AN EFFICIENT NUMERICAL SCHEME FOR BURGERS' EQUATION

Y.C. Hon[1] X.Z. Mao[2]

[1]Department of Mathematics
City University of Hong Kong, Hong Kong.
[2]Zhejiang Provincial Institute of Estuarine and
Coastal Engineering Research, China.
(visiting City University of Hong Kong)

ABSTRACT: This paper applies the multiquadric (MQ) as a spatial approximation scheme for solving the nonlinear Burgers' equation. For comparison purposes, a low order explicit finite difference approximation of the time derivative is employed. By decreasing the time step of the computation, it is shown that the major numerical error is from the time integration instead of the MQ spatial approximation. The numerical results indicate that this MQ offers an excellent approximation for all possible values of Reynolds number. An adaptive algorithm is also developed to adjust the MQ interpolation points to the peak of the shock wave which is shown to provide an improved numerical result. Numerical comparisons are made with most of the existing numerical schemes for solving the Burgers' equation.

Key words: Burgers' equation; nonlinear PDE; multiquadric.

1. INTRODUCTION

Burgers [1] firstly proposed the following nonlinear partial differential equation (now called Burgers' equation)

$$u_t + uu_x = \frac{1}{R}u_{xx}, \qquad R > 0 \tag{1}$$

as a mathematical model of free turbulence. Since then this model has been studied by many researchers for the following reasons: (1) it contains the simplest form of nonlinear advection term uu_x and dissipation term u_{xx}/R for simulating the physical phenomena of wave motion; (2) its analytical solution was obtained by Cole [2] so that numerical comparison can be made; and (3) its shock wave behavior when the Reynolds number R is large. Various numerical techniques have been applied to solve numerically equation (1) under the following boundary conditions

$$u(0,t) = 0 = u(1,t), \qquad t > 0 \tag{2}$$

and the initial condition

$$u(x,0) = f(x), \qquad 0 \le x \le 1. \tag{3}$$

It is not a purpose of this paper to exhaust all of existing numerical schemes for solving Burgers' equation. To mention some of them: cubic spline and finite differences by Jain & Holla [3], compact differencing technique by Hirsh [4] and Ciment et al. [5], finite element method (FEM) by Arminjon & Beanchamp [6], FEM with splitting technique by Jain & Raja [7] and Iskanda & Mohsen [8], FEM with moving nodes technique by Herbst et al. [9] and Caldwell et al. [10], Tau method by Ortiz & Pun [11] and method of lines by Sincovec & Madsen [12]. In this paper a numerical scheme is devised by using Hardy's multiquadric (MQ), which is a special kind of radial basis functions (RBF), to solve the nonlinear Burgers' equation. Hardy [13] firstly developed this MQ to approximate two-dimensional geographical surfaces. In Franke's [14] review paper, the MQ was rated one of the best methods among 29 scattered data interpolation schemes based on their accuracy, stability, efficiency,

memory requirement, and ease of implementation. Recently, Kansa [15, 16] successfully modified the MQ for solving PDE problems. Since then more researchers have been attracted by this mesh-free, scattered data approximation scheme. As an interpolation method, the MQ always produces a minimal semi-norm error as proven by Madych and Nelson [17]. In this paper we propose a new mechanism to adjust the values of the shape parameters r_j's whose magnitudes are a key factor for attaining high accuracy. Numerical computations show that this method offers an accuracy better than FEM with moving nodes when the Reynold number R is small and much better result than FEM with splitting technique when R is large. It is well known that there is no known numerical scheme which is good for arbitrary values of R. It was shown in Caldwell's [18] paper that the finite difference method (FDM) is good when R is small but gives incorrect answers when R is large whilst in the latter case FEM is superior to FDM. Numerical computations for a wide range of values of R show that this MQ offers better accuracy in comparison with all of the above mentioned methods. To further illustrate the benefit of this mesh-free MQ, an adaptive points choosing algorithm based on 'chasing the peak' of the shock wave is developed and is shown to provide an improved numerical result.

Furthermore, this MQ proposed here is of a general nature and can be used for solving nonlinear PDEs arising in other areas.

2. NUMERICAL SCHEME USING MULTIQUADRIC

For comparison purposes, we first discretize equation (1) by using a low order forward difference approximation scheme for the time derivative to obtain

$$u^m + k(u^{m-1}u_x^m - \frac{1}{R}u_{xx}^m) = u^{m-1}, \quad m \geq 1, \tag{4}$$

where k is the length of time step and u^m denotes the m^{th} iterate of the solution. It has been shown in our numerical computations that by decreasing only the magnitude of the time step k, the numerical error has been reduced to an approximate order of $O(k)$. This indicates that the MQ spatial approximation are highly accurate which is

particularly useful because this usually can only be achieved by using costly methods like Fourier transform. As FEM typically uses low order polynomials as basic linear elements, the faster convergent MQ can represent a steeper function because of its higher degree than FEM.

