

## Local radial basis function collocation method for Stokes equations with interface conditions

Masood Ahmad, Siraj-ul-Islam\*, Baseer Ullah

*Department of Basic Sciences, Faculty of Architecture, Allied Sciences & Humanities, University of Engineering and Technology, Peshawar, Pakistan*



### ARTICLE INFO

**Keywords:**

Local meshless method  
Radial basis functions  
Stokes equations  
Sparse matrix  
Collocation  
Interface PDEs

### ABSTRACT

In the present work, local radial basis function (RBF) collocation method is used to solve the Stokes problem numerically. Benefits of the local meshless method is its flexibility of handling complex computational domain and interface. Local meshless method produces sparse coefficient matrix, though its structure is depending on the placement of nodes, and type of boundary conditions. The Stokes equations are collocated in a rectangular region with different interface conditions and direct solvers are used instead of iterative algorithm. Due to the unknown boundary conditions for the pressure term in the Stokes equations, value of the pressure is obtained by integrating the Stokes equations and eliminating the constant of integration. In the numerical implementation, circular, elliptic and Deltoid like interfaces are taken into consideration to test performance of the proposed approach for complex problems. Meshless solution of the Stokes equations for the first two problems are compared with the published results in the literature. The last problem is different form the first two problems in the sense that the continuity equation is also discontinuous and the velocity has nonzero value in both the inner and outer sub-domains.

### 1. Introduction

There are several computational methods available in the literature for the numerical solution of Stokes problem. The most prominent amongst these are the finite difference, finite element and finite volume methods. Many engineering problems have been successfully solved using these methods. These algorithms have contributed a lot to state of the art modelling and simulation processes. These methods have been gradually upgraded computationally and theoretically to cope up the existing challenges up to some extent. There are still some grey areas and unanswered question for the existing methods which need further research investigations. Specifically, one of the major difficulty of these methods is the selection of an efficient mesh generation technique, when applied to computational fluid mechanics problems. At times, accuracy of these methods is strictly dependent on the mesh quality while dealing with complex interface modelling. Since mesh moulding as per geometric conformity is a difficult and time-consuming task, thus, speed of the numerical simulations using these methods is often hampered by slow mesh generation processes.

To overcome or reduce such difficulties, many prominent researchers have contributed to alternate numerical methods including meshless methods. Towards discretization of the domain, meshless procedure replaces mesh generation by nodes generation in a scattered or uniformed manner, without prior connectivity requirement. Various forms of mesh-

less methods have been reported in the literature in general applications as well as specific applications related to interface phenomena [1–10]. The RBF collocation meshless methods with strong formulation became popular due to their ease of implementation and truly meshless character. In this truly meshless formulation, each dependent variable is collocated as a linear combination of RBFs at specific locations, called centres.

Since RBFs are composite real functions of radial distances and sometimes shape parameter, meshless schemes based on the RBFs can be easily extended to higher dimensions. Due to the adoption of collocation procedure in strong formulation, no background grid is needed to perform any integrations. A variety of partial differential equations, such as initial value problems [11], integral equations [12–15], the convection diffusion equation [16,17], Burger's equation [18,19], financial models [20–22], have been solved using meshless approaches.

In recent years, a local RBF collocation method is an active research area for the numerical solution of partial differential equations. A variety of partial differential equations for complex-shaped domain have been successfully solved using the local RBF collocation method, e.g., a two-dimensional fractional evolution equation for the arbitrary fractional order in complex-shaped domains [23], a time fractional diffusion-wave equation [24], a semi-Lagrangian meshless framework [25], a second-order explicit Runge-Kutta method for two-dimensional sloshing phenomenon, a three dimensional multi-physics problem [26],

\* Corresponding author.

E-mail addresses: [masood.suf@uetpeshawar.edu.pk](mailto:masood.suf@uetpeshawar.edu.pk) (M. Ahmad), [siraj-ul-Islam@uetpeshawar.edu.pk](mailto:siraj-ul-Islam@uetpeshawar.edu.pk) (Siraj-ul-Islam).

a multi-term variable-order time fractional partial differential equations [27] using kernel-based collocation method, etc. In [28], a new adaptive two-stage algorithm is developed for solving elliptic partial differential equations via a radial basis function collocation method, which is based on the use of a leave-one-out cross validation technique and residual sub-sampling method allowing good detection of the areas that need to be refined with added flexibility to add or remove any point adaptively. The combination of the upwind local radial basis function-differential quadrature method with the space-splitting idea for solving some conservation laws equations is discussed in [29]. In [30], a simple explicit radial basis function collocation method is used to solve the shallow water equations for flows over irregular frictional topography involving wetting and drying.

