

A Comparative Study of Global and Local Meshless Methods for Diffusion-Reaction Equation

Guangming Yao¹, Siraj-ul-Islam², and Božidar Šarler²

Abstract: This paper focuses on the comparative study of global and local meshless methods based on collocation with radial basis functions for solving two dimensional initial boundary value diffusion-reaction problem with Dirichlet and Neumann boundary conditions. A similar study was performed for the boundary value problem with Laplace equation by Lee, Liu, and Fan (2003). In both global and local methods discussed, the time discretization is performed in explicit and implicit way and the multiquadric radial basis functions (RBFs) are used to interpolate diffusion-reaction variable and its spatial derivatives. Five and nine noded sub-domains are used in the local support of the local method. Uniform and non-uniform space discretization is used. Accuracy of global and local approaches is assessed as a function of the time and space discretizations, and value of the shape parameter. One can observe the convergence with denser nodes and with smaller time-steps in both methods. The global method is prone to instability due to ill-conditioning of the collocation matrix with the increase of the number of the nodes in cases $N \approx 3000$. On the other hand, the global method is more stable with respect to the time-step length. Numerical tests with and without noise are conducted based on the methodology proposed in Younga, Fana, Hua, and Atluri (2009). The results show larger stability of the local versions of the method in comparison with the global ones. The accuracy of the local method is comparable with the accuracy of the global method. The local method is more efficient because we solve only a small system of equations for each node in explicit case and a sparse system of equations in implicit case. Hence the local method represents a preferable choice to its global counter part.

Keywords: Diffusion-Reaction Equation, GRBFCM, LRBFCM, Meshless Methods, Collocation.

¹ IMPOL Aluminium Industry, Slovenska Bistrica, Slovenia

² Laboratory for Multiphase Processes, University of Nova Gorica, Slovenia

1 Introduction

The use of radial basis functions (RBFs) in the numerical solution of partial differential equations (PDEs) has gained popularity in the engineering and science community as it is meshless and can readily be extended to multidimensional problems. The motivation for the comparative study of the Global Radial Basis Function Collocation Method (GRBFCM) with the Local Radial Basis Function Collocation Method (LRBFCM) is to broaden understanding of the behavior of meshless methods in various applications keeping in view the difficulties that may occur during their use. In recent years RBFs have been extensively used in different context Buhmann (2003); Atluri (2004); Atluri and Shen (2002); Atluri, Liu, and Han (2006) and emerged as a potential alternative in the field of numerical solution of PDEs.

The radial basis functions interpolation was introduced in Hardy (1971), to approximate two-dimensional geographical surfaces based on scattered data. Kansa (1990) derived GRBFCM based on multiquadrics RBFs, for the meshless numerical solution of PDEs. This idea was extended later on by Golberg, Chen, and Karur (1996). The existence, uniqueness, and convergence of the RBFs approximation was discussed by Micchelli (1986), Madych and Nelson (1990), and Franke and Schaback (1998). The importance of shape parameter c in the MQ method was elaborated by Tarwater (1985). Micchelli (1986) has proved that for distinct interpolation points, the system of equations obtained by this method is always solvable. The authors Siraj-ul-Islam, Haq, and Ali (2009); Siraj-ul-Islam, Haq, and Uddin (2009); Ali, Siraj-ul-Islam, and Haq (2009) have very recently used the GRBFCM to obtain meshless numerical solution of the nonlinear coupled PDEs.

Contrary to the mesh based methods like the FEM, FVM and FDM, meshless methods use a set of uniform or random points which are not necessarily interconnected in the form of a mesh. Due to this advantageous feature, meshless methods have got increased prominence since mesh generation in multi-dimensional problems is a non-trivial task. The benefits of meshless approximation by the RBFs is somehow over shadowed by the dense and ill-conditioned matrix, especially in the large scale simulations. The non-singularity of the RBF's interpolation matrix depends on the shape parameter and the size of the domain. In the GRBFCM the collocation matrix is constructed by taking into consideration the whole domain. This limits the applicability of the GRBFCM to solve large scale problems. Many remedies like point collocation, local symmetric weak form and local boundary-integral-equation formulation, domain decomposition by Mai-Duy and Tran-Cong (2002), multi-grid approach and compactly supported RBFs by Chen, Ganesh, Golberg, and Cheng (2002) have been suggested in the literature to circumvent this problem. These approaches result in a substantial complication of the original simple method on one hand with a very limited advantages on the other hand. Var-

ious localized meshless methods such as Lee, Liu, and Fan (2003); Liu (2003); Shu, Ding, and Yeo (2003); Šarler and Vertnik (2006) have been successfully used in many practical problems for localization of the domain size while maintaining simplicity of the RBF approach. The LRBFCM was first introduced for diffusion problems in Šarler and Vertnik (2006). The results in the paper show accuracy and efficiency. Many authors have applied the LRBFCM to more complex problems such as convection-diffusion problems with phase-change Vertnik and Šarler (2006), continuous casting Vertnik, Založnik, and Šarler (2006), solid-solid phase transformations Kovačević and Šarler (2005), heat transfer and fluid flow Šarler (2005), Navier Stokes equations Divo and Kassab (2007), etc. The main idea of the LRBFCM is the collocation on overlapping sub-domains of the whole domain. The overlapping sub-domains drastically reduce the collocation matrix size at the expense of solving many small matrices with the dimension of the number of nodes included in the domain of influence for each node instead of a large collocation matrix. Circular and rectangular domains are most commonly used in the literature which can either be overlapping or non-overlapping. Detailed discussions on meshless methods and their applications to many complex PDEs, industrial and large-scale problems can be found in Atluri (2004), Fasshauer (2008), Kovačević and Šarler (2005); Liu (2003); Vertnik, Založnik, and Šarler (2006); Lorbiecka, Vertnik, Gjerkeš, Manojlovič, Senčič, Cesar, and Šarler (2009); Šarler, Kosec, Lorbiecka, and Vertnik (2010); Kosec and Šarler (2009); Vertnik and Šarler (2009); Kosec and Šarler (2010) and the references therein.

The explicit and implicit, global and local radial basis function collocation methods are considered in this paper. These methods are henceforth abbreviated as GRBFECM (Global Radial Basis Function Explicit Collocation Method), GRBFNCM (Global Radial Basis Function Crank-Nicolson Collocation Method), GRBFICM (Global Radial Basis Function Implicit Collocation Method), LRBFECEM (Local Radial Basis Function Explicit Collocation Method), and LRBFIEM (Local Radial Basis Functions Implicit Collocation Method). The structure of the present paper is organized as follows. In Section 2, we introduce the governing equations considered in this paper. In Section 3, we discuss the time discretization. In Section 4, the global and local meshless methods, the GRBFECM and LRBFCM are described. Section 5 is devoted to the numerical tests of both methods on the diffusion reaction PDEs with Dirichlet and Neumann's boundary conditions. At the end, we draw conclusions of the study.

2 Governing Equations

Consider the dimensionless form of two dimensional diffusion-reaction equation defined on domain Ω with boundary Γ

$$\frac{\partial u(\mathbf{x}, t)}{\partial t} = \mathcal{L}[u(\mathbf{x}, t)] + \kappa u(\mathbf{x}, t) + g(\mathbf{x}, t), \quad (1)$$

with initial condition

$$u(\mathbf{x}, t_0) = u_0, \mathbf{x} \in \Omega \cup \Gamma, \quad (2)$$

and Dirichlet or Neumann boundary conditions

$$\mathcal{B}[u(\mathbf{x}, t)] = f(\mathbf{x}, t), \quad t_0 + \Delta t \geq t \geq t_0, \mathbf{x} \in \Gamma, \quad (3)$$

where $u, t, \mathbf{x} = [x, y]^tr$ are the diffusion, time and space variables respectively, tr represents the transpose, g and f are the known functions of \mathbf{x} and t , $\Gamma = \Gamma_D + \Gamma_N$. Γ_N and Γ_D are Neumann and Dirichlet parts at the boundaries, κ is a real constant, \mathcal{L} is differential operator consisting of first and second-order derivatives of space variables and \mathcal{B} is the first-order differential operator with respect to space variables in the case of the Neumann boundary conditions and is identity operator in the case of the Dirichlet boundary conditions.

3 Time Discretization

Let Δt be the time step size, and $t = t^{(n)} = t_0 + n\Delta t$ be the time discretization. The time derivative in Eq.(1) is approximated by Euler formula in the following manner

$$\frac{\partial u(\mathbf{x}, t)}{\partial t} = \frac{u(\mathbf{x}, t) - u(\mathbf{x}, t_0)}{\Delta t}. \quad (4)$$

The parameter θ is used in the time discretization of Eq. (1) as

$$u(\mathbf{x}, t_0 + \theta \Delta t) \approx \theta u(\mathbf{x}, t_0 + \Delta t) + (1 - \theta)u(\mathbf{x}, t_0), \quad (5)$$

$$g(\mathbf{x}, t_0 + \theta \Delta t) \approx \theta g(\mathbf{x}, t_0 + \Delta t) + (1 - \theta)g(\mathbf{x}, t_0), \quad (6)$$

$$\mathcal{L}[u(\mathbf{x}, t_0 + \theta \Delta t)] \approx \theta \mathcal{L}[u(\mathbf{x}, t_0 + \Delta t)] + (1 - \theta)\mathcal{L}[u(\mathbf{x}, t_0)]. \quad (7)$$

Then $u(\mathbf{x}, t)$ in Eq. (1) can be approximated as

$$(1 - \kappa \Delta t)u - \Delta t \mathcal{L}u - \Delta t g = u_0 + \Delta t \mathcal{L}u_0 + \Delta t g_0, \quad \text{for } x \in \Omega \quad (8)$$

where $u = u(\mathbf{x}, t_0 + \theta \Delta t)$, $u_0 = u(\mathbf{x}, t_0)$, $g = g(\mathbf{x}, t_0 + \theta \Delta t)$, $g_0 = g(\mathbf{x}, t_0)$, $\mathcal{L}u = \mathcal{L}u(\mathbf{x}, t_0 + \theta \Delta t)$, $\mathcal{L}u_0 = \mathcal{L}u(\mathbf{x}, t_0)$, such that t_0 , u_0 and g_0 will be updated in

each time iteration. This is commonly used two-level time stepping strategy that approximates $u(\mathbf{x}, t)$ in Ω from the values given at the initial time t_0 and further time $t_0 + \Delta t$. Note that Eq. (8) reduces to the explicit, Crank-Nicolson and fully implicit method for $\theta = 0, 0.5, 1$, respectively. In the global cases derived in this paper, we take $\theta = 0, 0.5$ and 1 whereas in the local RBFs approximation we choose $\theta = 0, 1$.

