



ELSEVIER

Contents lists available at [SciVerse ScienceDirect](http://www.sciencedirect.com)

Engineering Analysis with Boundary Elements

journal homepage: www.elsevier.com/locate/enganabound

Assessment of global and local meshless methods based on collocation with radial basis functions for parabolic partial differential equations in three dimensions

Guangming Yao^a, Siraj-ul-Islam^{b,*}, Božidar Šarler^{b,c}^a Division of Mathematics and Computer Science, Clarkson University, Potsdam, NY 13699-5815, USA^b Laboratory for Multiphase Processes, University of Nova Gorica, Nova Gorica, Slovenia^c Laboratory for Advanced Materials Systems, Centre of Excellence for Biosensorics, Instrumentation and Process Control, Solkan, Slovenia

ARTICLE INFO

Article history:

Received 2 January 2012

Accepted 25 April 2012

Available online 21 June 2012

Keywords:

Parabolic partial differential equation

Meshless method

RBF collocation method

Explicit scheme

Implicit scheme

Crank–Nicolson scheme

ABSTRACT

A comparison of the performance of the global and the local radial basis function collocation meshless methods for three dimensional parabolic partial differential equations is performed in the present paper. The methods are structured with multiquadrics radial basis functions. The time-stepping is performed in a fully explicit, fully implicit and Crank–Nicolson ways. Uniform and non-uniform node arrangements have been used. A three-dimensional diffusion–reaction equation is used for testing with the Dirichlet and mixed Dirichlet–Neumann boundary conditions. The global methods result in discretization matrices with the number of unknowns equal to the number of the nodes. The local methods are in the present paper based on seven-noded influence domains, and reduce to discretization matrices with seven unknowns for each node in case of the explicit methods or a sparse matrix with the dimension of the number of the nodes and seven non-zero row entries in case of the implicit method. The performance of the methods is assessed in terms of accuracy and efficiency. The outcome of the comparison is as follows. The local methods show superior efficiency and accuracy, especially for the problems with Dirichlet boundary conditions. Global methods are efficient and accurate only in cases with small amount of nodes. For large amount of nodes, they become inefficient and run into ill-conditioning problems. Local explicit method is very accurate, however, sensitive to the node position distribution, and becomes sensitive to the shape parameter of the radial basis functions when the mixed boundary conditions are used. Performance of the local implicit method is comparatively better than the others when a larger number of nodes and mixed boundary conditions are used. The paper represents an extension of our recently made similar study in two dimensions.

Published by Elsevier Ltd.

1. Introduction

In recent years radial basis functions (RBFs) have been extensively used in different applications [2,3,5,17,32,36,37,39,43] and emerged as a potential alternative in the field of numerical solution of partial differential equations (PDEs). A detailed discussions on meshless methods and their applications to many complex PDEs, industrial and large-scale problems can also be found in [8,11,12,22–24,40,42] and the references therein.

Different types of meshless methods, based on RBFs, have gained popularity in the engineering and science community for a number of reasons. The most attractive features of the meshless methods are (i) they provide an alternative numerical tool, free from extensive and costly mesh generation or manipulation related

problems; (ii) they are flexible in dealing with complex geometries, and are easily extendible to multi-dimensional problems. Meshless methods have been proved successful for solving PDEs on both regular and irregular node arrangements. They use functional basis which allows arbitrary placement of points. Traditional numerical methods, such as the finite difference method (FDM), the finite volume method (FVM), and the finite element method (FEM), are based on the local mesh based interpolation to find the solution and its derivatives. In contrary to these mesh-based methods, meshless methods use a set of uniform or random points which are not interconnected in the form of a classical mesh. Meshless methods actually reduce to multivariate data fitting between the points and related calculation of the derivatives and/or integrals. In the case of meshless methods, interpolation can be accomplished both locally and globally with high efficiency.

In 1971, Hardy introduced radial basis functions interpolation [13] to approximate two-dimensional geographical surfaces based on scattered data. Later on, meshless methods, based on

* Corresponding author.

E-mail address: siraj.islam@gmail.com (Siraj-ul-Islam).

Multiquadric (MQ) RBFs [15], were derived for numerical solutions of different types of PDEs. The idea was extended by [10] afterwards. The existence, uniqueness, and convergence of the RBFs approximation was discussed in [9,26,28]. The importance of shape parameter c in the MQ RBF was elaborated in [35]. Solvability of the system of equations with respect to distinct interpolation points was discussed in [28]. All of these methods can be called the global radial basis function collocation method (GRBFCM). The most recent applications of the GRBFCM can be found in [1,17,32–34]. The main disadvantage of the GRBFCM is that it involves full matrices that result from the discretization of the PDEs. These matrices are often ill-conditioned and extremely sensitive to the choice of the shape parameters in RBFs.

To overcome the problems of ill-conditioning and shape parameter sensitivity of the GRBFCM, the local radial basis function collocation method (LRBFCM) was first introduced for diffusion problems in [43] with improved results in terms of accuracy and efficiency of the method. Subsequently, due to handiness of this approach, the LRBFCM has been applied to more complex problems such as convection–diffusion problems with phase-change [37], continuous casting [40], solid–solid phase transformations [22], heat transfer and fluid flow [41], Navier–Stokes equations [7], Darcy flow [19], turbulent flow [39], etc.

The main idea of LRBFCM is the collocation on the overlapping sub-domains of influence instead of the whole domain which drastically reduces the size of the collocation matrix at the expense of solving many small matrices. The size of each small matrix is the same as the number of nodes included in the domain of influence of each node.

The main disadvantage of the LRBFCM is that the method does not work for elliptic problems in a straightforward way. Another kind of RBF-based meshless methods use the integration of RBFs instead of the differentiation of the RBFs. They are known as indirect RBF collocation methods. This class includes indirect RBFN method (IRBFN) [27], the method of approximate particular solutions (MAPS) [6], the localized method of approximate particular solutions (LMAPS) [47], and others. The recent studies can be found in [49]. The LMAPS works well for elliptic PDEs, and can be extended to time-dependent problems as well [46]. This approach yields sparse matrices instead of full matrices, which makes the LMAPS suitable for solving large-scale problems. However, in this paper, we will focus on the LRBFCM only.

PDEs govern physical problems like transport processes, including heat transfer and fluid flows, wave propagation or interaction between fluids and solids, and option pricing. Unlike lower-dimensional problems, the numerical simulation of three-dimensional problems [4,44,45] is much more computationally intensive in terms of CPU time and huge memory requirements. Local meshless methods are not that much prone to these problems since the coefficient matrix is of the same size as the size of the local sub-domain, which is usually relatively small. In the case of uniform node arrangement, the small matrix needs to be inverted only once outside the time-loop for time-dependent problems. This saves a considerable amount of CPU time and consumes less memory as well. The computational efficiency of the local meshless methods in the case of two-dimensional problems and its usefulness in large-scale simulations can be found in [19–21,25,38,39,48].

This paper is an extension of work in [48] to three-dimensional problems. The main motivation for this work is that literature on the numerical methods for three-dimensional problems is sparse compared with lower-dimensional problems. This is particularly true in the field of meshless methods. Some of the related work can be found in Refs. [4,14,18,44,45]. We compare performances of the following five meshless collocation methods: the global implicit radial basis function collocation method (GI), the global Crank–Nicolson radial basis function collocation method (GCN),

the global explicit radial basis function collocation method (GE), the local explicit radial basis function collocation method (LE), and the local implicit radial basis function collocation method (LI).

The structure of the rest of the paper is as follows. In Section 2, we introduce the governing equations. In Section 3, we discuss the time discretization technique from implicit and explicit points of view. In Section 4, the numerical methods are introduced from the local and global views. Section 5 is devoted to discussion regarding the scaling technique of the shape parameter c of MQ RBF and the numerical tests on benchmark problems. At the end, conclusions are drawn.

