Neural, Parallel, and Scientific Computations 25 (2017) 61-78

LEAST SQUARES INTEGRATION-BASED RBF METHOD FOR SOLVING PARTIAL DIFFERENTIAL EQUATIONS

DONGFANG YUN

Department of Mathematics and Statistics, McMaster University
Hamilton, Ontario L8S 4K1, Canada

ABSTRACT. In this paper, the method of least square and the recently developed integration-based radial basis function method are applied for solving partial differential equations. Several boundary value problems defined in rectangular and L-shaped domains with uniform and random nodes are studied. Superiorities like higher accuracy and convergency are shown through comparisons with existing results. Furthermore, a two-dimensional Burgers' equation is taken as an example to indicate the superior stability and higher accuracy of this proposed method. Numerical results demonstrate that our method works better than classical Kansa's method and adaptive meshing technique.

AMS (MOS) Subject Classification. 39A10.

1. Introduction

Methods for numerical solutions of partial differential equations (PDEs) have been dominated by either finite difference method (FDM), finite element method (FEM) or finite volume method (FVM). These methods are derived by the assumption of local interpolation schemes. Before the implementation of these methods, discretisation of the domain into a number of finite elements is needed, which is not a straightforward task.

Consequently, meshless methods have been introduced in the last decades, which were developed with the objective of eliminating the mesh used in the popular mesh-dependent methods. As a novel class of numerical techniques for solving PDEs, meshless methods have attracted considerable attentions in recent years [1, 2, 3, 4, 5]. The mesh-independent property makes meshless methods very useful, especially when dealing with high-dimensional problems with complex shaped domains.

Radial basis functions (RBFs) have played an important role in the development of meshless methods when solving PDEs, its many applications include surface fitting, turbulence analysis, neural network and so forth. The development of RBFs is due to Hardy [6] for multi-variate interpolation problems with scattered data. Theoretical analysis concerning solvability, accuracy, convergency and error bound

Received June 30, 2016

of RBF approximation (interpolation) have also been built up and vastly studied [7, 8, 9, 10, 11, 12, 13].

The concept of using RBF and collocation method to solve PDEs was firstly introduced and implemented by Kansa [14, 15], which is now called RBFCM or Kansa method. This method is truly meshless and easy-to-use for a broad range of PDEs. [16, 17, 18] provided theoretical foundations of RBF method for solving PDE and derived error estimates. To get more accurate results and capable to solve large scale problem, some special technique has been developed, such as quasi-interpolation method [19], compactly supported RBFs [20], local RBFs [21]. Numerical results have been proved that the method of RBF admits higher error convergence rate, exponential convergence in contrast to the algebraic convergence of Galerkin FEM and superconvergence of $h-p$ FEM.

It is well-known that the differentiation lower the accuracy and is very sensitive to the round-off error. In FDM, the central difference scheme gives the first-order numerical derivatives with the round-offer error $O(h^2)$ while approximating the second-order derivatives with the round-offer error $O(h)$, where h stands for the distance between nodes. However, the integration preserves the approximation accuracy. Moreover, the integration is a smoothing process compared with differentiation. If we plug the RBF approximation into the PDE directly, the approximation accuracy decreases due to differential operators. Wen et al. [22] recently proposed a method called finite integration method (FIM) and successfully applied to study material mechanic problems. By FIM, we firstly transformed the given PDE into an equivalent integration equation. Instead of finite difference schemes, the numerical integration method is then applied. By the simple numerical trapezoidal rule for approximating integrations, FIM was studied and applied to solve one- and two-dimensional PDEs [23, 24]. Even for PDEs of fractional order, results show that FIM gives results with higher accuracy and convergency.

When solving two-dimensional problems by FIM with RBF, it is found that the resultant linear system is nearly singular and rank deficient, which makes the method unstable. To overcome such disadvantages, the method of least squares is applied. Several numerical examples are given where results are compared. Note the special advantage of integration versus differentiation, a two-dimensional Burgers' equation is studied to show superiority of FIM-RBF with least squares for problems with shock wave and layer properties.

In this paper, the integration-based meshless method is further investigated for solving PDEs. This paper is organized as follows. In Section 2, the methodology of integration-based method with RBF approximation is introduced and applied to solve PDEs. In Section 3, further discussions on the linear system that are deduced by integration-based method with RBF are given. Numerical examples are given in

Section 4. Some concluding remarks with suggested future works are given in the final conclusion section.

2. **Finite integration method with RBFs in one- and two-dimensional spaces**

2.1. **The methodology of integration-based RBF method.** Suppose $u(\mathbf{x})$ is an unknown function, a set of pairwise distinct nodes $\boldsymbol{X} \triangleq \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\}$ are given in $\Omega \cup \partial\Omega$, where $\mathbf{x} = (x, y)$, $\mathbf{x}_j = (x_j, y_j)$, $j = 1, 2, \ldots, N$ stand for nodes, $\Omega \in \mathbb{R}^2$. By the idea of RBF approximation, $u(\mathbf{x})$ has an approximation as:

$$\tilde{u}(\mathbf{x}) = \sum_{j=1}^{N} \lambda_j \, \phi(\|\mathbf{x} - \mathbf{x}_j\|) \,, \tag{2.1}$$

where $\phi(\|\mathbf{x} - \mathbf{x}_j\|)$ is referred to as RBF centered at $\mathbf{x}_j$, $\|\mathbf{x} - \mathbf{x}_j\|$ is the usual Euclidean distance between $\mathbf{x}$ and $\mathbf{x}_j$, λ_j, $j = 1, 2, \ldots, N$ are unknown coeficients to be determined. This set of nodes $\{\mathbf{x}_j\}_{j=1}^{N}$ are called source nodes (or centers), $\mathbf{x}$ is called collocation node (or approximation node).

Usually, source nodes are given and fixed. If the collocation node $\mathbf{x}$ is selected to coincide with centers, unknown coefficients $\{\lambda_j\}_{j=1}^{N}$ can be determined from interpolation conditions $u^*(\mathbf{x}_i) = u_i$, $i = 1, 2, \ldots, N$, which leads to the following symmetric linear system in the matrix form

$$\boldsymbol{\Phi} \, \boldsymbol{\Lambda} = \mathbf{U} \,, \tag{2.2}$$

where the coefficient matrix $\boldsymbol{\Phi} = (\phi_{ij})_{N\times N} = (\phi(\|\mathbf{x}_i - \mathbf{x}_j\|))_{N\times N}$, $\Lambda = [\lambda_1, \lambda_2, \ldots, \lambda_N]^T$, $\mathbf{U} = [u_1, u_2, \ldots, u_N]^T$ and u_j stands for the function value at $\mathbf{x}_j$, $j = 1, 2, \ldots, N$. In this paper, multiquadric (MQ) is used as RBF. The unknown coefficients can be computed by

$$\boldsymbol{\Lambda} = \boldsymbol{\Phi}^{-1} \, \mathbf{U} \,, \tag{2.3}$$

where $\boldsymbol{\Phi}^{-1} \triangleq (\phi_{ij}^{-1})_{N\times N}$ is denoted as the inverse of $\boldsymbol{\Phi}$ with elements are denoted by ϕ_{ij}^{-1}.

On the other hand, if the collocation nodes

$$\mathbf{X}^* \triangleq \{\mathbf{x}_1^*, \mathbf{x}_2^*, \ldots, \mathbf{x}_M^*\} \,, \tag{2.4}$$

are chosen to be different from source nodes, i.e., $\mathbf{X}^* \neq \mathbf{X}$, see Figure 1. Substitution of $\mathbf{x}_i^*$ into (2.1) yields a linear system $\boldsymbol{\Phi}^* \, \boldsymbol{\Lambda} = \mathbf{U}^*$, where $\boldsymbol{\Phi}^* \triangleq (\phi_{ij})_{M\times N} = (\phi(\|x_i^* - x_j\|))_{M\times N}$, $\mathbf{U}^* = [u(\mathbf{x}_1^*), u(\mathbf{x}_2^*), \ldots, u(\mathbf{x}_M^*)]^T$. The coefficient matrix $\boldsymbol{\Phi}^*$ is overdetermined ($M > N$) or underdetermined ($M < N$). In this case, the method of least squares, QR decomposition or singular value decomposition are used to get $\boldsymbol{\Lambda}$.