The well known incompressible Navier-Stokes equations have been solved numerically in the literature using various methods. For instance, local Petrov-Galerkin method with the Rankin source solution has been applied for the numerical solution of two dimensional incompressible Navier-Stokes equation [31]. A new interface treatment technique is proposed in the form of multi-phase meshless methods, enabling the interface conditions to be applied effectively at the interface points involving interface surface tension and high viscosity. In [32], a two-dimensional stationary Stokes problems was solved numerically by Galerkin boundary node method for circular cavity, flow around a rotating cylinder, flow between two rotating cylinders and flow around a cylinder between two parallel plates. Incompressible two- and three-dimensional Stokes flows were solved using meshless hybrid boundary node method [33] for single phase problems like a square cavity, circular cavity, flow between two rotating cylinders and rectangular cavity with wave-shaped bottom and a lid-driven cubic cavity flow. In [34], a comparison of the global and the localized RBF meshless methods was discussed for incompressible fluid flow with heat transfer. Meshless local Petrov-Galerkin method was used for the numerical solution of incompressible Navier-Stokes equation in [35]. The simulation of the time dependent incompressible Navier-Stokes equation with variable density is discussed via local radial basis functions-finite difference technique in [36].

In this paper, we present a meshless method based on RBF collocation approach for the solution of Stokes equations using uniform as well as scattered nodes, generated through a node placing algorithm described in [1,37]. The primary novelty of the current work is that both the pressure and the velocities are approximated directly by localized RBFs with interface conditions. To the best of our knowledge no such work has been reported earlier in the meshless framework (both local and global). The current work is the extension of our recently published work [1], where elliptic PDEs with interface conditions, having weak singularity and sharp corners in the domain are accurately solved. In the previous work [38], an interface-fitting technique was combined with an adaptive mesh approach in the finite element framework to solve the Stokes flow. However, the current implementation provides high accuracy results without using any adaptive mesh strategy, thus considerably reducing the computational efforts due to non-adaptivity. This implementation also provides a viable alternative to a comparatively complex technique, i.e., the Immersed interface method [39] for the solution of Stokes equations using both uniform and scattered nodes. A unique methodology is adopted in this study for nodes generations that avoids clustering at corners, thus, minimizing the effects of ill-conditioning for the coefficient matrix. Accuracy of the meshless method and the issue of the condition number versus different types of boundary conditions (for pressure) are discussed in details in the context of accuracy of the computed solution. Strength of the meshless method with reference to selecting uniform or scattered nodes is fully utilized. The missing boundary condition for the pressure term is derived by the integration of momentum equation with pressure gradient and utilizing the integrals of RBF function. Inside the domain, the field variables are separately approximated by the RBF. Numerical results are presented for Stokes problem with different interfaces and different values of velocity and pressure.

## 2. Stokes equations

In the present work, we consider meshless solution of the two dimensional steady state Stokes PDEs by localized meshless method in a strong form. The two dimensional Stokes equations along with boundary and interface conditions can be written as. The momentum equation is

$$\mu \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) - \frac{\partial p}{\partial x} = \begin{cases} f_u^+ & \text{if } (x, y) \in \Omega^+, \\ f_u^- & \text{if } (x, y) \in \Omega^-, \end{cases} \quad (1a)$$

$$\mu \left( \frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) - \frac{\partial p}{\partial y} = \begin{cases} f_v^+ & \text{if } (x, y) \in \Omega^+, \\ f_v^- & \text{if } (x, y) \in \Omega^-, \end{cases} \quad (1b)$$

whereas the continuity equation is

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = \begin{cases} f_c^+ & \text{if } (x, y) \in \Omega^+, \\ f_c^- & \text{if } (x, y) \in \Omega^-, \end{cases} \quad (1c)$$

$$u|_{\partial\Omega} = u_B, \quad (1d)$$

$$v|_{\partial\Omega} = v_B. \quad (1e)$$

The symbols  $u$  and  $v$  are the velocity components,  $p$  is the pressure,  $\mu$  is the viscosity and  $\vec{f} = (f_u, f_v)$  is the external force. In the present work, we assume  $\mu$  is constant in all the cases except one test problem and  $\vec{u}_B$  is the prescribed value of the velocity at the boundary of the computational domain.

