Middle-East Journal of Scientific Research 11 (12): 1628-1635, 2012
ISSN 1990-9233

DOI: 10.5829/idosi.mejsr.2012.11.12.1415

A New Approach for Solving Partial Differential Equations in Large Domains using Radial Basis Functions

[1]M. Esmaeilbeigi, [1]M.M. Hosseini and [2]Syed Tauseef Mohyud-Din

[1]Faculty of Mathematics, Yazd University, P.O. Box 89195-74, Yazd, Iran
[2]HITEC University, Taxila Cantt., Pakistan

Abstract: In this paper, a stable method is proposed for solving time dependent partial differential equations in large domains using radial basis functions. In this new approach, a domain decomposition scheme is applied by using collocation points and thin plate splines. The scheme works in a similar fashion as finite difference methods. The merit of the proposed approach is that it is capable to reduce condition number of the matrices resulting from discretization of the equations and easily overcome the difficulty arising in solving complicated algebraic systems. The new method is applied to linear hyperbolic telegraph and nonlinear Klein-Gordon equations and the obtained results confirm the accuracy and efficiency of this method. The results of numerical experiments are presented with and without using domain decomposition method.

Key words: Collocation method . radial basis functions . partial differential equations . thin plate splines . domain decomposition

INTRODUCTION

The Radial Basis Function (RBF) method has been actively used for solving Partial Differential Equations (PDEs) [1-3]. The RBF method for time dependent PDEs enjoy large advantages in accuracy over other flexible, but low order methods, such as finite differences, finite volumes and finite elements. However, RBF method shares the ease of implementation and flexibility of these lower order methods [4]. The traditional RBFs are globally defined functions which result in a full resultant coefficient matrix. In addition, in large domains, we require a large number of collocation points to get desirable accuracy, but we must also mention that the matrices which result from the discretization of the equations are usually ill-conditioned, in this case. This hinders the application of the RBFs to solve large domain problems due to severe ill-conditioning of the coefficient matrix.

In this article, to overcome this ill-conditioning conditioning problem in large domains and to get good accuracy in a stable structure, a new approach of the RBF method is constructed based on decomposition the domain to a few suitable subdomains. In this approach we use the finite-difference methods and employ the collocation method and approximate the solution directly by thin plate spline RBFs [5]. This structure without domain decomposition, used by Zerroukat et al. for solving heat transfer problem [6] and by Dehghan and Shokri for solving hyperbolic telegraph equation [2] and non-linear Klein-Gordon equation [7]. Here, we gain insight in one dimension before proceeding to higher dimensions. The implementation and complexity of RBF methods in higher dimensions are essentially the same as in one dimension.

The layout of the article is as follows: In Section 2 we show that how the RBFs is used to approximate the solution. In Section 3 we apply the proposed method on time dependent PDEs in linear and nonlinear cases. The results of numerical experiments are presented in Section 4. Section 5 is dedicated to a brief conclusion. The numerical results are obtained by using MATLAB programming.

RADIAL BASIS FUNCTION APPROXIMATION

In the interpolation of the scattered data using radial basis functions the approximation of a function u(x) at the centers $X = \{x_1,\ldots,x_N\}$, may be written as a linear combination of N RBFs; usually it takes the following form:

$$s_{u,X}(x) = \sum_{j=1}^{N} \alpha_j \phi\left(x - x_j\right) + \sum_{k=1}^{Q} \beta_k p_k(x) \qquad (2.1)$$

Here, Q denotes the dimension of the polynomial space $\pi_{m-1}(\mathbb{R}^d)$, $p_1,\ldots,p_Q$ denote a basis of $\pi_{m-1}(\mathbb{R}^d)$, $x = \{x_1, x_2,\ldots,x_d\}$, d is the dimension of the problem, α's and β's are coefficients to be determined, ϕ is the RBF. Some well-known RBFs are listed in Table 1.

To cope with additional degrees of freedom, the interpolation conditions

Corresponding Author: Dr. Syed Tauseef Mohyud-Din, HITEC University Taxila Cantt., Pakistan

Table 1: Some well-known functions that generate RBFs

Name of function	Definition
Multiquadrics (MQ)	$\phi(x)=\sqrt{\|x\|_2^2+c^2}$
Inverse multiquadrics (IMQ)	$\phi(x)=\left(\sqrt{\|x\|_2^2+c^2}\right)^{-1}$
Gaussian (GA)	$\phi(x)=\exp\left(-c\|x\|_2^2\right)$
Thin plate splines (TPS)	$\phi(x)=(-1)^{k+1}\|x\|_2^{2k}\log\|x\|_2$
Conical splines	$\phi(x)=\|x\|_2^{2k+1}$