2. Governing equations

Consider a dimensionless form of the three-dimensional diffusion–reaction equation, defined on domain Ω with boundary Γ

$$\frac{\partial u(\mathbf{x},t)}{\partial t} = \mathcal{L}[u(\mathbf{x},t)] + \mu u(\mathbf{x},t) + g(\mathbf{x},t), \quad \mathbf{x} \in \Omega, \quad t > t_0, \quad (1)$$

with the initial condition

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

and Dirichlet or Neumann boundary conditions

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

where $u, t, \mathbf{x} = [x, y, z]^T$ are the diffusion, time and space variables, respectively, tr represents the matrix transpose, g and f are the known functions of \mathbf{x} and t , $\Gamma = \Gamma_D + \Gamma_N$, where Γ_N and Γ_D are the boundaries that satisfy Neumann and Dirichlet boundary conditions, respectively. μ is a real constant, \mathcal{L} is a differential operator consisting of first- or second-order derivatives of space variables and \mathcal{B} is a 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 $t = t_0 + \Delta t$ be the time discretization, where t_0 refers to the beginning time of every time step, and t refers to the end of the time step. For a time period $[t_0, t]$, the time derivative in Eq. (1) is approximated by Euler formula:

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

Let $\theta \in (0, 1]$. The parameter θ is used in the time discretization of Eq. (1) as

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

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

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

Then Eq. (1) can be discretized in time-space as

$$\begin{aligned} (1 - \mu\theta\Delta t)u(\mathbf{x},t) - \theta\Delta t\mathcal{L}[u(\mathbf{x},t)] - \theta\Delta tg(\mathbf{x},t) \\ = (1 + \mu(1-\theta)\Delta t)u(\mathbf{x},t_0) + (1-\theta)\Delta t\mathcal{L}[u(\mathbf{x},t_0)] \\ + (1-\theta)\Delta tg(\mathbf{x},t_0), \end{aligned} \quad (8)$$

note that $t = t_0 + \Delta t$. Similarly, for $\mathbf{x} \in \Gamma$, Eq. (3) can be discretized in time-space as

$$\theta\mathcal{B}[u(\mathbf{x},t)] - \theta f(\mathbf{x},t) = -(1-\theta)\mathcal{B}[u(\mathbf{x},t_0)] + (1-\theta)f(\mathbf{x},t_0). \quad (9)$$

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:

$$\gamma_{\Omega}^{\mathbf{x}} = \begin{cases} 1, & \mathbf{x} \in \Omega, \\ 0, & \mathbf{x} \notin \Omega, \end{cases} \quad \gamma_{\Gamma}^{\mathbf{x}} = \begin{cases} 1, & \mathbf{x} \in \Gamma, \\ 0, & \mathbf{x} \notin \Gamma, \end{cases} \quad (10)$$

where $\mathbf{x} \in \Omega \cup \Gamma$. By using Eqs. (8) and (9), Eqs. (1)–(3) can be approximated as

$$\begin{aligned} &\gamma_{\Gamma}^{\mathbf{x}}[\theta B[u(\mathbf{x},t)] - \theta f(\mathbf{x},t)] + \gamma_{\Omega}^{\mathbf{x}}[(1 - \mu\theta\Delta t)u(\mathbf{x},t) \\ &\quad - \theta\Delta t\mathcal{L}[u(\mathbf{x},t)] - \theta\Delta t\mathbf{g}(\mathbf{x},t)] \\ &= \gamma_{\Gamma}^{\mathbf{x}}[-(1 - \theta)B[u(\mathbf{x},t_0)] + (1 - \theta)f(\mathbf{x},t_0)] \\ &\quad + \gamma_{\Omega}^{\mathbf{x}}[(1 + \mu(1 - \theta)\Delta t)u(\mathbf{x},t_0) \\ &\quad + (1 - \theta)\Delta t\mathcal{L}[u(\mathbf{x},t_0)] + (1 - \theta)\Delta t\mathbf{g}(\mathbf{x},t_0)]. \end{aligned} \quad (11)$$

More specifically, if $\theta = 0$, i.e., the explicit method:

$$\begin{aligned} &\gamma_{\Gamma}^{\mathbf{x}}B[u(\mathbf{x},t_0)] + \gamma_{\Omega}^{\mathbf{x}}u(\mathbf{x},t) \\ &= \gamma_{\Gamma}^{\mathbf{x}}f(\mathbf{x},t_0) + \gamma_{\Omega}^{\mathbf{x}}[(1 + \mu\Delta t)u(\mathbf{x},t_0) \\ &\quad + \Delta t\mathcal{L}[u(\mathbf{x},t_0)] + \Delta t\mathbf{g}(\mathbf{x},t_0)]; \end{aligned} \quad (12)$$

if $\theta = 0.5$, i.e., Crank–Nicolson method:

$$\begin{aligned} &0.5\gamma_{\Gamma}^{\mathbf{x}}[B[u(\mathbf{x},t) - f(\mathbf{x},t)] + \gamma_{\Omega}^{\mathbf{x}}[(1 - 0.5\mu\Delta t)u(\mathbf{x},t) \\ &\quad - 0.5\Delta t\mathcal{L}[u(\mathbf{x},t)] - 0.5\Delta t\mathbf{g}(\mathbf{x},t)] \\ &= 0.5\gamma_{\Gamma}^{\mathbf{x}}[-B[u(\mathbf{x},t_0)] + f(\mathbf{x},t_0)] \\ &\quad + \gamma_{\Omega}^{\mathbf{x}}[(1 + 0.5\mu\Delta t)u(\mathbf{x},t_0) \\ &\quad + 0.5\Delta t\mathcal{L}[u(\mathbf{x},t_0)] + 0.5\Delta t\mathbf{g}(\mathbf{x},t_0)]; \end{aligned} \quad (13)$$

if $\theta = 1$, i.e., implicit method:

$$\begin{aligned} &\gamma_{\Gamma}^{\mathbf{x}}B[u(\mathbf{x},t)] + \gamma_{\Omega}^{\mathbf{x}}[(1 - \mu\Delta t)u(\mathbf{x},t) - \Delta t\mathcal{L}[u(\mathbf{x},t)] \\ &\quad - \Delta t\mathbf{g}(\mathbf{x},t)] = \gamma_{\Gamma}^{\mathbf{x}}f(\mathbf{x},t) + \gamma_{\Omega}^{\mathbf{x}}u(\mathbf{x},t_0). \end{aligned} \quad (14)$$

These are the commonly used two-level time stepping strategies which approximate $u(\mathbf{x},t)$ in Ω from the values given at the initial time t_0 and further time $t = t_0 + \Delta t$. In the global cases derived in this paper, we take $\theta = 0, 0.5$ and 1 , whereas in the local approximations, we choose $\theta = 0$ and 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 the number of interior points, and N_{Γ} denotes the number of boundary points.

4.1. Global method

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

$$u(\mathbf{x},t) = \sum_{k=1}^N \phi(\|\mathbf{x} - \mathbf{x}_k\|)\alpha_k(t), \quad \mathbf{x} \in \Omega, \quad (15)$$

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

$$\phi(\|\mathbf{x} - \mathbf{x}_k\|) = \sqrt{\|\mathbf{x} - \mathbf{x}_k\|^2 + c^2}, \quad (16)$$

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

$$\mathcal{L}u(\mathbf{x},t) = \sum_{k=1}^N \mathcal{L}\phi(\|\mathbf{x} - \mathbf{x}_k\|)\alpha_k(t). \quad (17)$$

The coefficients $\alpha_k(t), k = 1, 2, \dots, N$ can be found from the collocation as

$$\begin{bmatrix} u(\mathbf{x}_1,t) \\ u(\mathbf{x}_2,t) \\ \vdots \\ u(\mathbf{x}_N,t) \end{bmatrix} = \begin{bmatrix} \phi(\|\mathbf{x}_1 - \mathbf{x}_1\|) & \phi(\|\mathbf{x}_1 - \mathbf{x}_2\|) & \dots & \phi(\|\mathbf{x}_1 - \mathbf{x}_N\|) \\ \phi(\|\mathbf{x}_2 - \mathbf{x}_1\|) & \phi(\|\mathbf{x}_2 - \mathbf{x}_2\|) & \dots & \phi(\|\mathbf{x}_2 - \mathbf{x}_N\|) \\ \vdots & \vdots & \ddots & \vdots \\ \phi(\|\mathbf{x}_N - \mathbf{x}_1\|) & \phi(\|\mathbf{x}_N - \mathbf{x}_2\|) & \dots & \phi(\|\mathbf{x}_N - \mathbf{x}_N\|) \end{bmatrix} \begin{bmatrix} \alpha_1(t) \\ \alpha_2(t) \\ \vdots \\ \alpha_N(t) \end{bmatrix}. \quad (18)$$