In the case of meshless method one can take irregular domain  $\Omega$ , but in the current work, we assume  $\Omega$  to be a regular domain, such as a rectangle, in order to compare numerical results of the proposed method with the methods reported in the literature. The interface  $\Gamma$  divides the domain into two portions  $\Omega^-$  and  $\Omega^+$ ,  $\Omega^-$  represents the inside domain and  $\Omega^+$  represents the outside domain. In order to solve (1), the following jump conditions are imposed

$$[u]_\Gamma = g_u(x, y), \quad (2a)$$

$$\left[ \frac{\partial u}{\partial \vec{n}} \right]_\Gamma = h_u(x, y), \quad (2b)$$

$$[v]_\Gamma = g_v(x, y), \quad (2c)$$

$$\left[ \frac{\partial v}{\partial \vec{n}} \right]_\Gamma = h_v(x, y), \quad (2d)$$

$$[p]_\Gamma = g_p(x, y), \quad (2e)$$

$$\left[ \frac{\partial p}{\partial \vec{n}} \right]_\Gamma = h_p(x, y), \quad (2f)$$

where  $g_u$ ,  $g_v$ ,  $g_p$ ,  $h_u$ ,  $h_v$  and  $h_p$  are the known functions and  $\vec{n}$  is the unit normal vector. In the next section, the meshless numerical procedure will be explained in detail.

## 3. Numerical procedure

In this section we describe the node generation process, meshless interpolation, derivatives approximation through RBFs and RBF collocation used for approximation of (1). We also discuss the stencil design at the interface and the overall construction of the meshless method on the overlapping small stencils along with different types of interfaces.

### 3.1. Generation and identification of nodal points

The local meshless differential quadrature procedure has many benefits over the mesh based methods. One of these is the arrangement of the nodal points according to the dictates of the given geometry, which makes meshless methods simple and flexible as compared to other mesh based methods. In the proposed local meshless method, we consider a

set of nodes in the computational domain  $\Omega = [-1, 1] \times [-1, 1]$ . Two types of arrangements in the form of uniform nodes and scattered nodes are taken into consideration. Uniform nodes are generated in the computational domain by taking equal mesh size both in  $x$ - and  $y$ -directions i.e.,  $N = N_x = N_y$  with  $h = \Delta x = \Delta y$ .

On the other hand, the scattered nodes are generated by selecting a global approximate nodal distance  $h$ . The sub-domains  $\Omega_1$  and  $\Omega_2$  are initially filled with nodes using the node placing algorithm described in [37]. The nodes that are closer than the distance  $h/4$  to the boundary or interface are pruned, and then finally a few iterations of an electrostatic node repulsion algorithm is applied to the interior nodes in each sub-domain while the boundary and interface nodes stay fixed(see Fig. 3).

To solve the two-dimensional Stokes equations on a set of nodes, we need to specify different locations of the nodes i.e., we represent the boundary set of nodes by  $\xi_{\partial\Omega}$  having  $N_b$  nodes on the boundary,  $\xi_\Gamma$  be the set of nodes belonging to interface having  $N_\Gamma$  nodes and  $\xi_{\Omega^+}$  and  $\xi_{\Omega^-}$  be the set of nodes belonging to  $\Omega^+$  and  $\Omega^-$  having  $N_2$  and  $N_1$  points respectively. We further assume that all these subsets are non-empty. In this case, we place nodes uniformly on each segment of the outer boundary and the interface to make sure that nodes are placed at all corner points. Stencil design at the boundary, in the interior and at the interface are shown in Fig. 2. In the current work we select the stencil size equals to 30.

### 3.2. Meshless interpolation

For the meshless approximation of the Stokes equations, the nodes are arranged in ascending order with respect to their distances from the centers. We select a set of nearest points (called local support domain) for each given center  $\mathbf{x}_j = (x_j, y_j)$ ,  $j = 1, \dots, N^2$  with respect to the Euclidean norm. We represent the total number of such nearest points by  $ns$ . Let  $u^+, v^+, p^+, u^-, v^-, p^-$  be the required approximation solutions in the respective sub-domains, then

