefficient, cell Peclet number, and effective FD diffusion coefficient.

Since the cell Peclet number varies linearly with grid size, Δx should be 0.002 rather than 0.05 for the case $D = 0.001$. Figures 1–3 show the FD solutions at time, $t = 1.0$, for $D = 0.1$, 0.01 and 0.001 given by the long dashed lines. The exact solution is given by solid lines. The plots show a clear need for grid refinement as the diffusion coefficient is reduced. For both the MQ and FD schemes, we used an implicit time scheme with $\theta = 0.5$.

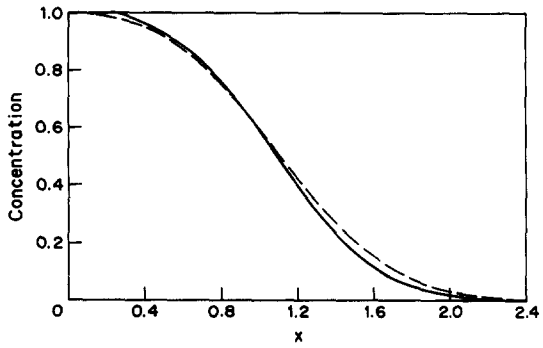

Fig. 1. FD solution of the linear advection-diffusion equation using upwinding run to time = 1.0, $\Delta x = 0.05$, $u = 1.0$, $D = 0.1$ and Pe = 0.5.

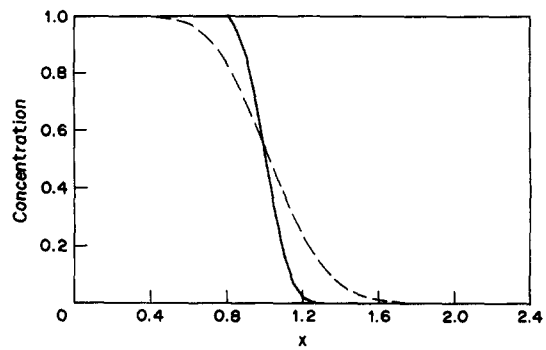

Fig. 2. FD solution of the linear advection-diffusion equation using upwinding run to time = 1.0, $\Delta x = 0.05$, $u = 1.0$, $D = 0.01$ and Pe = 5.0.

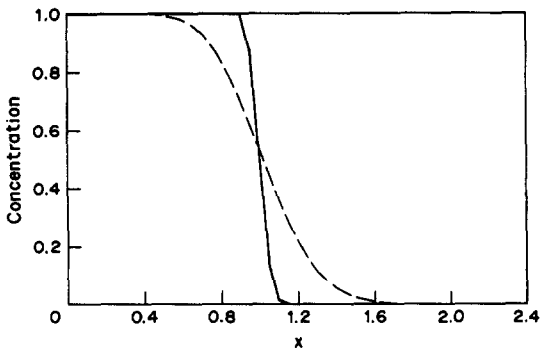

Fig. 3. FD solution of the linear advection-diffusion equation using upwinding run to time = 1.0, $\Delta x = 0.05$, $u = 1.0$, $D = 0.001$ and Pe = 50.0.

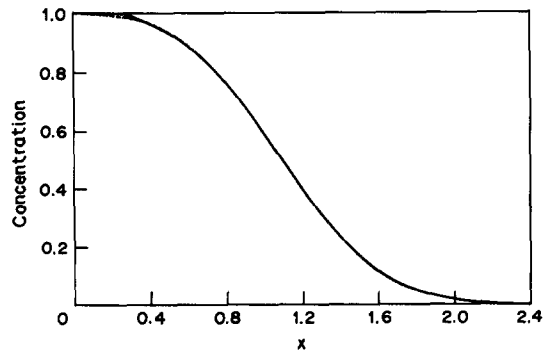

Fig. 4. MQ solution of the linear advection-diffusion equation run to time = 1.0 $\Delta x = 0.05$, $u = 1.0$, $D = 0.1$ and Pe = 0.5.

Figures 4–6 show the exact solutions which are indicated by the solid line and the MQ solutions which are indicated by the short dashed line. The MQ solutions were run for $D = 0.1$ 0.01 and 0.001. The results for MQ at $D = 0.1$ and 0.01 are excellent whereas the results for $D = 0.001$ show good agreement at the sharp front, but not at the end points where rounding has occurred.

Figure 7 shows the results for the finite difference solution for $D = 0.001$ run with a pure central difference algorithm. Note the ringing behind the front. Figure 8 shows the results run with 55% central differencing and 45% upwinding for the $D = 0.01$ and $\Delta x = 0.05$ problem. Note the dramatic reduction of the frontal spreading. Figure 9 shows the results for the $D = 0.001$ using FDs, 55% central differencing and 45% upwind differencing executed for a grid spacing of $\Delta x = 0.005$ for a total of 510 points.