This can be written in the matrix notation as

$$\mathbf{u} = \Phi\boldsymbol{\alpha}, \quad (19)$$

where $\mathbf{u} = [u(\mathbf{x}_1,t), u(\mathbf{x}_2,t), \dots, u(\mathbf{x}_N,t)]^T$, $\boldsymbol{\alpha} = [\alpha_1(t), \alpha_2(t), \dots, \alpha_N(t)]^T$ and $\Phi_{sk} = \phi(\|\mathbf{x}_s - \mathbf{x}_k\|)$ is the matrix element of $N \times N$ matrix Φ . By inserting Eqs. (15)–(17) in Eq. (11) we get

$$\begin{aligned} &\theta\gamma_{\Gamma}^{\mathbf{x}_j} \sum_{k=1}^N B\phi(\|\mathbf{x}_j - \mathbf{x}_k\|)\alpha_k(t) - \gamma_{\Gamma}^{\mathbf{x}_j}\theta f(\mathbf{x}_j,t) \\ &\quad + \gamma_{\Omega}^{\mathbf{x}_j} \left[(1 - \theta\mu\Delta t) \sum_{k=1}^N \phi(\|\mathbf{x}_j - \mathbf{x}_k\|)\alpha_k(t) \right. \\ &\quad \left. - \theta\Delta t \left(\sum_{k=1}^N \mathcal{L}\phi(\|\mathbf{x}_j - \mathbf{x}_k\|)\alpha_k(t) - \mathbf{g}(\mathbf{x}_j,t) \right) \right] \\ &= \gamma_{\Gamma}^{\mathbf{x}_j}(1 - \theta)f(\mathbf{x}_j,t_0) - \gamma_{\Gamma}^{\mathbf{x}_j}(1 - \theta) \sum_{k=1}^N B\phi(\|\mathbf{x}_j - \mathbf{x}_k\|)\alpha_k(0) \\ &\quad + \gamma_{\Omega}^{\mathbf{x}_j} \left[(1 + \mu(1 - \theta)\Delta t)u(\mathbf{x}_j,t_0) + (1 - \theta)\Delta t \right. \\ &\quad \left. \left(\sum_{k=1}^n \mathcal{L}\phi(\|\mathbf{x}_j - \mathbf{x}_k\|)\alpha_k(0) + \mathbf{g}(\mathbf{x}_j,t_0) \right) \right] \end{aligned} \quad (20)$$

for $j = 1, \dots, N$, and $0 \leq \theta \leq 1$ and $\boldsymbol{\alpha}_0 = [\alpha_1(0), \alpha_2(0), \dots, \alpha_N(0)]^T$ is the value of α_k at t_0 which is updated in every time iteration. The above linear system of equations can be written in a matrix notation as

$$\Psi\boldsymbol{\alpha} = \mathbf{b}, \quad (21)$$

where Ψ is the matrix in N by N system of linear equations Eq. (20), the unknown coefficients on the left-hand side of Eq. (20) are denoted by $\boldsymbol{\alpha} = [\alpha_1(t), \alpha_2(t), \dots, \alpha_N(t)]^T$ and $\mathbf{b} = [b_1, b_2, \dots, b_N]^T$ is the right-hand side of Eq. (20). The matrices Ψ and \mathbf{b} are defined as

$$\begin{aligned} \Psi_{sk} &= \theta\gamma_{\Gamma}^{\mathbf{x}_s} B\phi(\|\mathbf{x}_s - \mathbf{x}_k\|) + \gamma_{\Omega}^{\mathbf{x}_s} [(1 - \theta\mu\Delta t)\phi(\|\mathbf{x}_s - \mathbf{x}_k\|) \\ &\quad - \theta\Delta t\mathcal{L}\phi(\|\mathbf{x}_s - \mathbf{x}_k\|)], \end{aligned}$$

$$\begin{aligned} b_s &= \gamma_{\Gamma}^{\mathbf{x}_s}\theta f(\mathbf{x}_s,t) + \gamma_{\Omega}^{\mathbf{x}_s}\theta\Delta t\mathbf{g}(\mathbf{x}_s,t) - \gamma_{\Gamma}^{\mathbf{x}_s}(1 - \theta)B\Phi\boldsymbol{\alpha}_0 \\ &\quad + \gamma_{\Gamma}^{\mathbf{x}_s}(1 - \theta)f(\mathbf{x}_s,t_0) + \gamma_{\Omega}^{\mathbf{x}_s}(1 - \theta)\Delta t\mathbf{g}(\mathbf{x}_s,t_0) \\ &\quad + \gamma_{\Omega}^{\mathbf{x}_s}[(1 + \mu(1 - \theta)\Delta t)u(\mathbf{x}_s,t_0) \\ &\quad - (1 - \theta)\Delta t\mathcal{L}\Phi\boldsymbol{\alpha}_0] \quad s, k = 1, 2, \dots, m, \end{aligned}$$

where Φ is given in (19). We determine the coefficients $\boldsymbol{\alpha}$ by inverting Ψ :

$$\boldsymbol{\alpha} = \Psi^{-1}\mathbf{b}, \quad (22)$$

which implies that

$$\alpha_s(t) = \sum_{k=1}^N \Psi_{sk}^{-1} b_k, \quad s = 1, 2, \dots, N, \quad (23)$$

where Ψ_{sk}^{-1} denotes the matrix element of the matrix Ψ^{-1} . The diffusion variable u , at the next time $t = t_0 + \Delta t$ and position \mathbf{x} , can

be approximated in the following form:

$$u(\mathbf{x}, t) = \sum_{k=1}^N \phi(\|\mathbf{x} - \mathbf{x}_k\|) \alpha_k(t), \quad \mathbf{x} \in \Omega. \tag{24}$$

4.2. Local methods

In order to reduce the size of the dense matrices arising from the global scheme, we propose a local meshless scheme instead of the global methods, as discussed in the last section and compare its performance with global method in the context of three-dimensional time dependent PDEs.

For each $\mathbf{x}_j \in \Omega \cup \Gamma$, $j = 1, 2, \dots, N$ we choose n nearest neighboring points contained in the sub-domain $j\Omega = \{\mathbf{x}_k\}_{k=1}^n$, where k denotes the local indexing for each collocation point \mathbf{x}_j being center of $j\Omega$ instead of these whole domains. The schematic diagram of the local domain of influence containing seven points, i.e., $n=7$, is shown in Fig. 1.

To approximate $u(\mathbf{x}, t)$ through the local explicit radial basis function collocation method (LE), consider collocation on the sub-domain $j\Omega = \{\mathbf{x}_k\}_{k=1}^n$, $j = 1, \dots, N$ instead of the whole domain Ω . The diffusion variable u can be approximated on each sub-domain in the following form:

$$u(j\mathbf{x}, t_0) = \sum_{k=1}^n \phi(\|\mathbf{x} - \mathbf{x}_k\|) \alpha_k, \quad j\mathbf{x} \in j\Omega. \tag{25}$$

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

$$\mathcal{L}u(j\mathbf{x}, t_0) = \sum_{k=1}^n \mathcal{L}\phi(\|\mathbf{x} - \mathbf{x}_k\|) \alpha_k, \quad j\mathbf{x} \in j\Omega, \tag{26}$$

$$B u(j\mathbf{x}, t_0) = \sum_{k=1}^n B\phi(\|\mathbf{x} - \mathbf{x}_k\|) \alpha_k = f(j\mathbf{x}, t_0), \quad j\mathbf{x} \in j\Omega. \tag{27}$$

The coefficients $j\alpha_k$ are determined by collocation in the following form:

$$\begin{aligned} \gamma_{\Omega}^{\mathbf{x}_j} \sum_{k=1}^n \phi(\|\mathbf{x}_s - \mathbf{x}_k\|) \alpha_k + \gamma_{\Gamma}^{\mathbf{x}_j} \sum_{k=1}^n B\phi(\|\mathbf{x}_s - \mathbf{x}_k\|) \alpha_k \\ = \gamma_{\Omega}^{\mathbf{x}_j} u(j\mathbf{x}_s, t_0) + \gamma_{\Gamma}^{\mathbf{x}_j} f(j\mathbf{x}_s, t_0), \end{aligned} \tag{28}$$

where $s = 1, 2, \dots, n$, $\gamma_{\Omega}^{\mathbf{x}_j}$ and $\gamma_{\Gamma}^{\mathbf{x}_j}$ are defined indicators of \mathbf{x}_s . The above linear system can be written in matrix notation as

$$j\Phi_j \mathbf{j}\alpha = j\mathbf{b}, \tag{29}$$

where $j\alpha = [j\alpha_1, j\alpha_2, \dots, j\alpha_n]^T$, $j\mathbf{b} = [j b_{1,j}, j b_{2,j}, \dots, j b_{n,j}]^T$ is the right-hand side of Eq. (28). The matrices $j\Phi$ and $j\mathbf{b}$ are defined as

$$j\Phi_{sk} = \gamma_{\Omega}^{\mathbf{x}_j} \phi(\|\mathbf{x}_s - \mathbf{x}_k\|) + \gamma_{\Gamma}^{\mathbf{x}_j} B\phi(\|\mathbf{x}_s - \mathbf{x}_k\|), \quad s, k = 1, 2, \dots, n,$$

$$j b_s = \gamma_{\Omega}^{\mathbf{x}_j} u(j\mathbf{x}_s, t_0) + \gamma_{\Gamma}^{\mathbf{x}_j} f(j\mathbf{x}_s, t_0), \quad s = 1, 2, \dots, n,$$

