

THE TIME-MARCHING MESHLESS METHOD OF FUNDAMENTAL SOLUTIONS FOR WAVE EQUATION

MARJAN UDDIN¹, AND HAZRAT ALI²

¹Department of Basic Sciences, University of Engineering and Technology Peshawar, Pakistan.
marjankhan1@hotmail.com

² Department of Basic Sciences, University of Engineering and Technology Peshawar, Pakistan.
alimathematics007@gmail.com

ABSTRACT. *We extended a local meshless approximation method for solving numerically time-dependent 2D wave equations. The present numerical method is meshfree and is capable of solving large scale time-dependent PDEs in multi-dimension for arbitrary shaped domain.*

Keywords: Local Meshless Method; Radial Kernel Functions; 2D Wave Equation.

1. **Introduction.** Roland Hardy, an Iowa State geodesist, studied RBFs approximation methods in 1968. Hardy found that his developed interpolation methods were effective for solving two-dimensional and higher dimensional scattered data problems. Hardy method is known as the multiquadric (MQ) radial basis function method [7]. In 1986 Charles Micchelli, proved that the system matrix for the MQ method was invertible, which means that the RBF scattered data interpolation problem is well-posed [8]. later on, physicist Edward Kansa first used the MQ method to solve partial differential equations [9]. The convergence theory of Kansa's approach was provided by Schaback [10]. During the past two decades the meshless methods have been developed and effectively applied to solve many engineering and science problems [11, 12, 13, 14, 15, 16, 17, 19, 20, 21, 22, 24, 25, 26, 27, 23]. The key point of Kansa's method for solving the PDEs is the approximation by a set of global approximation functions. The main advantage of using the RBF approximation method for solution of PDEs is its simplicity, applicability to various PDEs, and effectiveness in dealing with high dimensional problems and complicated domains. The main disadvantage of RBF approximation method comes from the related full matrices that are very sensitive to the choice of the free parameter in RBFs and difficult to solve for problems with a large number of unknowns. This is because the use of the radial basis function interpolation increases the condition numbers of the related matrices with increasing number of nodes. There are several methods to circumvent this issue such as domain decomposition [28, 29], the greedy algorithm [30]. One of the possibilities for mitigating computational cost for large-scale problems is to employ the domain decomposition by Mai-Duy et al. [31], the multi-grid approach and compactly supported RBFs by Chen et al. [32]. An other way is to use the local radial basis function methods such as [33, 17, 34, 16], etc. In the present work, we extended local radial basis function method and developed a local meshless numerical scheme for solving time-dependent 2D wave equation.

2. Local RBF approximation method for time-dependent PDEs. In this section, we describe a local RBF approximation method for time-dependent PDEs of the form

$$u_t(x, t) = \mathcal{L}u(x, t), x \in \Omega \subset \mathcal{R}^d, t > 0. \quad (1)$$

2.1. Spatial approximation. For given a set of N centers $\{x_1, \dots, x_N\} \subset \Omega$. Where Ω is arbitrary shaped domain and the centers can be chosen any where in the domain. In the local RBF approximation method, at each center $x_i \in \Omega$, the local interpolant takes the form

$$u(x_i, t) = \sum_{j=i_1}^{i_n} \lambda_j(t) \phi(\|x_i - x_j\|), j = i_1, i_2, \dots, i_n \in I_i(n) \subset \Omega, \quad (2)$$

where λ is a vector of expansion coefficients, $r = \|x - x_j\|$, denotes the Euclidean distance between two centers x and x_j , $\phi(r)$ is a radial basis function defined for $r \geq 0$ and $I_i(n)$ is a vector corresponding to the center x_i , contains the index i_1 of the center x_i as well as the indices of the reaming $n - 1$ centers. The set of centers corresponding to the vector $I_i(n)$ is called a stencil, two such stencils, one correspond to an interior center, and another to boundary center, have been shown in Fig. 1. So corresponding to each stencil, we have N number of $n \times n$ linear systems,