Following the idea of modified MQ by Kansa [16], we propose to approximate, at each iteration m, the u^m by

$$u^m(x) = \sum_{j=0}^{N} \lambda_j^m [(x - x_j)^2 + r_j^2]^{1/2} + \lambda_{N+1}^m x + \lambda_{N+2}^m, \tag{5}$$

where $x_j = j/N, j = 0, 1, \ldots, N$ are $(N + 1)$ distinct uniformly distributed points in $[0, 1]$. The r_j's are called shape parameters whose magnitudes are a key factor for obtaining accurate solution. Tarwater [19] in her recent numerical experiments observed that there exists an optimal miminum r, in the case of constant shape parameter, due to the ill-conditioning effect when solving systems of linear equations. Kansa [16] proposed an exponential variation in r_j's to give a better conditioned coefficient matrix resulted from collocation using a formula similar to equation (5). However, our numerical computations indicate that a better condition number does not necessarily lead to an optimal accuracy. In fact, we observe that the optimal solution for the Burgers' equation is attained with a resultant matrix having large condition number ($\simeq 10e12$). This is consistent with the numerical results obtained by Golberg & Chen [20], Bogomonly [21], and Cheng [22] . In this paper, we propose the following formula for choosing the values of r_j's

$$r_j = Mj + b, \qquad j = 0, 1, \ldots, N, \tag{6}$$

where M and b are input parameters. Our numerical observations show that the accuracy of the solution depends on the magnitudes of M and b in such a way that the error drops to a minimum by choosing first the value of M so that the numerical solution provides a reasonable approximation to the exact solution. The accuracy can then further be 'fine-tuned' by adjusting the value of b. In fact, the value of M is equal to a fixed constant -0.2 in all our numerical computations with various Reynold number R. This constant M is easily found by entering M for 1, 10, -1, -0.1, and finally -0.2 successfully because the numerical solutions are far from the

exact solution for all the values other than -0.2. A monotonical convergent behavior is observed when adjusting the value of b for better accuracy in which the value of b decreases with respect to the increasing magnitude of the Reynold number R. This observation guides us to adjust the value of b for optimal accuracy. Here, we have a similar curiosity to Golberg & Chen [20] on the excellent performance of this MQ in spite of huge condition number. This can be explained by Christiansen & Saranen [23] which in their paper they pointed out that in solving a general matrix equation $Ax = b$, the numerical error is in general related to the condition number of the matrix A. However, there are two types of condition number: $K(global)$ and $K(local)$ defined by

$$K(global) = \frac{largest \quad eigenvalue \quad of \quad A}{smallest \quad eigenvalue \quad of \quad A}, \tag{7}$$

and

$$K(local) = \frac{\| A^{-1} \| \cdot \| b \|}{\| x \|}. \tag{8}$$

In many cases, $K(local) << K(global)$. In our numerical computations, K(global) is approximately $10e12$ but K(local) is only approximately $10e5$. This explains the excellent numerical results obtained by using this MQ despite large condition number (global). For each iteration m, to determine the $(N + 3)$ coefficients λ_j^m's, the boundary conditions (2) already give the following 2 equations

$$u^m(x_0) = 0 = u^m(x_N). \tag{9}$$

We then collocate u^m at $(N+1)$ distinct uniformly distributed points $\hat{x}_i = i/(N + 2)$ in (0, 1) using equation (4) to obtain, for $i = 1, 2, \ldots, N + 1$,

$$u^m(\hat{x}_i) + k[u^{m-1}(\hat{x}_i)\frac{d}{dx}u^m(\hat{x}_i) - \frac{1}{R}\frac{d^2}{dx^2}u^m(\hat{x}_i)] = u^{m-1}(\hat{x}_i), \quad m \geq 1, \tag{10}$$

where $u^0(\hat{x}_i)$ is taken to be $f(\hat{x}_i)$ from the initial condition (3). The system of equations (9) and (10) can then be solved by using Gaussian elimination with partial pivoting to obtain the coefficients λ_j^m's. We note here that the points x_j in (5) are different from the points $\hat{x}_i$ in (10). The following section gives numerical results to verify the accuracy and efficiency of this MQ.