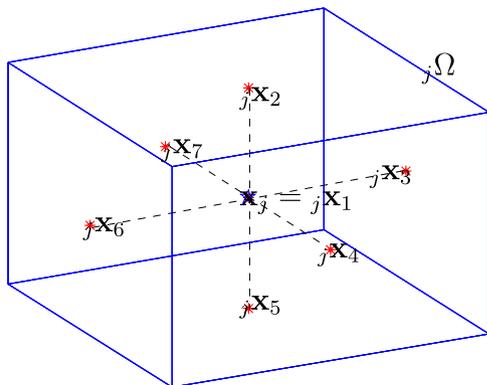


Fig. 1. Seven-node local domain schematics.

where $j\Phi = [j\Phi_{sk}] \in \mathbb{R}^{n \times n}$. We determine the coefficients $j\alpha$ by inverting $j\Phi$

$$j\alpha = j\Phi^{-1} j\mathbf{b}, \tag{30}$$

which implies that

$$j\alpha_s = \sum_{k=1}^n j\Phi_{sk}^{-1} j b_k, \quad s = 1, 2, \dots, n, \tag{31}$$

where $j\Phi_{sk}^{-1}$ denotes the matrix element of the matrix $j\Phi^{-1}$. Then,

$$\mathcal{L}u(\mathbf{x}_j, t_0) = \sum_{s=1}^n \mathcal{L}\phi(\|\mathbf{x}_j - \mathbf{x}_s\|) \sum_{k=1}^n j\Phi_{sk}^{-1} j b_k. \tag{32}$$

Let $\theta = 0$. The diffusion–reaction variable u for $\theta = 0$ at the interior point \mathbf{x}_j , $u(\mathbf{x}_j, t)$, can be approximated by using Eq. (26) in (11):

$$\begin{aligned} u(\mathbf{x}_j, t) = (1 + \mu\Delta t)u(\mathbf{x}_j, t_0) + \Delta t \sum_{s=1}^n \mathcal{L}\phi(\|\mathbf{x}_j - \mathbf{x}_s\|) \\ \sum_{k=1}^n j\Phi_{sk}^{-1} j b_k + \Delta t g(\mathbf{x}_j, t_0). \end{aligned} \tag{33}$$

For boundary point \mathbf{x}_j , from Eq. (25), we have

$$u(\mathbf{x}_j, t) = \sum_{s=1}^n \phi(\|\mathbf{x}_j - \mathbf{x}_s\|) \sum_{k=1}^n j\Phi_{sk}^{-1} j b_k. \tag{34}$$

This completes the formulation of the local explicit radial basis function collocation method (LE). The formulations are similar for the global and the local methods, however, the global methods involve inversion of a full matrix, but the local methods invert N small $n \times n$ matrices.

In the case of local implicit radial basis function collocation method (LI), let $\theta = 1$ in Eq. (11),

$$\begin{aligned} \gamma_{\Omega}^{\mathbf{x}_j} \left[(1 - \mu\Delta t)u(\mathbf{x}_j, t) - \Delta t \sum_{s=1}^n \mathcal{L}\phi(\|\mathbf{x}_j - \mathbf{x}_s\|) \sum_{k=1}^n j\Phi_{sk}^{-1} u(j\mathbf{x}_k, t) \right] \\ + \gamma_{\Gamma}^{\mathbf{x}_j} \sum_{k=1}^n B\phi(\|\mathbf{x}_j - \mathbf{x}_k\|) \sum_{k=1}^n j\Phi_{sk}^{-1} u(j\mathbf{x}_k, t) \\ = \gamma_{\Omega}^{\mathbf{x}_j} [u(\mathbf{x}_j, t_0) + \Delta t g(\mathbf{x}_j, t)] + \gamma_{\Gamma}^{\mathbf{x}_j} f(\mathbf{x}_j, t). \end{aligned} \tag{35}$$

Note that $\{\{\mathbf{x}_k\}_{k=1}^n, j = 1, 2, \dots, N\} = \{\mathbf{x}_j\}_{j=1}^N$, Eq. (35) leads to a linear system of N equations with N unknowns $\{u(\mathbf{x}_j, t)\}_{j=1}^N$ with all entries in each row 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.

5. Numerical results

The performance of the explicit/implicit and global/local collocation methods, both on uniform and random node arrangements, is investigated in this paper. Two test problems are considered for the numerical validation. The MQ RBF

$$\phi(r) = \sqrt{r^2 + c^2} \tag{36}$$

are used throughout this text. The differentiation operator involved is $\mathcal{L} = \nabla^2$. Thus, the differentiation of ϕ involved in all the numerical methods is

$$\nabla^2 \phi(r) = \frac{2r^2 + 3c^2}{(r^2 + c^2)^{3/2}}. \tag{37}$$

We investigate the performance of the global and local RBFM that can both be implemented on evenly or randomly distributed nodes. The computer program has been coded in MATLAB. Three kinds of error measures: absolute error, 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, where u and \hat{u} represent exact and numerical solutions of the given PDE, respectively. N is the total number of collocation points considered. However, the points on the eight corners of the cube domain are not included. Let \mathcal{N} be the number of points on unit length $(0, 1)$ (note this is not a closed interval) when a unit cube domain with evenly distributed nodes is considered, the total number of interior collocation points is $N_i = \mathcal{N}^3$, and total number of boundary collocation points is $N_b = 6\mathcal{N}^2$. Thus, the total number of collocation points is $N = N_i + N_b = \mathcal{N}^3 + 6\mathcal{N}^2$. For example, $N = 2^3 + 6(2^2) = 32$ when $\mathcal{N} = 2$, and $N = 3^3 + 6(3^2) = 81$ when $\mathcal{N} = 3$. The random nodes are generated from the uniform nodes through the following transformation:

$$\mathbf{x}_j = \mathbf{x}_j + c_{rand} \eta r_{min}, \tag{38}$$

where $\mathbf{x}_j = (x_j, y_j, z_j)$, is j th collocation node, c_{rand} is a random number between 0 and 1, r_{min} denotes the minimum distance among different points which are uniformly distributed throughout the domain, η stands for a displacement factor. The number η is chosen from 0 to 0.8 in this paper, the uniform nodes are used if not stated otherwise (i.e., $\eta = 0$). A scaling technique similar to the one introduced in [43] is used to alleviate the difficulty of choosing different values of shape parameter in MQ RBFs. The scaling parameter ${}_j r_0$ is the maximum nodal distance in the sub-domains:

$${}_j r_0 = \max_{j=1}^N \max_{k=1}^n \sqrt{\|\mathbf{x}_j - \mathbf{x}_k\|^2}. \tag{39}$$

The parameter c in all RBFs and the corresponding derivatives is replaced by $c r_0$. Hence, a larger shape parameter of the MQ RBF can be used in the numerical implementation. Scaling of the shape parameter is performed to make MQ RBFs approximation insensitive to various dimensions of the domain. Thus, the LRBFCM is less sensitive with respect to the shape parameter unlike the GRBFCM. In this paper, the shape parameter in the local methods is chosen as $c_l = 100$ if not otherwise specified. The shape parameter c for the GRBFCM (GCN, GI, GE) is chosen as $c = 1.6/\sqrt{N}$ [45], where N is the total number of collocation points. The number of nodes in each sub-domain in the case of LRBFCM is chosen as $n = 7$, which is the simplest possible 3D local implementation. As n increases, all the three kinds of errors are expected to improve while the computational efficiency will decrease, and we may run into ill-conditioning problem.

Example 1. Consider the three-dimensional problem:

$$\frac{\partial u(x,y,z,t)}{\partial t} = \frac{1}{\pi^2} \nabla^2 u(x,y,z,t) - 2 \exp(t - \pi(x+y+z)), \tag{40}$$

where $0 \leq x, y, z \leq 1$, with the initial condition

$$u(x,y,z,t_0) = \exp(-\pi(x+y+z)) + x + y + z \tag{41}$$

and subject to the boundary condition

$$u(0,y,z,t) = \exp(t - \pi(y+z)) + y + z,$$

$$u(x,0,z,t) = \exp(t - \pi(x+z)) + x + z,$$

$$u(x,y,0,t) = \exp(t - \pi(x+y)) + x + y,$$

$$u(1,y,z,t) = \exp(t - \pi(1+y+z)) + 1 + y + z,$$

$$u(x,1,z,t) = \exp(t - \pi(x+1+z)) + x + 1 + z,$$

$$u(x,y,1,t) = \exp(t - \pi(x+y+1)) + x + y + 1.$$

The analytical solution is given by

$$u(x,y,t) = \exp(t - \pi(x+y+z)) + x + y + z. \tag{42}$$

The numerical results of the above problem are shown in Tables 1–3 and Figs. 2 and 3.