The FD result for the $D = 0.001$ case using 510 points has comparable accuracy to the MQ result run with 51 points. At each time step, the MQ solution requires 567 operations. The number of operations using Thomas's algorithm for tridiagonal matrices requires about $8N$ operations or 4080 operations. In this case, MQ is about seven times more efficient. Figure 8 shows, however, that FD on a coarse grid is more efficient than MQ provided a stable balance between central and upwind differencing can be found for the FD scheme.

In this set of advection-diffusion problems, we used, in both the MQ and FD spatial schemes, the identical second order accurate Crank–Nicholson time integration scheme. In the FD schemes, a weighted average of upwind and central differencing was used since some amount of upwinding is required to stabilize the advection term. Furthermore, FD results are reasonably accurate for cell Peclet numbers $\leqslant 5.0$. On the other hand, MQ does not require upwinding and is accurate for all cell Peclet numbers $\leqslant 50.0$. Our operation count shows that FD is more efficient for low cell Peclet numbers than MQ for a given mesh, but MQ is more accurate and efficient for larger cell Peclet numbers. Furthermore, we perturbed in input boundary condition at $x = 0.0$ and changed the boundary value from 1.0 to 1.0001 and ran both the FD and MQ schemes for the cell Peclet

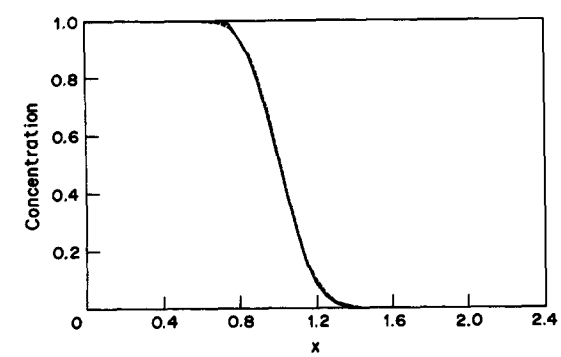

Fig. 5. MQ solution of the linear advection-diffusion equation run to time = 1.0 $\Delta x = 0.05$, $u = 1.0$, $D = 0.01$ and Pe = 5.0.

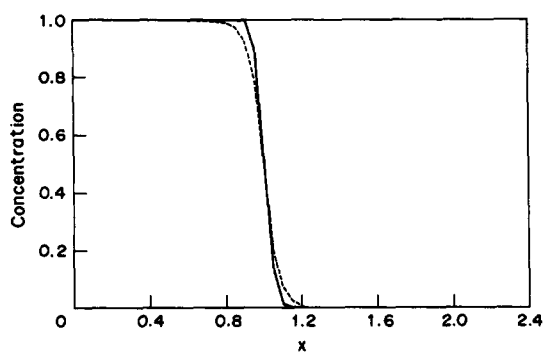

Fig. 6. MQ solution of the linear advection-diffusion equation run to time = 1.0 $\Delta x = 0.05$, $u = 1.0$, $D = 0.001$ and Pe = 50.0.

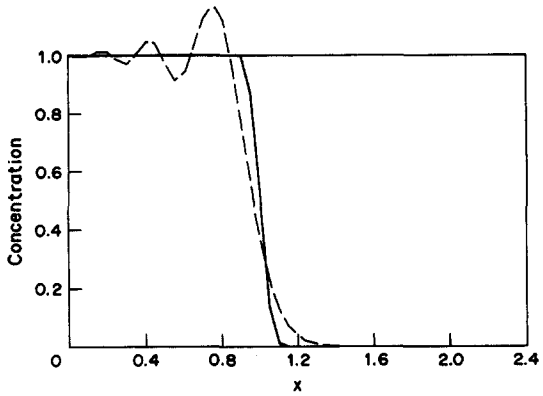

Fig. 7. FD solution of the linear advection-diffusion equation using central differencing run to time = 1.0 $\Delta x = 0.05$, $u = 1.0$, $D = 0.001$ and Pe = 50.0.

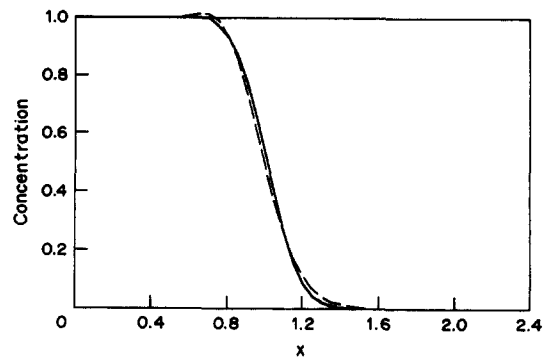

Fig. 8. FD solution of the linear advection-diffusion equation using 55% central and 45% upwind differencing to time = 1.0, $\Delta x = 0.05$, $u = 1.0$, $D = 0.01$ and Pe = 5.0.

number of 5.0. No numerical instabilities were seen for either case and the solutions are graphically indistinguishable from Figs 2 and 5.