3. NUMERICAL COMPUTATIONS

The analytical solution given by Cole [2] for equation (1) subject to the boundary conditions (2) and the initial condition (3) is

$$u(x,t) = \frac{2\pi\nu \sum_{n=1}^{\infty} nA_n sin(n\pi x) exp(-n^2\nu\pi^2 t)}{A_0 + \sum_{n=1}^{\infty} A_n cos(n\pi x) exp(-n^2\nu\pi^2 t)} \tag{11}$$

where $\nu = 1/R$ and

$$A_n = 2\int_0^1 cos(n\pi x) exp\left(-\frac{1}{2\nu}\int_0^x f(y)dy\right) dx, \qquad n \geq 1, \tag{12}$$

$$A_0 = \int_0^1 exp\left(-\frac{1}{2\nu}\int_0^x f(y)dy\right) dx. \tag{13}$$

In the case when $f(x) = sin\pi x$, Caldwell & Smith [24] derived that

$$u(x,t) = \frac{4\pi\nu \sum_{n=1}^{\infty} nI_n(1/2\pi\nu) sin(n\pi x) exp(-n^2\nu\pi^2 t)}{I_0(1/2\pi\nu) + 2\sum_{n=1}^{\infty} I_n(1/2\pi\nu) cos(n\pi x) exp(-n^2\nu\pi^2 t)} \tag{14}$$

where I_n denotes the modified Bessel function of order n. It can be observed that the difficulty in evaluating an accurate solution $u(x,t)$ based on the analytical formula (14) is due to the exponential increasing term $I_n(1/2\pi\nu)$ in the series when ν is small (i.e. when R is large). Using MATLAB [25] in Sun Sparc model 20 workstation (with 64 bits double precision), we can compute $u(x,t)$ to four decimal digits by evaluating the two series in formula (14) for $R = 0.1$, 1, 10, and 100 respectively. Unless an asymptotic formula is used, it is not possible to compute $u(x,t)$ using formula (14) when R is large. In fact, the magnitude of $I_n(1/2\pi\nu)$ already exceeds the limit of the 64 bit computation when $R \geq 4500$. For comparison purposes, in the case when $R = 10000$, we adopt the accurate solution computed by Christie & Mitchell [26] using the Galerkin method with fully upwinded cubic functions and a particularly small value of spatial step h. The numerical solution obtained by using a compact differencing technique is also quoted from the book of Mitchell & Griffiths [27] for comparison.

In order to compare with other algorithms, we apply the MQ to solve the nonlinear Burgers' equation (1) subject to the boundary conditions (2) and initial condition

$f(x) = sin\pi x$ for R = 0.1, 1, 10, 100, and 10000 respectively. All the computations are performed using MATLAB in Sun Sparc model 20 workstation. Due to the explanation on the accuracies of various methods given in section 1 and the availability of data, the comparison is made according to Table 1. The numerical results for various cases of R are summarized in the followings:

Case i. $R = 0.1$. This is the case when the dissipation term dominates the advection term. In fact, the curve of the solution u drops dramatically from $sin\pi x$ to zero within the first 0.05 seconds. To compare with the numerical value of the analytical solution, we compute using our MQ the approximation $u^m(x)$ by taking time step k = 0.001, 0.0001, and 0.00001 respectively. In this computation, we let $N = 10$ in equation (5) so that $x_j = 0(0.1)1$. M is taken to be -0.2 and $b = 2.5$ in equation (6). The result of comparison at $x = 0.1(0.1)0.9$ and t = 0.01 and 0.02 is shown in Table 2. It can be observed that with fixed number of spatial points, the accuracy of the numerical solution has been increased to an approximate order of $O(k)$. This indicates that the MQ spatial approximation is highly accurate. All our numerical computations in the following cases have the same finding.

Case ii. $R = 1$. This is also the case when the dissipation term dominates the advection term. In order to compare with the numerical values obtained by using FDM and FEM with moving node technique (Caldwell & Smith [24]), we let $N = 4$ in equation (5) so that $x_j = 0(0.25)1$. The time step k is taken to be 0.01 for comparison. M is taken to be -0.2 and $b = 2.5$ in equation (6). The result of comparison at $x = 0.25(0.25)0.75$ and t = 0.01, 0.05(0.05)0.25 is shown in Table 3. Numerical comparison in this case shows that our MQ offers slightly better results than FEM with moving nodes. We note here that our MQ has an advantage of ease of implementation compared to FDM. For illustrative purposes, we also list our numerical result when k = 0.0001 for comparison with the analytical solution at the end of Table 3.

Case iii. $R = 10$. This is the case when both the dissipation term and the advection term have a balanced influence on the solution. Again, the numerical value of the analytical solution is available. We compute using our MQ the approximation $u^m(x)$ by taking time steps k = 0.1, 0.01, and 0.001 respectively. In this computation, we let $N = 10$ in equation (5) so that $x_j = 0(0.1)1$. M is taken to be -0.2 and b = 2.4 in equation (6). The result of comparison at $x = 0.1(0.1)0.9$ and t = 0.5 and 1.0

is shown in Table 4.