$$\mathbf{u}^i = \mathbf{A}^i \lambda^i, i = 1, 2, \dots, N. \quad (3)$$

where the entries of the matrix \mathbf{A}^i are $a_{kj}^i = \phi(\|x_k - x_j\|)$, $k, j \in I_i(n)$, the matrix \mathbf{A}^i is called the interpolation matrix, and each system have to be solved for the expansion coefficients. Now to approximate $\mathcal{L}u(x, t)$, where \mathcal{L} is a linear differential operator, we have

$$\mathcal{L}u(x_i, t) = \sum_{j=i_1}^{i_n} \lambda_j(t) \mathcal{L}\phi(\|x_i - x_j\|), j = i_1, i_2, \dots, i_n \in I_i(n) \subset \Omega, \quad (4)$$

Eq. (3) can be written as a dot product of two vectors, as

$$\mathcal{L}u(x_i, t) = \mathbf{v}^i \cdot \lambda^i, \quad (5)$$

where λ^i is the $n \times 1$ vector of expansion coefficients, and \mathbf{v}^i is $1 \times n$ vector with entries

$$\mathbf{v}^i = \mathcal{L}\phi(\|x_i - x_j\|), j = i_1, i_2, \dots, i_n \in I_i(n) \quad (6)$$

To eliminate the expansion coefficients, we have from Eq. (2)

$$\lambda^i = (\mathbf{A}^i)^{-1} \mathbf{u}^i \quad (7)$$

We substitute the values of λ^i from (7) in (5) to get,

$$\mathcal{L}u(x_i, t) = \mathbf{v}^i (\mathbf{A}^i)^{-1} \mathbf{u}^i = \mathbf{w}^i \mathbf{u}^i \quad (8)$$

where,

$$\mathbf{w}^i = \mathbf{v}^i (\mathbf{A}^i)^{-1}, \quad (9)$$

is the weight corresponding to center x_i . Hence for all centers locations, we have

$$\mathcal{L}u = \mathbf{W} \mathbf{u}, \quad (10)$$

where \mathbf{W} is $N \times N$ sparse differentiation matrix, each row of the matrix \mathbf{W} contains n non-zeros elements and $N-n$ zeros elements, where n is the number of elements in a stencil.

2.2. Temporal approximation. After spatial local RBF approximation, we obtained the following system of ODEs

$$u_t = F(u), \quad (11)$$

In our case $F(\mathbf{u}) = \mathbf{W}\mathbf{u}$. To discretize in time we can use any ODE solver like ode113, ode45 from Matlab. The starting vector will be the initial solution u_0 .

ode45 is based on an explicit Runge-Kutta (4,5) formula, the Dormand-Prince pair [35]. It is a one-step solver for computing $u(t_n)$, and it needs only the solution at the immediately preceding time point, $u(t_{n-1})$. In general, ode45 is the best function to apply as a first try for most problems.

ode113 is a variable order Adams-Bashforth-Moulton PECE solver [36]. It may be more efficient than ode45 for stringent tolerances and when the ODE file function is particularly expensive to evaluate. ode113 is a multistep solver; it normally requires the solution at several preceding time points to compute the current solution.

A good ODE solver will automatically select a reasonable time step δt and detect stiffness of the ODE system. For this ODE computation we have used a fourth-order Runge-Kutta method and selected the time step δt manually.

$$\begin{aligned} \mathbf{r}_1 &= \mathbf{F}(\mathbf{u}^n), \\ \mathbf{r}_2 &= \mathbf{F}\left(\mathbf{u} + \frac{\delta t}{2}\mathbf{r}_1\right), \\ \mathbf{r}_3 &= \mathbf{F}\left(\mathbf{u} + \frac{\delta t}{2}\mathbf{r}_2\right), \\ \mathbf{r}_4 &= \mathbf{F}(\mathbf{u} + \delta t\mathbf{r}_3), \\ \mathbf{u}^{n+1} &= \mathbf{u}^n + \frac{\delta t}{6}(\mathbf{r}_1 + 2\mathbf{r}_2 + 2\mathbf{r}_3 + \mathbf{r}_4). \end{aligned} \quad (12)$$

2.3. Choosing a good value of shape parameter. A variety of kernel functions are available in the literature. In our computation we used the the multiquadrics, $\phi(r) = \sqrt{1 + (cr)^2}$. As usual these RBFs contain a shape parameter c and solution accuracy greatly depends on this parameter. In this paper we used the the following well known algorithm for a good value of the shape parameter [16].