4 Space discretization

Let $\{\mathbf{x}_i\}_1^N \in \Omega \cup \Gamma$ be the space discretization where $N = N_\Omega \cup N_\Gamma$ and N denotes the total number of points, N_Ω denotes number of the interior points and N_Γ denotes the number of boundary points. An introduction of the GRBFICM, GRBFECM, LRBFIEM and LRBFEEM is given in Section 4.1 and 4.2.

4.1 GRBFECM

In this approach, the formulation of the problem starts with the representation of u with RBFs on the entire domain. The derivatives are then calculated by differentiation of the RBF representation. The RBFs approximation for $u(\mathbf{x}, t)$ is given in the following form

$$u(\mathbf{x}, t) = \sum_{l=1}^N \phi(\|\mathbf{x} - \mathbf{x}_l\|) \alpha_l(t), \quad \mathbf{x} \in \Omega, \tag{9}$$

where $\alpha_l(t), l = 1, 2, \dots, N$ are the real RBFs coefficients. MQ RBFs are defined as

$$\phi(\|\mathbf{x} - \mathbf{x}_l\|) = \sqrt{\|\mathbf{x} - \mathbf{x}_l\|^2 + c^2} \tag{10}$$

where c is the shape parameter. RBFs approximation for the derivatives of $u(\mathbf{x}, t)$ can be represented by

$$\mathcal{L}u(\mathbf{x}, t) = \sum_{l=1}^N \mathcal{L}\phi(\|\mathbf{x} - \mathbf{x}_l\|) \alpha_l(t). \tag{11}$$

The coefficients $\alpha_l(t), l = 1, 2, \dots, N$ can be found from collocation as,

$$\begin{bmatrix} u(\mathbf{x}_1, t) \\ u(\mathbf{x}_2, t) \\ \vdots \\ u(\mathbf{x}_N, t) \end{bmatrix} = \begin{bmatrix} \phi(\|\mathbf{x}_1 - \mathbf{x}_1\|) & \phi(\|\mathbf{x}_1 - \mathbf{x}_2\|) & \dots & \phi(\|\mathbf{x}_1 - \mathbf{x}_N\|) \\ \phi(\|\mathbf{x}_2 - \mathbf{x}_1\|) & \phi(\|\mathbf{x}_2 - \mathbf{x}_2\|) & \dots & \phi(\|\mathbf{x}_2 - \mathbf{x}_N\|) \\ \vdots & \vdots & \vdots & \vdots \\ \phi(\|\mathbf{x}_N - \mathbf{x}_1\|) & \phi(\|\mathbf{x}_N - \mathbf{x}_2\|) & \dots & \phi(\|\mathbf{x}_N - \mathbf{x}_N\|) \end{bmatrix} \begin{bmatrix} \alpha_1(t) \\ \alpha_2(t) \\ \vdots \\ \alpha_N(t) \end{bmatrix}. \tag{12}$$

This can be written in the matrix notation as

$$\mathbf{u} = \Phi \alpha, \tag{13}$$

where $\mathbf{u} = [u(\mathbf{x}_1, t), u(\mathbf{x}_2, t), \dots, u(\mathbf{x}_N, t)]^{tr}$, $\alpha = [\alpha_1(t), \alpha_2(t), \dots, \alpha_N(t)]^{tr}$ and $\Phi_{sl} = \phi(\|\mathbf{x}_s - \mathbf{x}_l\|)$ is the matrix element of $N \times N$ matrix Φ . To represent the approximate solution of Eqs. (1)-(3) in a single equation at the interior and boundary points, we define the following domain and boundary indicators:

$$\gamma_{\Omega}^l = \begin{cases} 1, & x_l \in \Omega, \\ 0, & x_l \notin \Omega, \end{cases} \quad \gamma_{\Gamma}^l = \begin{cases} 1, & x_l \in \Gamma, \\ 0, & x_l \notin \Gamma, \end{cases} \tag{14}$$

where $l = 1, 2, \dots, N$. Using Eqs. (9)-(10) in Eqs. (3) and (8) we get

$$\begin{aligned} & \gamma_{\Omega}^j \left[(1 - \kappa \Delta t) \sum_{l=1}^N \phi(\|x_j - x_l\|) \alpha_l - \theta \Delta t \sum_{l=1}^N \mathcal{L} \phi(\|x_j - x_l\|) \alpha_l - \Delta t g(x_j, t) \right] \\ & + \gamma_{\Gamma}^j \sum_{l=1}^N \mathcal{B} \phi(\|x_j - x_l\|) \alpha_l - \gamma_{\Gamma}^j f(x_j, t) = \\ & \gamma_{\Omega}^j \left[u(x_j, t_0) + \Delta t (1 - \theta) \sum_{l=1}^n \mathcal{L} \phi(\|x_j - x_l\|) \alpha_{0l} + \Delta t g(x_j, t_0) \right] + \gamma_{\Gamma}^j f(x_j, t_0) \end{aligned} \tag{15}$$

for $j = 1, \dots, N$, and $0 \leq \theta \leq 1$ and α_{0l} (where α_{0l} is value of α_l at t_0) will be updated in each of the time iteration. The above linear system of equations can be written in matrix notation as

$$\Phi \alpha = \mathbf{b}, \tag{16}$$

where $\alpha = [\alpha_1(t), \alpha_2(t), \dots, \alpha_N(t)]^{tr}$, $\mathbf{b} = [b_1, b_2, \dots, b_N]^{tr}$. We determine the coefficients α by inverting Φ

$$\alpha = \Phi^{-1} \mathbf{b} \tag{17}$$

which implies that

$$\alpha_s = \sum_{l=1}^N \Phi_{sl}^{-1} b_l, \quad s = 1, 2, \dots, N, \tag{18}$$

where Φ_{sl}^{-1} denotes the matrix element of the matrix Φ^{-1} . The matrices Φ and \mathbf{b} are defined as,

$$\begin{aligned} \Phi_{sl} &= \gamma_{\Omega}^s [(1 - \kappa \Delta t) \phi(\|x_s - x_l\|) \alpha_l - \theta \Delta t \mathcal{L} \phi(\|x_s - x_l\|) \alpha_l - \Delta t g(x_s, t)] \\ &+ \gamma_{\Gamma}^s \mathcal{B} \phi(\|x_s - x_l\|) \alpha_l - \gamma_{\Gamma}^s f(x_l, t), \\ b_s &= \gamma_{\Omega}^s [u(x_s, t_0) + \Delta t (1 - \theta) \phi(\|x_s - x_l\|) \alpha_{0l} + \Delta t g(x_s, t_0)] + \gamma_{\Gamma}^s f(x_s, t_0), \end{aligned}$$

where $s, l = 1, 2, \dots, N$. The diffusion-reaction variable u , at the next time $t_0 + \Delta t$ and position x , can be approximated in the following form,

$$u(\mathbf{x}, t) = \sum_{l=1}^N \phi(\|\mathbf{x} - \mathbf{x}_l\|) \alpha_l, \quad \mathbf{x} \in \Omega. \tag{19}$$

4.2 LRBFCM

In order to reduce the size of the dense matrices arising from the global scheme, we use a local meshless scheme instead of the GRBFCM as discussed in the last section. For each $\mathbf{x}_j \in \Omega \cup \Gamma$, $j = 1, 2, \dots, N$ we choose m nearest neighboring points contained in the sub-domain ${}_j\Omega = \{\mathbf{x}_l\}_{l=1}^m$, where l denotes the local indexing for each collocation point x_j being center of ${}_j\Omega$ instead of the whole domain. The schematic diagram of 11×11 uniform node distribution, and a number of sub-domains of influence containing five points, i.e. $m = 5$, at the interior, near the boundary and corner points are shown in Fig. 1(left) and the overlapping sub-domains are shown in Fig. 1(right). To approximate $u(\mathbf{x}, t)$ through the

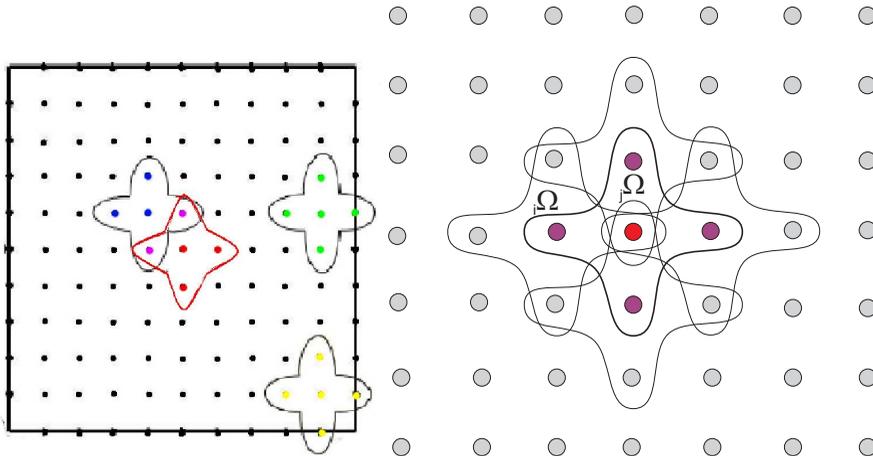


Figure 1: (left) Scheme of 11×11 uniform nodes, sub-domains of influence in the interior, near boundary and corner points using $m = 5$, (right) Scheme of overlapping sub-domains.