In Table 1, we compute the L_∞ and L_{rms} errors at $t=1$ by using different numbers of collocation points, N , fixed and time-step size $\Delta t = 10^{-3}$. RBF collocation methods have been shown [9,16] to be very accurate even for a small number of collocation points. As shown in [31], when the shape of the basis functions matches the shape/feature of the PDE solution, a small number of basis functions

Table 1

GI, LI, GE, LG, L_∞ and L_{rms} , Example 1, $t=1$, with different node densities, $\Delta t = 10^{-3}$.

N	GI	GCN	GE	LI	LE
L_∞					
32	8.42×10^{-3}	8.41×10^{-3}	4.36×10^{-2}	6.46×10^{-3}	6.38×10^{-3}
81	9.24×10^{-3}	9.22×10^{-3}	5.56×10^{-2}	5.36×10^{-3}	5.28×10^{-3}
160	8.22×10^{-3}	8.20×10^{-3}	5.89×10^{-2}	3.91×10^{-3}	3.88×10^{-3}
1600	1.58×10^{-3}	1.57×10^{-3}	6.61×10^{-2}	1.21×10^{-3}	8.81×10^{-4}
9025	–	–	–	1.84×10^{-3}	2.05×10^{-4}
L_{rms}					
32	1.61×10^{-3}	1.61×10^{-3}	1.02×10^{-2}	1.58×10^{-3}	1.56×10^{-3}
18	1.45×10^{-3}	1.45×10^{-3}	1.22×10^{-2}	1.16×10^{-3}	1.15×10^{-3}
160	9.46×10^{-4}	9.48×10^{-4}	1.38×10^{-2}	8.40×10^{-4}	8.32×10^{-4}
1600	1.35×10^{-4}	1.37×10^{-4}	1.73×10^{-2}	2.70×10^{-4}	1.96×10^{-4}
9025	–	–	–	4.20×10^{-4}	4.67×10^{-5}
CPU (s)					
32	0.52	0.29	0.30	0.19	0.28
18	0.88	0.88	0.82	0.42	0.59
160	3.10	3.18	3.41	0.77	1.21
1600	1068.54	1058.78	828.18	40.75	14.92
9025	–	–	–	2470.00	2505.95

Table 2

GI, LI, GE, LG, L_∞ and L_{rms} , Example 1, $t=1$, with different time-steps, $N=160$.

Δt	GI	GCN	GE	LI	LE
L_∞					
10^{-1}	9.58×10^{-3}	8.10×10^{-3}	8.53×10^{-2}	5.92×10^{-3}	6.06×10^{-3}
10^{-2}	8.53×10^{-3}	8.19×10^{-3}	5.83×10^{-2}	4.16×10^{-3}	2.98×10^{-3}
10^{-3}	8.22×10^{-3}	8.20×10^{-3}	5.89×10^{-2}	3.91×10^{-3}	3.88×10^{-3}
10^{-4}	8.41×10^{-3}	8.20×10^{-3}	5.90×10^{-2}	2.84×10^{-3}	3.97×10^{-3}
L_{rms}					
10^{-1}	1.75×10^{-3}	1.03×10^{-3}	2.07×10^{-2}	1.27×10^{-3}	1.38×10^{-3}
10^{-2}	1.62×10^{-3}	9.54×10^{-4}	1.37×10^{-2}	8.93×10^{-4}	6.39×10^{-4}
10^{-3}	9.46×10^{-4}	9.48×10^{-4}	1.38×10^{-2}	8.40×10^{-4}	8.32×10^{-4}
10^{-4}	1.61×10^{-3}	9.47×10^{-4}	1.38×10^{-2}	6.09×10^{-4}	8.51×10^{-4}

Table 3

GI, LI, GE, LG, L_∞ and L_{rms} , Example 1, $t=1$, with different displacement factors, η , $N=160$.

η	GI	GCN	GE	LI	LE
L_∞					
0.0	8.22×10^{-3}	8.20×10^{-3}	5.89×10^{-2}	3.91×10^{-3}	3.88×10^{-3}
0.2	1.07×10^{-2}	1.24×10^{-2}	6.52×10^{-2}	3.37×10^{-2}	3.50×10^{-2}
0.4	1.29×10^{-2}	1.86×10^{-2}	6.17×10^{-2}	4.58×10^{-2}	4.16×10^{-2}
0.6	1.35×10^{-2}	1.36×10^{-2}	5.71×10^{-2}	5.86×10^{-2}	3.06×10^{-2}
0.8	9.90×10^{-2}	5.58×10^{-2}	5.51×10^{-1}	6.16×10^{-2}	8.51×10^{-2}
L_{rms}					
0.0	9.46×10^{-4}	9.48×10^{-4}	1.38×10^{-2}	8.40×10^{-4}	8.32×10^{-4}
0.2	1.29×10^{-3}	1.20×10^{-3}	1.35×10^{-2}	4.16×10^{-3}	4.61×10^{-3}
0.4	1.22×10^{-3}	1.87×10^{-3}	1.22×10^{-2}	6.36×10^{-3}	4.31×10^{-3}
0.6	1.30×10^{-3}	1.51×10^{-3}	1.22×10^{-2}	6.44×10^{-3}	4.91×10^{-3}
0.8	1.61×10^{-3}	6.38×10^{-3}	4.68×10^{-2}	6.44×10^{-3}	9.58×10^{-3}

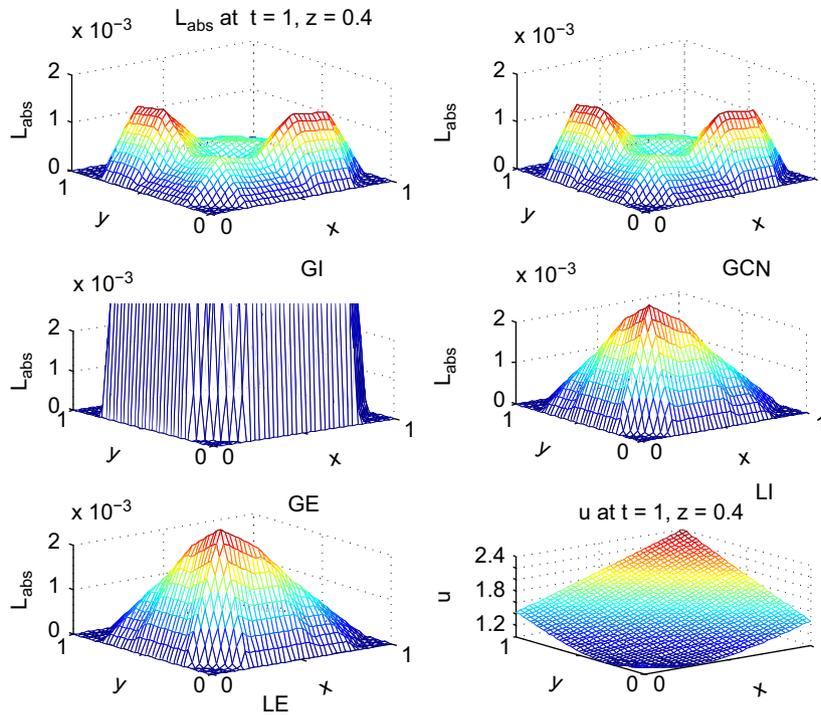


Fig. 2. The analytical solution and the approximate solutions using GI, GCN, GE, LI, and LE at $z=0.4$, $t=1$ in Example 1, $N=160$, $\Delta t=10^{-3}$.