Algorithm

$$\kappa = 0$$

$$10^{13} < \kappa < 10^{15}$$

while $\kappa < \kappa_{min}$ and $\kappa > \kappa_{max}$

$$\mathbf{U}, \mathbf{S}, \mathbf{V} = \text{svd}(\mathbf{A}^i)$$

$$\kappa = \max(S)/\min(S)$$

$$\text{if } \kappa < \kappa_{min}, c = c - \delta c$$

$$\text{if } \kappa > \kappa_{max}, c = c + \delta c$$

$$c \text{ (optimal)} = c.$$

Here svd is the singular value decomposition of the interpolation matrix \mathbf{A}^i . \mathbf{U} , \mathbf{V} are $N \times N$ orthogonal matrices and \mathbf{S} is $N \times N$ diagonal matrix contains the N singular values of \mathbf{A}^i , and $\kappa = \|\mathbf{A}^i\| \|(\mathbf{A}^i)^{-1}\| = \max(S)/\min(S)$ is the condition number of the matrix \mathbf{A}^i .

When an acceptable value of shape parameter is returned by the above algorithm, then the svd is used to compute $(\mathbf{A}^i)^{-1} = (\mathbf{U}\mathbf{S}\mathbf{V}^T)^{-1} = \mathbf{V}\mathbf{S}^{-1}\mathbf{U}^T$ (see [18]). Note that for orthogonal matrices the inverse of the matrix is equal to its transpose. Consequently, we can compute the weights \mathbf{w}^i in (9).

3. Numerical experiments. In this section, we apply the local meshless approximation method described in the above section to check the accuracy and validity of the local meshless approximation method for the wave equations.

	n	L_∞	κ	c	C.time(s)
RK4(MQ)	5	1.5537e-003	4.8730e+013	0.0200	2.465325
	10	1.4628e-003	3.3301e+014	0.4300	2.658144
	15	3.1315e-004	4.2923e+013	1.0800	3.018656
	20	2.9421e-004	5.2723e+013	1.4600	2.996878

TABLE 1. Numerical results with different stencil sizes n , corresponding to example 1: when $N = 625$, $t = 1$, $\delta t = 0.001$, $\alpha = 1$, domain $[0, 1]^2$.

3.1. The 2D membrane vibrating problem. Here we consider time-dependent 2D membrane vibration problem

$$u_{tt}(x, y, t) = \alpha^2 \nabla^2 u(x, y, t), \quad x, y \in \Omega, t > 0, \quad (13)$$

which may be transformed into a system of two partial differential equations given by

$$u_t = v, \quad v_t = \alpha^2 \nabla^2 u(x, y, t), \quad x, y \in \Omega, t > 0, \quad (14)$$

and having exact solutions

$$u(x, y, t) = \frac{64}{\pi^6} \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} \frac{1}{m^3 n^3} \sin(m\pi x) \sin(n\pi y) \cos(\sqrt{m^2 + n^2} \pi \alpha t), \quad (15)$$

to validate the capability of the local meshless method. The initial conditions $u(x, y, 0) = f(x, y)$, and $u_t(x, y, 0) = g(x, y, 0)$ can be extracted from the analytical solution. We have used the boundary conditions $u = 0$ on the boundary $\partial\Omega$. The solution is advanced in time by fourth-order Runge-Kutta method with time step size $\delta t = 0.001$. The problem domain is discretized with $N = 625$ equally spaced centers and stencil size $5 \leq n \leq 20$ have been used as shown in Fig. 1. The optimal value of n is mainly a problem dependent, however for a sparse differentiation matrix we have to choose n much smaller than N . The L_∞ error norm, the stencil size n , the condition number κ , of the interpolation matrix for the last center and its corresponding shape parameter c , and the computation time in seconds are shown in Table 1.