LRBFECM, consider collocation on the sub-domain ${}_j\Omega = \{\mathbf{x}_l\}_{l=1}^m$, $j = 1, \dots, N$ instead of the whole domain Ω . The diffusion variable u can be approximated on

each sub-domain in the following form,

$$u(j\mathbf{x}, t) = \sum_{l=1}^m \phi(\|j\mathbf{x} - j\mathbf{x}_l\|)_j \alpha_l, \quad j\mathbf{x} \in j\Omega. \quad (20)$$

It follows that for $j = 1, 2, \dots, N$,

$$\mathcal{L}u(j\mathbf{x}, t) = \sum_{l=1}^m \mathcal{L}\phi(\|j\mathbf{x} - j\mathbf{x}_l\|)_j \alpha_l, \quad (21)$$

$$\mathcal{B}u(j\mathbf{x}, t) = \sum_{l=1}^m \mathcal{B}\phi(\|j\mathbf{x} - j\mathbf{x}_l\|)_j \alpha_l = f(j\mathbf{x}, t). \quad (22)$$

The coefficients ${}_j\alpha_l$ are determined by collocation in the following form

$$\begin{aligned} & \gamma_{\Omega}^s \sum_{l=1}^m \phi(\|j\mathbf{x}_s - j\mathbf{x}_l\|)_j \alpha_l + \gamma_{\Gamma}^s \sum_{l=1}^m \mathcal{B}_i \phi(\|j\mathbf{x}_s - j\mathbf{x}_l\|)_j \alpha_l \\ & = \gamma_{\Omega}^s [u(j\mathbf{x}_s, t_0) + g(j\mathbf{x}_s, t_0)] + \gamma_{\Gamma}^s f(j\mathbf{x}_s, t_0), \end{aligned} \quad (23)$$

where $s = 1, 2, \dots, m$, γ_{Ω}^s and γ_{Γ}^s are defined in the last section. The above linear system can be written in matrix notation as

$${}_j\Phi_j \alpha = {}_j\mathbf{b}, \quad (24)$$

where ${}_j\alpha = [{}_j\alpha_1, {}_j\alpha_2, \dots, {}_j\alpha_m]^{tr}$, ${}_j\mathbf{b} = [{}_j\mathbf{b}_1, {}_j\mathbf{b}_2, \dots, {}_j\mathbf{b}_m]^{tr}$ is the right-hand side of Eq. (23). The matrices ${}_j\Phi$ and ${}_j\mathbf{b}$ are defined as

$${}_j\mathbf{b}_s = \gamma_{\Omega}^s (u(j\mathbf{x}_s, t_0) + g(j\mathbf{x}_s, t_0)) + \gamma_{\Gamma}^s f(j\mathbf{x}_s, t_0),$$

$${}_j\Phi_{sl} = \gamma_{\Omega}^s \phi(\|j\mathbf{x}_s - j\mathbf{x}_l\|) + \gamma_{\Gamma}^s \mathcal{B}_i \phi(\|j\mathbf{x}_s - j\mathbf{x}_l\|), \quad s, l = 1, 2, \dots, m,$$

where ${}_j\Phi = [{}_j\Phi_{sl}] \in \mathbb{R}^{m \times m}$. We determine the coefficients ${}_j\alpha$ by inverting ${}_j\Phi$

$${}_j\alpha = {}_j\Phi^{-1} {}_j\mathbf{b} \quad (25)$$

which implies that

$${}_j\alpha_s = \sum_{l=1}^m {}_j\Phi_{sl}^{-1} {}_j\mathbf{b}_l, \quad s = 1, 2, \dots, m, \quad (26)$$

where ${}_j\Phi_{sl}^{-1}$ denotes the matrix element of the matrix ${}_j\Phi^{-1}$. The diffusion-reaction variable u can be approximated for $\theta = 0$ at interior point \mathbf{x}_j by using Eq. (21) in Eqs. (8) in the following form:

$$u(\mathbf{x}_j, t) = \frac{u(\mathbf{x}_j, t_0) + \Delta t g(\mathbf{x}_j, t_0) + \Delta t \sum_{s=1}^m \mathcal{L}\phi(\|\mathbf{x}_j - j\mathbf{x}_s\|) \sum_{l=1}^m {}_j\Phi_{sl}^{-1} {}_j\mathbf{b}_l}{1 - \kappa \Delta t}. \quad (27)$$

For boundary point \mathbf{x}_j , from Eq. (20) we have

$$u(\mathbf{x}_j, t) = \sum_{s=1}^m \phi(\|\mathbf{x}_j - \mathbf{x}_s\|) \sum_{l=1}^m j\Phi_{sl}^{-1} j b_l. \quad (28)$$

This completes the formulation of the LRBFEEM. The formulations are similar for the GRBFEEM and LRBFEEM but the global method involves inversion of a full matrix during one time step, but the local method inverts N small $m \times m$ matrices through entire time period considered.

In the case of LRBFEEM, we use Eqs. (8), (20)–(22) to get RBFs approximation corresponding to $\theta = 1$ at nodal points $\mathbf{x}_j \in \Omega, j = 1, \dots, N$ in the following form

$$\begin{aligned} & \gamma_{\Omega}^j \left[(1 - \kappa \Delta t) u(\mathbf{x}_j, t) - \Delta t \sum_{s=1}^m \mathcal{L} \phi(\|\mathbf{x}_j - \mathbf{x}_s\|) \sum_{l=1}^m j\Phi_{sl}^{-1} u(j\mathbf{x}_l, t) \right] \\ & + \gamma_{\Gamma}^j \sum_{l=1}^N \mathcal{B} \phi(\|\mathbf{x}_j - \mathbf{x}_s\|) \sum_{l=1}^m j\Phi_{sl}^{-1} u(j\mathbf{x}_l, t) \\ & = \gamma_{\Omega}^j [u(\mathbf{x}_j, t_0) + \Delta t g(\mathbf{x}_j, t)] + \gamma_{\Gamma}^j f(\mathbf{x}_j, t) \end{aligned} \quad (29)$$

Note that $\{\{j\mathbf{x}_l\}_{l=1}^m, j = 1, 2, \dots, N\} = \{\mathbf{x}_j\}_{j=1}^N$, Eq. (29) leads to a linear system of N equations with N unknown $\{u(\mathbf{x}_j, t)\}_{j=1}^N$ with all entries in each row are equal to zero except those related to the sub-domain. This leads to $N \times N$ sparse system which can be solved by an efficient sparse matrix solver.

4.3 Continuity of the methods

In the global methods (GRBFEEM), the global continuity of function and its derivatives is assured. However, in the local methods (LRBFEEM), the global continuity of the function is assured but not of its derivatives. The overlapping sub-domains $j\Omega$ cover the whole domain Ω such that $\Omega = \cup_j \Omega$ and $\cap_j \Omega \neq \emptyset, j = 1, 2 \dots N$. The continuity of the LRBFEEM at the same nodal points that belong to different sub-domains behaves as

$$u_i(\mathbf{x}_k, t) = u_j(\mathbf{x}_k, t) \text{ if } \mathbf{x}_k \in i\Omega \cap j\Omega \text{ for } i \neq j = 1, 2, \dots, N. \quad (30)$$

$$\mathcal{L} [u_i(\mathbf{x}_k, t)] \neq \mathcal{L} [u_j(\mathbf{x}_k, t)] \text{ if } \mathbf{x}_k \in i\Omega \cap j\Omega \text{ for } i \neq j = 1, 2, \dots, N. \quad (31)$$

Function values in the points which belong to several sub-domains are the same. However, the derivative at the point depends on the sub-domain from which the RBFs coefficients are calculated.

5 Numerical results

Throughout this section, we investigate the performance of the global and local RBFCM that can both be implemented on evenly or randomly distributed nodes. The computer program has been coded in MATLAB. Three kinds of errors, absolute errors, maximum absolute error and root mean squared error

$$L_{abs} = |u(\mathbf{x}_j, t) - \hat{u}(\mathbf{x}_j, t)|, \quad j = 1, 2, \dots, N,$$

$$L_{\infty} = \max L_{abs},$$

$$L_{rms} = \left[\frac{1}{N} \sum_{j=1}^N |u(\mathbf{x}_j, t) - \hat{u}(\mathbf{x}_j, t)|^2 \right]^{1/2}$$

are considered in this paper. $u(\mathbf{x}_j, t)$ and $\hat{u}(\mathbf{x}_j, t)$ in the above equations represent exact and numerical solutions of the given partial differential equation, respectively. The random nodes are generated from the uniform nodes through the following transformation

$$\mathbf{x}_j = \mathbf{x}_j + c_{rand} \eta r_{min} \quad (32)$$

where \mathbf{x}_j is coordinate of node $\mathbf{x}_j = (x_j, y_j)$, c_{rand} is a random number between 0 to 1, r_{min} denotes the minimum distance among different uniform points, η stands for a displacement factor. The $\eta = 0.15$ is chosen in this paper, the uniform nodes are used if not stated otherwise. A scaling technique similar to the one introduced in Šarler and Vertnik (2006) is used to alleviate the difficulty of choosing different values of shape parameter in MQ RBFs. The scaling parameter r_0 is the maximum nodal distance in the sub-domain

$$r_0 = \max_{j=1}^N \max_{k=1}^m \sqrt{\|\mathbf{x}_j - \mathbf{x}_k\|^2}. \quad (33)$$

The parameter c in all RBFs and corresponding derivatives are replaced by cr_0 . Hence, a large shape parameter of the MQ RBF can be used in the numerical implementation. Scaling of the shape parameter is performed to make Multi-quadratic (MQ) RBFs approximation insensitive to various dimensions of the domain. Thus, the LRBFM is less sensitive with respect to the shape parameter unlike the GRBFM. The number of nodes in each sub-domain is chosen as $m = 5$ and $m = 9$ in one case in the LBRFICM. For numerical validation, the following following examples are considered.

5.1 Numerical Results for GRBFM

In this section we investigate the performance of the GRBFM in explicit, implicit and Crank-Nicolson's implementation. The method is applied on two benchmark problems Jang (2007) (Ex. 1), Nie, Wana, Zhang, and Liu (2008) (Ex. 2) with

Dirichlet and Neumann boundary conditions. The numerical results of the GRBFC-NCM, GRBFICM and the GRBFECM are presented in Figs. 2–10 and Tabs. 1–10.