$$\begin{aligned} u^+ &= \sum_{i=1}^{ns} \lambda_i^+ \phi(\|\mathbf{x} - \mathbf{x}_i\|_2) + a_1^+ x + a_2^+ y + a_3^+, & \mathbf{x} \in \Omega^+ \\ u^- &= \sum_{i=1}^{ns} \lambda_i^- \phi(\|\mathbf{x} - \mathbf{x}_i\|_2) + a_1^- x + a_2^- y + a_3^-, & \mathbf{x} \in \Omega^- \\ v^+ &= \sum_{i=1}^{ns} \zeta_i^+ \phi(\|\mathbf{x} - \mathbf{x}_i\|_2) + b_1^+ x + b_2^+ y + b_3^+, & \mathbf{x} \in \Omega^+ \\ v^- &= \sum_{i=1}^{ns} \zeta_i^- \phi(\|\mathbf{x} - \mathbf{x}_i\|_2) + b_1^- x + b_2^- y + b_3^-, & \mathbf{x} \in \Omega^- \\ p^+ &= \sum_{i=1}^{ns} \eta_i^+ \phi(\|\mathbf{x} - \mathbf{x}_i\|_2) + c_1^+ x + c_2^+ y + c_3^+, & \mathbf{x} \in \Omega^+ \\ p^- &= \sum_{i=1}^{ns} \eta_i^- \phi(\|\mathbf{x} - \mathbf{x}_i\|_2) + c_1^- x + c_2^- y + c_3^-, & \mathbf{x} \in \Omega^- \end{aligned} \quad (3)$$

with the following constraints

$$\begin{aligned} \sum_{i=1}^{ns} \lambda_i^+ &= \sum_{i=1}^{ns} \lambda_i^+ x_i = \sum_{i=1}^{ns} \lambda_i^+ y_i = 0, & x_i, y_i \in \Omega^+, \\ \sum_{i=1}^{ns} \lambda_i^- &= \sum_{i=1}^{ns} \lambda_i^- x_i = \sum_{i=1}^{ns} \lambda_i^- y_i = 0, & x_i, y_i \in \Omega^-, \\ \sum_{i=1}^{ns} \zeta_i^+ &= \sum_{i=1}^{ns} \zeta_i^+ x_i = \sum_{i=1}^{ns} \zeta_i^+ y_i = 0, & x_i, y_i \in \Omega^+, \\ \sum_{i=1}^{ns} \zeta_i^- &= \sum_{i=1}^{ns} \zeta_i^- x_i = \sum_{i=1}^{ns} \zeta_i^- y_i = 0, & x_i, y_i \in \Omega^-, \\ \sum_{i=1}^{ns} \eta_i^+ &= \sum_{i=1}^{ns} \eta_i^+ x_i = \sum_{i=1}^{ns} \eta_i^+ y_i = 0, & x_i, y_i \in \Omega^+, \end{aligned}$$

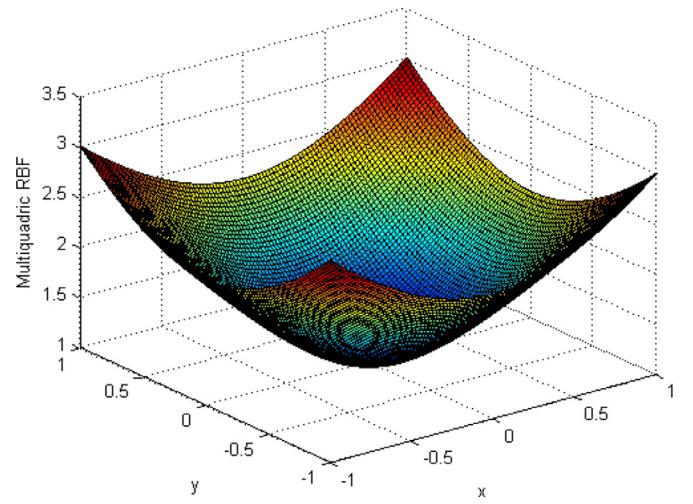


Fig. 1. Surface plot of the Multi-quadric (4) RBF.

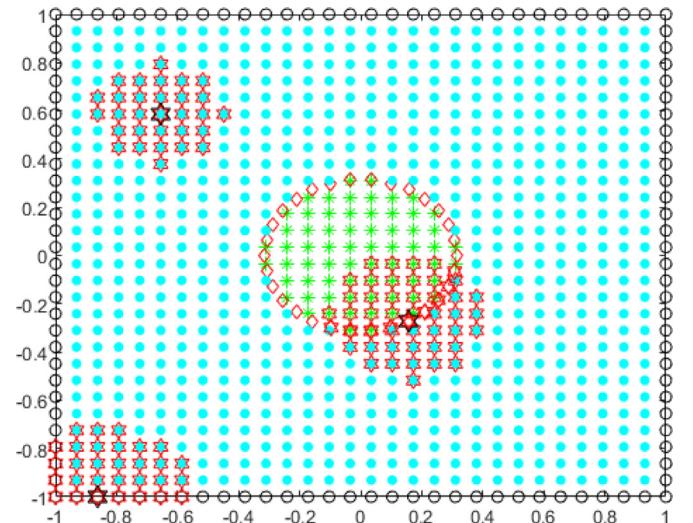


Fig. 2. Selection of stencils in the subregions  $\Omega^+$ ,  $\partial\Omega$  and the interface  $\Gamma$ .