4. Conclusions. In this work we have extended the local radial basis functions approximation method and developed a local meshless numerical scheme for solving 2D wave equation. Due to the radial basis functions the applied method have a great flexibility to solve multi-dimensional problems with arbitrary shaped domain. For large scale problems it is not possible to implement the global RBF method due to the dense differentiation matrices, but the local RBF method have the capability of solving large scale problems with a small stencil size to keep the differentiation matrix sparse. The applied local method is equally applicable for large scale time-dependent multi-dimensional problems with irregular shaped domain.

Acknowledgment

We are thankful to UET Peshawar and HEC Pakistan for their financial support under the grant no: 112-25237-2PSI-127 (50021443).

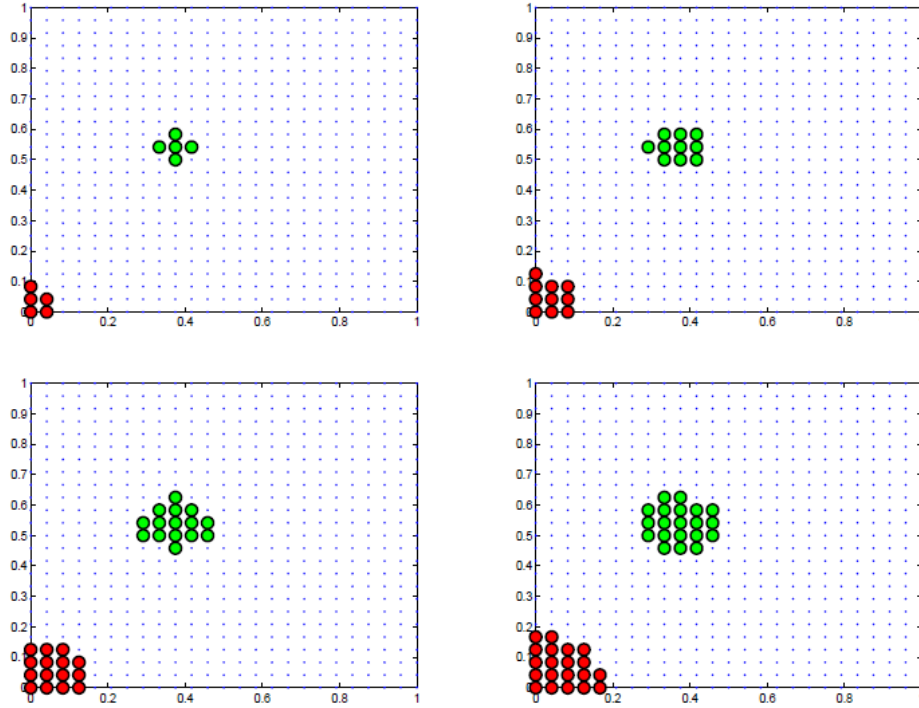


FIGURE 1. Centers distributions and a typical stencil associated with a boundary center (red) and an interior center (green)