Example 1. Consider the non-dimensional form of a two-dimensional problem

$$\frac{\partial u}{\partial t} = \left[\frac{2x+1}{x+y+1} \right] \frac{\partial^2 u}{\partial x^2} + \left[\frac{2y+1}{x+y+1} \right] \frac{\partial^2 u}{\partial y^2} + \cos(2x+2y+t) + 8 \sin(2x+2y+t), \tag{34}$$

where $x_a \leq x \leq x_b, y_a \leq y \leq y_b, t \geq 0, x_a = y_a = t_0 = 0$ and $x_b = y_b = 1$, with the initial condition

$$u(x, y, t_0) = \sin(2x+2y) + \exp(x+y) \tag{35}$$

and subject to the boundary condition

$$\begin{aligned} u(x_a, y, t) &= \sin(2y+t) + \exp(y+2t), \\ u(x, y_a, t) &= \sin(2x+t) + \exp(x+2t), \\ u(x_b, y, t) &= \sin(2y+t+2) + \exp(y+2t+1), \\ u(x, y_b, t) &= \sin(2x+t+2) + \exp(x+2t+1). \end{aligned}$$

The analytical solution is given by

$$u(x, y, t) = \sin(2x+2y+t) + \exp(x+y+2t). \tag{36}$$

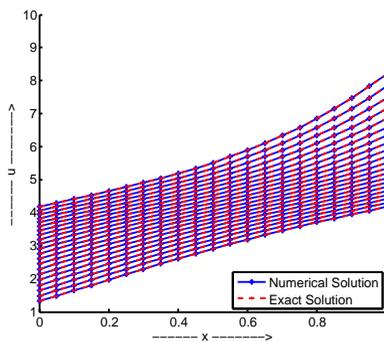


Figure 2: GRBFCNCM, Exact versus Numerical solution, Ex. 1, $t = 0.1, \Delta t = 10^{-3}, c = 0.37, y = 0, 0.1, 0.2, \dots, 0.9, 1.0, 21 \times 21$ nodes.

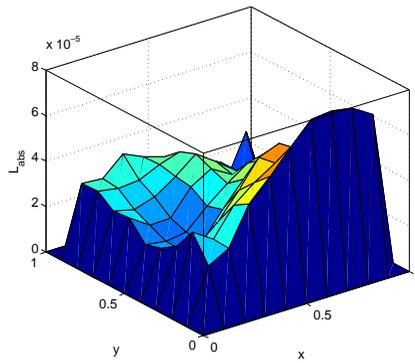


Figure 3: GRBFCNCM, L_{abs} , Ex. 1, $t = 0.1$, $\Delta t = 10^{-3}$, $c = 0.37$, 21×21 nodes.

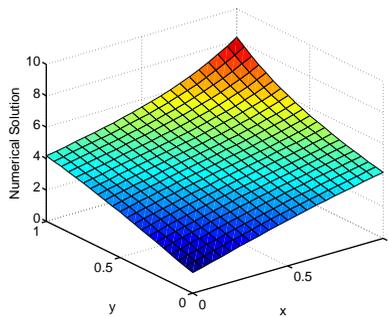


Figure 4: GRBFCNCM, Numerical solution, Ex. 1, $t = 0.1$, $\Delta t = 10^{-3}$, 21×21 nodes.

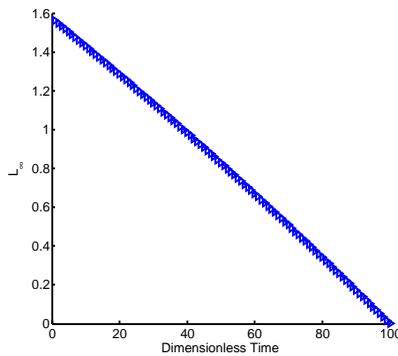


Figure 5: GRBFCNCM, L_∞ error norm versus time, Ex. 1, $t = 0.1$, $\Delta t = 10^{-3}$, $c = 0.95$, 11×11 nodes.

Table 1: GRBFCNCM, L_∞ and L_{rms} , Ex. 1, $t = 0.1$, 11×11 nodes.

c	$\Delta t = 10^{-4}$		$\Delta t = 10^{-3}$		$\Delta t = 10^{-2}$	
	L_∞	L_{rms}	L_∞	L_{rms}	L_∞	L_{rms}
0.50	1.04×10^{-3}	4.32×10^{-4}	1.01×10^{-3}	4.41×10^{-4}	1.30×10^{-3}	6.77×10^{-4}
0.80	1.27×10^{-4}	5.63×10^{-5}	1.39×10^{-4}	8.08×10^{-5}	1.23×10^{-3}	5.01×10^{-4}
0.85	9.10×10^{-5}	4.17×10^{-5}	1.33×10^{-4}	6.94×10^{-5}	1.24×10^{-3}	4.98×10^{-4}
0.90	3.70×10^{-5}	3.35×10^{-5}	1.29×10^{-4}	6.26×10^{-5}	1.24×10^{-3}	4.96×10^{-4}
0.95	1.55×10^{-5}	1.14×10^{-5}	1.26×10^{-4}	5.72×10^{-5}	1.27×10^{-3}	4.79×10^{-4}
0.99	div	div	1.22×10^{-4}	5.33×10^{-5}	1.24×10^{-3}	4.93×10^{-4}

Table 2: GRBFICM, L_∞ and L_{rms} , Ex. 1, $t = 0.1$, 11×11 nodes.

c	$\Delta t = 10^{-4}$		$\Delta t = 10^{-3}$		$\Delta t = 10^{-2}$	
	L_∞	L_{rms}	L_∞	L_{rms}	L_∞	L_{rms}
0.50	1.06×10^{-3}	4.20×10^{-4}	1.17×10^{-3}	3.69×10^{-4}	3.64×10^{-3}	1.71×10^{-3}
0.80	1.42×10^{-4}	4.71×10^{-5}	3.74×10^{-4}	1.72×10^{-4}	3.79×10^{-3}	1.88×10^{-3}
0.85	1.09×10^{-4}	3.31×10^{-5}	3.76×10^{-4}	1.77×10^{-4}	3.80×10^{-3}	1.89×10^{-3}
0.90	div	div	6.45×10^{-4}	2.74×10^{-4}	3.80×10^{-3}	1.90×10^{-3}
0.95	8.24×10^{-5}	2.48×10^{-5}	3.80×10^{-4}	1.81×10^{-4}	5.70×10^{-2}	1.43×10^{-2}
0.99	div	div	div	div	3.81×10^{-3}	1.91×10^{-3}

Table 3: GRBFECM, L_∞ and L_{rms} , Ex. 1, $t = 0.1$, 11×11 nodes.

c	$\Delta t = 10^{-4}$		$\Delta t = 10^{-3}$		$\Delta t = 10^{-2}$	
	L_∞	L_{rms}	L_∞	L_{rms}	L_∞	L_{rms}
0.50	1.03×10^{-4}	4.45×10^{-4}	8.60×10^{-4}	6.12×10^{-4}	div	div
0.80	1.15×10^{-4}	7.30×10^{-5}	6.51×10^{-4}	3.16×10^{-4}	div	div
0.85	8.85×10^{-5}	5.90×10^{-5}	6.44×10^{-4}	3.07×10^{-4}	div	div
0.90	div	div	6.39×10^{-4}	3.01×10^{-4}	div	div
0.95	div	div	6.36×10^{-4}	2.98×10^{-4}	div	div
0.99	div	div	6.38×10^{-4}	2.98×10^{-4}	div	div

Table 4: GRBFCNCM, L_∞ and L_{rms} , Ex. 1, $t = 0.1$, with different sets of nodes.

c	N	Δt	L_∞	L_{rms}
0.90	11×11	10^{-4}	7.55×10^{-5}	3.13×10^{-5}
0.38	21×21	10^{-4}	7.00×10^{-5}	2.70×10^{-5}
0.15	41×41	10^{-4}	1.12×10^{-4}	3.58×10^{-5}
0.10	55×55	10^{-4}	1.34×10^{-4}	6.96×10^{-5}
0.10	100×100	-	div	div

Table 5: GRBFICM, L_∞ and L_{rms} , Ex. 1, $t = 0.1$, with different sets of nodes.

c	N	Δt	L_∞	L_{rms}
0.90	11×11	10^{-4}	8.24×10^{-5}	2.48×10^{-5}
0.38	21×21	10^{-4}	1.17×10^{-4}	3.38×10^{-5}
0.15	41×41	10^{-4}	1.35×10^{-4}	2.96×10^{-5}
0.10	55×55	10^{-4}	1.29×10^{-4}	3.14×10^{-5}
0.10	100×100	–	div	div

Table 6: GRBFECM, L_∞ and L_{rms} , Ex. 1, $t = 0.1$, with different sets of nodes.

c	N	Δt	L_∞	L_{rms}
0.8	11×11	10^{-4}	1.15×10^{-4}	9.28×10^{-5}
0.3	21×21	10^{-4}	2.35×10^{-4}	1.03×10^{-4}
0.1	41×41	10^{-4}	div	div
0.1	55×55	–	div	div

Example 2. Consider the non-dimensional form of a linear diffusion-reaction equation

$$\frac{\partial u}{\partial t} = 0.2 \left[\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right] + 0.1u, \quad x_a \leq x \leq x_b, y_a \leq y \leq y_b, t \geq t_0 \tag{37}$$

with the initial condition

$$u(x, y, t_0) = \cos(x) + \sin(y) \tag{38}$$

and subject to the boundary conditions

$$u(x, y_a, t) = u(x, y_b, t) = \exp(-0.1t) \cos(x),$$

$$\frac{\partial u(x_a, y, t)}{\partial x} = \frac{\partial u(x_b, y, t)}{\partial x} = 0,$$

where $x_a = y_a = t_0 = 0$ and $x_b = y_b = 2\pi$.

Exact solution of the above equation is given by

$$u(x, y, t) = \exp(-0.1t)(\cos(x) + \sin(y)). \tag{39}$$

Accuracy of the GRBFCNCM and GRBFICM is shown as a function of different time step lengths, number of nodes in Tabs. 1, 2, 4, 7 and 9 for examples 1 and 2. The performance of GRBFCNCM is marginally better than GRBFICM. It is clear from these tables that when the time step restriction eases the accuracy is

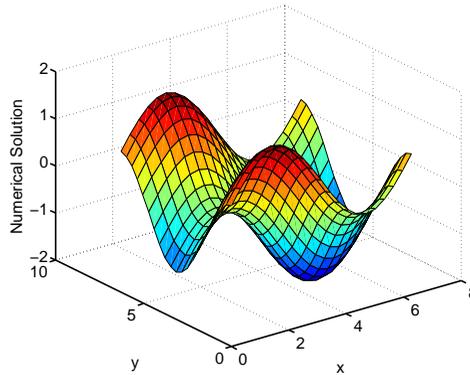


Figure 6: The GRBFCNCM, Numerical solution, Ex. 2, $t = 0.1$, $\Delta t = 10^{-4}$, $c = 1.99$, 21×21 nodes.