$$s_{u,X}\left(x_j\right)=u\left(x_j\right),\qquad 1\le j\le N \tag{2.2}$$

are completed by the additional conditions

$$\sum_{j=1}^{N}\alpha_j p_k\left(x_j\right)=0,\qquad 1\le k\le Q \tag{2.3}$$

Solvability of this system is therefore equivalent to solvability of the system

$$\begin{pmatrix} A_{\phi,X} & P \\ P^T & 0\end{pmatrix}\begin{pmatrix}\alpha\\ \beta\end{pmatrix}=\begin{pmatrix}u|X\\ 0\end{pmatrix} \tag{2.4}$$

where

$$A_{\phi,X}=\left(\phi\left(x_j-x_k\right)\right)\in\mathbb{R}^{N\times N}\ \text{ and }\ P=\left(p_k\left(x_j\right)\right)\in\mathbb{R}^{N\times Q}$$

This last system is obviously solvable if the coefficient matrix on the left-hand side is invertible. Equation (2.1) can be written without the additional polynomial $\Sigma_{k=1}^{Q}\beta_k p_k(x)$. In that case, ϕ must be unconditionally positive definite to guarantee the solvability of the resulting system (e.g. Gaussian or inverse multiquadrics). However $\Sigma_{k=1}^{Q}\beta_k p_k(x)$ is usually required when ϕ is conditionally positive definite, i.e. when ϕ has a polynomial growth towards infinity. For instance, suppose ϕ is thin plate splines. Moreover, since these functions are globally supported, the interpolation matrix is full and may be very ill-conditioned for some RBFs.

In a similar representation as (2.1), for any linear partial differential operator $\mathcal{L}$, $\mathcal{L}u$ may be approximated by [8]

$$\mathcal{L}u(x)\simeq\sum_{j=1}^{N}\alpha_j\mathcal{L}\phi\left(x-x_j\right)+\sum_{k=1}^{Q}\beta_k\mathcal{L}p_k(x) \tag{2.5}$$

We use the thin plate spline RBFs in our method. The reason is that it has been shown by Franke [9], that MQ and thin plate spline give the most accurate results for scattered data approximations. Furthermore, the accuracy of the MQ method depends on a shape parameter and as yet there is no mathematical theory about how to choose its optimal value. Hence, most applications of the MQ use experimental tuning parameters or expensive optimization techniques to evaluate the optimum shape parameter [10]. While the thin plate spline method gives good agreement without requiring such additional parameters and is based on sound mathematical theory [11].

$$\phi(x)=(-1)^{k+1}\|x\|_2^{2k}\log\|x\|_2,\ k\in\mathbb{N}$$

from $\mathbb{R}^d$ to $\mathbb{R}$ that generates thin plate spline RBFs is conditionally positive definite of order m = k+1, [12]. Since ϕ is C^{2k-1} continuous, a higher-order thin plate spline must be used, for higher-order partial differential operators.

To avoid problems at x = 0 (since log(0) = -∞), we implement

$$\phi(x)=(-1)^3\|x\|_2^3\log\|x\|_2^{\|x\|_2}\ \text{for } k=2$$

Definition 1: The points

$$X=\{x_1,\ldots,x_N\}\subseteq\mathbb{R}^d\ \text{ with }\ N\ge Q=\dim\pi_m\left(\mathbb{R}^d\right)$$

are called $\pi_{m-1}(\mathbb{R}^d)$-unisolvent if the zero polynomial is the only polynomial from $\pi_m(\mathbb{R}^d)$ that vanishes on all of them.

Theorem 1: Suppose that ϕ is conditionally positive definite of order m and X is a $\pi_{m-1}(\mathbb{R}^d)$-unisolvent set of centers. Then the system (2.4) is uniquely solvable.

Proof: [12].

The numerical solution of PDEs by RBF methods is based on a scattered data interpolation problem which was reviewed in this section.

IMPLEMENTATION OF THE NEW APPROACH

Linear case: Telegraph equation: Let us consider the following hyperbolic telegraph equation [2]:

$$\frac{\partial^2 u}{\partial t^2}+\alpha\frac{\partial u}{\partial t}+\beta u=\frac{\partial^2 u}{\partial x^2}+f(x,t)$$

$$x\in\Omega=[a,b]\subset\mathbb{R},\quad 0< t\le T \tag{3.1.1}$$

with initial conditions

$$\begin{cases}u(x,0)=g_1(x), & x\in O\\ u_t(x,0)=g_2(x), & x\in O\end{cases} \tag{3.1.2}$$

and Dirichlet boundary condition

$$u(x,t)=h(x,t),\ x\in\partial O,\ 0<t\le T \tag{3.1.3}$$

where α and β are known constant coefficients, f, g_1, g_2 and h are known functions and the function u is unknown. In this method, we decompose Ω into a few subdomains