A natural question to raise is the behavior of MQ as the diffusion coefficient tends toward zero, giving rise to a front which tends toward a step-function. We have observed that the flat regions before and after the shock are very noisy using MQ. These flat regions require very large r^2 parameter base functions which in turn, give rise to very ill-conditioned MQ coefficient matrices. Consequently, the flat regions are very noisy. We recommend that regions which behave as step-functions be approximated by step-functions, and moving node schemes be used, see Ref. [9].

2.2. Dynamic one-dimensional von Neumann blast wave

The problem to be solved is a one-dimensional spherical symmetry problem of von Neumann [10]. In a sphere of radius one, a gas is suddenly heated and a shock is formed. Immediately ahead of the shock $u = p = 0$ and the gas density is one. The pressure behind the shock is 100 and its density is 4 for a $\gamma = 5/3$ ideal gas. Because of spherical divergence, the gas velocity and pressure behind the shock eventually decay, but the density remains at 4. The exact time dependent solutions for the $\gamma = 5/3$ gas are listed in Ref. [11].

From the primitive dependent variables, one can construct the momentum and total energy densities given by

$$m = \rho u, \tag{52}$$

$$E = p/(\gamma - 1) + 1/2\rho u^2. \tag{53}$$

In spherical coordinates, the conservation equations are

$$\partial\rho/\partial t + \partial m/\partial r + (2/r)m = 0, \tag{54}$$

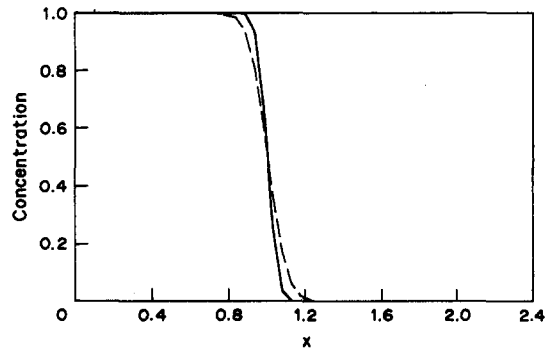

Fig. 9. FD solution of the linear advection-diffusion equation using 55% central and 45% upwinding differencing run to time = 1.0, $u = 1.0$, $D = 0.001$ and Pe = 5.0. Here the grid was refined to $\Delta x = 0.005$.

$$\partial m/\partial t + \partial p/\partial r + \partial(m^2/\rho)/\partial r + (2/r)m^2/\rho = 0, \tag{55}$$

$$\partial E/\partial t + \partial(u[E+p])/\partial r + (2/r)u[E+p] = 0. \tag{56}$$

At the origin, the momentum profile is asymmetric at $r = 0$, while ρ and E are symmetric. The boundary conditions at $r = 0$ are $m = (\partial\rho/\partial r) = (\partial E/\partial r) = 0$. To avoid the singularity at the origin, L'Hôpital's rule is invoked for the $(2/r)$ terms.

Equations (54)–(56) may be written compactly as:

$$\partial U/\partial t + \nabla \cdot \mathbf{F} = 0, \tag{57}$$

where $U = [\rho, m, E]$ and $\mathbf{F}$ is the vector of the corresponding fluxes. We have two sources of truncation errors; one from the spatial differencing, and the other from the time integration scheme. In this exercise, we wish to minimize spatial truncation errors which propagate along characteristics. This model problem was chosen because we can compare the exact solution of three coupled non-linear equations with the numerical solutions, and it is typical of a fluid dynamics problem involving coupled, non-linear partial differential equations.

In order to minimize the time truncation errors, we choose a moving node scheme so that equation (57) becomes

$$\partial U/\partial t + \nabla \cdot F - \mathbf{v} \cdot \nabla U = 0. \tag{58}$$

We track the shock as in Ref. [11], and in the interior, we move the nodes. The nodal velocity, $\mathbf{v}$, is chosen to transform equation (58) into a nearly stationary set of conservation equations in the least squares sense, [11]. In addition, a fourth order Runge–Kutta (RK) scheme is used to time integrate equation (58) in the following manner:

$$U_k^{n+1/4} = U_k^n + (\Delta t/4)G_k^n, \tag{59a}$$

$$U_k^{n+1/3} = U_k^n + (\Delta t/3)G_k^{n+1/4}, \tag{59b}$$

$$U_k^{n+1/2} = U_k^n + (\Delta t/2)G_k^{n+1/3}, \tag{59c}$$

$$U_k^{n+j} = U_k^n + \Delta t G_k^{n+1/2}, \tag{59d}$$

where k denotes the node number, $n + j$, $j = \{0, 1/4, 1/3, 1,/2\}$ denotes the time step.

$$G_k^{n+k} = \mathbf{v}_k^{n+j} \cdot \nabla U_k^{n+j} - \nabla \cdot F_k^{n+j}, \tag{60}$$

and

$$\Delta t = 0.9 \cdot (\Delta r)_{\min}/(|u| + c)_{\max} \tag{61}$$

is the explicit CFL time step, and c is the sound speed.