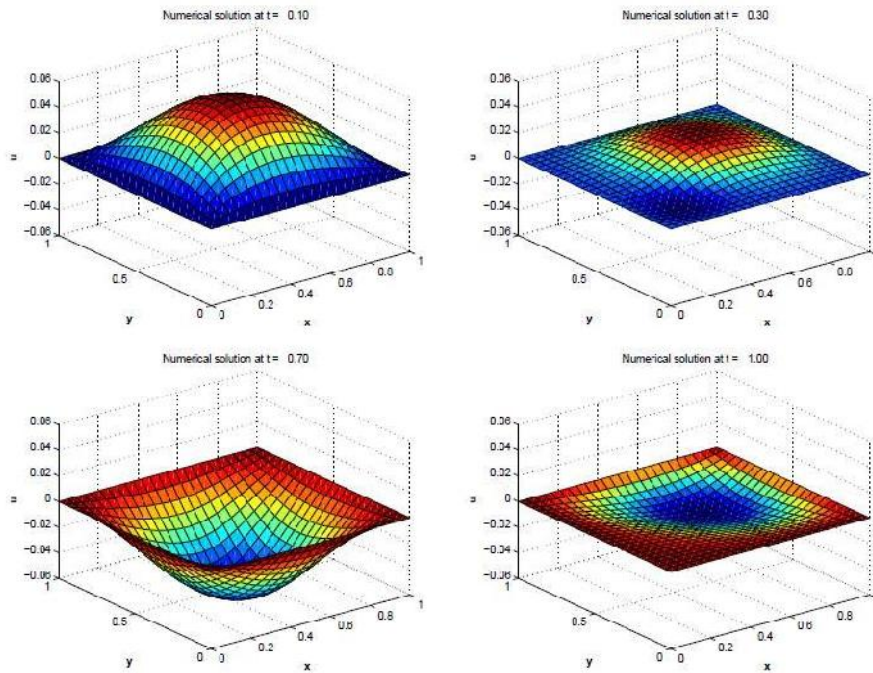


FIGURE 2. 2D membrane vibration problem: Numerical results, when stencil size $n = 20$, centers $N = 625$, $\delta t = 0.001$, $\alpha = 1$ and domain $[0, 1]^2$.

REFERENCES

- [1] Smiley, M. W. (2009). An efficient implementation of a numerical method for a chemotaxis system. *International Journal of Computer Mathematics*, 86(2), 219-235.
- [2] Keller, E. F., & Segel, L. A. (1970). Initiation of slime mold aggregation viewed as an instability. *Journal of Theoretical Biology*, 26(3), 399-415.
- [3] Epshteyn, Y., & Kurganov, A. (2008). New interior penalty discontinuous Galerkin methods for the Keller-Segel chemotaxis model. *SIAM Journal on Numerical Analysis*, 47(1), 386-408.
- [4] Chertock, A., & Kurganov, A. (2008). A second-order positivity preserving central-upwind scheme for chemotaxis and haptotaxis models. *Numerische Mathematik*, 111(2), 169-205.
- [5] Smiley, M. W. (2009). An efficient implementation of a numerical method for a chemotaxis system. *International Journal of Computer Mathematics*, 86(2), 219-235.
- [6] Tyson, R., Stern, L. G., & LeVeque, R. J. (2000). Fractional step methods applied to a chemotaxis model. *Journal of mathematical biology*, 41(5), 455-475.
- [7] Hardy, R. L. (1971). Multiquadric equations of topography and other irregular surfaces. *Journal of geophysical research*, 76(8), 1905-1915.
- [8] Micchelli, C. A. (1984). *Interpolation of scattered data: distance matrices and conditionally positive definite functions* (pp. 143-145). Springer Netherlands.
- [9] Kansa, E. J. (1990). Multiquadrics—A scattered data approximation scheme with applications to computational fluid-dynamics—II solutions to parabolic, hyperbolic and elliptic partial differential equations. *Computers & mathematics with applications*, 19(8), 147-161.
- [10] Schaback, R. (2007). Convergence of unsymmetric kernel-based meshless collocation methods. *SIAM Journal on Numerical Analysis*, 45(1), 333-351.
- [11] Mai-Duy, N., & Tran-Cong, T. (2001). Numerical solution of Navier–Stokes equations using multiquadric radial basis function networks. *International Journal for Numerical Methods in Fluids*, 37(1), 65-86.
- [12] Liu, G. R. (2009). *Meshfree methods: moving beyond the finite element method*. CRC press.
- [13] Buhmann, M. D. (2000). Radial basis functions. *Acta Numerica 2000*, 9, 1-38.
- [14] Wendland, H. (2004). *Scattered data approximation* (Vol. 17). Cambridge university press.
- [15] Fasshauer, G. F. (2007). *Meshfree approximation methods with MATLAB*. World Scientific Publishing Co., Inc..
- [16] Sarra, S. A., & Kansa, E. J. (2009). Multiquadric radial basis function approximation methods for the numerical solution of partial differential equations. *Advances in Computational Mechanics*, 2, 2.
- [17] Sarra, S. A. (2012). A local radial basis function method for advection–diffusion–reaction equations on complexly shaped domains. *Applied Mathematics and Computation*, 218(19), 9853-9865.
- [18] Trefethen, L. N., & Bau III, D. (1997). *Numerical linear algebra* (Vol. 50). Siam.
- [19] Fasshauer, G. E., Khaliq, A. Q. M., & Voss, D. A. (2004). Using meshfree approximation for multi-asset American options. *Journal of the Chinese Institute of Engineers*, 27(4), 563-571.
- [20] Uddin, M., & Haq, S. (2009). Numerical solution of complex modified Korteweg–de Vries equation by mesh-free collocation method. *Computers & Mathematics with Applications*, 58(3), 566-578.
- [21] Uddin, M., & Haq, S. (2011). RBFs approximation method for time fractional partial differential equations. *Communications in Nonlinear Science and Numerical Simulation*, 16(11), 4208-4214.
- [22] Haq, S., & Uddin, M. (2011). RBFs approximation method for Kawahara equation. *Engineering Analysis with Boundary Elements*, 35(3), 575-580.
- [23] Uddin, M., & Haq, S. (2011). Application of a Numerical Method Using Radial Basis Functions to Nonlinear Partial Differential Equations. *Selcuk Journal of Applied Mathematics*, 12(1), 77-93.
- [24] Haq, S., & Uddin, M. (2009). Numerical solution of nonlinear schrodinger equations by collocation method using radial basis functions. *CMES: Computer Modeling in Engineering & Sciences*, 44(2), 115-136.
- [25] Uddin, M., & Kassem, G. (2012). Radial basis functions approximation method for numerical solution of good Boussinesq equation. *SL: Structural Longevity*, 8(3), 91-103.
- [26] Uddin, M., Haq, S., & Ishaq, M. (2012). RBF-Pseudospectral Method for the Numerical Solution of Good Boussinesq Equation. *Applied Mathematical Sciences*, 6(49), 2403-2410.