$$\Omega_1=\left[a_1,b_1\right],\ldots,\Omega_m=\left[a_m,b_m\right]$$

uniformly. Then, let us discretize (3.1.1) according to the following θ-weighted scheme

$$\frac{u(x,t+\delta t)-2u(x,t)+u(x,t-\delta t)}{(\delta t)^2}+\alpha\frac{u(x,t+\delta t)-u(x,t-\delta t)}{2\delta t}=\theta\left[\nabla^2u(x,t+\delta t)-\beta u(x,t+\delta t)\right]+(1-\theta)\left[\nabla^2u(x,t)-\beta u(x,t)\right]+f(x,t+\delta t) \tag{3.1.4}$$

where ∇ is the gradient differential operator, $0\le\theta\le1$ and δt is the time step size. Rearranging (3.1.4), by the notation $u^n=u(x,t^n)$ where $t^n=t^{n-1}+\delta t$, we obtain

$$\left(1+\frac{\alpha\delta t}{2}+\beta\theta(\delta t)^2\right)u^{n+1}-\theta(\delta t)^2\nabla^2u^{n+1}=\left(2-\beta(1-\theta)(\delta t)^2\right)u^n+(1-\theta)(\delta t)^2\nabla^2u^n+\left(\frac{\alpha\delta t}{2}-1\right)u^{n-1}+(\delta t)^2f^{n+1} \tag{3.1.5}$$

$u_k(x,t^n)$, i.e., solution in $x\in\Omega_k$ and $t=t^n$, can be approximated by

$$u_k^n(x)\simeq\sum_{j=1}^{N-3}\lambda_{k,j}^n\phi(x-x_{k,j})+\lambda_{k,N-2}^n x^2+\lambda_{k,N-1}^n x+\lambda_{k,N}^n,\quad k=1,\ldots,m \tag{3.1.6}$$

Let $\left\{x_{k,j}\right\}_{j=1}^{N-3},k=1,\ldots,m$

be a set of N-3 scattered nodes in Ω_k. Here, we take the Chebyshev-Gauss-Lobatto nodes in Ω_k as follows:

$$x_{k,j}=\frac{b_k-a_k}{2}\left(\cos\left(\frac{(j-1)\pi}{N-4}\right)\right)+\frac{b_k+a_k}{2},\ k=1,\ldots,m,\ j=1,\ldots,N-3 \tag{3.1.7}$$

Now, to determine the coefficients

$$\left(\lambda_{k,j}\right)_{j=1}^{N},k=1,\ldots,m$$

in each step, the collocation method may be used. For this reason, we put (3.1.6) into (3.1.5) in every subdomains Ω_k. Then by substituting collocation points $x_{k,j}$, $j=1,2,\ldots,N-4$ into obtained equation, we have

$$\begin{aligned}&\left(1+\frac{\alpha\delta t}{2}+\beta\theta(\delta t)^2\right)u_k\left(x_{k,j},t^{n+1}\right)-\theta(\delta t)^2\nabla^2u_k\left(x,t^{n+1}\right)|_{x=x_{k,j}}=\left(2-\beta(1-\theta)(\delta t)^2\right)u_k\left(x_{k,j},t^n\right)+\\&(1-\theta)(\delta t)^2\nabla^2u_k\left(x,t^n\right)|_{x=x_{k,j}}+\left(\frac{\alpha\delta t}{2}-1\right)u_k\left(x_{k,j},t^{n-1}\right)+(\delta t)^2f\left(x_{k,j},t^{n+1}\right),\quad \begin{array}{l}k=1,\ldots,m\\ j=2,\ldots,N-4\end{array}\end{aligned} \tag{3.1.8}$$

The additional conditions due to (2.3) are written as:

$$\sum_{j=1}^{N-3}\lambda_{k,j}^{n+1}=\sum_{j=1}^{N-3}\lambda_{k,j}^{n+1}x_{k,j}=\sum_{j=1}^{N-3}\lambda_{k,j}^{n+1}x_{k,j}^2=0,\quad k=1,\ldots,m \tag{3.1.9}$$

For $\alpha_1\in\Omega_1$ and $b_m\in\Omega_m$, we have the boundary conditions as follows:

$$\sum_{j=1}^{N-3}\lambda_{1,j}^{n+1}\phi\left(a_1-x_{1,j}\right)+\lambda_{1,N-2}^{n+1}a_1{}^2+\lambda_{1,N-1}^{n+1}a_1+\lambda_{1,N}^{n+1}=h\left(a_1,t^{n+1}\right) \tag{3.1.10}$$

$$\sum_{j=1}^{N-3}\lambda_{m,j}^{n+1}\phi\left(b_m-x_{1,j}\right)+\lambda_{m,N-2}^{n+1}b_m{}^2+\lambda_{m,N-1}^{n+1}b_m+\lambda_{m,N}^{n+1}=h\left(b_m,t^{n+1}\right) \tag{3.1.11}$$