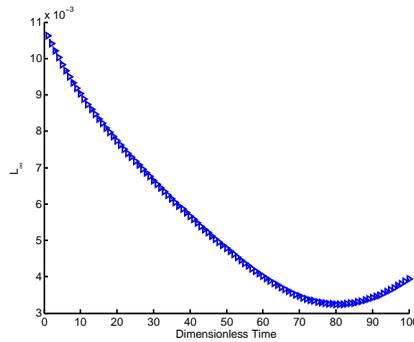


Figure 7: The GRBFCNCM, L_{rms} errors verses time, Ex. 2, $t = 0.1$, $\Delta t = 10^{-3}$, $c = 2.1$, 21×21 nodes.

Table 7: The GRBFCNCM, L_{∞} and L_{rms} , Ex. 2, $t = 0.1$, 11×11 nodes.

c	$\Delta t = 10^{-4}$		$\Delta t = 10^{-3}$		$\Delta t = 10^{-2}$	
	L_{∞}	L_{rms}	L_{∞}	L_{rms}	L_{∞}	L_{rms}
3.20	2.21×10^{-2}	5.79×10^{-3}	2.21×10^{-2}	5.78×10^{-3}	2.21×10^{-3}	5.78×10^{-3}
3.00	8.99×10^{-3}	2.48×10^{-3}	8.99×10^{-3}	2.48×10^{-3}	8.96×10^{-3}	2.47×10^{-3}
2.99	8.33×10^{-3}	2.46×10^{-3}	3.33×10^{-3}	2.45×10^{-3}	8.31×10^{-3}	2.45×10^{-3}
2.80	2.06×10^{-2}	5.23×10^{-3}	2.05×10^{-2}	5.23×10^{-3}	2.06×10^{-2}	5.23×10^{-3}
2.00	7.53×10^{-2}	2.48×10^{-2}	7.53×10^{-2}	2.48×10^{-2}	7.51×10^{-2}	2.48×10^{-2}
1.50	1.02×10^{-1}	3.64×10^{-2}	1.02×10^{-1}	3.64×10^{-2}	1.02×10^{-1}	3.63×10^{-2}

reasonably good in the case of the GRBFICM. However when the coefficient matrix size exceeds 60×60 the solution diverges. The numerical results of the GRBFECM

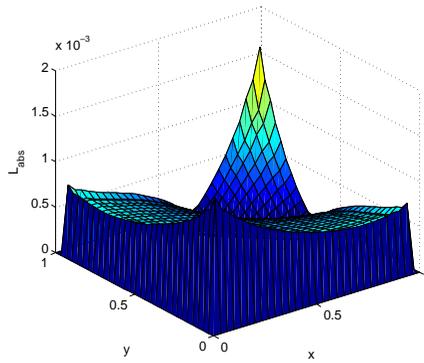


Figure 8: GRBFCNCM, L_{abs} , Ex. 2, $t = 0.1$ $c = 0.1$, $\Delta t = 10^{-3}$, 31×31 nodes.

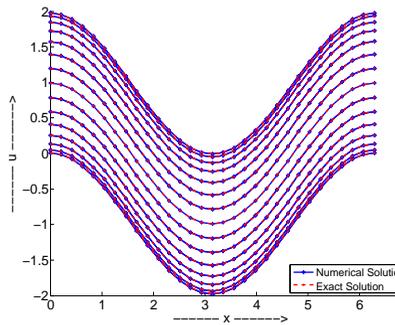


Figure 9: The GRBFCNCM, approximate versus analytical solutions, Ex. 2, $t = 0.1$, $\Delta t = 10^{-3}$, $c = 10$, $m = 9$, $y = \pi/15, 2\pi/15, \dots, 14\pi/15$, 31×31 nodes.

Table 8: The GRBFICM, L_∞ and L_{rms} , Ex. 2, $t = 0.1$, 11×11 nodes.

c	$\Delta t = 10^{-4}$		$\Delta t = 10^{-3}$		$\Delta t = 10^{-2}$	
	L_∞	L_{rms}	L_∞	L_{rms}	L_∞	L_{rms}
3.20	2.21×10^{-2}	5.78×10^{-3}	2.21×10^{-2}	5.78×10^{-3}	2.20×10^{-3}	5.78×10^{-3}
3.00	8.99×10^{-3}	2.48×10^{-3}	8.99×10^{-3}	2.48×10^{-3}	8.96×10^{-3}	2.47×10^{-3}
2.99	8.33×10^{-3}	2.46×10^{-3}	3.33×10^{-3}	2.45×10^{-3}	8.31×10^{-3}	2.45×10^{-3}
2.80	2.06×10^{-2}	5.23×10^{-3}	2.05×10^{-2}	5.23×10^{-3}	2.06×10^{-2}	5.23×10^{-3}
2.00	7.53×10^{-2}	2.48×10^{-2}	7.53×10^{-2}	2.48×10^{-2}	7.51×10^{-2}	2.48×10^{-2}
1.50	1.02×10^{-1}	3.64×10^{-2}	1.02×10^{-1}	3.64×10^{-2}	1.02×10^{-1}	3.63×10^{-2}

are shown in Tabs. 3 and 6. Excellent agreement of exact versus numerical solution is shown in Figs. 2, 3, 6. Absolute errors are shown in Fig. 3 and 8. In Figs. 5 and 7, L_∞ errors are shown converging as the solution advances in time. Overall

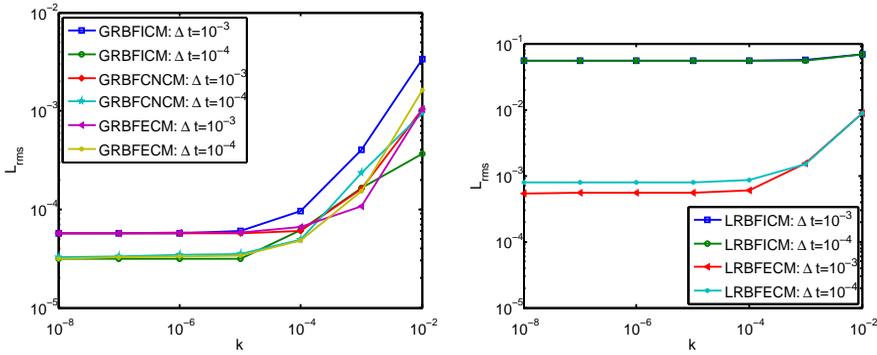


Figure 10: Noisy data, Ex. 1, $t = 0.1$, L_{rms} , 11×11 uniform nodes, (left) GRBFICM, $c = 0.95$, (right) LBRFCM, $c = 100$.

Table 9: The GRBFCNCM, L_∞ and L_{rms} , Ex. 2, $t = 0.1$, different sets of nodes.

c	N	Δt	L_∞	L_{rms}
2.99	11×11	10^{-4}	8.34×10^{-3}	2.46×10^{-3}
1.50	21×21	10^{-4}	6.57×10^{-2}	1.48×10^{-2}
0.50	41×41	10^{-4}	8.17×10^{-2}	1.37×10^{-2}
0.50	55×55	10^{-4}	7.92×10^{-2}	1.19×10^{-2}
0.50	100×100	–	div	div

Table 10: The GRBFICM, L_∞ and L_{rms} , Ex. 2, $t = 0.1$, different sets of nodes.

c	N	Δt	L_∞	L_{rms}
2.99	11×11	10^{-4}	8.34×10^{-3}	2.46×10^{-3}
1.5	21×21	10^{-4}	6.58×10^{-2}	1.48×10^{-2}
0.50	41×41	10^{-4}	8.18×10^{-2}	1.37×10^{-2}
0.50	55×55	10^{-4}	7.95×10^{-2}	1.20×10^{-2}
0.50	100×100	–	div	div

performance of the GRBFCNCM and GRBFICM is better than the GRBFECM as the later one suffers from small time step restriction in addition to ill-conditioning of the global collocation matrix. In terms of stability, the GRBFCNCM is better than the GRBFICM and GRBFCEM.

Next we consider Ex. 1 with initial noisy data as proposed in Younga, Fana, Hua, and Atluri (2009) as follows:

$$u(x, y, 0) = (1 + k\lambda) \sin(2x + 2y) + \exp(x + y). \tag{40}$$

The boundary conditions can be extracted from the analytical solution

$$u(x,y,t) = (1 + k\lambda) \sin(2x + 2y + t) + \exp(x + y + 2t), \tag{41}$$

where $-1 \leq \lambda \leq 1$ are uniformly distributed random numbers generated through Matlab command *rand* and k is the amplitude of noise level. In the present case we consider the range $[10^{-8}, 10^{-2}]$ for the noise level k with the initial profile given in the Eq. (40). Comparison of L_{rms} corresponding to the GRBFECM, GRBFICM and GRBFCNCM for different time steps varying from 10^{-2} to 10^{-5} is shown in Fig. 10 (left). From these results it can be concluded that the initial perturbations do not produce instabilities when solution marches into final time.

Next section is focused on numerical results obtained form local meshless methods implemented on a local domain instead of the whole set of points as discussed in the previous section.

5.2 Numerical Results for LRBFCM

In this section numerical results of the LRBFCM corresponding to examples 1 and 2 are presented in their explicit and implicit formulations. The numerical results corresponding to examples 1 and 2 of LRBFCM and LRBFICM are shown in Figs. 10-18 and Tabs. 11-21.

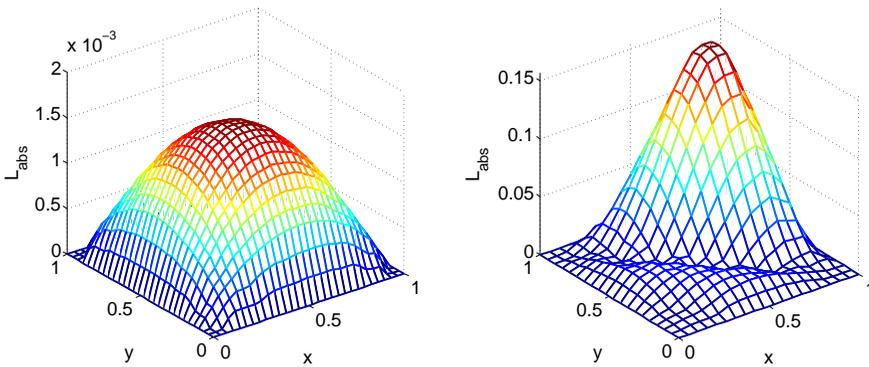


Figure 11: LRBFECM on the left and LRBFICM on the right, Ex. 1, $t = 0.1$, $\Delta t = 10^{-5}$, 11×11 nodes.