Case iv. $R = 100$. This is the case when the advection term dominates the dissipation term. The behavior of the solution u is very different when advection dominates. It can be observed that the solution u incurs a shock wave to the right with speed proportional to the magnitude of u. We compare the numerical values of the analytical solution and our MQ approximation $u^m(x)$ by taking time step $k =$ 0.1, 0.01, and 0.001 respectively. In this computation, we let $N = 10$ in equation (5) so that $x_j = 0(0.1)1$. M is taken to be -0.2 and $b = 2.22$ in equation (6). The result of comparison at $x = 0.1(0.1)0.9$ and $t = 0.5$ and 1.0 is shown in Table 5.

Case v. $R = 10000$. This is also the case when the advection term dominates the dissipation term. In fact, the peak of the shock wave remains high and moves to the right during the first 0.5 seconds. In order to compare with the numerical values obtained using compact difference, FEM with splitting technique (Iskandar & Mohsen [8]), and FEM with moving nodes (Caldwell et al. [10]), we let $N = 10$ in equation (5) and the time step k is taken to be 0.1 and 0.001 for comparison. M is taken to be -0.2 and $b = 2.205$ in equation (6). The result of comparison at $x = 0(1/18)1$ and $t = 1$ is shown in Table 6. Numerical comparison in this case shows that our MQ offers better results than FEM with moving nodes and much better results than compact difference and FEM with splitting.

Also, in Figure 1(a-e), we display the graphs of our approximation u^m to the solution u by evaluating the last iterate u^m at $x = 0(1/300)1$ using the global formula (5) for the above five cases. This shows that our proposed MQ has an advantage of offering a global formula to the solution. Figure 1d can also be compared with the Figure 1 of Ortiz & Pun [11].