To achieve a smooth solution on common boundary points from adjacent subdomains, the solutions in adjacent subdomains are imposed to reach the same amount and derivative in their common boundary points. Thus, we have

$$\sum_{j=1}^{N-3}\lambda_{i-1,j}^{n+1}\phi\left(a_i - x_{i-1,j}\right) + \lambda_{i-1,N-2}^{n+1}a_i^{\,2} + \lambda_{i-1,N-1}^{n+1}a_i + \lambda_{i-1,N}^{n+1}$$
$$= \sum_{j=1}^{N-3}\lambda_{i,j}^{n+1}\phi\left(a_i - x_{i-1,j}\right) + \lambda_{i,N-2}^{n+1}a_i^{\,2} + \lambda_{i,N-1}^{n+1}a_i + \lambda_{i,N}^{n+1}, i = 2,\ldots,m \tag{3.1.12}$$

$$\sum_{j=1}^{N-3}\lambda_{i-1,j}^{n+1}\frac{d}{dx}\phi\left(x - x_{i-1,j}\right)\big|_{x=a_i} + 2\lambda_{i-1,N-2}^{n+1}a_i + \lambda_{i-1,N-1}^{n+1}$$
$$= \sum_{j=1}^{N-3}\lambda_{i,j}^{n+1}\frac{d}{dx}\phi\left(x - x_{i-1,j}\right)\big|_{x=a_i} + 2\lambda_{i,N-2}^{n+1}a_i + \lambda_{i,N-1}^{n+1}, \quad i = 2,\ldots,m \tag{3.1.13}$$

Equations (3.1.8)-(3.1.13) lead to a linear system of $m\times N$ equations with $m\times N$ unknowns. We use the LU factorization to the coefficient matrix and use this factorization in our algorithm. For n = 0, the Eq. (3.1.8) has the following form

$$\left(1+\frac{\alpha\delta t}{2}+\beta\theta(\delta t)^2\right)u_k\left(x_{k,j},t^1\right)-\theta(\delta t)^2\nabla^2u_k\left(x,t^1\right)|_{x=x_{k,j}}=\left(2-\beta(1-\theta)(\delta t)^2\right)u_k\left(x_{k,j},t^0\right)+$$
$$(1-\theta)(\delta t)^2\nabla^2u_k\left(x,t^0\right)|_{x=x_{k,j}}+\left(\frac{\alpha\delta t}{2}-1\right)u_k\left(x_{k,j},t^{-1}\right)+(\delta t)^2 f\left(x_{k,j},t^1\right),\ \begin{matrix}k=1,\ldots,m\\ j=2,\ldots,N-4\end{matrix} \tag{3.1.14}$$

Through initial conditions, we know that:

$$u_k\left(x,t^0\right)=g_1(x), \qquad x\in O_k \tag{3.1.15}$$

To approximate $u_k(x_{k,j}, t^{-1})$ the second initial condition can be used. For this purpose we discretize the second initial condition as

$$\frac{u_k\left(x,t^1\right)-u_k\left(x,t^{-1}\right)}{2\delta t}=g_2(x), \qquad x\in O_k \tag{3.1.16}$$

Thus, we have

$$\left(1+\frac{\alpha\delta t}{2}+\beta\theta(\delta t)^2\right)u_k\left(x_{k,j},t^1\right)-\theta(\delta t)^2\nabla^2u_k\left(x,t^1\right)|_{x=x_{k,j}}=\left(2-\beta(1-\theta)(\delta t)^2\right)g_1\left(x_{k,j}\right)+$$
$$(1-\theta)(\delta t)^2\nabla^2g_1(x)|_{x=x_{k,j}}+\left(\frac{\alpha\delta t}{2}-1\right)\left(u_k\left(x_{k,j},t^1\right)-2(\delta t)g_2\left(x_{k,j}\right)\right)+(\delta t)^2 f\left(x_{k,j},t^1\right),\ \begin{matrix}k=1,\ldots,m\\ j=2,\ldots,N-4\end{matrix} \tag{3.1.17}$$

At other steps, no problems will be confronted and the solutions obtained from the two previous steps are used for the next step.