The object of this exerise is only vary the method by which derivatives are approximated. Using MQ to construct partial derivatives, a total of 35 points were used in the domain. Domain decomposition and blending was used to simplify the problem more. Also, three (FD) calculations of 50, 500 and 5000 points were used having second order spatial accuracy. Table 2 shows the initial gradients using the MQ and FD schemes. Table 3 gives the solution after marching in time to $t/t_0 = 1.1523$.

From Tables 2 and 3, we see that the MQ results are excellent approximations to the exact solution whereas the FD scheme converges rather slowly. To determine whether the extra effort of using MQ for the spatial scheme is warranted in the von Neumann problem, let us estimate the total operations required assuming all simple arithmetic operations are comparable.

First, let us consider the operations count for the 35 point, MQ—moving node—fourth order RK scheme. Using domain decomposition, we have two rank 18 matices to be inverted at the beginning of the RK cycle for a total of 10,710 operations. An estimate of 510 operations are required to form the basis vectors and corresponding derivatives. At each RK subcycle, we require 7560 operations to find the expansion coefficients, and 816 operations to form the gradients and flux divergences. An additional 455 operations are required to form the nodal velocities and 105 operations are required to obtain an estimate of the RK dependent variable. Considering the four

Table 2. Initial derivative at $t = t_0$ for the von Neumann blast wave problem

Method	r	$\partial\rho/\partial r$	$\partial m/\partial r$	$\partial E/\partial r$
Exact		0.053666	0.11713	0.030429
MQ—35 pts		0.053667	0.11713	0.030421
FD—50pts	0.2577	0.040696	0.08231	0.19831
FD—500pts		0.052224	0.11310	0.29147
FD—5000pts		0.053519	0.11823	0.30289
Exact		0.580824	2.49562	12.6258
MQ—35pts		0.580824	2.49562	12.6258
FD—50pts	0.50516	0.496276	2.03000	9.7852
FD—500pts		0.572414	2.44943	12.3448
FD—5000pts		0.579979	2.49098	12.5976
Exact		3.25215	21.8323	153.517
MQ—35pts		3.25215	21.8323	153.517
FD—50pts	0.75277	2.79231	18.0928	125.191
FD—500pts		2.20719	21.4631	150.717
FD—5000pts		3.25304	21.8315	153.511
Exact		48.0000	484.974	3600.000
MQ—35pts		48.0000	484.974	3599.999
FD—50pts	1.00000	41.4423	412.826	3066.345
FD—500pts		47.2719	476.913	3540.387
FD—5000pts		47.9335	484.225	3594.457

RK subcycles, we estimate 42,026 operations/time cycle and 70 operations to update the positions after one time cycle to give

$$N_{\text{cycle}}^{\text{MQ}} = 42{,}096 \text{ operations/time cycle}, \tag{62}$$

or 1202 operations/time cycle/node.

Next, let us consider the operation count to one full time cycle using FD scheme for the spatial derivative approximation. The weight factor for a three point local derivative approximation from a quadratic Lagrange polynomial is 15 operations/node. There are three gradients, three flux divergences per node, also requiring 36 operations/node. 13 operations/node are required for the local grid velocity, and three operations/node/dependent variables are required to update the dependent variables.

The positions are updated at the end of the RK cycle requiring two operations/node. The estimated operation count for the FD scheme is

$$N_{\text{cycle}}^{\text{FD}} = 249 \text{ operations/node/time cycle} \cdot N_{\text{FD}}. \tag{63}$$

The total number of complete time cycles to run to completion is

$$N_{\text{time cycles}} = (t_F - t_0)\,\Delta t = (t_F - t_0) \cdot (|u| + c)_{\max}/(0.9\Delta r_{\min}). \tag{64}$$

In this von Neumann blast wave problem, each grid velocity is positive so that we may take the initial minimum velocity throughout the calculation. For the FD example, $\Delta r_{\min} = 1/N_{\text{FD}}$, and for the MQ example, $\Delta r_{\min} = 0.045$.

Table 3. Comparison of numerical and exact solution at $t/t_0 = 1.1523$

Method	r	ρ	u	p
Exact		0.00599	1.9026	25.840
MQ—35pts		0.00599	1.9028	25.841
FD—50pts	0.3164	0.00813	2.3141	28.374
FD—500pts		0.00412	1.7178	23.019
FD—5000pts		0.00517	1.8734	24.770
Exact		0.17811	4.0579	27.207
MQ—35pts		0.17811	4.0579	27.206
FD—50pts	0.6656	0.19409	4.1121	27.070
FD—500pts		0.17827	4.0566	27.171
FD—5000pts		0.18103	4.0611	27.248
Exact		3.1492	7.6322	71.176
MQ—35pts		3.1492	7.6321	71.173
FD—50pts	1.0365	3.8476	7.4644	66.588
FD—500pts		3.1248	7.6436	71.468
FD—5000pts		3.1417	7.5501	71.809
Exact		4.0000	7.9540	84.355
MQ—35pts		4.0000	7.9538	84.356
FD—50pts	1.0584	3.9244	8.0203	82.453
FD—500pts		3.9291	8.0129	82.711
FD—5000pts		3.9315	8.0140	82.878