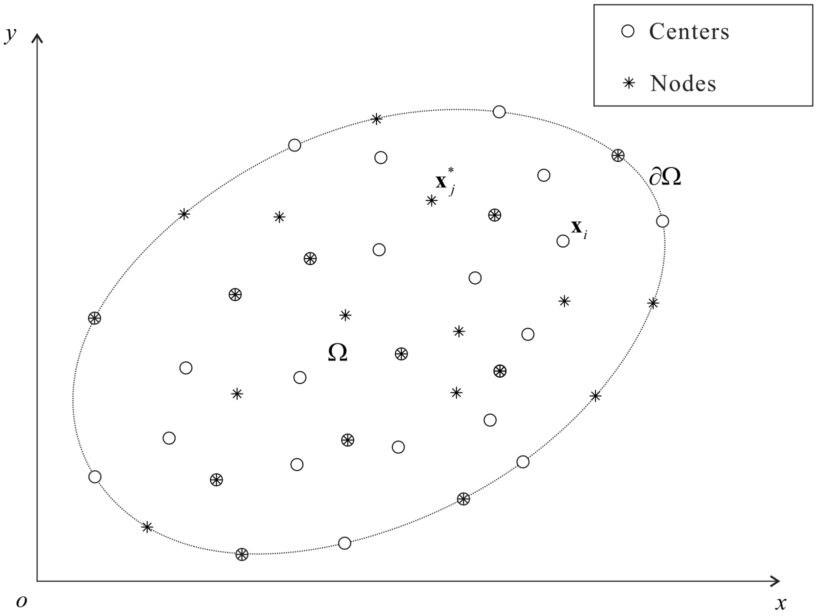


FIGURE 1. Distribution of centers and nodes.

As (2.1) gives an approximation of $u(\mathbf{x})$ and RBFs are smooth, we have

$$\frac{\partial}{\partial\tau}\tilde{u}(\mathbf{x}) = \sum_{j=1}^{N}\lambda_j \frac{\partial}{\partial\tau}\phi(\|\mathbf{x}-\mathbf{x}_j\|)\,, \tag{2.5}$$

is an approximation of $\frac{\partial}{\partial\tau}u(\mathbf{x})$, where $\tau = x$ or y. Evaluation of the equation (2.5) at collocation nodes $\mathbf{x}_i^*$ gives a linear system as:

$$\frac{\partial}{\partial\tau}\mathbf{U}^* = \frac{\partial}{\partial\tau}\mathbf{\Phi}^*\,\mathbf{\Lambda}\,, \tag{2.6}$$

where $\frac{\partial}{\partial\tau}\mathbf{U}^* = [\frac{\partial}{\partial\tau}u(\mathbf{x}_1^*), \frac{\partial}{\partial\tau}u(\mathbf{x}_2^*), \ldots, \frac{\partial}{\partial\tau}u(\mathbf{x}_M^*)]^T$, and the coefficient matrix $\frac{\partial}{\partial\tau}\mathbf{\Phi}^* = (\frac{\partial}{\partial\tau}\phi(\|\mathbf{x}_i^*-\mathbf{x}_j\|))_{M\times N}$:

$$\frac{\partial}{\partial\tau}\mathbf{\Phi}^* = \begin{pmatrix} \frac{\partial}{\partial\tau}\phi(\|\mathbf{x}_1^*-\mathbf{x}_1\|) & \frac{\partial}{\partial\tau}\phi(\|\mathbf{x}_1^*-\mathbf{x}_2\|) & \cdots & \frac{\partial}{\partial\tau}\phi(\|\mathbf{x}_1^*-\mathbf{x}_N\|) \\ \frac{\partial}{\partial\tau}\phi(\|\mathbf{x}_2^*-\mathbf{x}_1\|) & \frac{\partial}{\partial\tau}\phi(\|\mathbf{x}_2^*-\mathbf{x}_2\|) & \cdots & \frac{\partial}{\partial\tau}\phi(\|\mathbf{x}_2^*-\mathbf{x}_N\|) \\ \frac{\partial}{\partial\tau}\phi(\|\mathbf{x}_3^*-\mathbf{x}_1\|) & \frac{\partial}{\partial\tau}\phi(\|\mathbf{x}_3^*-\mathbf{x}_2\|) & \cdots & \frac{\partial}{\partial\tau}\phi(\|\mathbf{x}_3^*-\mathbf{x}_N\|) \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial}{\partial\tau}\phi(\|\mathbf{x}_M^*-\mathbf{x}_1\|) & \frac{\partial}{\partial\tau}\phi(\|\mathbf{x}_M^*-\mathbf{x}_2\|) & \cdots & \frac{\partial}{\partial\tau}\phi(\|\mathbf{x}_M^*-\mathbf{x}_N\|) \end{pmatrix}_{M\times N},$$

Replace Λ by (2.3) in (2.6):

$$\frac{\partial}{\partial\tau}\mathbf{U}^* = \frac{\partial}{\partial\tau}\mathbf{\Phi}^*\,\mathbf{\Lambda} = \frac{\partial}{\partial\tau}\mathbf{\Phi}^*\,\mathbf{\Phi}^{-1}\mathbf{U} \triangleq \boldsymbol{D}_\tau^*\mathbf{U}\,. \tag{2.7}$$

where $\boldsymbol{D}_\tau^* \triangleq \frac{\partial}{\partial\tau}\mathbf{\Phi}^*\,\mathbf{\Phi}^{-1}$. It's not difficult to see the size of this differentiation matrix is $M \times N$, where M is the number of collocation nodes and N is the number of source nodes. The differentiation matrix $\boldsymbol{D}_\tau^*$ is either underdetermined ($M < N$) or overdetermined ($M > N$) matrix.

Differential matrices corresponding to higher-order derivatives, partial derivatives, and mixed partial derivatives can be derived in the similar manner.

In contrast to differentiation, we consider the integration of RBF approximation. $\int_{x_0}^{x} u(\xi,y)\,d\xi$ and $\int_{y_0}^{y} u(x,\eta)\,d\eta$ are approximated by

$$\begin{aligned}
\int_{x_0}^{x} \tilde{u}(\xi,y)\,d\xi &= \sum_{j=1}^{N} \lambda_j \int_{x_0}^{x} \phi(\|\boldsymbol{\xi}-\mathbf{x}_j\|)\,d\xi\,, \quad \boldsymbol{\xi}=(\xi,y),\\
\int_{y_0}^{y} \tilde{u}(x,\eta)\,d\eta &= \sum_{j=1}^{N} \lambda_j \int_{y_0}^{y} \phi(\|\boldsymbol{\eta}-\mathbf{x}_j\|)\,d\xi\,, \quad \boldsymbol{\eta}=(x,\eta).
\end{aligned} \tag{2.8}$$

Substituting $\mathbf{x}_i^* \in \mathbf{X}_i^*$ into (2.8), we have

$$\begin{aligned}
\int_{x_0}^{\mathbf{X}^*} \tilde{u}(\xi,y)\,d\xi &\triangleq \overline{\mathbf{\Phi}}_x^* \mathbf{\Lambda}\,,\\
\int_{y_0}^{\mathbf{X}^*} \tilde{u}(x,\eta)\,d\eta &\triangleq \overline{\mathbf{\Phi}}_y^* \mathbf{\Lambda}\,,
\end{aligned} \tag{2.9}$$

where the left-hand side of above equalities are column vectors given by

$$\begin{aligned}
\int_{x_0}^{\mathbf{X}^*} u^*(\xi,y)\,d\xi &\triangleq \left[\int_{x_0}^{x_1^*} u^*(\xi,y_i^*)\,d\xi, \int_{x_0}^{x_2^*} u^*(\xi,y_i^*)\,d\xi, \ldots, \int_{x_0}^{x_M^*} u^*(\xi,y_i^*)\,d\xi\right]^T,\\
\int_{y_0}^{\mathbf{X}^*} u^*(x,\eta)\,d\eta &\triangleq \left[\int_{y_0}^{y_1^*} u^*(x_i^*,\eta)\,d\eta, \int_{y_0}^{y_2^*} u^*(x_i^*,\eta)\,d\eta, \ldots, \int_{y_0}^{y_M^*} u^*(x_i^*,\eta)d\eta\right]^T,
\end{aligned}$$

and coefficient matrices are $\overline{\mathbf{\Phi}}_x^* = \left(\int_{x_0}^{x_i^*} \phi(\|\boldsymbol{\xi}_i^*-\mathbf{x}_j\|)\,d\xi\right)_{N\times N}$, $\boldsymbol{\xi}_i^* = (\xi, y_i)$, $\overline{\mathbf{\Phi}}_y^* = \left(\int_{y_0}^{y_i^*} \phi(\|\boldsymbol{\eta}_i^*-\mathbf{x}_j\|)\,d\eta\right)_{N\times N}$, $\boldsymbol{\eta}_i^* = (\eta, y_i)$:

$$\overline{\mathbf{\Phi}}_x^* = \begin{pmatrix}
\int_{x_0}^{x_1^*} \phi(\|\boldsymbol{\xi}_1^*-\mathbf{x}_1\|)\,d\xi & \int_{x_0}^{x_1^*} \phi(\|\boldsymbol{\xi}_1^*-\mathbf{x}_2\|)\,d\xi & \cdots & \int_{x_0}^{x_1^*} \phi(\|\boldsymbol{\xi}_1^*-\mathbf{x}_N\|)\,d\xi\\
\int_{x_0}^{x_2^*} \phi(\|\boldsymbol{\xi}_2^*-\mathbf{x}_1\|)\,d\xi & \int_{x_0}^{x_2^*} \phi(\|\boldsymbol{\xi}_2^*-\mathbf{x}_2\|)\,d\xi & \cdots & \int_{x_0}^{x_2^*} \phi(\|\boldsymbol{\xi}_2^*-\mathbf{x}_N\|)\,d\xi\\
\int_{x_0}^{x_3^*} \phi(\|\boldsymbol{\xi}_3^*-\mathbf{x}_1\|)\,d\xi & \int_{x_0}^{x_3^*} \phi(\|\boldsymbol{\xi}_3^*-\mathbf{x}_2\|)\,d\xi & \cdots & \int_{x_0}^{x_3^*} \phi(\|\boldsymbol{\xi}_3^*-\mathbf{x}_N\|)\,d\xi\\
\vdots & \vdots & \ddots & \vdots\\
\int_{x_0}^{x_M^*} \phi(\|\boldsymbol{\xi}_M^*-\mathbf{x}_1\|)\,d\xi & \int_{x_0}^{x_M^*} \phi(\|\boldsymbol{\xi}_M^*-\mathbf{x}_2\|)\,d\xi & \cdots & \int_{x_0}^{x_M^*} \phi(\|\boldsymbol{\xi}_M^*-\mathbf{x}_N\|)\,d\xi
\end{pmatrix}_{N\times N},$$

$$\overline{\mathbf{\Phi}}_y^* = \begin{pmatrix}
\int_{y_0}^{y_1^*} \phi(\|\boldsymbol{\eta}_1^*-\mathbf{x}_1\|)\,d\eta & \int_{y_0}^{y_1^*} \phi(\|\boldsymbol{\eta}_1^*-\mathbf{x}_2\|)\,d\eta & \cdots & \int_{y_0}^{y_1^*} \phi(\|\boldsymbol{\eta}_1^*-\mathbf{x}_N\|)\,d\eta\\
\int_{y_0}^{y_2^*} \phi(\|\boldsymbol{\eta}_2^*-\mathbf{x}_1\|)\,d\eta & \int_{y_0}^{y_2^*} \phi(\|\boldsymbol{\eta}_2^*-\mathbf{x}_2\|)\,d\eta & \cdots & \int_{y_0}^{y_2^*} \phi(\|\boldsymbol{\eta}_2^*-\mathbf{x}_M\|)\,d\eta\\
\int_{y_0}^{y_3^*} \phi(\|\boldsymbol{\eta}_3^*-\mathbf{x}_1\|)\,d\eta & \int_{y_0}^{y_3^*} \phi(\|\boldsymbol{\eta}_3^*-\mathbf{x}_2\|)\,d\eta & \cdots & \int_{y_0}^{y_3^*} \phi(\|\boldsymbol{\eta}_3^*-\mathbf{x}_N\|)\,d\eta\\
\vdots & \vdots & \ddots & \vdots\\
\int_{y_0}^{y_M^*} \phi(\|\boldsymbol{\eta}_M^*-\mathbf{x}_1\|)\,d\eta & \int_{y_0}^{y_M^*} \phi(\|\boldsymbol{\eta}_M^*-\mathbf{x}_2\|)\,d\eta & \cdots & \int_{y_0}^{y_M^*} \phi(\|\boldsymbol{\eta}_M^*-\mathbf{x}_N\|)\,d\eta
\end{pmatrix}_{N\times N}.$$

Replacing the vector of unknowns $\mathbf{\Lambda}$ by (2.3), we have

$$\begin{aligned} \int_{x_0}^{\mathbf{X}^*} u^*(\xi, y)\, d\xi &= \overline{\mathbf{\Phi}}_x^*\, \mathbf{\Phi}^{-1}\, \mathbf{U} \triangleq \boldsymbol{A}_x^*\, \mathbf{U}\,, \\ \int_{y_0}^{\mathbf{X}^*} u^*(x, \eta)\, d\eta &= \overline{\mathbf{\Phi}}_y^*\, \mathbf{\Phi}^{-1}\, \mathbf{U} \triangleq \boldsymbol{A}_y^*\, \mathbf{U}\,, \end{aligned} \tag{2.10}$$

where $\boldsymbol{A}_\tau^* \triangleq \overline{\mathbf{\Phi}}_\tau^* \mathbf{\Phi}^{-1}$ is called integration matrices, $\tau = x, y$.

For double-layer integrations, $\int_{x_0}^{x} \int_{x_0}^{\xi_2} u(\xi_1, y)\, d\xi_1 d\xi_2$ and $\int_{y_0}^{y} \int_{x_0}^{x} u(\xi_1, \eta_1)\, d\xi_1 d\eta_1$ are taken as examples. Following the same integration method (2.8)–(2.10), we know there exist two matrices such that

$$\begin{aligned} \boldsymbol{A}_{xx}^* &= \overline{\mathbf{\Phi}}_{xx}^* \mathbf{\Phi}^{-1}\,, \\ \boldsymbol{A}_{xy}^* &= \overline{\mathbf{\Phi}}_{xy}^* \mathbf{\Phi}^{-1}\,, \end{aligned} \tag{2.11}$$

where

$$\begin{aligned} \overline{\mathbf{\Phi}}_{xx}^* &= \left(\int_{x_0}^{x_i^*} \int_{x_0}^{\xi_2} \phi\left(\|\boldsymbol{\xi}_i - \mathbf{x}_j\| \right)\, d\xi_1 d\xi_2 \right)_{M\times N}, \quad \boldsymbol{\xi}_i = (\xi_1, y_i^*), \mathbf{x}_j = (x_j, y_j)\,, \\ \overline{\mathbf{\Phi}}_{xy}^* &= \left(\int_{y_0}^{y_i^*} \int_{x_0}^{x_i^*} \phi\left(\|\mathbf{x} - \mathbf{x}_j\| \right)\, d\xi_1 d\eta_1 \right)_{M\times N}, \quad \mathbf{x} = (\xi_1, \eta_1), \mathbf{x}_j = (x_j, y_j)\,. \end{aligned}$$

Similarly, multi-layer integration can be obtained easily following the same formula as above.

In practice, integration is more complicated and time consuming than differentiation. Reason lies in the global property of the integral operator compared with the local property of differential operator. In this paper, it is worth pointing out that integration matrix corresponding to the multi-layer integration can be approximated by multiplications of integration matrices that corresponding to lower-order integral operators. As an example, we rewrite the double integration as

$$\begin{aligned} \overline{\overline{u}}(x_i^*, y_i^*) &\triangleq \int_{x_0}^{x_i^*} \int_{x_0}^{\xi_2} u(\xi_1, y_i^*)\, d\xi_1 d\xi_2 \\ &= \int_{x_0}^{x_i^*} \left(\int_{x_0}^{\xi_2} u(\xi_1, y_i^*)\, d\xi_1 \right) d\xi_2 \\ &\triangleq \int_{x_0}^{x_i^*} \overline{u}(\xi_2, y_i^*)\, d\xi_2, \quad i = 1, 2, \ldots, M\,. \end{aligned} \tag{2.12}$$

where $\overline{u}(\xi_2, y) \triangleq \int_{x_0}^{\xi_2} u(\xi_1, y)\, d\xi_1$. Then from (2.10) we have

$$\overline{\overline{\mathbf{U}}} = \boldsymbol{A}_x^*\, \overline{\mathbf{U}}\,,$$

where $\overline{\overline{\mathbf{U}}} = [\overline{\overline{u}}(x_1^*, y_1^*), \overline{\overline{u}}(x_2^*, y_2^*), \ldots, \overline{\overline{u}}(x_M^*, y_M^*)]^T$, $\overline{\mathbf{U}} = [\overline{u}(x_1, y_1), \overline{u}(x_2, y_2), \ldots,$ $\overline{u}(x_N, y_N)]^T$. Moreover, $\overline{u}$ is an integration of u, so according to (2.10) we have

$$\overline{\mathbf{U}} = \boldsymbol{A}_x\, \mathbf{U}\,,$$

where $\boldsymbol{A}_x$ instead of $\boldsymbol{A}_x^*$ is used to represent integration matrix if $\mathbf{X}^* = \mathbf{X}$, i.e., source nodes are used as collocation nodes. Hence we have

$$\overline{\overline{\overline{\mathbf{U}}}} = \boldsymbol{A}_x^* \, \boldsymbol{A}_x \, \mathbf{U}, \tag{2.13}$$

from which we see the corresponding integration matrix is $\boldsymbol{A}_x^* \boldsymbol{A}_x$. Considering (2.11), we have $\boldsymbol{A}_{xx}^* \approx \boldsymbol{A}_x^* \boldsymbol{A}_x$. Following the same analysis, $\boldsymbol{A}_{xy}^*$ can be replaced by $\boldsymbol{A}_y^* \boldsymbol{A}_x$ or $\boldsymbol{A}_x^* \boldsymbol{A}_y$. Following the same analysis, for triple integration $\int_{y_0}^{y} \int_{y_0}^{\eta_2} \int_{x_0}^{x} \cdot \, d\xi d\eta_1 d\eta_2$, it's corresponding interpolation could be computed through either $\boldsymbol{A}_y^* \boldsymbol{A}_y \boldsymbol{A}_x$, $\boldsymbol{A}_{yy}^* \boldsymbol{A}_x$ or A_{yyx}^*.