Nonlinear case: Klein-Gordon equation: Let us consider the following one-dimensional nonlinear Klein-Gordon equation [7]:

$$\frac{\partial^2u}{\partial t^2}+\alpha\frac{\partial^2u}{\partial x^2}+\beta u+\gamma u^k=f(x,t), \quad x\in O=[a,b]\subset\mathbb{R}, \quad 0< t\le T \tag{3.2.1}$$

with initial conditions

$$\begin{cases}u(x,0)=g_1(x), & x\in O\\ u_t(x,0)=g_2(x), & x\in O\end{cases} \tag{3.2.2}$$

and Dirichlet boundary condition

$$u(x,t) = h(x,t), \quad x \in \partial O, \quad 0 < t \le T \tag{3.2.3}$$

where α, β and γ are known constants. The f, g_1, g_2 and h are known functions and the function u is unknown. In the same way as mentioned in telegraph equation, we decompose Ω into a few subdomains

$$\Omega_1 = [a_1, b_1], \ldots, \Omega_m = [a_m, b_m]$$

uniformly and discretize (3.2.1) according to the following θ-weighted schem

$$\frac{u(x,t+\delta t) - 2u(x,t) + u(x,t-\delta t)}{(\delta t)^2} + \theta\left[\alpha \nabla^2 u(x,t+\delta t) + \beta u(x,t+\delta t)\right] + (1-\theta)\left[\alpha \nabla^2 u(x,t) + \beta u(x,t)\right] + \gamma\left(u(x,t)\right)^k = f(x,t+\delta t) \tag{3.2.4}$$

Rearranging (3.2.4), we obtain

$$\left(1+\beta\,\theta(\delta t)^2\right)u^{n+1} + \alpha\,\theta(\delta t)^2\nabla^2 u^{n+1} = \left(2-\beta(1-\theta)(\delta t)^2\right)u^n + \alpha(1-\theta)(\delta t)^2\nabla^2 u^n - \gamma(\delta t)^2\left(u^n\right)^k - u^{n-1} + (\delta t)^2 f^{n+1} \tag{3.2.5}$$

In every subdomains Ω_k, the collocation method is used by putting (3.1.6) into (3.2.5) and substituting collocation points $x_{k,j}$, $j = 2,\ldots,N\text{-}4$ into obtained equation. Thus, we have

$$\begin{aligned}&\left(1+\beta\,\theta(\delta t)^2\right)u_k\left(x_{k,j},t^{n+1}\right) + \alpha\,\theta(\delta t)^2\nabla^2 u_k\left(x,t^{n+1}\right)|_{x=x_{k,j}} = \left(2-\beta(1-\theta)(\delta t)^2\right)u_k\left(x_{k,j},t^n\right)\\ &+\alpha(1-\theta)(\delta t)^2\nabla^2 u_k\left(x,t^n\right)|_{x=x_{k,j}} - \gamma(\delta t)^2\left(u_k\left(x_{k,j},t^n\right)\right)^k - u_k\left(x_{k,j},t^{n-1}\right) + (\delta t)^2 f\left(x_{k,j},t^{n+1}\right), \quad \begin{matrix} k=1,\ldots,m \\ j=2,\ldots,N-4\end{matrix}\end{aligned} \tag{3.2.6}$$

Again, we have the additional conditions (3.1.9), the boundary conditions (3.1.10) and (3.1.11) and the equations (3.1.12) and (3.1.13). Therefore, equations (3.2.6), (3.1.9)-(3.1.13) lead to a system of linear equations, which is solved using the LU factorization to the coefficient matrix in order to determine the coefficients

$$\left(\lambda_{k,j}\right)_{j=1}^{N}, k = 1,\ldots,m$$

in each step. For n = 0 the Eq. (3.2.6) has the following form

$$\begin{aligned}&\left(1+\beta\,\theta(\delta t)^2\right)u_k\left(x_{k,j},t^1\right) + \alpha\,\theta(\delta t)^2\nabla^2 u_k\left(x,t^1\right)|_{x=x_{k,j}} = \left(2-\beta(1-\theta)(\delta t)^2\right)u_k\left(x_{k,j},t^0\right)\\ &+\alpha(1-\theta)(\delta t)^2\nabla^2 u_k\left(x,t^0\right)|_{x=x_{k,j}} - \gamma(\delta t)^2\left(u_k\left(x_{k,j},t^0\right)\right)^k - u_k\left(x_{k,j},t^{-1}\right) + (\delta t)^2 f\left(x_{k,j},t^1\right), \quad \begin{matrix} k=1,\ldots,m \\ j=2,\ldots,N-4\end{matrix}\end{aligned} \tag{3.2.7}$$

according to relations (3.1.15) and (3.1.16), we have

$$\begin{aligned}&\left(1+\beta\,\theta(\delta t)^2\right)u_k\left(x_{k,j},t^1\right) + \alpha\theta(\delta t)^2\nabla^2 u_k\left(x,t^1\right)|_{x=x_{k,j}} = \left(2-\beta(1-\theta)(\delta t)^2\right)g_1\left(x_{k,j}\right)\\ &+\alpha(1-\theta)(\delta t)^2\nabla^2 g_1(x)|_{x=x_{k,j}} - \gamma(\delta t)^2\left(g_1\left(x_{k,j}\right)\right)^k - \left(u_k\left(x_{k,j},t^1\right) - 2(\delta t)g_2\left(x_{k,j}\right)\right) + (\delta t)^2 f\left(x_{k,j},t^1\right), \quad \begin{matrix} k=1,\ldots,m \\ j=2,\ldots,N-4\end{matrix}\end{aligned} \tag{3.2.8}$$

At other steps, no problems will be confronted and the solutions obtained from the two previous steps are used for the next step.