The total operations count for the MQ scheme is:

$$N_{\text{tot}}^{\text{MQ}} = 42{,}096 \cdot (t_F - t_0) \cdot (|u| + c)_{\max}/(0.9 \cdot 0.045). \tag{65}$$

For the FD scheme, we have

$$N_{\text{tot}}^{\text{FD}} = 249 \cdot N_{\text{FD}}^2 (t_F - t_0) \cdot (|u| + c)_{\max}/0.9. \tag{66}$$

Taking the ratios of equations (65) and (66) gives

$$N_{\text{tot}}^{\text{MQ}}/N_{\text{tot}}^{\text{FD}} = 1.645,\ 0.01645 \text{ and } 0.0001645. \tag{67}$$

for $N_{\text{FD}} = 50{,}500$ and 5000, respectively. The operations count of this 35 point MQ scheme is equal to a 64 point FD scheme.

There are many variations of familiar FD schemes. Local mesh refinement may drastically reduce the number of FD nodes required, but we must realize low order FD schemes still are slowly convergent. Just how the operations counts compare between MQ and FD schemes in the two and three dimensions, with and without adaptive mesh refinement, etc. requires further research.

2.3. The solution of a two-dimensional elliptic Poisson's equation

In this section, we will examine the solution technique used to solve the elliptic Poisson's equation subject to either Dirichlet or Neumann boundary conditions.

The sample problem to be solved is the following elliptic Poisson's equation

$$\partial^2 f/\partial x^2 + \partial^2 f/\partial y^2 = (\lambda^2 + \mu^2)\exp(\lambda x + \mu y), \tag{68}$$

in the interior. On the boundary, we have either Dirichlet conditions

$$f = \exp(\lambda x + \mu y), \tag{69}$$

or Neumann boundary conditions

$$\partial f/\partial x = \lambda \exp(\lambda x + \mu y) \tag{70a}$$

or

$$\partial f/\partial y = \mu \exp(\lambda x + \mu y). \tag{70b}$$

In the MQ example, we used 30 nodal values, 12 scattered points in the interior and 18 along the boundary. The domain was a unit square on $[0, 1] \times [0, 1]$. The location of the data points will be presented later.

The expansion used is the Madych–Nelson expansion with an appended constant, i.e.

$$f(\mathbf{x}) = a_1 + \sum_{j=2}^{N} \hat{g}(\mathbf{x} - \mathbf{x}_j) a_j, \tag{71a}$$

where

$$\hat{g}(\mathbf{x} - \mathbf{x}_j) = g(\mathbf{x} - \mathbf{x}_j) - g(\mathbf{x} - \mathbf{x}_1) \tag{71b}$$

and

$$g(\mathbf{x} - \mathbf{x}_j) = [(\mathbf{x} - \mathbf{x}_j)^2 + (y - y_j)^2 + r_j^2]^{1/2}. \tag{71c}$$

The partial derivatives of the basis functions have been presented previously, see equations (37) and (38). We partitioned the points $\mathbf{X}_i$ into three classes: (1) those points belonging to the interior set, I; (2) those boundary points who are Dirichlet points, D and (3) those boundary points who are Neumann, N, points.

In order to solve for the N expansion coefficients, $\mathbf{a}$, we construct an appropriate MQ coefficient matrix, $\mathbf{S}$, and the corresponding column vectors, $\mathbf{b}$. The coefficient matrix is constructed in the following manner

$$\begin{aligned} S_{i,1} &= 0, \\ S_{i,j} &= (\partial^2 \hat{g}/\partial x^2 + \partial^2 \hat{g}/\partial y^2)_{i,j}, \quad \text{for } 2 \leqslant j \leqslant N, \\ b_i &= (\lambda^2 + \mu^2)\exp(\lambda x_i + \mu y_i), \end{aligned} \tag{72}$$

if $i \in I$

$$S_{i,1} = 1,$$
$$S_{i,j} = \hat{g}(x_i - x_j), \text{ for } 2 \leqslant j \leqslant N, \tag{73}$$
$$b_i = \exp(\lambda x_i + \mu y_i),$$

if $i \in D$.

$$S_{i,l} = 0,$$
$$S_{i,j} = (\partial \hat{g}/\partial x)_{ij}, \tag{74}$$

or for $2 \leqslant j \leqslant N$

$$(\partial \hat{g}/\partial y)_{ij},$$
$$b_i = \lambda \exp(\lambda x_i + \mu y_i),$$

or

$$\mu \exp(\lambda x_i + \mu y_i) \quad \text{if } i \in N$$

The MQ expansion coefficients are found as the solution of

$$\mathbf{Sa} = \mathbf{b}, \tag{75}$$

where

$$\mathbf{a} = \mathbf{S}^{-1}\mathbf{b}, \quad \text{assuming } \mathbf{S} \text{ is invertible.} \tag{76}$$

The values of the function, f, at $\mathbf{x}_i$ are given by equations (76) and (71).