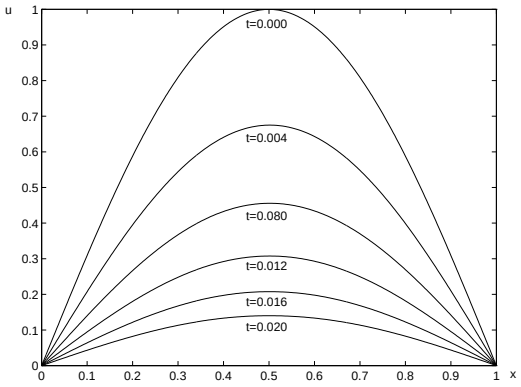

Figure 1a. R = 0.1

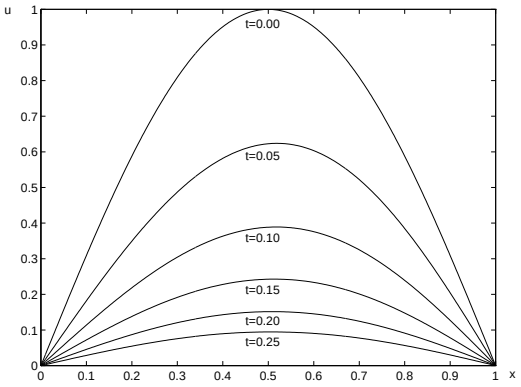

Figure 1b. R = 1

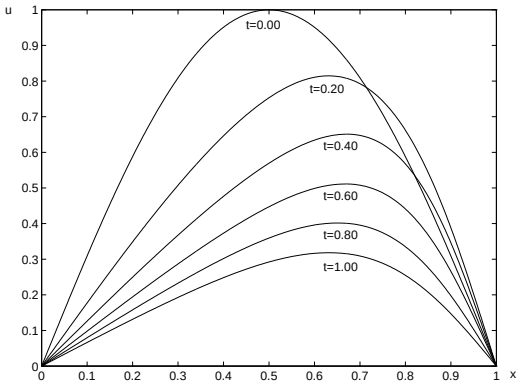

Figure 1c. R = 10

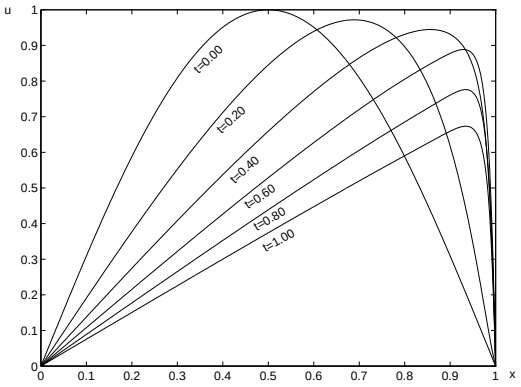

Figure 1d. R = 100

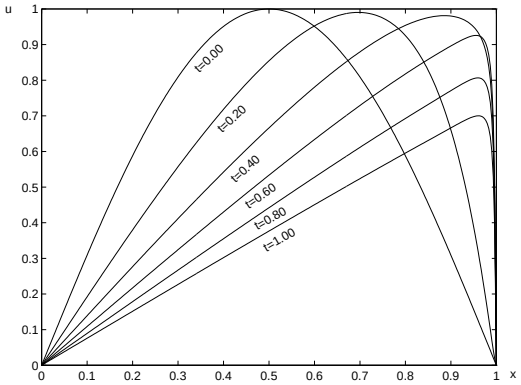

Figure 1e. R = 10000

Figure 1. Curves of MQ approximation of Burgers' equation for $R = 0.1$, 1, 10, 100, and 10000 respectively in $0 \leq x \leq 1$.

Finally, to further illustrate the benefit of this mesh-free MQ, we develop an adaptive MQ points choosing algorithm based on 'chasing the peak' of the shock wave which occurs when the Reynold number R is large. The idea in using FEM in solving this Burger's equation by Caldwell, Wanless and Cook [10] is to alter the size of the elements at each stage using information from the previous step so that more elements are closer to the peak. However, the expense of this element alteration at each time step in FEM can be very costly which makes it impractical for problems with complex geometry. Since the MQ is mesh free, the re-allocation of the collocation points is comparatively simple and ease to implement. We re-allocate at each time iteration m the points x_j in (5) such that it always contains a point x^* where $du^m(x^*)/dx$ is zero. This point x^* can easily be found by using the following Newton's iterative scheme

$$x^*_{New} = x^*_{Old} - \frac{\frac{d}{dx}u^m(x^*_{Old})}{\frac{d^2}{dx^2}u^m(x^*_{Old})} \tag{15}$$

because the first and second derivatives of u^m are readily computed by using (5). This is definitely an advantage of this MQ since its basis functions are C^∞ and globally defined. In contrast, this high order Newton iterative scheme cannot be used in FEM because the numerical solution is defined locally in each element. In each iteration, the initial iterate at each time step m in (15) is the x^* computed from the last time step $(m-1)$. After x^* is obtained, the re-allocation of the points are done by first computing how many L points should be on the left of x^*:

$$L = round(Nx^* - 0.5). \tag{16}$$

The $(N+1)$ points x_j in (5) are then re-computed by letting $x_j = jx^*/L, j = 0, 1, \ldots, L$ and $x_j = x^* + (j-L)(1-x^*)/(N-L), j = (L+1), \ldots, N$. Numerical verification of this adaptive algorithm is performed to our last case (v) computation with $R = 10000$. The time step k is again taken to be 0.1 and 0.001 for comparison. The parameters M and b remain the same as -0.2 and 2.205 respectively although the points x_j are now changing at each time step. This shows that the location of the

MQ points x_j has little effect on the choice of the shape parameter which affects the accuracy of the optimal solution. This phenomena is also reported by Carlson and Foley [28]. Numerical computations show that this adaptive MQ offers much better results near the peak of the shock wave. Compared with FEM with moving nodes, this adaptive MQ is much easier to implement and, as shown in Table 7, offers much better numerical results.

4. CONCLUSION

Numerical results show that the multiquadric offers a very high accuracy in computation. Unlike the finite element method which interpolates the solution by using local basis functions, this multiquadric provides a global interpolation formula not only for the solution but also for the derivatives of the solution. Unfortunately, the relationship between the magnitudes of the shape parameters r_j's and the rate of convergence is still yet to be studied. However, in our numerical computations, we observe that the magnitudes of the b's in the formula for r_j's decreases monotonically from 2.5 to 2.205 as the Reynolds number R increases from 0.1 to 10000 for optimal accuracy. This highlights an existence of radius of convergence of this multiquadric corresponding to the derivatives of the solution which we hope could be obtained in the future. For comparison purpose, this paper applies only low order explicit finite difference approximation for the time derivatives. Our numerical results show that the major numerical error is from the time integration instead of the MQ spatial approximation. A much better result should be obtained by taking a numerical Laplace inversion scheme to the time derivative as reported by Moridis & Kansa [29] because the time marching process can contaminate the solutions by means of time truncation errors. In the meantime, we expect that this multiquadric can provide an alternative mesh-free computational algorithm supplement to finite element and boundary element methods.

5. ACKNOWLEDGMENTS

The authors would like to thank the Research Grants Council and the City University of Hong Kong for their research grants support.

References

[1] J.M. Burgers, Application of a model system to illustrate some points of the statistical theory of free turbulence, Proc. Acad. Sci. Amsterdam, 43:2–12 (1940).