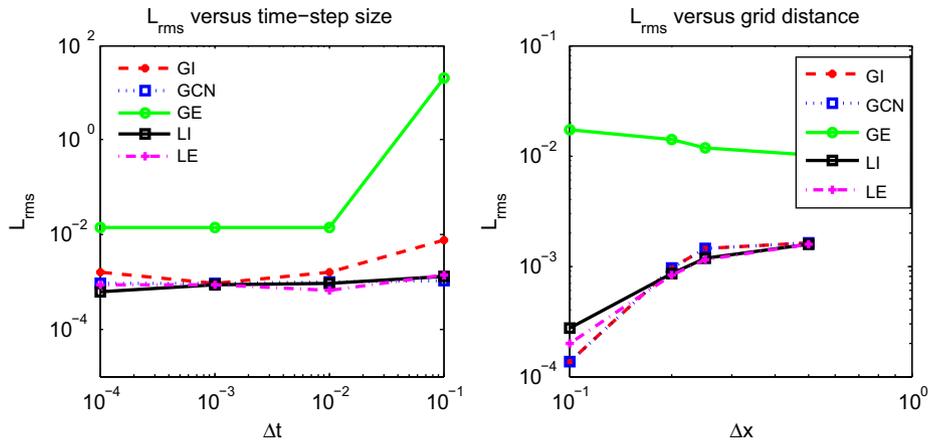


Fig. 3. The rate of convergence with respect to the time-step size and the grid distance in Example 1.

approximates the solution of PDE very well. As N increases, both L_∞ and L_{rms} are seen improving a little, while the computational efficiency decreases as N gets larger. As shown in Table 1, CPU times (in second) of GI are 15%, 650%, 1042%, 1837% more than LI for $N=32, 81, 160, 1600$, respectively. It is clear that in three-dimensional space, LE and LI have shown accurate solution for $N=9025$ whereas in the case of global methods (GCN, GI and GE) no solution is obtained due to excessive memory requirement and large CPU time.

Table 2 shows the accuracy of all methods at $t=1$ regarding different time-step sizes with $N=160$, where the results of GE are not as accurate as the results of other methods. As we have seen, L_∞ and L_{rms} improves slightly as Δt decreases. In Table 3 a comparative performance of the global and the local methods are given for random nodes with random parameter η ranging from 0.0 to 0.8, where $N=160$. This table shows that both the global as well as local methods produce accurate approximation

to the exact solution. As discussed in Introduction, this attribute is only specific to meshless methods [30].

Fig. 2 shows the absolute errors of the five methods together with the exact solution u when $t=1$, $z=0.4$, $\Delta t=10^{-3}$, $N=160$. The pattern of absolute errors of GI, GCN are similar, and the pattern of absolute errors of GE, LI, and LE resembles each other. Clearly, GI, GCN, LI, and LE show smaller absolute errors in the entire domain, where LE is slightly better than LI when larger number of nodes are used. Fig. 3 shows the rate of convergence of the five methods with respect to time-step size and minimum grid distance in the case of uniformly spaced collocation nodes. All the methods perform with a similar rate of convergence except GE.

It is clear that the performance LRBFCM (LE, LI) is marginally better than GRBFCM (GCN, GI, GE) for three-dimensional Dirichlet boundary value problems, especially when large number of collocation points is used. Apart from some improvement in the

accuracy of the LRBFCM, there is a considerable reduction in the memory requirement and CPU time as well.

Another advantage of LRBFCM in the case of uniform node arrangement is that a small matrix (of the size of sub-domain which is 7×7 in the present case) needs to be inverted only once outside the time-loop. Among the global methods (GCN, GI, GE) performances of GCN and GI are similar, whereas accuracy of GE is little lower than both GCN and GI. The global methods are prone to the problem of ill-conditioning and large CPU time in case of larger nodal density.

Example 2. Consider a three-dimensional problem [29]

$$\frac{\partial u(x,y,z,t)}{\partial t} = 0.2 \nabla^2 u(x,y,z,t) + 0.1u, \quad (x,y,z) \in \Omega, \quad t > 0, \quad (43)$$

where $\Omega = [0, \pi] \times [0, \pi] \times [0, \pi]$, with the initial condition:

$$u(x,y,z,0) = \cos x + \sin y + \cos z, \quad (x,y,z) \in \Omega \quad (44)$$

and subject to the boundary condition

$$u(x,y,0,t) = \exp(-0.1t)(\cos x + \sin y + 1), \quad t > 0,$$

$$u(x,y,\pi,t) = \exp(-0.1t)(\cos x + \sin y - 1), \quad t > 0,$$

$$\frac{\partial u(x,0,z,t)}{\partial y} = \exp(-0.1t), \quad t > 0,$$

$$\frac{\partial u(x,\pi,z,t)}{\partial y} = -\exp(-0.1t), \quad t > 0,$$

$$u(0,y,z,t) = \exp(-0.1t)(1 + \sin y + \cos z), \quad t > 0,$$

$$u(\pi,y,z,t) = \exp(-0.1t)(-1 + \sin y + \cos z), \quad t > 0.$$

The analytical solution is given by

$$u(x,y,z,t) = \exp(-0.1t)(\cos x + \sin y + \cos z). \quad (45)$$

The numerical results are shown in Tables 4–6 and Figs. 4 and 5.

In Table 4, we compute the L_∞, L_{rms} errors at $t=1$ and CPU time using different sets of collocation points, N for the time-step size $\Delta t = 10^{-3}$. As we observe from Example 1, as N increases, both L_∞ and L_{rms} are recorded to improve marginally while the computational efficiency decreases at the same time. It is clear that the

Table 4 GI, GCN, GE, LI, LE, L_∞ and L_{rms} , Example 2, $t=1$, with different node densities, $\Delta t = 10^{-3}$.

N	GI	GCN	GE	LI	LE
L_∞					
32	5.13×10^{-1}	4.97×10^{-1}	9.18×10^{-1}	1.29×10^0	6.69×10^{-1}
81	3.09×10^{-1}	2.54×10^{-1}	5.60×10^{-1}	7.13×10^{-1}	5.74×10^{-1}
160	2.22×10^{-1}	1.78×10^{-1}	3.50×10^{-1}	2.89×10^{-1}	4.73×10^{-1}
432	1.15×10^{-1}	8.98×10^{-2}	1.15×10^{-1}	6.83×10^{-2}	3.32×10^{-1}
1215	5.44×10^{-2}	4.34×10^{-2}	5.44×10^{-2}	4.14×10^{-2}	2.16×10^{-1}
6647	–	–	–	2.39×10^{-2}	1.28×10^{-1}
L_{rms}					
32	1.73×10^{-1}	1.57×10^{-1}	3.02×10^{-1}	6.44×10^{-1}	2.89×10^{-1}
81	6.69×10^{-2}	5.14×10^{-2}	1.17×10^{-1}	1.38×10^{-1}	2.64×10^{-1}
160	4.52×10^{-2}	3.14×10^{-2}	6.99×10^{-2}	6.83×10^{-2}	1.72×10^{-1}
432	2.19×10^{-2}	1.30×10^{-2}	2.85×10^{-2}	1.40×10^{-2}	8.67×10^{-2}
1215	1.09×10^{-2}	5.51×10^{-3}	1.09×10^{-2}	7.67×10^{-3}	4.81×10^{-2}
6647	–	–	–	4.40×10^{-3}	2.16×10^{-2}
CPU (s)					
32	0.26	0.29	0.30	1.22	1.39
81	0.80	0.81	0.82	2.72	3.05
160	2.54	2.66	2.30	5.23	6.78
432	14.00	14.43	14.00	7.55	17.01
1215	191.00	192.84	190.00	35.04	52.06
6647	–	–	–	643.26	117.5

Table 5 GI, LI, GE, LG, L_∞ and L_{rms} , Example 2, $t=1$, with different time-steps, $N=1215$.

Δt	GI	GCN	GE	LI	LE
L_∞					
10^{-1}	5.54×10^{-2}	5.43×10^{-2}	5.54×10^{-2}	4.38×10^{-2}	2.16×10^{-1}
10^{-2}	5.55×10^{-2}	5.44×10^{-2}	5.55×10^{-2}	4.41×10^{-2}	2.14×10^{-1}
10^{-3}	5.55×10^{-2}	5.44×10^{-2}	5.55×10^{-2}	4.08×10^{-2}	2.14×10^{-1}
10^{-4}	5.55×10^{-2}	5.44×10^{-2}	5.55×10^{-2}	9.23×10^{-2}	2.14×10^{-1}
L_{rms}					
10^{-1}	1.09×10^{-2}	1.07×10^{-2}	1.09×10^{-2}	8.04×10^{-3}	4.81×10^{-2}
10^{-2}	1.11×10^{-2}	1.09×10^{-2}	1.11×10^{-2}	8.13×10^{-3}	4.79×10^{-2}
10^{-3}	1.11×10^{-2}	1.09×10^{-2}	1.11×10^{-2}	7.65×10^{-3}	4.80×10^{-2}
10^{-4}	1.11×10^{-2}	1.09×10^{-2}	1.11×10^{-2}	2.79×10^{-2}	4.80×10^{-2}

Table 6 GI, LI, GE, LG, L_∞ and L_{rms} , Example 2, $t=1$, with different displacement factors, $\eta, \Delta t = 10^{-2}, N=1215$.