Our results for Poisson's equations, equation (68) are presented in Tables 4 and 5 run with $\lambda = 2.0$ and $\mu = 3.0$. Table 4 was run with all boundaries being Dirichlet boundaries. Table 5 was with Dirichlet boundaries at $y = 0.0$ and $y = 1.0$ and Neumann boundaries at $x = 0.0$ and $x = 1.0$.

Table 4. Scattered data solution of Poisson's equation over a unit square with all Dirichlet boundary conditions

X	Y	MQ—F	EXACT—F
0.0	0.0	1.000	1.000
0.0	0.25	2.117	2.117
0.0	0.50	4.482	4.482
0.0	0.75	9.448	9.488
0.0	1.0	20.09	20.09
1.0	0.0	7.389	7.389
1.0	0.25	15.64	15.64
1.0	0.50	33.12	33.12
1.0	0.75	70.11	70.22
1.0	1.0	148.4	148.4
0.20	0.0	1.492	1.492
0.20	1.0	29.96	29.96
0.40	0.0	2.226	2.226
0.40	1.0	44.70	44.70
0.60	0.0	3.320	3.320
0.60	1.0	66.69	66.69
0.80	0.0	4.953	4.593
0.80	1.0	99.48	99.48
0.05	0.05	1.299	1.284
0.13	0.26	2.794	2.829
0.46	0.16	3.998	4.055
0.31	0.42	6.229	6.554
0.07	0.58	6.516	6.554
0.12	0.73	11.28	11.36
0.42	0.91	35.45	35.52
0.51	0.57	14.61	15.33
0.68	0.82	45.48	45.60
0.84	0.37	15.14	16.28
0.97	0.68	55.39	53.52
0.17	0.93	22.83	22.87

Max norm error = 1.877; L-2 norm error = 0.074.

Table 5. Scattered data solution of Poisson's equation over a unit square using both Neumann and Dirichlet boundary conditions

X	Y	MQ—F	EXACT—F
0.0	0.0	1.000	1.000
0.0	0.25	2.572	2.117
0.0	0.50	4.906	4.482
0.0	0.75	9.949	9.488
0.0	1.0	20.09	20.09
1.0	0.	7.389	7.389
1.0	0.25	15.64	15.64
1.0	0.50	33.12	33.12
1.0	0.75	70.11	70.22
1.0	1.0	148.4	148.4
0.20	0.0	1.492	1.492
0.20	1.0	29.96	29.96
0.40	0.0	2.226	2.226
0.40	1.0	44.70	44.70
0.60	0.0	3.320	3.320
0.60	1.0	66.69	66.69
0.80	0.0	4.953	4.593
0.80	1.0	99.48	99.48
0.05	0.05	1.408	1.284
0.13	0.26	3.245	2.829
0.46	0.16	5.040	4.055
0.31	0.42	7.023	6.554
0.07	0.58	6.783	6.554
0.12	0.73	11.39	11.36
0.42	0.91	33.34	35.52
0.51	0.57	15.94	15.94
0.68	0.82	47.38	45.60
0.84	0.37	17.89	16.28
0.97	0.68	54.94	53.52
0.17	0.93	22.93	22.87

Max norm error = 1.772; L-2 norm error = 0.106.

The results presented in Tables 4 and 5 are quite good, considering that over the unit square the exact function varies from 1 to 148. In addition, the interior points were scattered. Because of the global, rather than local, connectivity, we are guaranteed that the MQ coefficient matrix is always invertible for distinct points, see Ref. [6].

In order to estimate the computational effort required to achieve the same accuracy as our MQ results, we experimented with the grid size to an l_2 norm error of 2.0 in the estimates of the first and second partial derivatives throughout the domain $[0, 1] \times [0, 1]$. We found that a minimum uniform grid spacing $\Delta x = \Delta y = 0.03$ or a 33×33 grid was required, whereas only 30 points were used for the MQ solution.

In the next set of examples, we used another scattered data arrangement with 10 boundary points and 20 interior points. We ran two cases,

$$\nabla^2 f = 13 \exp(-2x + 3y) \quad (\lambda = -2 \text{ and } \mu = 3.0)$$

and

$$\nabla^2 f = 32.16 \exp(4.01x + 4.01y) \quad (\lambda = \mu = 4.01).$$

These two cases are presented in Tables 6 and 7. The errors change in magnitude with the steepness of the function. In the first case, f varies from 0.25 to 20 in $[0, 1] \times [0, 1]$ with a max error of 0.44. In the second case, f varies from 1.0 to 3041 in $[0, 1] \times [0, 1]$. This function is too steep to be represented by only 30 data points, and much of the structure is missing. To test this hypothesis, we excluded the last two interior points given in Table 12. We found for a total of 28 points, this problem gives an l_2 error of 14.9 and a max error of 255. With 29 points, the l_2 error was 9.82 and the max error was 122. Franke [2] also observed the goodness-of-fit improves as more sample points were included.