2.2. **Application of integration-based RBF method for solving PDEs.** Given a boundary value problem (BVP):

$$\mathcal{L}u(\mathbf{x}) = f(\mathbf{x}) \,, \quad \text{in } \Omega \,, \tag{2.14}$$

$$\mathcal{B}u(\mathbf{x}) = g(\mathbf{x}) \,, \quad \text{on } \partial\Omega. \tag{2.15}$$

where $\mathcal{L}$ and $\mathcal{B}$ are arbitrary differential operators in the domain Ω and on $\partial\Omega$. The operator $\mathcal{B}$ can specify Dirichlet, Neumann, Robin, or mixed boundary conditions. Time dependent PDEs can be solved by the method of lines (MOL) technique. For clarity we only focus on steady state (elliptic type) problems. Moreover, one may assume $\mathcal{L}$ is a linear differential operator while some kinds of linearization techniques is needed to seek the solution iteratively.

To better illustrate how to apply the integration-based RBF method to solve a PDE, we study the following simple linear PDE:

$$\begin{aligned} & a_1(x,y)\frac{\partial^2 u}{\partial x^2} + a_2(x,y)\frac{\partial^2 u}{\partial y^2} + b_1(x,y)\frac{\partial u}{\partial x} \\ & \quad + b_2(x,y)\frac{\partial u}{\partial y} + c(x,y)u = f(x,y), \quad (x,y) \in \Omega \,, \end{aligned} \tag{2.16}$$

with Dirichlet boundary condition $u(x,y) = g(x,y)$ on $\partial\Omega$, where a_1, a_2, b_1, b_2, c, f and g are all smooth functions defined in Ω and bounded above. Integrating (2.16) twice for x and y respectively, the differential equation is transformed into equivalent integral function through integration by parts

$$\begin{aligned} & \int_{y_0}^{y} \int_{y_0}^{\eta_2} \Bigg[a_1(x,\eta_1)u(x,\eta_1) - 2\int_{x_0}^{x} \frac{\partial a_1}{\partial x}(\xi_2,\eta_1)u(\xi_2,\eta_1)d\xi_2 \\ & \quad + \int_{x_0}^{x} \int_{x_0}^{\xi_2} \frac{\partial^2 a_1}{\partial x^2}(\xi_1,\eta_1)u(\xi_1,\eta_1)d\xi_1 d\xi_2 \Bigg] d\eta_1 d\eta_2 \\ & \quad + \int_{x_0}^{x} \int_{x_0}^{\xi_2} \Bigg[a_2(\xi_1,y)u(\xi_1,y) - 2\int_{y_0}^{y} \frac{\partial a_2}{\partial y}(\xi_1,\eta_2)u(\xi_1,\eta_2)d\eta_2 \\ & \quad + \int_{y_0}^{y} \int_{y_0}^{\eta_2} \frac{\partial^2 a_2}{\partial y^2}(\xi_1,\eta_1)u(\xi_1,\eta_1)d\eta_1 d\eta_2 \Bigg] d\xi_1 d\xi_2 \end{aligned}$$

$$
\begin{aligned}
&+\int_{y_0}^{y}\int_{y_0}^{\eta_2}\int_{x_0}^{x}\Bigg[b_1(\xi_2,\eta_1)u(\xi_2,\eta_1)\\
&\quad-\int_{x_0}^{\xi_2}\frac{\partial b_1}{\partial x}(\xi_1,\eta_1)u(\xi_1,\eta_1)d\xi_1 d\xi_2\Bigg]d\xi_2 d\eta_1 d\eta_2\\
&+\int_{x_0}^{x}\int_{x_0}^{\xi_2}\int_{y_0}^{y}\Bigg[b_2(\xi_1,\eta_2)u(\xi_1,\eta_2)\\
&\quad-\int_{y_0}^{\eta_2}\frac{\partial b_2}{\partial y}(\xi_1,\eta_1)u(\xi_1,\eta_1)d\eta_1 d\eta_2\Bigg]d\eta_2 d\xi_1 d\xi_2\\
&+\int_{y_0}^{y}\int_{y_0}^{\eta_2}\int_{x_0}^{x}\int_{x_0}^{\xi_2}c(\xi_1,\eta_1)u(\xi_1,\eta_1)d\xi_1 d\xi_2 d\eta_1 d\eta_2\\
=&\int_{y_0}^{y}\int_{y_0}^{\eta_2}\int_{x_0}^{x}\int_{x_0}^{\xi_2}f(\xi_1,\eta_1)d\xi_1 d\xi_2 d\eta_1 d\eta_2\\
&+xf_0(y)+f_1(y)+yg_0(x)+g_1(x)\,, \qquad (2.17)
\end{aligned}
$$

where $f_0(y)$, $f_1(y)$, $g_0(x)$, $g_1(x)$ are functions emerge in the process of integration.

One method to deal with these four one-dimensional functions is using RBF approximation method. Suppose there are respectively s_0, s_1, r_0 and r_1 centers used for RBF interpolation, i.e. $\{y_j^{(0)}\}_{j=1}^{s_0}$, $\{y_j^{(1)}\}_{j=1}^{s_1}$, $\{x_j^{(0)}\}_{j=1}^{r_0}$ and $\{x_j^{(1)}\}_{j=1}^{r_1}$:

$$
\begin{aligned}
f_0^*(y)&=\sum_{j=1}^{s_0}\alpha_j^{(0)}\phi_j^{(0)}(y), \quad f_1^*(y)=\sum_{j=1}^{s_1}\alpha_j^{(1)}\phi_j^{(1)}(y),\\
g_0^*(x)&=\sum_{j=1}^{r_0}\beta_j^{(0)}\phi_j^{(0)}(x), \quad g_1^*(x)=\sum_{j=1}^{r_1}\beta_j^{(1)}\phi_j^{(1)}(x)\,,
\end{aligned}
$$

where $\phi_j^{(0)}(y)=\phi(|y-y_j^{(0)}|)$, $\phi_j^{(1)}(y)=\phi(|y-y_j^{(1)}|)$, $\phi_j^{(0)}(x)=\phi(|x-x_j^{(0)}|)$, $\phi_j^{(1)}(x)=\phi(|x-x_j^{(1)}|)$ and $\{\alpha_j^{(0)}\}_{j=1}^{s_0}$, $\{\alpha_j^{(1)}\}_{j=1}^{s_1}$, $\{\beta_j^{(0)}\}_{j=1}^{r_0}$, $\{\beta_j^{(1)}\}_{j=1}^{r_0}$ are coefficients to be determined. Given a set of nodes $\mathbf{X}^*=\{\mathbf{x}_1^*,\mathbf{x}_2^*,\ldots,\mathbf{x}_M^*\}$, we have four linear systems:

$$
\mathbf{F}_0^*=\boldsymbol{\Phi}_y^{(0)}\,\boldsymbol{\alpha}^{(0)}, \quad \mathbf{F}_1^*=\boldsymbol{\Phi}_y^{(1)}\,\boldsymbol{\alpha}^{(1)}, \quad \mathbf{G}_0^*=\boldsymbol{\Phi}_x^{(0)}\,\boldsymbol{\beta}^{(0)}, \quad \mathbf{G}_1^*=\boldsymbol{\Phi}_x^{(1)}\,\boldsymbol{\beta}^{(1)}\,, \qquad (2.18)
$$

where $\mathbf{F}_0^*=[f_0^*(y_1^*),f_0^*(y_2^*),\ldots,f_0^*(y_1^*)]$, $\mathbf{F}_1^*=[f_1^*(y_1^*),f_1^*(y_2^*),\ldots,f_1^*(y_1^*)]$, $\mathbf{G}_0^*=[g_0^*(x_1^*), g_0^*(x_2^*),\ldots,g_0^*(x_1^*)]$, $\mathbf{G}_1^*=[g_1^*(x_1^*),g_1^*(x_2^*),\ldots,g_1^*(x_1^*)]$, $\boldsymbol{\alpha}^{(0)}=[\alpha_1^{(0)},\alpha_2^{(0)},\ldots,\alpha_{s_0}^{(0)}]$, $\boldsymbol{\alpha}^{(1)}=[\alpha_1^{(1)},\alpha_2^{(1)},\ldots,\alpha_{s_1}^{(1)}]$, $\boldsymbol{\beta}^{(0)}=[\beta_1^{(0)},\beta_2^{(0)},\ldots,\beta_{r_0}^{(0)}]$, $\boldsymbol{\beta}^{(1)}=[\beta_1^{(1)},\beta_2^{(1)},\ldots,\beta_{r_1}^{(1)}]$, and coefficients matrices are

$$
\begin{aligned}
\Phi_y^{(0)}&=\Big(\phi_j^{(0)}(y_i^*)\Big)_{M\times s_0}, \quad \Phi_y^{(1)}=\Big(\phi_j^{(1)}(y_i^*)\Big)_{M\times s_1},\\
\Phi_x^{(0)}&=\Big(\phi_j^{(0)}(x_i^*)\Big)_{M\times r_0}, \quad \Phi_x^{(1)}=\Big(\phi_j^{(1)}(x_i^*)\Big)_{M\times r_1}.
\end{aligned}
$$