Remark: Although equations (3.1.4) and (3.2.4) are valid for any value of $\theta \in [0,1]$, we will use $\theta = 1/2$ (the famous Crank-Nicolson scheme).

NUMERICAL RESULTS

In this section, we present some numerical results to test the efficiency of the new scheme for solving time dependent partial differential equations.

Example 1: Consider the hyperbolic telegraph Eq. (3.1.1) with $\alpha = 6$ and $\beta = 2$ in the interval Ω. The initial conditions are given by

$$\begin{cases} u(x,0) = g_1(x) = \sin(x), & x \in O \\ u_t(x,0) = g_2(x) = -\sin(x), & x \in O \end{cases}$$

and the analytical solution is given in [2] as

$$u(x,t) = \sin(x)\exp(-t)$$

In this case

$$f(x,t) = -2\exp(-t)\sin(x)$$

We extract the boundary function h(x,t) from the exact solution. We solve this equation in different computational domains (Table 2). The RMS and the maximum errors of the obtained numerical results are achieved at t = 1. The RMS and the maximum errors are defined as follow, respectively:

$$E^2 = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left|u(x_i) - s(x_i)\right|^2}$$

$$E^\infty = \max_{1 \le i \le N}\left|u(x_i) - s(x_i)\right|$$

where u(x) is the exact solution, s(x) is the approximate solution and $\{x_i\}_{i=1}^{N}$ are collocation points. In each row of the Table 2, we use the same number of collocation points. For example, in the first row, we use the algorithm without domain decomposition by 100 collocation points and with two subdomains by 50 collocation points in each subdomains.

The space-time graph of analytical solution is given in Fig. 1. We also demonstrate the space-time graph of absolute error with and without domain decomposition by the same number of collocation points in Fig. 2 and 3, respectively.

Example 2: Consider the nonlinear Klein-Gordon Eq. (3.2.1) with α = -1, (= 0. (=1 and k = 2 in the interval (. The initial conditions are given by

$$\begin{cases} u(x,0) = g_1(x) = 0, & x \in O \\ u_t(x,0) = g_2(x) = \frac{1}{5}\cos(x), & x \in O \end{cases}$$

The analytical solution is given as

Table 2: The RMS and the maximum errors, with and without domain decomposition, for example 1

Ω	dt	Number of subdomains	Number of collocation points in each subdomains	E^2	E^∞	Computational Time(s)
[0,60]	0.001	1	100	0.1526	0.4562	0.7
		2	50	1.7×10^{-4}	4.3×10^{-4}	0.7
[0,20]	0.001	1	160	0.0813	0.2305	1.3
		4	40	1.8×10^{-4}	2.7×10^{-4}	1.6
[0,30]	0.005	1	210	0.60	1.5	1.1
		6	35	9.3×10^{-4}	1.4×10^{-3}	1.6
[0,50]	0.0001	1	120	Fail	Fail	-
		4	30	8.6×10^{-5}	2.1×10^{-4}	6.8

Table 3: The RMS and the maximum errors, with and without domain decomposition, for example 2

Ω	dt	Number of subdomains	Number of collocation points in each subdomains	E^2	E^∞	Computational Time(s)
[0,50]	0.001	1	120	0.1350	0.3950	0.9
		2	60	2.4×10^{-5}	8.5×10^{-5}	0.8
[0,20]	0.001	1	180	0.9228	2.7188	1.6
		6	30	3.1×10^{-5}	5.5×10^{-5}	2.6
[0,80]	0.005	1	160	0.3429	0.9440	0.8
		4	40	1.1×10^{-4}	2.6×10^{-4}	1
[0,50]	0.0001	1	60	Fail	Fail	-
		2	30	3.6×10^{-4}	8.9×10^{-4}	2.6

Table 4: The RMS and the Max errors, with and without domain decomposition, for Example 2 when dt = 0.0001, Ω = [0, 50]

Number of subdomains	Number of collocation points in each subdomains	E^2	E^∞	Condition number
1	20	0.0071	0.0172	2.9×10^{15}
	30	0.0034	0.0061	4.3×10^{15}
	40	0.0034	0.0104	5.7×10^{15}
	50	0.0284	0.1006	7.1×10^{15}
2	30	3.6×10^{-4}	8.9×10^{-4}	3.4×10^{14}
4	30	4.5×10^{-5}	1.2×10^{-4}	1.6×10^{13}
8	30	4.9×10^{-6}	1.1×10^{-5}	2.3×10^{12}