Braess [12, 13] and Braess and Hackbusch [14], used multigrid methods for solving Poisson's equation in two dimensions. They experienced problems with convergence on reentrant corner which may produce pollution effects on the solution, and recommended extra relaxation steps to reduce the errors.

Table 6. Scattered data solution of Poisson's equation over a unit square $\nabla^2 F = 13.3 \exp(-2x + 3y)$ using all Dirichlet boundary conditions

X	Y	MQ—F	EXACT—F
0.0	0.0	1.000	1.000
0.0	0.22	1.935	1.935
0.0	0.50	4.482	4.482
0.0	1.0	20.09	20.09
0.50	0.0	0.3679	0.3679
0.50	1.0	7.389	7.389
1.0	0.0	0.1353	0.1353
1.0	1.0	2.718	2.718
1.0	0.50	0.6065	0.6065
1.0	0.70	1.105	1.105
0.13	0.26	1.627	1.682
0.31	0.42	1.720	1.896
0.77	0.58	1.195	1.221
0.22	0.73	5.464	5.755
0.93	0.18	0.2598	0.2671
0.86	0.78	1.997	1.859
0.42	0.91	6.490	6.619
0.51	1.57	1.910	1.994
0.68	0.82	3.024	3.004
0.72	0.62	1.514	1.522
0.64	0.33	0.7345	0.7483
0.84	0.37	0.5701	0.5655
0.97	0.68	1.127	1.105
0.27	0.93	9.040	9.488
0.53	0.11	0.4778	0.4819
0.15	0.21	1.338	1.391
0.69	0.95	4.385	4.349
0.48	0.44	1.225	1.433
0.21	0.57	3.462	3.633
0.59	0.94	5.272	5.155

Max norm error = 0.4480; L-2 norm error = 0.0225.

Table 7. Scattered data solution of Poisson's equation over a unit square, $\nabla^2 F = 32.16^* \exp(4.01x + 4.01y)$ using all Dirichlet boundary conditions

X	Y	MQ—F	EXACT—F
0.0	0.0	1.000	1.000
0.0	0.22	2.416	2.426
0.0	0.50	7.526	7.426
0.0	1.0	55.15	55.15
0.50	0.0	7.426	7.426
0.50	1.0	409.5	409.5
1.0	0.0	55.15	55.15
1.0	0.50	409.5	409.5
1.0	0.70	913.2	913.2
1.0	1.0	3041.0	3041.0
0.13	0.26	9.774	4.777
0.31	0.42	46.90	18.68
0.77	0.58	225.1	224.4
0.22	0.73	48.83	45.13
0.93	0.18	108.7	85.72
0.86	0.78	672.0	718.0
0.42	0.91	191.2	207.1
0.51	1.57	109.8	76.01
0.68	0.82	348.6	409.5
0.72	0.62	216.5	215.6
0.64	0.33	78.37	48.90
0.94	0.37	149.3	128.0
0.97	0.68	722.2	747.3
0.27	0.93	125.9	123.0
0.53	0.11	26.72	13.02
0.13	0.21	5.752	3.909
0.69	0.95	650.6	718.0
0.48	0.44	52.06	40.01
0.19	0.57	31.06	21.06
0.59	0.94	403.6	461.9

Max error = 67.40; L-2 error = 4.585.

Regarding the MQ solution of the two-dimensional Poisson's equation over scattered data, we find that Franke's [2] observations regarding the goodness of the approximation and the number of data ponts considered is relevant. Also to be taken into account is whether too many distances between points are similar which leads to poor coefficient matrix conditioning. The most effective strategy with MQ still remains to be developed since MQ is a fairly new tool in computational fluid-dynamics.

3. SUMMARY

Multiquadrics (MQ) has been used as a spatial approximation scheme for parabolic, hyperbolic and the elliptic partial differential equations (PDEs). When comparing the MQ results with the exact solutions, the agreement is very good.

We have shown with the parabolic linear advection-diffusion equation that MQ can be cast into an implicit time marching scheme. Furthermore, no special stabilizing treatment of the advective term is required with MQ as compared to standard finite difference (FD) approximations. Because MQ is a very high order scheme, we can obtain excellent results using a coarser distribution of data points than with an FD approximation. We have shown that as the FD distribution becomes increasingly refined, that the MQ scheme on the coarser distribution is more efficient and more accurate than the FD scheme.

We have demonstrated MQ is an excellent spatial approximation scheme for non-linear systems of hyperbolic PDEs. Using an explicit time marching scheme combined with moving nodes, we have shown MQ is not only far more efficient and accurate than FD. FD converges slowly to the exact solution. Even though the operational/node for the FD scheme is relatively small, the FD scheme becomes inefficient compared to the MQ scheme since many more FD nodes are required to achieve the same accuracy as with an MQ calculation. Note that grid staggering is unnecessary using MQ, simplifying our bookkeeping.