[2] J.D. Cole, On a quasi-linear parabolic equation occuring in aerodynamics, Q. Appl. Math., 9:225–236 (1951).

[3] P.C. Jain and D.N. Holla, Numerical solutions of coupled Burgers' equation, Int. J. Non-linear Mech., 13:213–222 (1978).

[4] R.S. Hirsh, Higher order accurate difference solutions of fluid mechanics problems by a compact differencing technique, J. Comp. Phys., 19:90–109 (1975).

[5] M. Ciment, S.H. Leventhal and B.C. Weinberg, The operator compact implicit method for parabolic equations, J. Comp. Phys., 28:135–166 (1978).

[6] P. Arminjon and C. Beauchamp, A finite element method for Burgers' equation in hydrodynamics, Int. J. Numer. Methods Eng., 12:415–428 (1978).

[7] P.C. Jain and M. Raja, Splitting-up technique for Burgers' equations, Indian J. Pure Appl. Math., 10:1543–1551 (1979).

[8] L. Iskandar and A. Mohsen, Some numerical experiments on the splitting of Burgers' equation, Numerical Methods for Partial Differential Equations, 8:267-276 (1992).

[9] B.M. Herbst, S.W. Schoombie, D.F. Griffiths and A.R. Mitchell, Generalized Petrov-Galerkin methods for the numerical solution of Burgers' equation, Int. J. Numer. Methods Eng., 20:1273–1289 (1984).

[10] J. Caldwell, P. Wanless and A.E. Cook, Solution of Burgers' equation for large Reynolds number using finite elements with moving nodes, J. Appl. Math. Modelling, 11:211–214 (1987).

[11] E.L. Ortiz and K.S. Pun, A bidimensional Tau-elements method for the numerical solution of nonlinear partial differential equations with an application to Burgers' equation, Comp. & Maths. with Appls., 12B:1225–1240 (1986).

[12] R.F. Sincovec and N.K. Madsen, Software for nonlinear partial differential equations, ACM TOMS, 1:222–260 (1975).

[13] R.L. Hardy, Multiquadric equations of topography and other irregular surfaces, J. Geophys. Res., 176:1905–1915 (1971).

[14] R. Franke, Scattered data interpolation: test of some methods, Math. Comput., 38:181–200 (1982).

[15] E.J. Kansa, Multiquadrics - a scattered data approximation scheme with applications to computational fluid dynamics - I, Computers Math. Applic., 19:127–145 (1990).

[16] E.J. Kansa E.J., Multiquadrics - a scattered data approximation scheme with applications to computational fluid dynamics - II, Computers Math. Applic., 19:147–161 (1990).

[17] W.R. Madych and S.A. Nelson, Multivariate interpolation and conditionally positive definite functions I, J. Approx. Theory and Its Applications, 4:77–89 (1988).

[18] J. Caldwell, Numerical approaches to the nonlinear Burgers' equation, J. Math. Sc., 5:19–32 (1994).

[19] A.E. Tarwater, A parameter study of Hardy's multiquadric method for scattered data interpolation, UCRL-54670, Sept. 1985.

[20] M.A. Golberg and C.S. Chen, On a method of Atkinson for evaluating domain integrals in the boundary element method, Appl. Math. Comput., 60:125–138 (1994).

[21] A. Bogomonly, Fundamental solutions method for elliptic boundary value problems, SIAM J. Numer. Anal., 22:644–669 (1985).

[22] R.S.C. Cheng, Delta-trigonometric and spline methods using the single-layer potential representation, Ph. D. dissertation, University of Maryland, 1987.

[23] S. Christiansen and J. Saranen, The conditioniny of some numerical methods for first kind of boundary integral equations, Comput. Appl. Math., 67:43-58 (1996).

[24] J Caldwell and P. Smith, Solution of Burgers' equation with a large Reynolds number, Appl. Math. Modelling, 6:381–385 (1982).

[25] MATLAB 4.2, The MathWorks, Inc., 24 Prime Park Way, Natick, MA, 1994.

[26] I. Christie and A.R. Mitchell, Upwinding of high order Galerkin methods in conduction-convection problems, Int. J. Num. Meth. Eng., 12:1764–1771 (1978).

[27] A.R. Mitchell and D.F. Griffiths, "The Finite Difference Method in Partial Differential Equations", John Wiley & Sons Ltd, 1980, pp.233.

[28] R.E. Carlson and T.A. Foley, "The parameter R^2 in multiquadric and related interpolations", Comput. Math. Applic., 21:29–42 (1991).

[29] G.J. Moridis and E.J. Kansa, "The Laplace transform multiquadrics method: highly accurate scheme for the numerical solution of linear partial differential equations", Appl. Sci. Comput., 1:375–407 (1994).