η	GI	GCN	GE	LI	LE
L_∞					
0.0	5.55×10^{-2}	5.44×10^{-2}	5.55×10^{-2}	4.41×10^{-2}	6.15×10^{-1}
0.1	6.85×10^{-2}	7.79×10^{-2}	7.73×10^{-2}	4.20×10^{-1}	$9.84 \times 10^{-1}(c_I=10)$
0.2	9.38×10^{-2}	8.51×10^{-2}	9.97×10^{-2}	7.03×10^{-2}	$9.68 \times 10^{-1}(c_I=10)$
0.3	1.32×10^{-1}	1.40×10^{-1}	1.10×10^{-1}	8.92×10^{-2}	$1.03 \times 10^{-0}(c_I=10)$
L_{rms}					
0.0	1.11×10^{-2}	1.09×10^{-2}	1.11×10^{-2}	8.13×10^{-3}	1.83×10^{-1}
0.1	1.20×10^{-2}	1.21×10^{-2}	1.20×10^{-2}	1.62×10^{-2}	$2.18 \times 10^{-1}(c_I=10)$
0.2	1.47×10^{-2}	1.32×10^{-2}	1.33×10^{-2}	8.79×10^{-3}	$2.12 \times 10^{-1}(c_I=10)$
0.3	1.66×10^{-2}	1.80×10^{-2}	1.74×10^{-2}	1.24×10^{-2}	$2.09 \times 10^{-1}(c_I=10)$

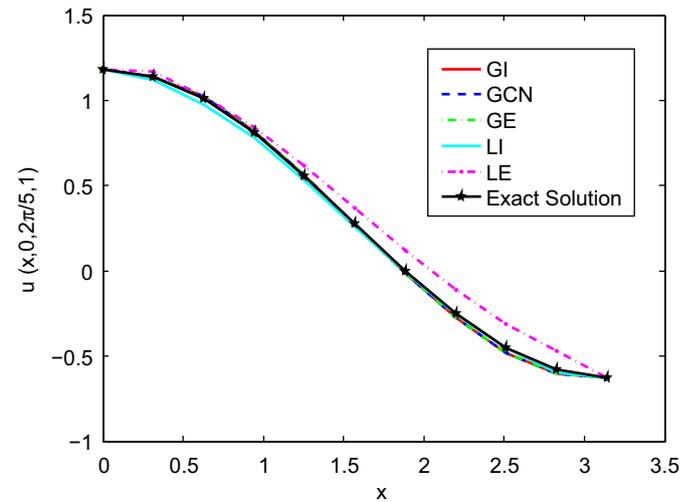


Fig. 4. The analytical solution and the approximate solutions using GI, GCN, GE, LE and LI, $y = \pi, z = 2\pi/5, t=1$ in Example 2, $N=1215, \Delta t = 10^{-3}$.

accuracy of LE is comparable with GRBFCM (GCN, GI, GE) for three-dimensional mixed problem, whereas LI gives very large errors when a very small number of collocation points ($N=32$) is used. On the other hand, with larger number of collocation points LI performs slightly better than their other counter parts. Same scenario has been observed in our earlier study in [48] for two-dimensional case with the mixed boundary conditions. In the case of LRBFCM, there is a considerable reduction in the memory requirement as well as CPU time.

Table 5 shows that all methods are stable with respect to time-step size in the given range where $N=1215, t=1$ are used. However, LE performs slightly less accurate for all the time-step

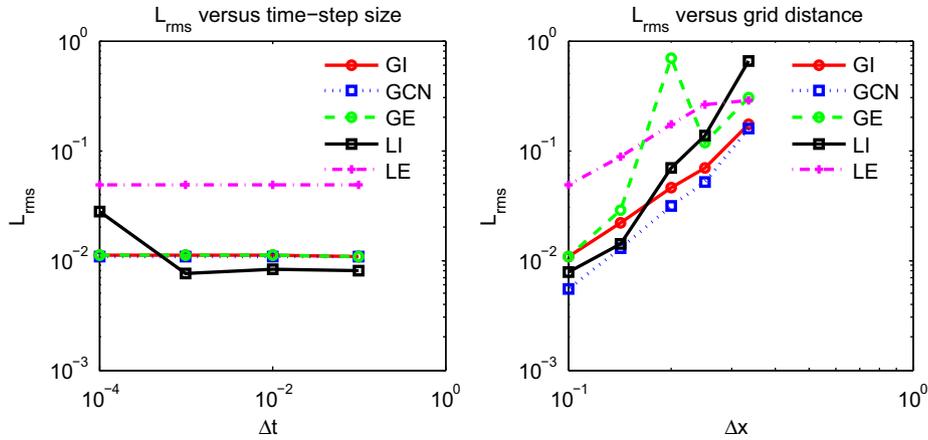


Fig. 5. L_{rms} at $t=1$ in Example 2. Left: the rate of convergence w.r.t. time-step size where $N=1215$. Right: the rate of convergence w.r.t. grid distance, where $\Delta t = 10^{-3}$.

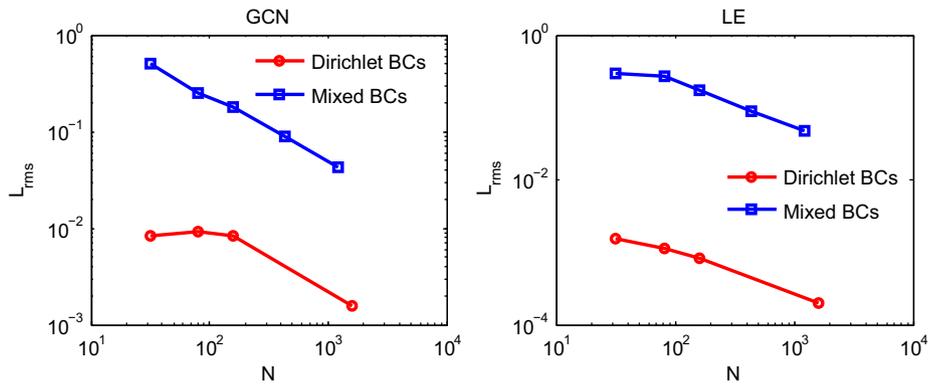


Fig. 6. Performance GCN and LE in terms of L_{rms} for Dirichlet boundary conditions in Example 1 and for the mixed boundary conditions in Example 2, with $\Delta t = 10^{-3}$.

sizes chosen, and LI performs slightly better with respect to L_{rms} when the time-step Δt is larger than 10^{-4} . All the global methods perform with a similar accuracy in all the cases.

In Table 6 a comparative performance of the global and local methods are given for 1215 random nodes with random parameter, η , ranging from 0.0 to 0.3. When the mixed boundary condition is used, the acceptable random parameter has smaller ranges than the problem with Dirichlet boundary condition. Furthermore, both the global and local methods produce enough accurate approximation to the exact solution except LE. LE cannot approximate the solution accurately when the shape parameter $c=100$ is used. However, as shown in the table, LE with a smaller shape parameter, $c=10$ yields acceptable solutions. Thus, LE is very sensitive to the shape parameter in random node distribution case.

Fig. 4 shows the exact solution versus numerical solution when $y = \pi$, where $z = 2\pi/5$, $x \in [0, \pi]$, $t=1$, $\Delta t = 10^{-3}$, $N=1215$. The distribution of absolute errors of GI, GCN, GE, and LI are similar to each other. Clearly, LE shows larger absolute errors throughout the entire domain in comparison with the other methods.

Fig. 5 shows the rate of convergence of the global and local methods with respect to time-step size and minimum grid distance. LI is more sensitive to the grid distance, and it performs better accuracy with smaller time-steps. All other methods perform similarly.

From Fig. 6, it is clear that the gap between the accuracy of the global method GCN in the case of Dirichlet and mixed type of conditions is comparatively less than its local counter part LE. Over all, performance of both the methods is satisfactory for both types of boundary conditions. In the case of global methods, the performance of GCN and GI are similar, whereas accuracy of GE is little lower than

both GCN and GI. Global methods run into troubles due to ill-conditioning and CPU runs out of memory for larger node density.

6. Conclusions

A comparison of the meshless methods based on the global and the local approximations with the three-dimensional parabolic PDEs is performed in the present paper. The global methods run into difficulties for cases with large number of collocation points and the coefficient matrix becomes ill-conditioned. The local methods are based on local interpolations and work well when a large number of nodes is used. They save CPU time since several small matrices are inverted only once outside the time-loop instead of a large matrix. Tests on different benchmark problems show superiority of the local collocation methods for problems with both Dirichlet and mixed type of boundary conditions. In the case of mixed type of boundary conditions, performance of the global methods is comparatively better than their local methods if small number of nodes are used. Nevertheless, global methods are more prone to ill-condition and hence cannot be used for the numerical solution of multi-dimensional problems with denser nodes.