In Figs. 11-12 maximum absolute errors of the LRBFECM and LRBFICM as well as numerical versus approximate solution are shown for different values of time and time step lengths. In Tabs. 11-17 L_∞ and L_{rms} errors of both the LRBFECM and LRBFICM are shown as a function various values of t , Δt and nodal density. It is observed from these tables that the LRBFICM can tolerate comparatively

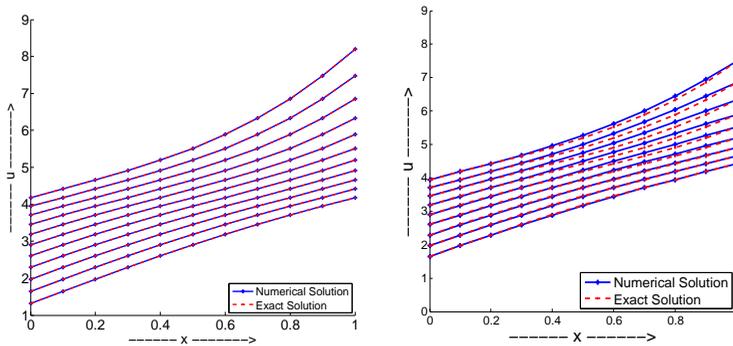


Figure 12: LRBFE on the left and LRBFI on the right, analytical versus approximate solution, Ex. 1, $t = 0.1$, $\Delta t = 10^{-3}$, $m = 5$, $c = 100$, $y = 0, 0.1, 0.2, \dots, 0.9, 1.0$, 11×11 nodes.

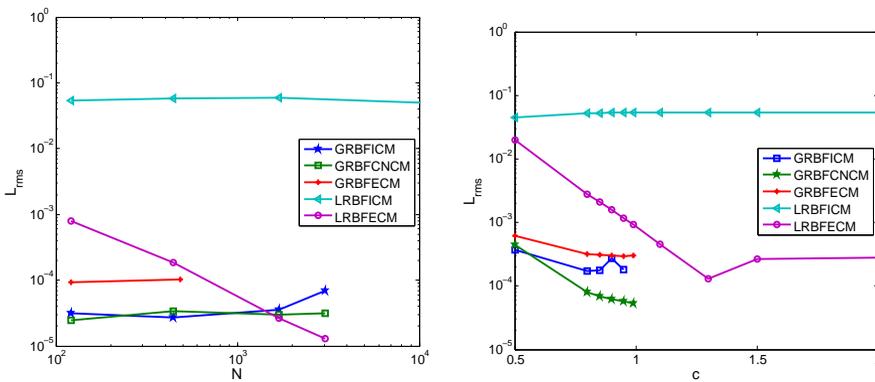


Figure 13: Left GRBF and LRBFE ($m = 5$, $c = 100$), L_{rms} errors, Ex. 1, $t = 0.1$, $\Delta t = 10^{-3}$, right GRBF and LRBFE, different values of c , 11×11 nodes.

Table 11: LRBFE, L_∞ and L_{rms} , Ex. 1, 11×11 nodes.

t	$\Delta t = 10^{-5}$		$\Delta t = 10^{-4}$		$\Delta t = 10^{-3}$	
	L_{rms}	L_∞	L_{rms}	L_∞	L_{rms}	L_∞
10^{-3}	1.97×10^{-5}	3.01×10^{-5}	1.92×10^{-5}	2.97×10^{-5}	1.56×10^{-5}	2.66×10^{-5}
10^{-2}	1.74×10^{-4}	2.92×10^{-4}	1.70×10^{-4}	2.87×10^{-4}	1.33×10^{-4}	2.53×10^{-4}
10^{-1}	8.17×10^{-4}	1.62×10^{-3}	7.91×10^{-4}	1.57×10^{-3}	5.43×10^{-4}	1.17×10^{-3}

large time step length where as the performance of the LRBFE is very good at small time, small time step lengths and denser nodes. The computational cost of

Table 12: LRBFCM, L_∞ and L_{rms} , Ex. 1, 11×11 nodes.

	$\Delta t = 10^{-4}$		$\Delta t = 10^{-3}$		$\Delta t = 10^{-2}$	
t	L_{rms}	L_∞	L_{rms}	L_∞	L_{rms}	L_∞
10^{-3}	2.76×10^{-3}	1.01×10^{-2}	2.94×10^{-3}	9.93×10^{-3}	–	–
10^{-2}	1.99×10^{-2}	6.73×10^{-2}	1.98×10^{-2}	6.64×10^{-2}	1.97×10^{-2}	6.16×10^{-2}
10^{-1}	5.44×10^{-2}	1.59×10^{-1}	5.46×10^{-2}	1.60×10^{-1}	5.55×10^{-2}	1.62×10^{-1}

Table 13: LRBFCM, L_∞ and L_{rms} , Ex. 1, 21×21 nodes.

	$t = 10^{-3}$		$t = 10^{-2}$		$t = 10^{-1}$	
Δt	L_{rms}	L_∞	L_{rms}	L_∞	L_{rms}	L_∞
10^{-5}	5.24×10^{-6}	7.44×10^{-6}	4.57×10^{-5}	7.22×10^{-5}	2.13×10^{-4}	4.08×10^{-4}
10^{-4}	4.74×10^{-5}	7.01×10^{-6}	4.10×10^{-5}	6.78×10^{-5}	1.86×10^{-4}	3.61×10^{-4}
10^{-3}	7.51×10^{-6}	2.03×10^{-5}	5.30×10^{-5}	1.39×10^{-4}	div	div

Table 14: LRBFCM, L_∞ and L_{rms} , Ex. 1, 21×21 nodes.

	$t = 10^{-3}$		$t = 10^{-2}$		$t = 10^{-1}$	
Δt	L_{rms}	L_∞	L_{rms}	L_∞	L_{rms}	L_∞
10^{-4}	3.13×10^{-3}	1.08×10^{-2}	2.15×10^{-2}	6.94×10^{-2}	5.75×10^{-2}	1.68×10^{-1}
10^{-3}	3.15×10^{-3}	1.03×10^{-2}	2.13×10^{-2}	6.82×10^{-2}	5.77×10^{-2}	1.68×10^{-1}
10^{-2}	–	–	1.98×10^{-2}	6.06×10^{-2}	5.85×10^{-2}	1.70×10^{-1}

Table 15: LRBFCM L_∞ and L_{rms} , Ex. 1, different number of nodes.

	$\Delta t = 10^{-5}$		$t = 10^{-3}$		$t = 10^{-2}$	
N	L_{rms}	L_∞	L_{rms}	L_∞	L_{rms}	L_∞
11×11	1.97×10^{-5}	3.01×10^{-5}	1.74×10^{-4}	2.92×10^{-4}	1.74×10^{-4}	2.92×10^{-4}
21×21	4.81×10^{-6}	7.44×10^{-6}	4.57×10^{-5}	7.22×10^{-5}	4.57×10^{-5}	7.22×10^{-5}
41×41	1.29×10^{-6}	1.72×10^{-6}	1.12×10^{-5}	1.71×10^{-5}	1.12×10^{-5}	1.71×10^{-5}
101×101	1.21×10^{-7}	1.69×10^{-7}	1.03×10^{-6}	1.64×10^{-6}	1.03×10^{-6}	1.64×10^{-6}

Table 16: LRBFCM, L_∞ and L_{rms} , Ex. 1, different number of nodes.

	$\Delta t = 10^{-2}$		$t = 10^{-2}$		$t = 10^{-1}$		$t = 10^0$	
N	L_{rms}	L_∞	L_{rms}	L_∞	L_{rms}	L_∞	L_{rms}	L_∞
11×11	1.97×10^{-2}	6.16×10^{-2}	5.55×10^{-2}	1.62×10^{-1}	7.62×10^{-2}	1.75×10^{-1}	7.62×10^{-2}	1.75×10^{-1}
21×21	1.98×10^{-2}	6.06×10^{-2}	5.85×10^{-2}	1.70×10^{-1}	8.05×10^{-2}	1.78×10^{-1}	8.05×10^{-2}	1.78×10^{-1}
41×41	1.97×10^{-2}	5.96×10^{-2}	5.99×10^{-2}	1.70×10^{-1}	8.27×10^{-2}	1.78×10^{-1}	8.27×10^{-2}	1.78×10^{-1}
101×101	1.96×10^{-2}	5.90×10^{-2}	6.07×10^{-2}	1.70×10^{-1}	8.39×10^{-2}	1.78×10^{-1}	8.39×10^{-2}	1.78×10^{-1}

Table 17: LRBFCM, L_∞ and L_{rms} , Ex. 1, different number of nodes.

$\Delta t = 10^{-3}$	$t = 10^{-3}$		$t = 10^{-2}$		$t = 10^{-1}$	
N	L_{rms}	L_∞	L_{rms}	L_∞	L_{rms}	L_∞
11×11	2.94×10^{-3}	9.93×10^{-3}	1.98×10^{-2}	6.64×10^{-2}	5.46×10^{-2}	1.60×10^{-1}
21×21	3.15×10^{-3}	1.03×10^{-2}	2.13×10^{-2}	6.82×10^{-2}	5.77×10^{-2}	1.68×10^{-1}
41×41	3.20×10^{-3}	1.04×10^{-2}	2.19×10^{-2}	6.96×10^{-2}	5.92×10^{-2}	1.68×10^{-1}
101×101	3.21×10^{-3}	1.03×10^{-2}	2.23×10^{-2}	6.97×10^{-2}	6.02×10^{-2}	1.69×10^{-1}

the LRBFCM is greater than the LRBFECM as the later needs solution of $N \times N$ sparse matrix at each time iteration. In Fig. 13, the L_{rms} errors are shown as a function of the number of uniform interpolation nodes and the shape parameter using the GRBFECM, GRBFCNCM, GRBFICM, LRBFECM and LRBFCM. The local methods work for larger amount of nodes and larger range of the shape parameter, but the global methods perform more accurate with smaller number of nodes when the shape parameter is chosen properly.