Table 1. Comparison between MQ and other methods for various R

R	Compared with
0.1	Cole's analytical solution
1	Cole, FDM, FEM with moving node
10	Cole
100	Cole, Tau method
10000	Christie's accurate solution, FEM with splitting, FEM with moving nodes, Compact difference method

Table 2. Comparison of results for $R = 0.1$

	time $t = 0.01$				time $t = 0.02$			
		MQ (time step k)				MQ (time step k)		
x	Anal	0.001	0.0001	0.00001	Anal	0.001	0.0001	0.00001
0.10	0.1146	0.1199	0.1152	0.1147	0.0428	0.0469	0.0433	0.0429
0.20	0.2182	0.2283	0.2192	0.2183	0.0815	0.0893	0.0823	0.0816
0.30	0.3006	0.3146	0.3021	0.3008	0.1122	0.1230	0.1133	0.1124
0.40	0.3539	0.3704	0.3556	0.3541	0.1320	0.1447	0.1333	0.1322
0.50	0.3727	0.3902	0.3745	0.3729	0.1389	0.1522	0.1403	0.1391
0.60	0.3550	0.3717	0.3567	0.3552	0.1322	0.1449	0.1335	0.1324
0.70	0.3024	0.3167	0.3039	0.3026	0.1125	0.1234	0.1136	0.1127
0.80	0.2200	0.2304	0.2211	0.2201	0.0818	0.0897	0.0826	0.0819
0.90	0.1157	0.1213	0.1163	0.1159	0.0430	0.0472	0.0434	0.0431

Table 3. Comparison of results for $R = 1$

	$x = 0.25$				$x = 0.50$			
	FEM	FDM	MQ		FEM	FDM	MQ	
t	(n=4)*	(exp)	(N=4)	Anal	(n=4)*	(exp)	(N=4)	Anal
0.01	0.6333	0.6267	0.6332	0.6290	0.9100	0.9063	0.9097	0.9057
0.05	0.4231	0.4099	0.4228	0.4131	0.6233	0.6100	0.6229	0.6091
0.10	0.2648	0.2525	0.2645	0.2536	0.3889	0.3729	0.3886	0.3716
0.15	0.1671	0.1565	0.1669	0.1566	0.2429	0.2281	0.2426	0.2268
0.20	0.1053	0.0967	0.1052	0.0964	0.1517	0.1395	0.1515	0.1385
0.25	0.0662		0.0661	0.0592	0.0948		0.0946	0.0845

* n is the number of elements

	$x = 0.75$				MQ ($k = 0.0001$)			
	FEM	FDM	MQ			x		
t	(n=4)*	(exp)	(N=4)	Anal	t	0.25	0.50	0.75
0.01	0.6539	0.6550	0.6537	0.6524	0.01	0.6291	0.9057	0.6525
0.05	0.4601	0.4556	0.4597	0.4502	0.05	0.4130	0.6092	0.4502
0.10	0.2862	0.2762	0.2858	0.2726	0.10	0.2535	0.3716	0.2726
0.15	0.1767	0.1663	0.1765	0.1644	0.15	0.1566	0.2268	0.1644
0.20	0.1093	0.1006	0.1092	0.0994	0.20	0.0964	0.1385	0.0995
0.25	0.0678		0.0677	0.0603	0.25	0.0592	0.0846	0.0604

Table 4. Comparison of results for $R = 10$

	time $t = 0.5$				time $t = 1.0$			
		MQ (time step k)				MQ (time step k)		
x	Anal	0.1	0.01	0.001	Anal	0.1	0.01	0.001
0.10	0.1099	0.1118	0.1104	0.1103	0.0663	0.0670	0.0664	0.0664
0.20	0.2180	0.2211	0.2186	0.2183	0.1312	0.1325	0.1314	0.1313
0.30	0.3222	0.3258	0.3227	0.3224	0.1928	0.1948	0.1930	0.1928
0.40	0.4190	0.4221	0.4194	0.4191	0.2480	0.2512	0.2483	0.2481
0.50	0.5028	0.5038	0.5028	0.5028	0.2919	0.2969	0.2923	0.2919
0.60	0.5623	0.5594	0.5618	0.5621	0.3161	0.3239	0.3167	0.3159
0.70	0.5759	0.5674	0.5744	0.5753	0.3081	0.3192	0.3090	0.3079
0.80	0.5055	0.4924	0.5030	0.5041	0.2537	0.2666	0.2548	0.2534
0.90	0.3093	0.2974	0.3059	0.3066	0.1461	0.1554	0.1468	0.1459