Acknowledgment

This paper represents a part of the programme group P2-0379 Modelling and Simulation of Materials and Processes and project J2-4120 Advanced Modelling and Simulation of Liquid-Solid Processes. Financial support of the Slovenian Grant Agency is kindly acknowledged. Centre of Excellence for Biosensors,

Instrumentation and Process Control is an operation funded by European Union. Financial support is kindly acknowledged.

References

- [1] Ali A, Siraj-ul-Islam, Haq S. A computational meshfree technique for the numerical solution of the two-dimensional coupled Burger's equations. *Int J Comput Methods Eng Sci Mech* 2009;10:406–412.
- [2] Atluri SN. The meshless method MLPG for domain and BIE discretization. USA: Tech Science Press; 2004.
- [3] Atluri SN, Shen S. The meshless method. USA: Forsyth: Tech Science Press; 2002.
- [4] Axelsson O. Iterative solution methods. Cambridge: Cambridge University Press; 1994.
- [5] Buhmann MD. Radial basis functions: theory and implementations. Cambridge University Press; 2003.
- [6] Chen CS, Fan CM, Wen PH. The method of approximate particular solutions for solving certain partial differential equations. *Num Methods Partial Differential Equations* 2011, <http://dx.doi.org/10.1002/num.20631>.
- [7] Divo E, Kassab AJ. An efficient localized RBF meshless method for fluid flow and conjugate heat transfer. *ASME J Heat Trans* 2007;129:124–136.
- [8] Fasshauer GF. Meshfree approximation methods with MATLAB. Singapore: World Scientific Press; 2008.
- [9] Franke C, Schaback R. Convergence order estimates of meshless collocation methods using radial basis functions. *Adv Comput Math* 1998;8:381–399.
- [10] Golberg MA, Chen CS, Karur SR. Improved multiquadric approximation for partial differential equations. *Eng Anal Boundary Elem* 1996;18:9–17.
- [11] Haq S, Siraj-ul-Islam, Ali A. A numerical meshfree technique for the solution of the MEW equation. *Comput Model Eng Sci* 2008;38:1–23.
- [12] Haq S, Siraj-ul-Islam, Uddin M. Numerical solution of nonlinear schrodinger equations by collocation method using radial basis functions. *Comput Model Eng Sci* 2009;44:115–135.
- [13] Hardy RL. Multiquadric equations of topography and other irregular surfaces. *J Geophys Res* 1971;176:1905–1915.
- [14] Cheng RJ, Liew KM. A meshless analysis of three-dimensional transient 01 conduction problems. *Eng Anal Boundary Elem* 2012;36:203–210.
- [15] Kansa EJ. Multiquadrics scattered data approximation scheme with applications to computational fluid-dynamics I, surface approximations and partial derivative estimates. *Comput Math Appl* 1990;19:127–145.
- [16] Kansa EJ. Multiquadrics scattered data approximation scheme with applications to computational fluid-dynamics II, solutions to parabolic, hyperbolic and elliptic partial differential equations. *Comput Math Appl* 1990;19(8–9): 147–161.
- [17] Kansa EJ. Exact explicit time integration of hyperbolic partial differential equations with mesh free radial basis functions. *Eng Anal Boundary Elem* 2007;31:577–585.
- [18] Kharab A. Use of multiple sheets for the solution of a three-dimensional transient heat conduction problem. *Comput Math Appl* 1979;34:71–79.
- [19] Kosec G, Šarler B. Local RBF collocation method for Darcy flow. *Comput Model Eng Sci* 2008;25(3):197–207.
- [20] Kosec G, Šarler B. Convection driven melting of anisotropic metals. *Int J Cast Met Res* 2009;22:279–282.
- [21] Kosec G, Šarler B. Meshless approach to solving freezing with natural convection. *Mater Sci Forum* 2010;22:205–210.
- [22] Kovačević I, Šarler B. 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* 2005;413–414:423–428.
- [23] Li S, Atluri SN. The MLPG mixed collocation method for material orientation and topology optimization of anisotropic solids and structures. *Comput Model Eng Sci* 2008;30:37–56.
- [24] Liu GR. Meshfree methods moving beyond the finite element method. CRC Press; 2003.
- [25] Lorbiecka A, Vertnik R, Gjerkeš H, Manojlović G, Senčić B, Cesar J, et al. Numerical modeling of grain structure and the continuous casting of steel. *Comput Mater Continua* 2009;8:195–208.
- [26] Madych WR, Nelson SA. Multivariate interpolation and conditionally positive definite functions II. *Math Comput* 1990;54:211–230.
- [27] Mai-Duy N, Tran-Cong T. Indirect RBFN method with thin plate splines for numerical solution of differential equations. *Comput Model Eng Sci* 2003;4: 85–102.
- [28] Micchelli CA. Interpolation of scattered data: distance matrix and conditionally positive definite functions. *Constr Approx* 1986;2:11–22.
- [29] Nie Q, Wana FYM, Zhang YT, Liu XF. Compact integration factor methods in high spatial dimensions. *J Comput Phys* 2008;227:5238–5255.
- [30] Reuther K, Šarler B, Rettenmayr M. Solving diffusion problems on an unstructured, amorphous grid by a meshless method. *Int J Thermal Sci* 2012;51: 16–22.
- [31] Sarra SA. Adaptive radial basis function methods for time dependent partial differential equations. *Appl Numer Math* 2005;54((1 June):79–94.
- [32] Siraj-ul-Islam, Ali A, Haq S. A computational modeling of the behavior of the two-dimensional reaction-diffusion Brusselator system. *Appl Math Model* 2010;34:3896–3909.
- [33] Siraj-ul-Islam, Haq S, Ali A. A meshfree method for the numerical solution of the RLW equation. *J Comput Appl Math* 2009;223:997–1012.
- [34] Siraj-ul-Islam, Haq S, Uddin M. A meshfree interpolation method for the numerical solution of the coupled nonlinear partial differential equations. *Eng Anal Boundary Elem* 2009;33:399–409.
- [35] Tarwater AE. A parameter study of Hardy's multiquadric method for scattered data interpolation, Lawrence Livermore National Laboratory, Technical Report UCRL-54670, 82; 1985.
- [36] Trobec R, Kosec G, Šterk M, Šarler B. Comparison of local weak and strong form meshless methods for 2-d diffusion equation. *Eng Anal Boundary Elem* 2012;36:310–321.
- [37] Vertnik R, Šarler B. Meshless local radial basis function collocation method for convective-diffusive solid-liquid phase change problems. *Int J Numer Methods Heat Fluid Flow* 2006;16:617–640.
- [38] Vertnik R, Šarler B. Simulation of continuous casting of steel by a meshless technique. *Int J Cast Met Res* 2009;22:311–313.
- [39] Vertnik R, Šarler B. Solution of incompressible turbulent flow by a mesh-free method. *Comput Model Eng Sci* 2009;44(1):65–96.
- [40] Vertnik R, Založnik M, Šarler B. Solution of transient direct-chill aluminum billet casting problem with simulations material and interphase moving boundaries by a meshless method. *Eng Anal Boundary Elem* 2006;30: 847–855.
- [41] Šarler B. A radial basis function collocation approach in computational fluid dynamics. *Comput Model Eng Sci* 2005;7:185–193.
- [42] Šarler B, Kosec G, Lorbiecka A, Vertnik R. A meshless approach and solution of multiscale solidification modeling. *Mater Sci Forum* 2010;649:211–216.
- [43] Šarler B, Vertnik R. Meshfree explicit local radial basis function collocation method for diffusion problems. *Comput Math Appl* 2006;1269–1282.
- [44] Wang Y, Zhang J. Fast and robust sixth-order multigrid computation for the three-dimensional convection diffusion equation. *Comput Model Eng Sci* 2010;59:127–154.
- [45] Xiang S, Jin YX, Jiang SX, Bi ZY, Wang KM. Meshless global radial point collocation method for three-dimensional partial differential equations. *Eng Anal Boundary Elem* 2011;35:289–297.
- [46] Yao G. Localized meshless method for solving partial differential equations. PhD thesis, University of Southern Mississippi; 2010.
- [47] Yao G, Kolibal J, Chen CS. A localized approach for the method of approximate particular solutions. *Comput Math Appl* 2011;61:2376–2387.
- [48] Yao G, ul Islam S, Šarler B. A comparative study of global and local meshless methods for diffusion-reaction equation. *Comput Model Eng Sci* 2010;59: 127–154.
- [49] Yao G, Šarler B, Chen CS. A comparison of three explicit local meshless methods using radial basis functions. *Eng Anal Boundary Elem* 2011;35: 600–609.