Comparison of L_{rms} for the LRBFECM, LRBFCM corresponding to Ex. 1 with noisy initial data given in Eq. (40) with different values of noise levels mentioned earlier and different time steps varying from 10^{-2} to 10^{-5} is shown in Fig. 10 (right). The LRBFECM and LRBFCM maintain their stability at the advance time levels for the whole range of noise introduced in the initial data. Performance of LRBFCM is more stable than GRBFCM for the whole range of k .

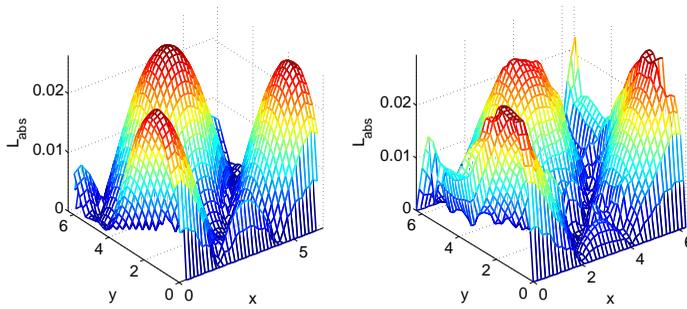


Figure 14: LRBFECM, L_{abs} , Ex. 2, $t = 0.1$, $\Delta t = 10^{-3}$, $m = 5$, $c = 100$, 21×21 nodes. on the left and random nodes on the right.

The numerical results of Ex. 2 produced through the LRBFCM and LRBFECM are shown Figs. 14-16 and Tabs. 18-21. In Figs. 14, a comparison of absolute errors of the LRBFECM for uniform and non-uniform nodes is shown. In the case of non-uniform nodes the behavior of absolute errors is comparatively more oscillatory and

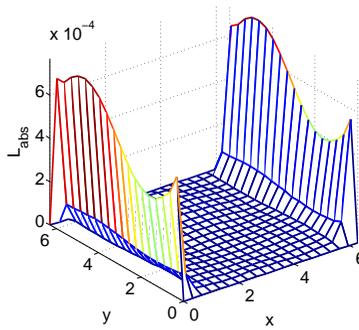


Figure 15: LRBFIEM, L_{abs} , Ex. 2, $t = 0.1$, $\Delta t = 10^{-3}$, $m = 9$, $c = 10$, 31×31 nodes.

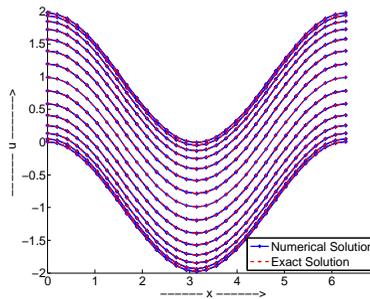


Figure 16: LRBFIEM, approximate versus analytical solutions, Ex. 2, $t = 0.1$, $\Delta t = 10^{-3}$, $c = 10$, $m = 9$, $y = \pi/15, 2\pi/15, \dots, 14\pi/15$, 31×31 nodes.

Table 18: LRBFEEM, L_∞ and L_{rms} , Ex. 2, $m = 5$, 11×11 and 21×21 nodes.

$N = 11 \times 11$		$t = 10^{-3}$		$t = 10^{-2}$		$t = 10^{-1}$	
Δt	L_{rms}	L_∞	L_{rms}	L_∞	L_{rms}	L_∞	
10^{-3}	9.33×10^{-3}	2.43×10^{-2}	8.96×10^{-3}	2.43×10^{-2}	1.25×10^{-2}	2.64×10^{-2}	
10^{-4}	9.33×10^{-3}	2.43×10^{-2}	8.96×10^{-3}	2.43×10^{-2}	1.25×10^{-2}	2.64×10^{-2}	
10^{-5}	9.33×10^{-3}	2.43×10^{-2}	8.96×10^{-3}	2.43×10^{-2}	1.25×10^{-2}	2.64×10^{-2}	
$N = 21 \times 21$		$t = 10^{-3}$		$t = 10^{-2}$		$t = 10^{-1}$	
10^{-3}	4.47×10^{-4}	1.60×10^{-3}	1.25×10^{-3}	2.68×10^{-3}	1.28×10^{-2}	2.67×10^{-2}	
10^{-4}	4.48×10^{-4}	1.60×10^{-3}	1.25×10^{-3}	2.68×10^{-3}	1.28×10^{-2}	2.67×10^{-2}	
10^{-5}	4.48×10^{-4}	1.60×10^{-3}	1.25×10^{-3}	2.68×10^{-3}	1.28×10^{-2}	2.67×10^{-2}	

large than the uniform nodes. In Figs. 15, a comparison of absolute errors of the LRBFIEM is shown. It is clear from this figure that the errors are maximum near the boundaries where the Neumann boundary condition are specified. In Fig. 16 a

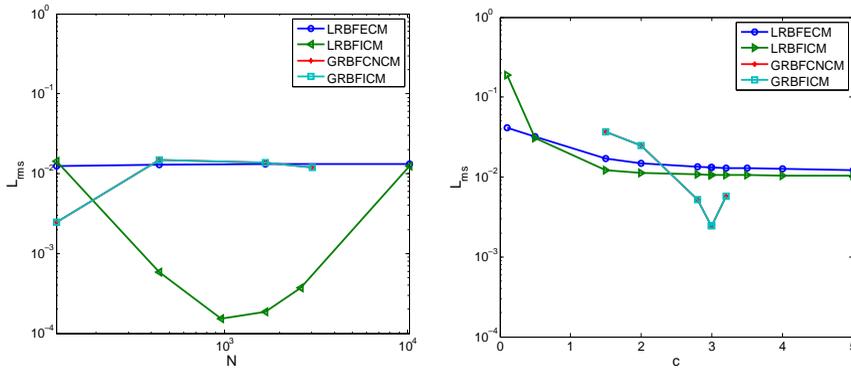


Figure 17: Left: GRBFICM and LRBFICM ($m = 5, c = 100$), L_{rms} errors, Ex. 2, $t = 0.1, \Delta t = 10^{-3}$. Right: GRBFICM and LRBFICM, different values of c , 11×11 nodes.

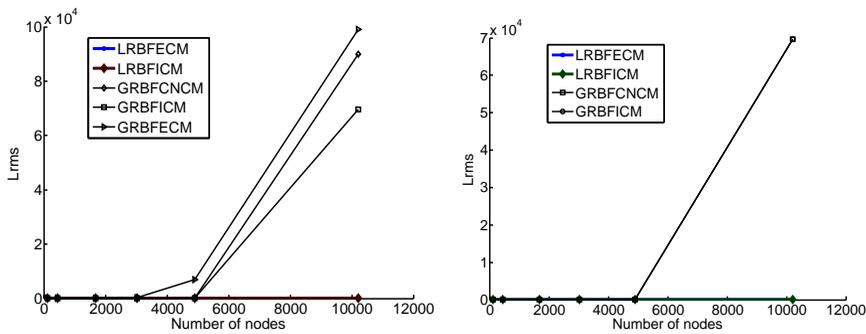


Figure 18: LRBFICM, LRBFE and GRBF, GRBFECM, GRBFECM left Ex. 1, right Ex. 2, L_{rms} , different sets of nodes.

Table 19: LRBFICM, L_∞ and L_{rms} , Ex. 2, $21 \times 21, m = 9, 31 \times 31$ nodes.

$N = 21 \times 21$		$t = 10^{-3}$		$t = 10^{-2}$		$t = 10^{-1}$	
Δt	L_{rms}	L_∞	L_{rms}	L_∞	L_{rms}	L_∞	
10^{-2}	–	–	5.08×10^{-4}	2.19×10^{-3}	5.96×10^{-4}	2.63×10^{-3}	
10^{-3}	4.99×10^{-4}	2.13×10^{-3}	5.07×10^{-4}	2.19×10^{-3}	5.87×10^{-4}	2.68×10^{-3}	
10^{-4}	4.99×10^{-4}	2.13×10^{-3}	5.05×10^{-4}	2.24×10^{-3}	6.73×10^{-4}	3.10×10^{-3}	
$N = 31 \times 31$		$t = 10^{-3}$		$t = 10^{-2}$		$t = 10^{-1}$	
10^{-2}	–	–	1.03×10^{-4}	5.42×10^{-4}	1.52×10^{-4}	7.66×10^{-4}	
10^{-3}	9.79×10^{-5}	5.15×10^{-4}	1.03×10^{-4}	5.42×10^{-4}	1.54×10^{-4}	7.68×10^{-4}	
10^{-4}	9.79×10^{-5}	5.15×10^{-4}	1.04×10^{-4}	5.42×10^{-4}	1.58×10^{-4}	7.69×10^{-4}	

Table 20: LRBFECM, L_∞ and L_{rms} , Ex. 2, $\Delta t = 10^{-3}$ with different sets of nodes.

$m = 5$	$t = 10^{-3}$		$t = 10^{-2}$		$t = 10^{-1}$	
N	L_{rms}	L_∞	L_{rms}	L_∞	L_{rms}	L_∞
11×11	9.33×10^{-3}	2.43×10^{-2}	8.96×10^{-3}	2.43×10^{-2}	1.25×10^{-2}	2.64×10^{-2}
21×21	4.47×10^{-4}	1.60×10^{-3}	1.25×10^{-3}	2.68×10^{-3}	1.28×10^{-2}	2.67×10^{-2}
41×41	1.29×10^{-4}	2.68×10^{-4}	1.32×10^{-3}	2.68×10^{-3}	1.31×10^{-2}	2.67×10^{-2}
101×101	1.34×10^{-4}	2.70×10^{-4}	1.34×10^{-3}	2.70×10^{-3}	1.33×10^{-2}	2.69×10^{-2}

Table 21: LRBFICM, L_∞ and L_{rms} , Ex. 2, $\Delta t = 10^{-3}$, with different sets of nodes.