Figure 2 shows how interpolations of $f_0(y)$, $f_1(y)$, $g_0(x)$ and $g_1(x)$ are implemented:

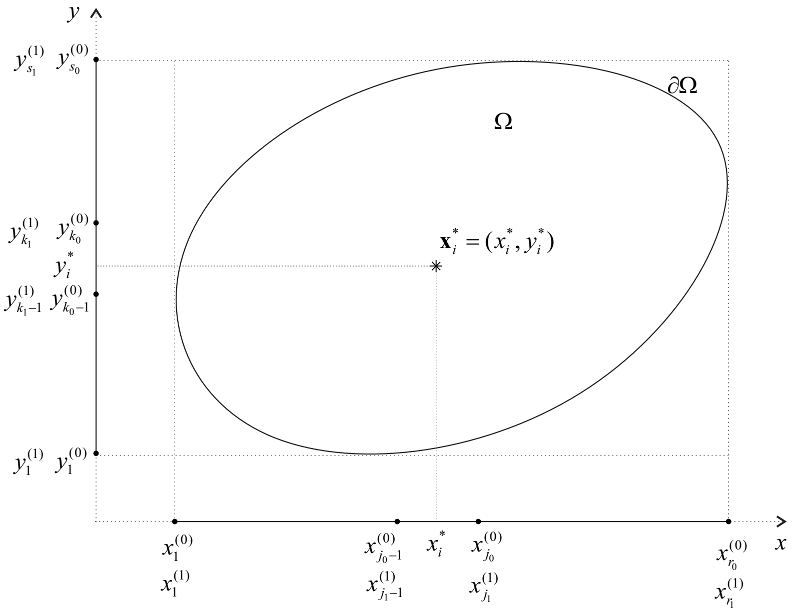

y
$y_{s_1}^{(1)}$ $y_{s_0}^{(0)}$
$\partial\Omega$
Ω
$y_{k_1}^{(1)}$ $y_{k_0}^{(0)}$
$\mathbf{x}_i^* = (x_i^*, y_i^*)$
y_i^*
$y_{k_1-1}^{(1)}$ $y_{k_0-1}^{(0)}$
$y_1^{(1)}$ $y_1^{(0)}$
$x_1^{(0)}$ $x_1^{(1)}$
$x_{j_0-1}^{(0)}$ $x_{j_1-1}^{(1)}$
x_i^*
$x_{j_0}^{(0)}$ $x_{j_1}^{(1)}$
$x_{r_0}^{(0)}$ $x_{r_1}^{(1)}$
x

Figure 2. Interpolation for one dimensional functions.

If replace u in (2.17) by its RBFs expansion (2.1), replace integrations by corresponding interpolation matrices with multiple integration matrices are replaces by multiplying lower integration matrices, we have M equalities by plugging $\mathbf{x}_i^*$, $i = 1, 2, \ldots, M$ into (2.17), which leads to the linear system:

$$\begin{pmatrix} \boldsymbol{A} & \boldsymbol{X}^*\boldsymbol{\Phi}_y^{(0)} & \boldsymbol{\Phi}_y^{(1)} & \boldsymbol{Y}^*\boldsymbol{\Phi}_x^{(0)} & \boldsymbol{\Phi}_x^{(1)} \end{pmatrix} \begin{pmatrix} \mathbf{U} \\ \boldsymbol{\alpha}^{(0)} \\ \boldsymbol{\alpha}^{(1)} \\ \boldsymbol{\beta}^{(0)} \\ \boldsymbol{\beta}^{(1)} \end{pmatrix} = \boldsymbol{A}_{yy}^*\boldsymbol{A}_{xx}\mathbf{F} \tag{2.19}$$

where $\mathbf{U} = [u_1, u_2, \ldots, u_N]^T$ are unknowns to be comptuted, $\boldsymbol{X}^* = diag\{x_1^*, x_2^*, \ldots, x_N^*\}$, $\boldsymbol{Y}^* = diag\{y_1^*, y_2^*, \ldots, y_N^*\}$, $\mathbf{F} = [f(\mathbf{x}_1^*), f(\mathbf{x}_2^*), \ldots, f(\mathbf{x}_M^*)]^T$, the submatrix A which is corresponding to integrations in (2.17) is given as

$$\begin{aligned} \mathrm{A} =& \boldsymbol{A}_{yy}^*(\boldsymbol{A}_1 - 2\boldsymbol{A}_x\boldsymbol{A}_{1,x} + \boldsymbol{A}_{xx}\boldsymbol{A}_{1,xx}) + \boldsymbol{A}_{xx}^*(\boldsymbol{A}_2 - 2\boldsymbol{A}_y\boldsymbol{A}_{2,y} + \boldsymbol{A}_{yy}\boldsymbol{A}_{2,yy}) \\ &+ \boldsymbol{A}_{yy}^*\boldsymbol{A}_x(\mathbf{b}_1 - \boldsymbol{A}_x\mathbf{b}_{1,x}) + \boldsymbol{A}_{xx}^*\boldsymbol{A}_y(\mathbf{b}_2 - \boldsymbol{A}_y\mathbf{b}_{2,x}) + \boldsymbol{A}_{yy}^*\boldsymbol{A}_{xx}^*\mathbf{c}, \end{aligned} \tag{2.20}$$

with diagonal matrices given by known coefficient functions as

$$\begin{aligned} \boldsymbol{A}_1 &= diag\{a_1(x_1, y_1), a_1(x_2, y_2), \ldots, a_1(x_N, y_N)\} \\ \boldsymbol{A}_{1,x} &= diag\{\frac{\partial a_1}{\partial x}(x_1, y_1), \frac{\partial a_1}{\partial x}(x_2, y_2), \ldots, \frac{\partial a_1}{\partial x}(x_N, y_N)\} \\ \boldsymbol{A}_{1,xx} &= diag\{\frac{\partial^2 a_1}{\partial x^2}(x_1, y_1), \frac{\partial^2 a_1}{\partial x^2}(x_2, y_2), \ldots, \frac{\partial^2 a_1}{\partial x^2}(x_N, y_N)\} \\ \boldsymbol{A}_2 &= diag\{a_2(x_1, y_1), a_2(x_2, y_2), \ldots, a_2(x_N, y_N)\} \end{aligned}$$

$$\begin{aligned}
\boldsymbol{A}_{2,x} &= diag\{\frac{\partial a_2}{\partial x}(x_1,y_1), \frac{\partial a_2}{\partial x}(x_2,y_2), \ldots, \frac{\partial a_2}{\partial x}(x_N,y_N)\} \\
\boldsymbol{A}_{2,xx} &= diag\{\frac{\partial^2 a_2}{\partial x^2}(x_1,y_1), \frac{\partial^2 a_2}{\partial x^2}(x_2,y_2), \ldots, \frac{\partial^2 a_2}{\partial x^2}(x_N,y_N)\} \\
\mathbf{b}_1 &= diag\{b_1(x_1,y_1), b_1(x_2,y_2), \ldots, b_1(x_N,y_N)\} \\
\mathbf{b}_{1,x} &= diag\{\frac{\partial b_1}{\partial x}(x_1,y_1), \frac{\partial b_1}{\partial x}(x_2,y_2), \ldots, \frac{\partial b_1}{\partial x}(x_N,y_N)\} \\
\mathbf{b}_2 &= diag\{b_2(x_1,y_1), b_2(x_2,y_2), \ldots, b_2(x_N,y_N)\} \\
\mathbf{b}_{2,x} &= diag\{\frac{\partial b_2}{\partial x}(x_1,y_1), \frac{\partial b_2}{\partial x}(x_2,y_2), \ldots, \frac{\partial b_2}{\partial x}(x_N,y_N)\} \\
\mathbf{c} &= diag\{c(x_1,y_1), c(x_2,y_2), \ldots, c(x_N,y_N)\}
\end{aligned}$$

Regarding the given Dirichlet boundary condition, there exist a matrix $\boldsymbol{B}$ with elements are either 0 or 1 such that $u_i = u(x_i,y_i) = g(x_i,y_i)$ when $(x_i,y_i) \in \partial\Omega$:

$$\boldsymbol{B}\mathbf{U} = \boldsymbol{G}\,, \tag{2.21}$$

where $\mathbf{G} = [g(x_1,y_1), g(x_2,y_2), \ldots, g(x_M,y_M)]^T$.