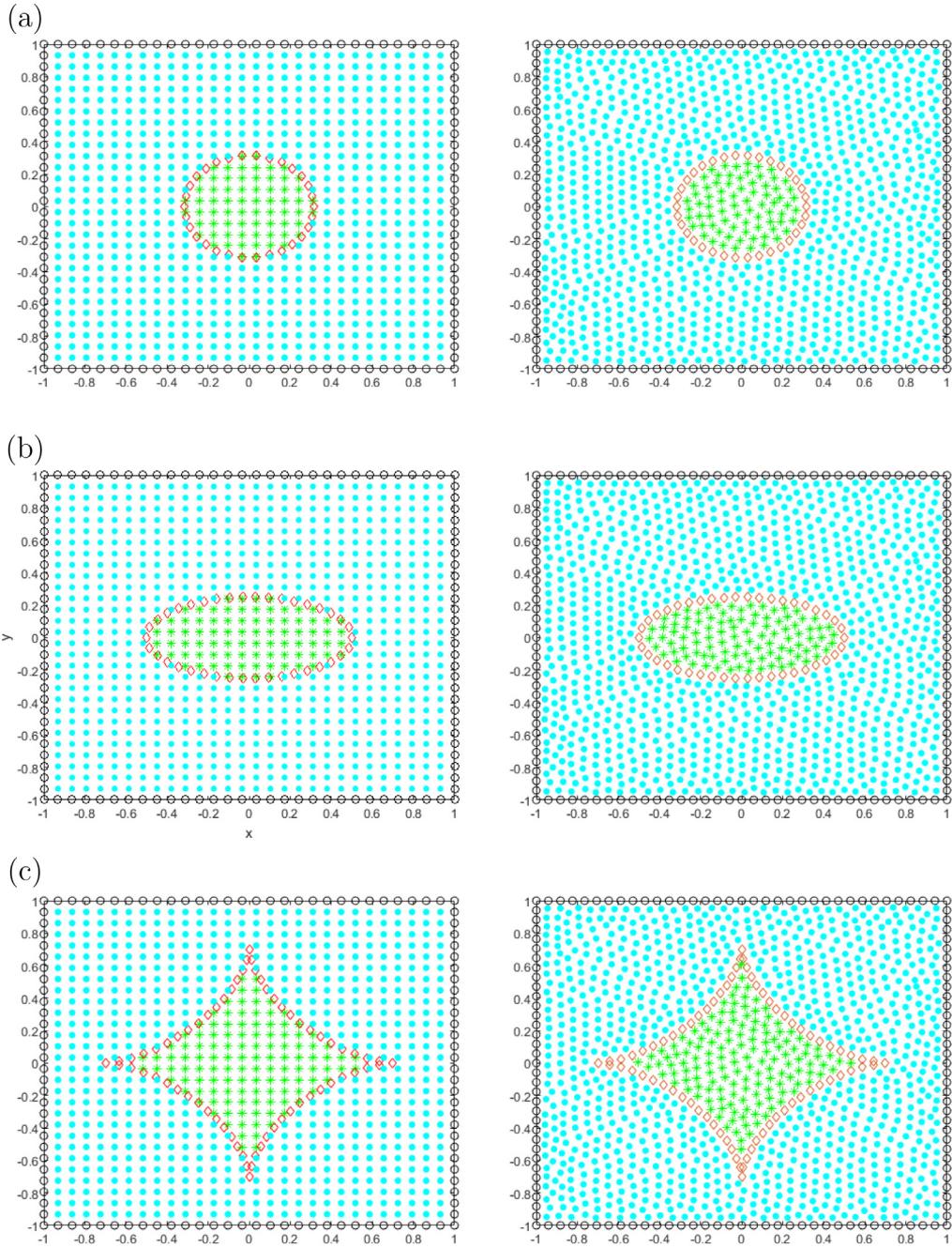
$$\sum_{i=1}^{ns} \eta_i^- = \sum_{i=1}^{ns} \eta_i^- x_i = \sum_{i=1}^{ns} \eta_i^- y_i = 0, \quad x_i, y_i \in \Omega^-.$$