$m = 9$	$t = 10^{-3}$		$t = 10^{-2}$		$t = 10^{-1}$	
N	L_{rms}	L_∞	L_{rms}	L_∞	L_{rms}	L_∞
11×11	9.88×10^{-3}	3.10×10^{-2}	9.43×10^{-3}	3.06×10^{-2}	1.44×10^{-2}	3.43×10^{-2}
21×21	4.99×10^{-4}	2.13×10^{-3}	5.07×10^{-4}	2.19×10^{-3}	5.87×10^{-4}	2.68×10^{-3}
31×31	9.79×10^{-5}	5.15×10^{-4}	1.03×10^{-4}	5.42×10^{-4}	1.54×10^{-4}	7.68×10^{-4}
41×41	4.42×10^{-5}	3.13×10^{-4}	5.25×10^{-5}	3.42×10^{-4}	1.87×10^{-4}	5.60×10^{-4}
51×51	3.31×10^{-5}	2.57×10^{-4}	6.18×10^{-5}	4.64×10^{-4}	3.71×10^{-4}	9.66×10^{-4}
101×101	1.42×10^{-4}	1.78×10^{-3}	1.30×10^{-3}	4.73×10^{-3}	1.25×10^{-2}	3.54×10^{-2}

comparison of exact versus numerical solution for various values of y is shown. In Fig. 17, the L_{rms} errors are shown as a function of the number of uniform interpolation nodes and the shape parameter using the GRBFECM, GRBFCNCM, GRBFICM, LRBFECM and LRBFICM for Ex. 2. In Fig. 18, the L_{rms} errors are shown as a function of the number of uniform interpolation nodes and the shape parameter using the GRBFECM, GRBFCNCM, GRBFICM, LRBFECM and LRBFICM for both examples 1 and 2. It is clear from these figures that the performance of the global methods is more sensitive to the shape parameter and the accuracy gets worse as the number of nodal points is increased. Like the example 1, the local methods work for a larger amount of nodes and a larger range of the shape parameter in this case as well. Global methods perform more accurately with a smaller number of nodes when the shape parameter is chosen properly.

In Tabs. 18-21 the L_∞ and L_{rms} errors are shown as a function of different time step lengths at various times and different number of uniform nodes for both the LRBFECM and LRBFICM. It is clear from these tables that the numerical results do not improve when the time step length is decreased for a comparatively large time. For denser nodes, the accuracy is improving for shorter times, and small time step lengths but not for longer times and large time step lengths. The denser discretization yields higher accuracy at shorter times.

6 Conclusions

In this paper we presented and compared truly meshless global and local approaches based on the MQ RBFs for solving two dimensional diffusion-reaction PDEs. The LRBFCM is different from traditional the GRBFCM in the sense that it is a pure local procedure. Based on the study presented in the previous sections, we summarize the outcomes as follows:

- i The GRBFCM uses nodes from the whole domain simultaneously while the LRBFCM needs local configuration of the nodes falling in the domain of influence.
- ii The LRBFCM approach is less sensitive to the shape parameter than the GRBFCM
- iii Both versions of meshless techniques (GRBFCM and LRBFCM) produce stable results for initial data with and without noise in the specified range of parameters k , c and Δt . The LRBFEFCM and LRBFCM perform more stable than their counter parts the GRBFECM, GRBFCNCM and GRBFICM for large values of k .
- iv The GRBFCNCM, GRBFICM and the LRBFCM are less sensitive to the time step size than its counter parts the GRBFECM and the LRBFEFCM.
- v The GRBFCNCM, is more stable in terms of sensitivity to the shape parameter c than the GRBFICM and GRBFECM.
- vi Computational cost of the local version is considerably smaller than the global one as it requires only solution of a small system of the same size as the number of nodes in the domain of influence.
- vii The matrix resulting from the LRBFCM is sparse with all entries in each row zero except those related to the sub-domain.
- viii For small systems both the GRBFCM and LRBFCM can be used but for large problems the LRBFCM is the only preferable choice.
- ix The study of Lee, Liu, and Fan (2003) has been extended and confirmed for time dependent PDEs.

Acknowledgement: The authors are grateful to the financial support provided from the 6th EU framework project INSPIRE and to the Slovenian Grant Agency through grant no. P2-0379 Programme Group Modeling of Material and Processes.

The second author is also thankful to NWFP UET Peshawar Pakistan for sabbatical leave.

References

- Ali, A.; Siraj-ul-Islam; Haq, S.** (2009): A computational meshfree technique for the numerical solution of the two-dimensional coupled burger's equations. *Int. J. Comput. Meth. Eng. Sci. Mech.*, vol. 10, pp. 406–412.
- Atluri, S.; Shen, S.** (2002): *The Meshless Method*. Forsyth: Tech. Sci. Press USA.
- Atluri, S. N.** (2004): *The Meshless Method MLPG for Domain and BIE Discretization*. Tech. Sci. Press USA.
- Atluri, S. N.; Liu, H. T.; Han, Z. D.** (2006): Meshless local petrov-galerkin MLPG mixed collocation method for elasticity problems. *CMES: Computer Modeling in Engineering and Sciences*, vol. 14, pp. 141–152.
- Buhmann, M. D.** (2003): *Radial Basis Functions: Theory and Implementations*. Cambridge University Press.
- Chen, C. S.; Ganesh, M.; Golberg, M. A.; Cheng, A. H. D.** (2002): Multi-level compact radial basis functions based computational scheme for some elliptic problems. *Computer Math. Appl.*, vol. 43, pp. 359–378.
- Divo, E.; Kassab, A. J.** (2007): An efficient localized rbf meshless method for fluid flow and conjugate heat transfer. *ASME J. Heat Trans.*, vol. 129, pp. 124–136.
- Fasshauer, G. F.** (2008): *Meshfree Approximation Methods with MATLAB*. World Scientific Press Singapore.
- Franke, C.; Schaback, R.** (1998): Convergence order estimates of meshless collocation methods using radial basis functions. *Adv. Comput. Math.*, vol. 8, pp. 381–399.
- Golberg, M. A.; Chen, C. S.; Karur, S. R.** (1996): Improved multiquadric approximation for partial differential equations. *Eng. Anal. Bound. Elem.*, vol. 18, pp. 9–17.
- Hardy, R.** (1971): Multiquadric equations of topography and other irregular surfaces. *Geo. Phys. Res.*, vol. 176, pp. 1905–1915.
- Jang, B.** (2007): Solutions to the non-homogeneous parabolic problems by the extended hadm. *Appl. Math. Comput.*, vol. 191, pp. 466–483.
- Kansa, E. J.** (1990): Multiquadrics scattered data approximation scheme with applications to computational fluid-dynamics I, surface approximations and partial derivative estimates. *Comput. Math. Appl.*, vol. 19, pp. 127–145.

- Kosec, G.; Šarler, B.** (2009): Convection driven melting of anisotropic metals. *Int. J. Cast. Met. Res.*, vol. 22, pp. 279–282.
- Kosec, G.; Šarler, B.** (2010): Meshless approach to solving freezing with natural convection. *Mater. Sci. Forum*, vol. 22, pp. 205–210.
- Kovačević, I.; Šarler, B.** (2005): Solution of a phase-field model for dissolution of primary particles in binary aluminum alloys by an r-adaptive mesh-free method. *Mater. Sci. Eng. A*, vol. 413–414, pp. 423–428.
- Lee, C. K.; Liu, X.; Fan, S. C.** (2003): Local multiquadric approximation for solving boundary value problems. *Computat. Mech.*, vol. 30, pp. 396–409.
- Liu, G. R.** (2003): *Meshfree Methods: Moving Beyond the Finite Element Method*. CRC Press.
- Lorbiecka, A.; Vertnik, R.; Gjerkeš, H.; Manojlovič, G.; Senčič, B.; Cesar, J.; Šarler, B.** (2009): Numerical modeling of grain structure and the continuous casting of steel. *CMC: Computers Material Continua*, vol. 8, pp. 195–208.
- Madych, W. R.; Nelson, S. A.** (1990): Multivariate interpolation and conditionally positive definite functions II. *Math. Comput.*, vol. 54, pp. 211–30.
- 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 equation. *Eng. Anal. Bound. Elemt.*, vol. 26, pp. 133–156.
- Micchelli, C. A.** (1986): Interpolation of scattered data: distance matrix and conditionally positive definite functions. *Construct. Approx.*, vol. 2, pp. 11–22.
- Nie, Q.; Wana, F. Y.; Zhang, Y.-T.; Liu, X.-F.** (2008): Compact integration factor methods in high spatial dimensions. *J. Comput. Phys.*, vol. 227, pp. 5238–5255.
- 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. Meth. Appl. Mech. Eng.*, vol. 192, pp. 941–954.
- Siraj-ul-Islam; Haq, S.; Ali, A.** (2009): A meshfree method for the numerical solution of the rlw equation. *J. Comput. Appl. Math.*, vol. 223, pp. 997–1012.
- Siraj-ul-Islam; Haq, S.; Uddin, M.** (2009): A meshfree interpolation method for the numerical solution of the coupled nonlinear partial differential equations. *Eng. Anal. Bound. Elemt.*, vol. 33, pp. 399–409.
- Tarwater, A.** (1985): A parameter study of Hardy's multiquadric method for scattered data interpolation. *Lawrence Livermore National Laboratory, Technical Report UCRL-54670*, vol. 82.

Vertnik, R.; Šarler, B. (2006): Meshless local radial basis function collocation method for convective-diffusive solid-liquid phase change problems. *Inter. J. Numer. Meth. Heat and Fluid Flow*, vol. 16, pp. 617–640.

Vertnik, R.; Šarler, B. (2009): Simulation of continuous casting of steel by a meshless technique. *Int. J. Cast. Met. Res.*, vol. 22, pp. 311–313.

Vertnik, R.; Založnik, M.; Šarler, B. (2006): Solution of transient direct-chill aluminum billet casting problem with simulations material and interphase moving boundaries by a meshless method. *Eng. Anal. Bound. Elemt.*, vol. 30, pp. 847–855.

Šarler, B. (2005): A radial basis function collocation approach in computational fluid dynamics. *CMES: Computer Modeling in Engineering and Sciences*, vol. 7, pp. 185–193.

Šarler, B.; Kosec, G.; Lorbiecka, A.; Vertnik, R. (2010): A meshless approach and solution of multiscale solidification modeling. *Mater. Sci. Forum*, vol. 649, pp. 211–216.

Šarler, B.; Vertnik, R. (2006): Meshfree explicit local radial basis function collocation method for diffusion problems. *Computer Math. Appl.*, pp. 1269–1282.

Younga, D.; Fana, C.; Hua, S.; Atluri, S. (2009): The Eulerian Lagrangian method of fundamental solutions for two dimensional unsteady Burgers' equations. *Eng. Anal. Bound. Elemt.*, vol. 32, pp. 395–412.