From (2.19) and (2.21), $\mathbf{U}$ can be obtained by solving a linear system

$$\begin{pmatrix} \mathbf{A} & \boldsymbol{X}^*\boldsymbol{\Phi}_y^{(0)} & \boldsymbol{\Phi}_y^{(1)} & \boldsymbol{Y}^*\boldsymbol{\Phi}_x^{(0)} & \boldsymbol{\Phi}_x^{(1)} \\ \boldsymbol{B} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} \end{pmatrix} \begin{pmatrix} \mathbf{U} \\ \boldsymbol{\alpha}^{(0)} \\ \boldsymbol{\alpha}^{(1)} \\ \boldsymbol{\beta}^{(0)} \\ \boldsymbol{\beta}^{(1)} \end{pmatrix} = \begin{pmatrix} \boldsymbol{A}_{yy}^*\boldsymbol{A}_{xx}\mathbf{F} \\ \mathbf{G} \end{pmatrix}, \tag{2.22}$$

where $\boldsymbol{O}$ stands for matrices with all elements are equal to zero.

2.3. **Least square method.** In (2.22), denote the coefficient matrix as

$$\boldsymbol{H} \triangleq \begin{pmatrix} \boldsymbol{A} & \boldsymbol{X}^*\boldsymbol{\Phi}_y^{(0)} & \boldsymbol{\Phi}_y^{(1)} & \boldsymbol{Y}^*\boldsymbol{\Phi}_x^{(0)} & \boldsymbol{\Phi}_x^{(1)} \\ \boldsymbol{B} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} & \boldsymbol{O} \end{pmatrix}. \tag{2.23}$$

The size of H is $(M + M_B) \times (N + r_0 + r_1 + s_0 + s_1)$, where M, N, M_B, r_0, r_1, s_0, s_1 represent the number of collocation nodes, source nodes, collocation nodes on $\partial\Omega$, interpolation centers of four free terms $g_0(x)$, $g_1(x)$, $f_0(y)$, $f_1(y)$, respectively

For a special case, if Ω is a rectangular with $N_1 \times N_2$ uniform centers, we let collocation nodes are chosen the same as source nodes, i.e., $M = N = N_1 \cdot N_2$, $M_B = 2(N_1 + N_2) - 4$, and let $r_0 = r_1 = N_2 - 1$, $s_0 = s_1 = N_1 - 1$. In this case, $\boldsymbol{H}$ is a square matrix of size $(N_1 \cdot N_2 + 2(N_1 + N_2) - 4) \times (N_1 \cdot N_2 + 2(N_1 + N_2) - 4)$ [24]. However, from several numerical experiments we found H is always warned to be nearly singular and rank deficient when using the backsplash order of Matlab to solve linear system (2.22). So to avoid singularity and rank deficiency, an overdetermined matrix $\boldsymbol{H}$ is preferred. On the other hand, overdetermined linear system is more stable.

There are two ways to achieve this: choose more collocation nodes, i.e. let M large, or use less interpolation centers for $f_0(y)$, $f_1(y)$, $g_0(x)$, $g_1(x)$, i.e. let r_0, r_1, s_0, s_1 small.

To better illustrate our ideas, let's take a rectangular domain as an example. If $N_1 \times N_2$ uniformly nodes are chosen as source nodes, with $N_3 \times N_4$ uniformly nodes are selected as collocation points, assume $r_0 = r_1 = r$ and $s_0 = s_1 = s$. The size of $\boldsymbol{H}$ becomes $(N_3 \cdot N_4 + 2(N_1 + N_2) - 4) \times (N_1 \cdot N_2 + 2(r + s))$. Our first idea using more collocation nodes means let N_3, N_4 large. The other idea is decrease r, s. As RBFs are globally defined, collocation nodes can be chosen arbitrarily to ensure the rank of coefficient matrix is overdetermined and full column ranked, i.e., $rank(\boldsymbol{H}) = N_1 \times N_2 + 2(r + s)$. So the matrix $\boldsymbol{H}^T\boldsymbol{H}$ is a square matrix and not singular. In this case, the mthod of least square is used to solve the linear system (2.22).

3. **Numerical Examples**

Example 1. We firstly consider the following PDE with variable coefficients and homogeneous Dirichlet boundary conditions:

$$x(1-x)\frac{\partial^2 u}{\partial x^2} + y(1-y)\frac{\partial^2 u}{\partial y^2} = -4xy(1-x)(1-y), \quad (x, y) \in (0,1) \times (0,1)\,, \tag{3.1}$$

$$u(0, y) = u(1, y) = u(x, 0) = u(x, 1) = 0\,. \tag{3.2}$$

This 2D BVP admits an analytical solution $u(x, y) = xy(1-x)(1-y)$.

According to the method of integration-based RBF, double-layer integrations with respect x, y variables should be applied on both sides of (3.1) to eliminate differentiations $\frac{\partial^2}{\partial x^2}$ and $\frac{\partial^2}{\partial y^2}$.

$N_1 \times N_2$ uniformly distributed nodes are chosen as source nodes with $(2N_1 - 1) \times (2N_2 - 1)$ uniform nodes are selected as source nodes. To approximated the four free univariate functions $g_0(x), g_1(x)$ and $f_0(y), f_1(y)$, N_1 and N_2 nodes are used as centers, respectively. We choose $N_1 = N_2 = N$ equal to 11, 21 and 31, respectively. Figure 3 shows the distribution of nodes when $N_1 = N_2 = 11$. MQ RBF is used with the shape parameter is chosen equals to $\frac{5}{N-1}$.

To compare with results in [24], the average relative error (RE) is used, which is defined as

$$RE = \frac{1}{M} \sum_{i=1}^{N} \frac{|u_i - u_i^*|}{|u_{\max}|} \tag{3.3}$$

where $N = N_1 \times N_2$ is the total number of source nodes in $[0, 1] \times [0, 1]$, $u_{\max} = u(\frac{1}{2}, \frac{1}{2}) = \frac{1}{16}$.

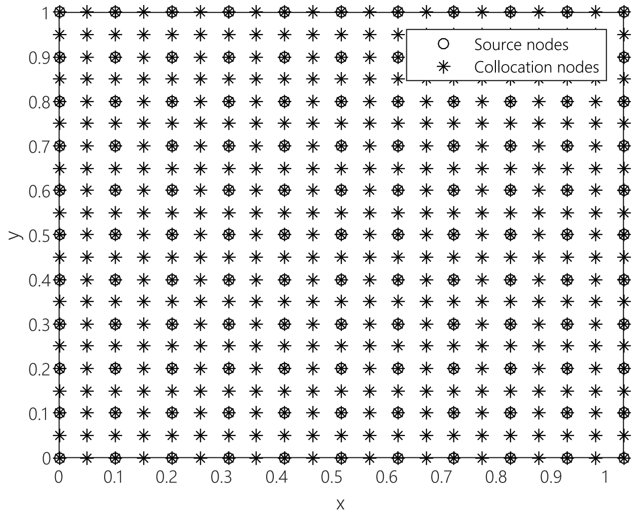


FIGURE 3. Regular nodes distribution.

Table 1 gives us numerical comparisons of relative errors with different number of source nodes, which implies FIM-RBF with least square technique works better than FIM-OLA, MQ and polynomial spline of radial basis function methods.

N	OLA	MQ	PSF	FIM-RBF
11	$1.371E-02$	$1.911E-02$	$1.299E-02$	$2.233E-03$
21	$5.365E-03$	$5.689E-03$	$4.496E-03$	$9.017E-04$
31	$3.083E-03$	$1.382E-02$	$2.373E-03$	$5.354E-04$

TABLE 1. Average errors for different methods and number of points.

Due to the meshless features of RBF, regular nodes distribution is not necessary. Suppose 100 nodes are randomly distributed in $(0,1)\times(0,1)$ with 36 nodes are uniformly located on the boundary. The integration nodes are chosen to be 200 nodes lie in $(0,1)\times(0,1)$ with 72 nodes uniformly on the boundary. Still use 10 nodes in x and y direction as integration centers for RBF approximation of $g_0(x), g_1(x)$ and $f_0(y), f_1(y)$. Figure 4 gives us nodes distribution in $[0,1]\times[0,1]$.