Table 5. Comparison of results for $R = 100$

	time $t = 0.5$				time $t = 1.0$			
		MQ (time step k)				MQ (time step k)		
x	Anal	0.1	0.01	0.001	Anal	0.1	0.01	0.001
0.10	0.1211	0.1211	0.1211	0.1211	0.0754	0.0754	0.0755	0.0755
0.20	0.2415	0.2413	0.2415	0.2415	0.1506	0.1506	0.1507	0.1507
0.30	0.3603	0.3589	0.3601	0.3603	0.2257	0.2253	0.2256	0.2257
0.40	0.4764	0.4724	0.4760	0.4764	0.3003	0.2995	0.3002	0.3003
0.50	0.5887	0.5799	0.5877	0.5886	0.3744	0.3726	0.3742	0.3744
0.60	0.6953	0.6780	0.6935	0.6951	0.4478	0.4444	0.4475	0.4478
0.70	0.7935	0.7615	0.7901	0.7932	0.5203	0.5143	0.5197	0.5202
0.80	0.8783	0.8189	0.8719	0.8776	0.5915	0.5813	0.5905	0.5913
0.90	0.9381	0.8128	0.9264	0.9381	0.6600	0.6438	0.6594	0.6607

Table 6. Comparison of results for $R = 10000$

	Christie	Compact	FEM	FEM	MQ	
	accurate	difference	splitting	moving node	($N = 10$)	
x	solution	k=0.001	k=0.1*	k=0.001*	k=0.1	k=0.001
0.00	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
0.05	0.0422	0.0501	0.0419	0.0422	0.0422	0.0424
0.11	0.0843	0.0753	0.0839	0.0844	0.0843	0.0843
0.16	0.1263	0.1471	0.1253	0.1266	0.1263	0.1263
0.22	0.1684	0.1359	0.1692	0.1687	0.1682	0.1684
0.27	0.2103	0.2611	0.2034	0.2108	0.2100	0.2103
0.33	0.2522	0.2091	0.2666	0.2527	0.2517	0.2522
0.38	0.2939	0.3340	0.2527	0.2946	0.2931	0.2939
0.44	0.3355	0.3048	0.3966	0.3362	0.3343	0.3355
0.50	0.3769	0.4173	0.2350	0.3778	0.3751	0.3769
0.55	0.4182	0.3741	0.5480	0.4191	0.4156	0.4182
0.61	0.4592	0.5059	0.2578	0.4601	0.4557	0.4592
0.66	0.5000	0.4634	0.6049	0.5009	0.4952	0.4999
0.72	0.5404	0.5808	0.6014	0.5414	0.5341	0.5404
0.77	0.5806	0.5369	0.4630	0.5816	0.5722	0.5805
0.83	0.6203	0.6671	0.7011	0.6213	0.6093	0.6201
0.88	0.6596	0.6201	0.6717	0.6605	0.6455	0.6600
0.94	0.6983	0.7410	0.7261	0.6992	0.6783	0.6957
1.00	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

* with 16 intervals

Table 7. Comparison of results for $R = 10000$ by using the adaptive algorithm

	Christie	FEM	Adaptive MQ	
	accurate	moving node	($N = 10$)	
x	solution	k=0.001*	k=0.1	k=0.001
0.00	0.0000	0.0000	0.0000	0.0000
0.05	0.0422	0.0422	0.0421	0.0421
0.11	0.0843	0.0844	0.0843	0.0843
0.16	0.1263	0.1266	0.1263	0.1264
0.22	0.1684	0.1687	0.1682	0.1684
0.27	0.2103	0.2108	0.2100	0.2103
0.33	0.2522	0.2527	0.2517	0.2522
0.38	0.2939	0.2946	0.2931	0.2939
0.44	0.3355	0.3362	0.3343	0.3355
0.50	0.3769	0.3778	0.3751	0.3769
0.55	0.4182	0.4191	0.4156	0.4182
0.61	0.4592	0.4601	0.4557	0.4592
0.66	0.5000	0.5009	0.4952	0.4999
0.72	0.5404	0.5414	0.5341	0.5404
0.77	0.5806	0.5816	0.5722	0.5805
0.83	0.6203	0.6213	0.6093	0.6202
0.88	0.6596	0.6605	0.6453	0.6595
0.94	0.6983	0.6992	0.6785	0.6972
1.00	0.0000	0.0000	0.0000	0.0000

* with 16 intervals

FIGURE CAPTIONS

Figure 1. Curves of MQ approximation of Burgers' equation for $R = 0.1$, 1, 10, 100, and 10000 respectively in $0 \leq x \leq 1$.

1a. $R = 0.1$.

1b. $R = 1$.

1c. $R = 10$.

1d. $R = 100$.

1e. $R = 10000$.