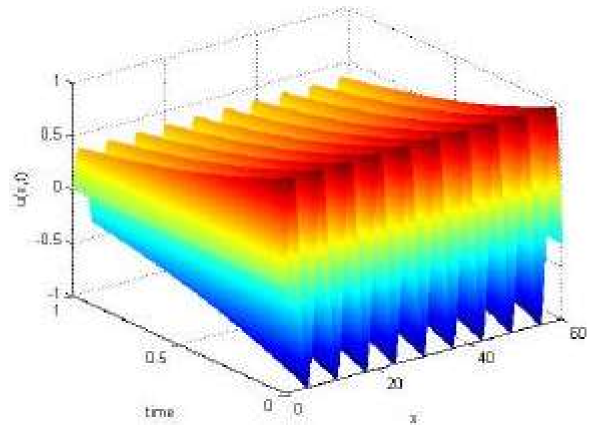

Fig. 1: Space-time graph of the analytical solution up to t = 1s, with dt = 0.01 and dx = 0.01, for Example 1

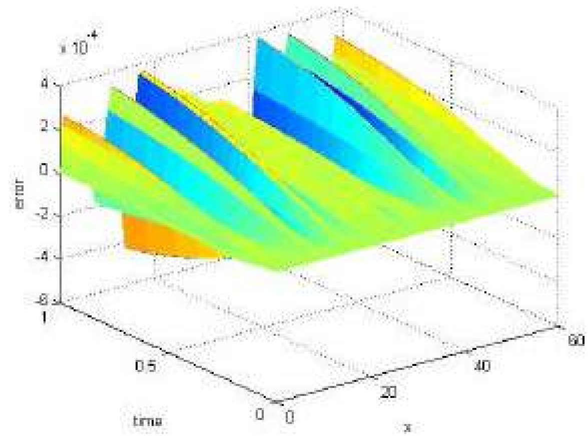

Fig. 2: Space-time graph of the absolute error up to t = 1s, with two subdomains (60 collocation points in each subdomain) and dt = 0.001, for Example 1

$$u(x,t)=\frac{1}{5}t\cos x$$

For this case we have

$$f(x,t)=\frac{t\cos x}{5}\left(1+\frac{1}{5}t\cos x\right)$$

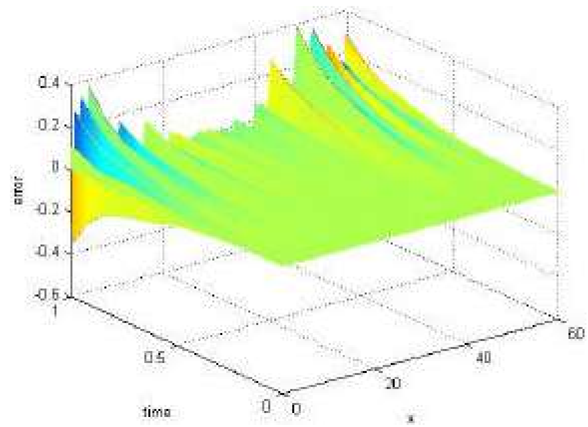

Fig. 3: Space-time graph of the absolute error up to t = 1s, with dt = 0.001 on 100 collocation points, without domain decomposition, for Example 1

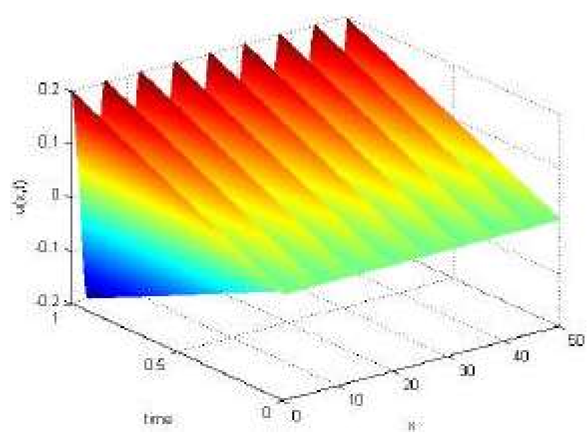

Fig. 4: Space-time graph of the analytical solution up to t = 1s, with dt = 0.01 and dx = 0.01, for Example 2

We extract the boundary function h(x,t) from the exact solution. Table 3 and 4, show the accuracy and efficiency of the proposed approach in different domains. As mentioned in Example 1, we use the same number of collocation points in each row of Table 3.