In this case, the shape parameter of MQ is chosen as $c = 0.4$. The size of the resultant matrix is 308×176 and is full column ranked. By the integration-based RBF method with least square method, the maximum pointwise error is $5.96E-04$ and the average relative error is $1.54E-03$.

Example 2. To further demonstrate the smoothness effect of the proposed integration-based RBF method, we consider the following two-dimensional Burgers' equation,

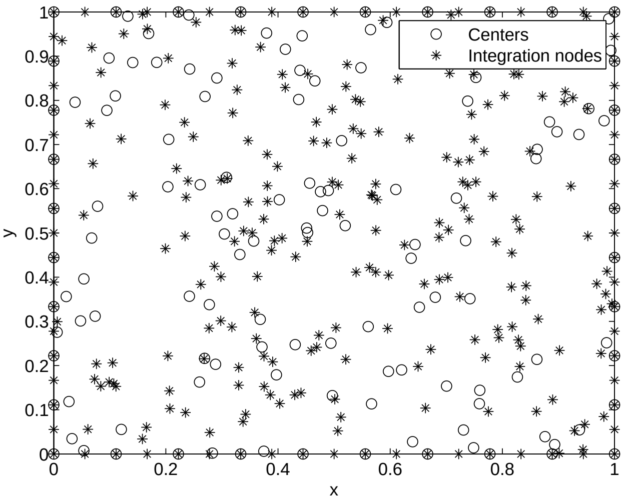


Figure 4. Randomly nodes distribution.

which contains a moving shock wave front

$$\frac{\partial u}{\partial t} = \frac{1}{R}\Delta u - (uu_x + uu_y), \quad (x, y) \in (0,1) \times (0,1)\,, \tag{3.4}$$

with boundary and initial conditions are set accordingly such that the exact solution is

$$u(x,y,t) = \frac{1}{1+\exp\left(\frac{x+y-t}{2}R\right)}\,. \tag{3.5}$$

The coefficient R is a physical parameter called Reynolds number. The solution contains a shock wave front along the line $x + y - t$, which is moving with time..

The Burgers' equation is time dependent and nonlinear, which is different from previous examples. The backward Euler scheme is adopted to discretize the temporal derivative and the method of linearization is used to deal with the nonlinear term:

$$\frac{u^m - u^{m-1}}{\Delta t} - \frac{1}{R}\left(\frac{\partial^2 u^m}{\partial x^2} + \frac{\partial^2 u^m}{\partial y^2}\right) = -\left(u^{m-1}u_x^{m-1} + u^{m-1}u_y^{m-1}\right)\,, \tag{3.6}$$

where Δt is denoted as the time step, u^m is the approximation of solution at $m\Delta t$. When $m = 0$, $u^0(x,y) = u(x,y,0) = \frac{1}{1+\exp\left(\frac{x+y}{2}R\right)}$.

Different from Example 1, we use $N_1 \times N_2$ uniform nodes in $[0,1] \times [0,1]$ as source nodes as well as collocation nodes. To ensure the coefficient matrix is overdetermined and full column ranked, we use $N_1 - 2, N_2 - 2$ nodes as centers for RBF approximation of $g_0(x), g_1(x)$ and $f_0(y), f_1(y)$.

To compare with results given by RBFCM in [25], we let $\frac{1}{R} = 0.05$, $\Delta t = 0.001$, and $N_1 = N_2 = N_3 = N_4 = N = 19$, $r = s = N - 2 = 17$. MQ is used with shape parameter $c = \frac{4}{N-1}$. At this situation, the size of the resultant matrix $\boldsymbol{H}$ is 433×429, which is obviously overdetermined and it's verified to be column full ranked, i.e. $rank(\boldsymbol{H}) = 429$. By algebraic analysis, $\boldsymbol{H}^T\boldsymbol{H}$ is not singular. Therefore the least squares method does work here.

Profiles of approximated solutions and pointwise errors at different time are shown in Figure 5, from which we can observe that our method can simulate the solution of the 2D Burgers' equation very well. At $t = 1.25$, the maximum pointwise error is 2.51×10^{-3}, compared with $7.03E - 02$ by MFS-DRM and $7.19E - 02$ by Kansa method [25].

As R becomes larger, the wave front becomes steeper, which gives more difficulty when compute numerical results. Refer to Figure 6 (a) to see profiles of $u(x, y)$ at the line $x = y$. It's obvious that an interior layer arised as R becomes larger.

So it's naturally to require more points locate near the wave front to get better approximation as R become larger. Sulman et al. [26] studied this problem when $R = 200$. We let $N_1 = N_2 = N_3 = N_4 = N = 41$, $r = s = N - 2 = 39$. At the time $t = 1.25$, the maximum pointwise error is $2.23E - 02$ and the discretized L^2 norm error is $4.66E - 03$, where the discretized version of L^2 norm error is defined as

$$\sqrt{\frac{\sum_{i=1}^{\overline{N}} |u_i - u_i^*|^2}{\overline{N}}},$$

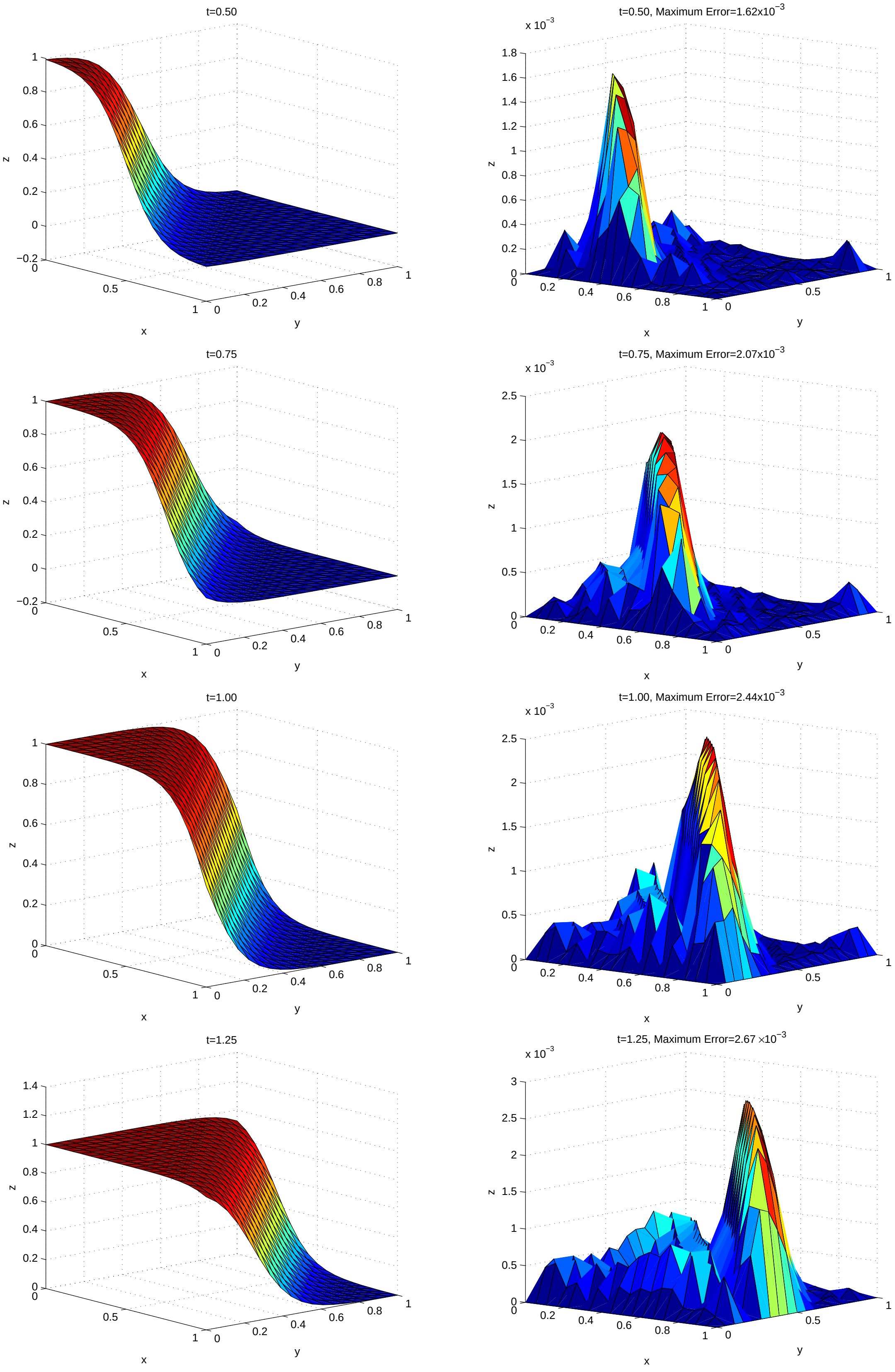
$\overline{N} \triangleq N_1 \cdot N_2$, u_i and u_i^* represent exact and numerical function value, respectively. Numerical solution and pointwise error are displayed in Figure 7, which is better than results in [26] by FDM with complicated adaptive grid generate technique.