The norm  $\|\mathbf{x} - \mathbf{x}_i\|$  is a Euclidean norm between the point  $\mathbf{x}$  to the point  $\mathbf{x}_i$ , the point  $\mathbf{x}_i$  belong to the local support domain  $\Omega_j$  associated with center  $\mathbf{x}_j$  that has nearest neighboring nodes  $\mathbf{x}_i$ ,  $i = 1, 2, 3, \dots, ns$ . The coefficients  $\lambda_i^+$ ,  $\zeta_i^+$ ,  $\eta_i^+$ ,  $\lambda_i^-$ ,  $\zeta_i^-$ , and  $\eta_i^-$  are the expansion coefficients, and  $\phi$  is some given RBF. In current work we use the Multi-quadric RBF [40], whose surface plot is shown in Fig. 1, having mathematical representation as

$$\phi(\|\mathbf{x} - \mathbf{x}_i\|_2) = \sqrt{1 + (s_p r)^2}, \quad r^2 = (x - x_i)^2 + (y - y_i)^2. \quad (4)$$

The parameter  $s_p$  in the RBF is a shape parameter controlling the shape of the Multi-quadric RBF. The expansion coefficients  $\lambda_i^+$ ,  $\zeta_i^+$ ,  $\eta_i^+$ ,  $\lambda_i^-$ ,  $\zeta_i^-$ , and  $\eta_i^-$  can be found by using the interpolation condition, which gives  $3N^2$  linear systems of equations, each having dimension  $(ns + 3) \times (ns + 3)$  of the form

$$\begin{aligned} \mathbf{u}^+ &= B_1 \lambda^+, & \mathbf{u}^- &= B_2 \lambda^-, \\ \mathbf{v}^+ &= B_1 \zeta^+, & \mathbf{v}^- &= B_2 \zeta^-, \\ \mathbf{p}^+ &= B_1 \eta^+, & \mathbf{p}^- &= B_2 \eta^-, \end{aligned} \quad (5)$$



**Fig. 3.** Computational domains used in the analysis with different interfaces. (a): Circular interface. (b): Elliptic interface. (c): Deltoid interface.

where,

$$\begin{aligned} \lambda^+ &= [\lambda_1^+, \lambda_2^+, \lambda_3^+, \dots, \lambda_{ns}^+, a_1^+, a_2^+, a_3^+]^T, & \lambda^- &= [\lambda_1^-, \lambda_2^-, \lambda_3^-, \dots, \lambda_{ns}^-, a_1^-, a_2^-, a_3^-]^T, \\ \zeta^+ &= [\zeta_1^+, \zeta_2^+, \zeta_3^+, \dots, \zeta_{ns}^+, b_1^+, b_2^+, b_3^+]^T, & \zeta^- &= [\zeta_1^-, \zeta_2^-, \zeta_3^-, \dots, \zeta_{ns}^-, b_1^-, b_2^-, b_3^-]^T, \\ \eta^+ &= [\eta_1^+, \eta_2^+, \eta_3^+, \dots, \eta_{ns}^+, c_1^+, c_2^+, c_3^+]^T, & \eta^- &= [\eta_1^-, \eta_2^-, \eta_3^-, \dots, \eta_{ns}^-, c_1^-, c_2^-, c_3^-]^T. \end{aligned}$$

In the above expressions  $T$  represents a transpose. Equations which are mentioned in (5) can be solved for the expansion coefficients as the interpolation matrices

$$B_1 = \begin{bmatrix} B_1^+ & P_1 \\ P_1^T & \mathbf{0} \end{bmatrix}, B_2 = \begin{bmatrix} B_2^- & P_1 \\ P_1^T & \mathbf{0} \end{bmatrix} \quad (6)$$

are invertible and the expansion coefficients are uniquely defined on each stencil. The entries of the matrices  $B_1^+$  and  $B_2^-$  are

$$b_{lk} = \begin{cases} \phi(\|\mathbf{x}_l - \mathbf{x}_k\|), & \mathbf{x}_l, \mathbf{x}_k \in \Omega_j \subset \Omega^+, \\ \phi(\|\mathbf{x}_l - \mathbf{x}_k\|), & \mathbf{x}_l, \mathbf{x}_k \in \Omega_j \subset \Omega^-, \end{cases}$$

respectively, and the matrix  $P_1^T$  is

$$P_1^T = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ x_1 & x_2 & x_3 & \dots & x_{ns} \\ y_1 & y_2 & y_3 & \dots & y_{ns} \end{bmatrix},$$

where  $\mathbf{0}$  is a zero matrix of size 3. Since, the matrices  $B_1$  and  $B_2$  are same in all equations mentioned in (5), we need to find inverse of only  $N^2$  small matrices instead of  $3N^2$ .