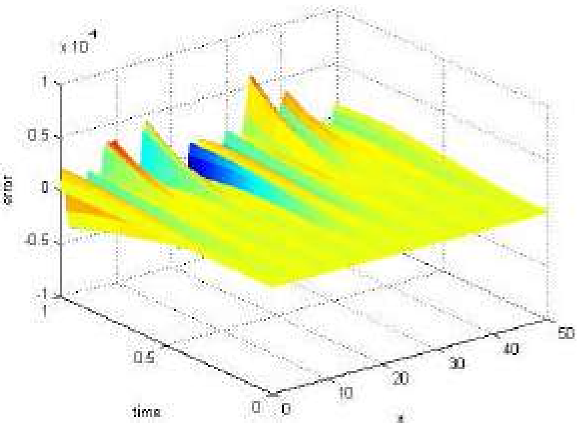

Fig. 5: Space-time graph of the absolute error up to t = 1s, with two subdomains (60 collocation points in each subdomain) and dt = 0.001, for Example 2

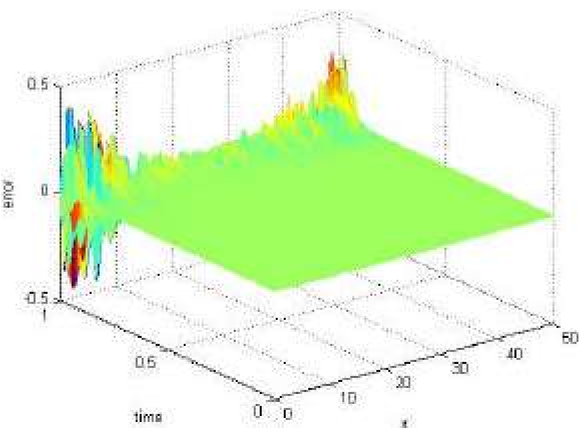

Fig. 6: Space-time graph of the absolute error up to t = 1s, with dt = 0.001 on *120* collocation points, without domain decomposition, for Example 2

The space-time graph of analytical solution is given in Fig. 4. We also demonstrate the space-time graph of absolute error with and without domain decomposition by the same number of collocation points in Fig. 5 and 6, respectively.

CONCLUSION

A major drawback of the RBF collocation method is that a large number of collocation points are required in order to obtain a desirable accuracy on large domains. This hinders the application of the RBFs to solve large domain problems due to severe ill-conditioning of the coefficient matrix. To overcome this difficulty, the domain was decomposed to a few suitable subdomains. For instance, the linear hyperbolic telegraph equation and the nonlinear Klein-Gordon equation were solved in different domains to demonstrate the effectiveness of the new approach. The numerical results showed the high accuracy of the proposed scheme in this research in comparison with the classical method lacking domain decomposition. Extension of the proposed method to solve time dependent PDEs in high dimensions is the subject of a future research work.

REFERENCES

1. Dehghan, M. and A. Shokri, 2007. A numerical method for two-dimensional Schrödinger equation using collocation and radial basis functions. Comput. Math. Appli., 54: 136-146.
2. Dehghan, M. and A. Shokri, 2008. A numerical method for solving the hyperbolic telegraph equation. Numer. Methods Partial Differential Eq., 24: 1080-1093.
3. Chinchapatnam, P.P., K. Djidjeli and P.B. Nair, 2007. Domain decomposition for time-dependent problems using radial based meshless methods. Numer. Methods Partial Differential Eq., 23: 38-59.
4. Sarra, S.A., 2005. Adaptive radial basis function methods for time dependent partial differential equations. Appl. Numer. Math., 54: 79-94.
5. Kansa, E.J., 1990. Multiquadrics-A scattered data approximation scheme with applications to computational fluid dynamics-I. Comput. Math. Appl., 19: 127-145.
6. Zerroukat, M., H. Power and C. S. Chen, 1998. A numerical method for heat transfer problems using collocation and radial basis functions. Int. J. Numer. Meth. Eng. 42: 1263-1278
7. Dehghan, M. and A. Shokri, 2009. Numerical solution of the nonlinear Klein-Gordon equation using radial basis functions. J. Comput. Appl. Math., 230: 400-410.
8. Dehghan, M. and A. Shokri, 2009. A meshless method for numerical solution of the one-dimensional wave equation with an integral condition using radial basis functions. Numer. Algor., 52: 461-477.
9. Franke, R., 1982. Scattered data interpolation: Test of some methods. Math. Comput., 38: 181-200.
10. Carlson, R.E. and T. Foley, 1991. The parameter R^2 in multiquatric interpolation. Comput. Math. Appl., 21: 29-42.
11. Boztosun, I. and A. Charafi, 2002. An analysis of the linear advection-diffusion equation using mesh-free and mesh-dependent methods. Engineering Analysis with Boundary Elements, 26: 889-895.
12. Wendland, H., 2005. Scattered data approximation. Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, Cambridge.