- [27] Uddin, M. (2013). RBF-PS scheme for solving the equal width equation. *Applied Mathematics and Computation*, 222, 619-631.
- [28] Hardy, R. L. (1977). Least squares prediction. *Photogrammetric Engineering and Remote Sensing*, 43(4).
- [29] Kansa, E. J., & Hon, Y. C. (2000). Circumventing the ill-conditioning problem with multiquadric radial basis functions: applications to elliptic partial differential equations. *Computers & Mathematics with applications*, 39(7), 123-137.
- [30] Hon, Y. C., Schaback, R., & Zhou, X. (2003). An adaptive greedy algorithm for solving large RBF collocation problems. *Numerical Algorithms*, 32(1), 13-25.
- [31] Mai-Duy, N., & Tran-Cong, T. (2002). Mesh-free radial basis function network methods with domain decomposition for approximation of functions and numerical solution of Poisson's equations. *Engineering Analysis with Boundary Elements*, 26(2), 133-156.
- [32] Chen, C. S., Ganesh, M., Golberg, M. A., & Cheng, A. D. (2002). Multilevel compact radial functions based computational schemes for some elliptic problems. *Computers & Mathematics with Applications*, 43(3), 359-378.
- [33] Shu, C., Ding, H., & Yeo, K. S. (2003). Local radial basis function-based differential quadrature method and its application to solve two-dimensional incompressible Navier–Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 192(7), 941-954.
- [34] Vertnik, R., & Šarler, B. (2006). Meshless local radial basis function collocation method for convective-diffusive solid-liquid phase change problems. *International Journal of Numerical Methods for Heat & Fluid Flow*, 16(5), 617-640.
- [35] Dormand, J. R., & Prince, P. J. (1980). A family of embedded Runge-Kutta formulae. *Journal of computational and applied mathematics*, 6(1), 19-26.
- [36] Shampine, L. F., & Gordon, M. K. (1975). Computer solution of ordinary differential equations.