### 3.3. Differential quadrature

The derivative of a function can be approximated at the center locations by applying a linear differential operator  $D^{(m)}$  to the local interpolant in the form of radial basis function. Moreover,  $D^{(m)}$  can be a linear combination of derivative operator. However, in the current work,

we introduce the following notations for  $m$ th-order partial derivatives:  $D_x^{(m)} = \frac{\partial^m}{\partial x^m}$  and  $D_y^{(m)} = \frac{\partial^m}{\partial y^m}$ . The following equations

$$\begin{aligned} D_x^{(m)} u^+ &= \sum_{i=1}^{ns} \lambda_i^+ D_x^{(m)} \phi(\|\mathbf{x} - \mathbf{x}_i\|_2) + D_x^{(m)}(a_1^+ x + a_2^+ y + a_3^+), & \mathbf{x} \in \Omega^+ \\ D_x^{(m)} u^- &= \sum_{i=1}^{ns} \lambda_i^- D_x^{(m)} \phi(\|\mathbf{x} - \mathbf{x}_i\|_2) + D_x^{(m)}(a_1^- x + a_2^- y + a_3^-), & \mathbf{x} \in \Omega^- \\ D_x^{(m)} v^+ &= \sum_{i=1}^{ns} \zeta_i^+ D_x^{(m)} \phi(\|\mathbf{x} - \mathbf{x}_i\|_2) + D_x^{(m)}(b_1^+ x + b_2^+ y + b_3^+), & \mathbf{x} \in \Omega^+ \\ D_x^{(m)} v^- &= \sum_{i=1}^{ns} \zeta_i^- D_x^{(m)} \phi(\|\mathbf{x} - \mathbf{x}_i\|_2) + D_x^{(m)}(b_1^- x + b_2^- y + b_3^-), & \mathbf{x} \in \Omega^- \\ D_x^{(m)} p^+ &= \sum_{i=1}^{ns} \eta_i^+ D_x^{(m)} \phi(\|\mathbf{x} - \mathbf{x}_i\|_2) + D_x^{(m)}(c_1^+ x + c_2^+ y + c_3^+), & \mathbf{x} \in \Omega^+ \\ D_x^{(m)} p^- &= \sum_{i=1}^{ns} \eta_i^- D_x^{(m)} \phi(\|\mathbf{x} - \mathbf{x}_i\|_2) + D_x^{(m)}(c_1^- x + c_2^- y + c_3^-), & \mathbf{x} \in \Omega^- \end{aligned} \quad (7)$$

are then obtained by evaluating each of them at the center where the stencil is based. These equations can be written into a matrix form as simplified into a dot product of the form

$$\begin{aligned} D_x^{(m)} u^+ &= H_1^{(m)} \lambda^+, & D_x^{(m)} u^- &= H_2^{(m)} \lambda^-, \\ D_x^{(m)} v^+ &= H_1^{(m)} \zeta^+, & D_x^{(m)} v^- &= H_2^{(m)} \zeta^-, \\ D_x^{(m)} p^+ &= H_1^{(m)} \eta^+, & D_x^{(m)} p^- &= H_2^{(m)} \eta^-. \end{aligned} \quad (8)$$

If the stencil is based at the node  $\mathbf{x}_j$ , then the row matrices  $H_1^{(m)}$  and  $H_2^{(m)}$  each having size  $1 \times (ns + 3)$  are of the form

$$\begin{aligned} H_1^{(m)} &= [D_x^{(m)} \phi(\|\mathbf{x}_j - \mathbf{x}_1\|) \quad D_x^{(m)} \phi(\|\mathbf{x}_j - \mathbf{x}_2\|) \dots D_x^{(m)} \phi(\|\mathbf{x}_j - \mathbf{x}_{ns}\|) \quad D_x^{(m)} x_j \quad 0 \quad 0], \\ &\mathbf{x}_j \in \Omega_j \subset \Omega^+, \\ H_2^{(m)} &= [D_x^{(m)} \phi(\|\mathbf{x}_j - \mathbf{x}_1\|) \quad D_x^{(m)} \phi(\|\mathbf{x}_j - \mathbf{x}_2\|) \dots D_x^{(m)} \phi(\|\mathbf{x}_j - \mathbf{x}_{ns}\|) \quad D_x^{(m)} x_j \quad 0 \quad 0], \\ &\mathbf{x}_j \in \Omega_j \subset \Omega^-. \end{aligned}$$