We have shown MQ is an excellent approximation scheme for solving a two-dimensional elliptic Poisson's equation especially with the interior nodes being scattered. When MQ is used in a scattered data scheme, we need not be concerned with long skinny triangles since the connectivity is shared with all points considered. We have a straightforward procedure for calculating either Neumann or Dirichlet boundary conditions by merely changing the rows of the appropriate MQ coefficient matrix.

Micchelli [6] and Madych and Nelson [7] have provided a theoretical justification for the performance of MQ. Micchelli showed that MQ belongs to a class of conditionally positive definite functions, and that the MQ coefficient matrix is invertible for distinct points. Madych and Nelson have shown for all functions in the space of conditionally positive-definite functions, a semi-norm exists which is minimized by all such functions.

Per se, MQ is a global approximation scheme which would be impractical for large scale fluid dynamics problems. However, we have shown that domain decomposition into many overlapping subdomains has been proven very effective and efficient. Over each subdomain, we deal with not only smaller coefficient matrices which could be partitioned to parallel processsors, but each coefficient matrix has a much better condition number thereby improving accuracy. By this process of subdomain decomposition and blending, MQ can be made more competitive than the corresponding simple slowly converging FD schemes.

As shown in the previous paper [1], MQ is an excellent spatial approximation scheme in steep regions, but is rather poor in the flat regions. For this reason, we advocate a hybrid scheme in which MQ would be the spatial approximation scheme in rapidly varying regions and the monotonic cubic spline can be used elsewhere. Because MQ does very well in steep regions, we can minimize the operations count because excessive grid refinement is unnecessary as with the slowly converging low order FD schemes. If we use different approximation schemes for different regions such as done with ordinary differential equation solvers, we can optimize both accuracy and computational efficiency.

The next phase of inquiry will be directed toward thoroughly understanding a suite of two-dimensional dynamic problems to gain more experience and understanding. MQ performs very well in steep regions where most low order polynomial schemes fail, but becomes noisy and

inefficient in shallow gradient regions. Therefore, we advocate a hybrid scheme in which either standard FD schemes or monotonic splines be used in shallow gradient regions, and MQ be used in steep regions. Furthermore, front tracking methods such as in Ref. [11] can be used for shocks, contact discontinuities and material interfaces. We believe that a more sophisticated approach would be to use spatial approximation schemes which are best suited for a particular situation. It appears that the brute force trend of relying on the number of processors, memory and arithmetic performance alone is not justified.

Acknowledgements—Work performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under Contract No. W-7405-ENG-48, and partially supported by the Army Research Office, Contract No. MIPR-ARO 124-84.

REFERENCES

1. E. J. Kansa, Multiquadrics—a scattered data approximation scheme with applications to computational fluid-dynamics—I. Surface approximations and partial derivative estimates. *Computers Math. Applic.* **19**(8/9), 127–145 (1990).
2. R. Franke, Scattered data interpolation: test of some other methods. *Math. Comput.* **38,** 181–200 (1982).
3. R. L. Hardy, Multiquadric equations of topography and other irregular surfaces. *J. geophys. Res.* **176,** 1905–1915.
4. R. L. Hardy, Research results in the application of multiquadratic equations to surveying and mapping problems. *Surv. Mapp.* **35,** 321–332 (1975).
5. S. Stead, Estimation of gradients from scattered data. *Rocky Mount. J. Math.* **14,** 265–279 (1984).
6. C. A. Micchelli, Interpolation of scattered data: distance matrices and conditionally positive definite functions. *Constr. Approx.* **2,** 11–22 (1986).
7. W. R. Madych and S. A. Nelson, Multivariate interpolation: a variational theory. *J. Approx. Theory Applic.* (in press).
8. E. Adams, *VII Int. Conf. Computational Methods in Water Resources.* MIT Press, Cambridge, Mass. June (1988).
9. E. J. Kansa, Highly accurate shock flow calculations with moving grids and mesh refinement, in *Numerical Mathematics and Applications* (Eds R. Vichnevetsky and J. Vignes). pp. 311–316. North Holland, New York, (1986).
10. J. von Neumann, *John von Neumann, Collected Works* (Ed. A. K. Traub), Vol. 6, pp. 219–237. MacMillan, New York (1963).
11. E. J. Kansa, Application of Hardy's multiquadric interpolation to hydrodynamics. *Proc. 1986 Simul Conf.*, Vol. 4, pp. 111–117 (1986).
12. D. Braess, The contraction number of a multigrid method for solving the Poisson equation. *Numer. Math.* **37,** 387–404 (1981).
13. D. Braess, The convergence rate of a multigrid method relaxation for the Poisson equation with Gauss–Seidel relaxation for the Poisson equation. *Math. Comput.* **42,** 505–519 (1984).
14. D. Braess and W. Hackbusch, A new convergence proof for the multigrid method including the V-cycle. *SIAM Jl numer. Analysis* **20,** 967–975 (1983).