4. Conclusion

In this paper, the integration-based RBF method is introduced. Note that RBFs are global defined, collocation nodes can be chosen different from source nodes. The method of least squares is adopted to eliminated possible singularities.

To verify this, two typical examples are given and discussed. The effectiveness of our proposed method is verified. Compared with differentiation, integration is more stable and accurate through computation by computer. For problems with shock waves and boundary (interior) layers, the solution's derivatives change largely in a small interval while its integral is smooth. Therefore, instead of solving PDE directly, solving its equivalent integral equation gives more accurate solution.

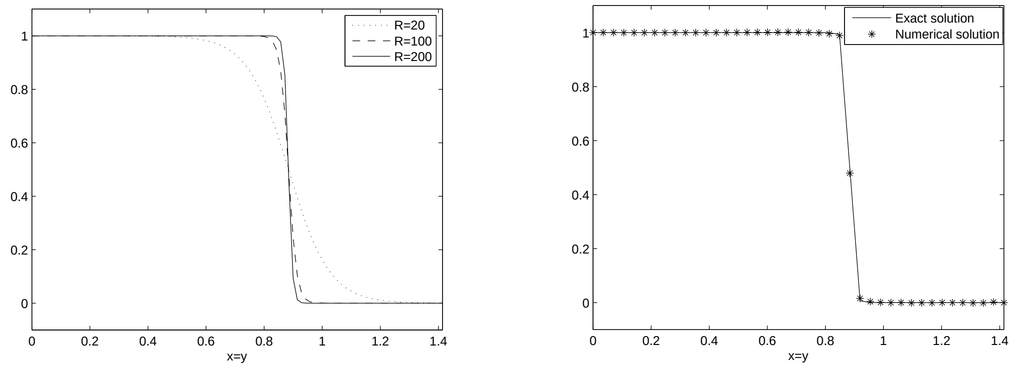
Another potential application is solving PDEs with singular terms. Singularity gives more difficulty theoretically and numerically. To avoid singularity, several common techniques have been applied. Firstly add a small perturbation value to the eliminate singularity. The solution of the singular PDE can be obtained as this perturbation goes to zero. Instead of this time consuming process, integration can eliminate singularity directly. We will continue our research on this problem in the future.



(a) Profiles of approximation.

(b) Profiles of maximum error.

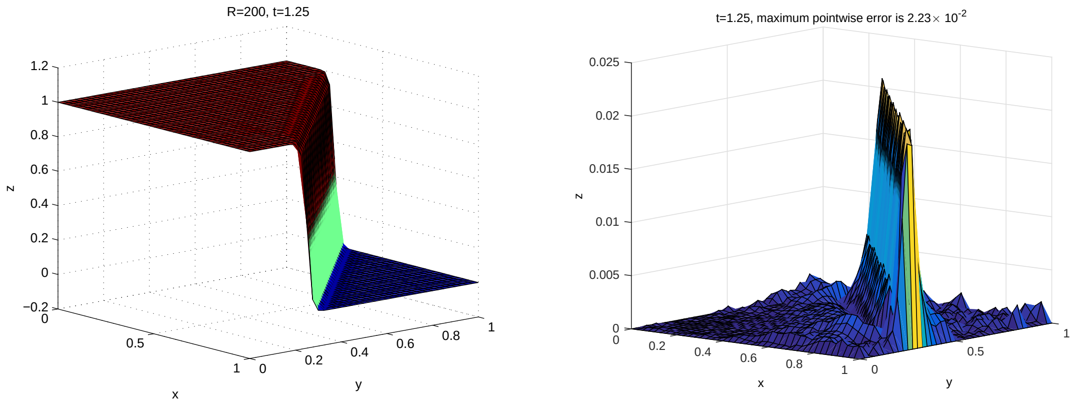
FIGURE 5. Profiles at $t = 0.50$, $t = 0.75$, $t = 0.10$ and $t = 1.25$ when $\frac{1}{R} = 0.05$, i.e. $R = 20$.



(a) Profiles of u(x,y) for different R.

(b) Profiles of numerical solution.

FIGURE 6. Profiles at the line $x = y$.



(a)

(b)

FIGURE 7. Profiles of approximation by FIM-RBF, $R = 200$, $t = 1.25$.

ACKNOWLEDGMENTS

The author thanks Prof. Benny Hon (City University of Hong Kong) and Prof. Bartosz Protas (McMaster University) for useful discussions and comments. This work was partially supported by the Fields-Ontario Postdoctoral Fellowships from Fields Institute, Toronto, Canada.

REFERENCES

[1] S. N. Atluri, S. Shen, The Meshless Local Petrov-Galerkin (MLPG) Method. Tech Science Press, 2002.

[2] A. Ferreira, E. J. Kansa, G. E. Fasshauer, V. Leito, Progress on Meshless Methods, Springer, 2008.

[3] V. Leito, C. Durate, Advances in meshfree techniques, Springer, 2007.

[4] G. R. Liu, Mesh Free Methods: Moving Beyond the Finite Element Method, CRC Press, 2002.

[5] J. Sladek, V. Sladek, Advances in Meshless Methods, Tech Science Press, 2006.

[6] R. L. Hardy, Multiquadric equations of topography and other irregular surfaces, J. Geophysics Res., 176: 1905–1915, 1971.

[7] M. D. Buhmann, Radial Basis Functions: Theory and Implementations, Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, 2003.

[8] H. Wendland, Scattered data approximation. Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, 2005.

[9] R. Schaback, Error estimates and condition numbers for radial basis function interpolation, Advances in Computational Mathematics, 3: 251–264, 1995.

[10] Z. M. Wu, Schaback R. Local error estimates for radial basis function interpolation of scattered data, IMA J. of Numerical Analysis, 13: 13–27, 1993.

[11] M. J. Johnson, An error analysis for radial basis function interpolation, Numer. Math. 98: 675–694, 2004.

[12] R. Schaback, Approximation by radial basis functions with finitely many centers, Constructive Approximation 12: 331–340, 1996.

[13] W. R. Madych, Miscellaneous error bounds for multiquadric and related interpolators, Computers Math. Applic. 24: 121–138, 1992.

[14] E. J. Kansa, Multiquadrics - a scattered data approximation scheme with application to computational fluid dynamics, part I, Computers and Mathematics with Applications, 19: 127–145, 1990.

[15] E. J. Kansa, Multiquadrics - a scattered data approximation scheme with application to computational fluid dynamics, part II, Computers and Mathematics with Applications, 19: 147–161, 1990.

[16] R. Franke, R. Schaback, Solving partial differential equations by collocation using radial functions, Applied Mathematics and Computation 93: 73-82, 1998.

[17] H. Wendland, Meshless Galerkin methods using radial basis functions, Math. Comp., 68: 1521–1531, 1999.

[18] L. Ling, R. Schaback, Stable and Convergent Unsymmetric Meshless Collocation Methods, SIAM J. Num. Anal., 46: 1097–1115, 2008.

[19] L. M. Ma, Z. M. Wu, Approximation to the k-th derivatives by multiquadric quasiinterpolation method, Journal of Computational and Applied Mathematics, 231: 925–932, 2009.

[20] H. Wendland. Error estimates for interpolation by compactly supported radial basis functions of minimal degree, J. Approx. Theory, 93: 258–272, 1998.

[21] B. Šarler, From global to local radial basis function collocation method for transport phenomena, Computational Methods in Applied Sciences, 257–282, 2007.

[22] M. Li, Y. C. Hon, T. Korakianitis, P. H. Wen, Finite integration method for nonlocal elastic bar under static and dynamic loads, Engineering Analysis with Boundary Elements, 37: 842–849, 2013.

[23] P. H. Wen, Y. C. Hon, M. Li, T. Korakianitis, Finite integration method for partial differential equations, Applied Mathematical Modelling, 37: 10092–10106, 2013.

[24] P. H. Wen, C. S. Chen, Y. C. Hon, M. Li, Finite integration method for multi-dimensional partial differential equations, Applied Mathematical Modelling, 39: 4979–4994, 2015.

[25] J. C. Li, Y. C. Hon, C. S. Chen, Numerical comparisons of two meshless methods using radial basis functions, Engineering Analysis with Boundary Elements, 26: 205–225, 2002.

[26] M. Sulman, J. F. Williams, R. D. Russell. Optimal mass transport for higher dimensional adaptive grid generation, Journal of Computational Physics, 230: 3302–3330, 2011.