By combining (5) and (8), we have

$$\begin{aligned} D_x^{(m)} u^+ &= (H_1^{(m)} B_1^{-1}) u^+, & D_x^{(m)} u^- &= (H_2^{(m)} B_2^{-1}) u^-, \\ D_x^{(m)} v^+ &= (H_1^{(m)} B_1^{-1}) v^+, & D_x^{(m)} v^- &= (H_2^{(m)} B_2^{-1}) v^-, \\ D_x^{(m)} p^+ &= (H_1^{(m)} B_1^{-1}) p^+, & D_x^{(m)} p^- &= (H_2^{(m)} B_2^{-1}) p^-. \end{aligned} \quad (9)$$

Thus the differential operator of order  $m$  with respect to  $x$  can be approximated as

$$D_x^{(m)} = \begin{cases} H_1^{(m)} B_1^{-1}, & \mathbf{x} \in \Omega^+, \\ H_2^{(m)} B_2^{-1}, & \mathbf{x} \in \Omega^-. \end{cases} \quad (10)$$

Similarly, the differential operator of order  $m$  with respect to  $y$  can be approximated as

$$D_y^{(m)} = \begin{cases} G_1^{(m)} B_1^{-1}, & \mathbf{x} \in \Omega^+, \\ G_2^{(m)} B_2^{-1}, & \mathbf{x} \in \Omega^-, \end{cases} \quad (11)$$

and if the stencil is based at nodes  $\mathbf{x}_j$ , then the row matrices  $G_1^{(m)}$  and  $G_2^{(m)}$  each having size  $1 \times (ns + 3)$  are of the form

$$\begin{aligned} G_1^{(m)} &= [D_y^{(m)} \phi(\|\mathbf{x}_j - \mathbf{x}_1\|) \quad D_y^{(m)} \phi(\|\mathbf{x}_j - \mathbf{x}_2\|) \dots D_y^{(m)} \phi(\|\mathbf{x}_j - \mathbf{x}_{ns}\|) \quad 0 \quad D_y^{(m)} y \quad 0], \\ &\mathbf{x}_j \in \Omega_j \subset \Omega^+, \\ G_2^{(m)} &= [D_y^{(m)} \phi(\|\mathbf{x}_j - \mathbf{x}_1\|) \quad D_y^{(m)} \phi(\|\mathbf{x}_j - \mathbf{x}_2\|) \dots D_y^{(m)} \phi(\|\mathbf{x}_j - \mathbf{x}_{ns}\|) \quad 0 \quad D_y^{(m)} y \quad 0], \\ &\mathbf{x}_j \in \Omega_j \subset \Omega^-. \end{aligned}$$

The stencil weights  $H_1^{(m)} B_1^{-1}$ ,  $H_2^{(m)} B_2^{-1}$ ,  $G_1^{(m)} B_1^{-1}$  and  $G_2^{(m)} B_2^{-1}$  can be found easily, because the matrices  $B_1$  and  $B_2$  are non-singular. We approximate the derivative by multiplying the stencil weight by the function values at the center.

### 3.4. Radial basis function collocation

After discretizing the PDE in the manner of collocation, a system of linear equations of the form

$$A\alpha = B, \quad (12)$$

is obtained by putting the corresponding derivatives (10) and (11) in the governing Eqs. (1a)-(1c), (2b), (2d) and (2f). We also incorporate the Dirichlet boundary conditions (1d), (1e) for velocities  $u$  and  $v$  and the interface conditions (2a), (2c), and (2e) in the linear system (12). It is pertinent to mention that we need boundary condition for the pressure term which is not mentioned in the model equation. To accomplish this, we integrate (1a) and (1b) and eliminate the constant term to get

$$\mu \left( \frac{\partial u^+}{\partial x} + \int \frac{\partial^2 u^+}{\partial y^2} dx \right) - p = \int f_u^+ dx. \quad (13)$$

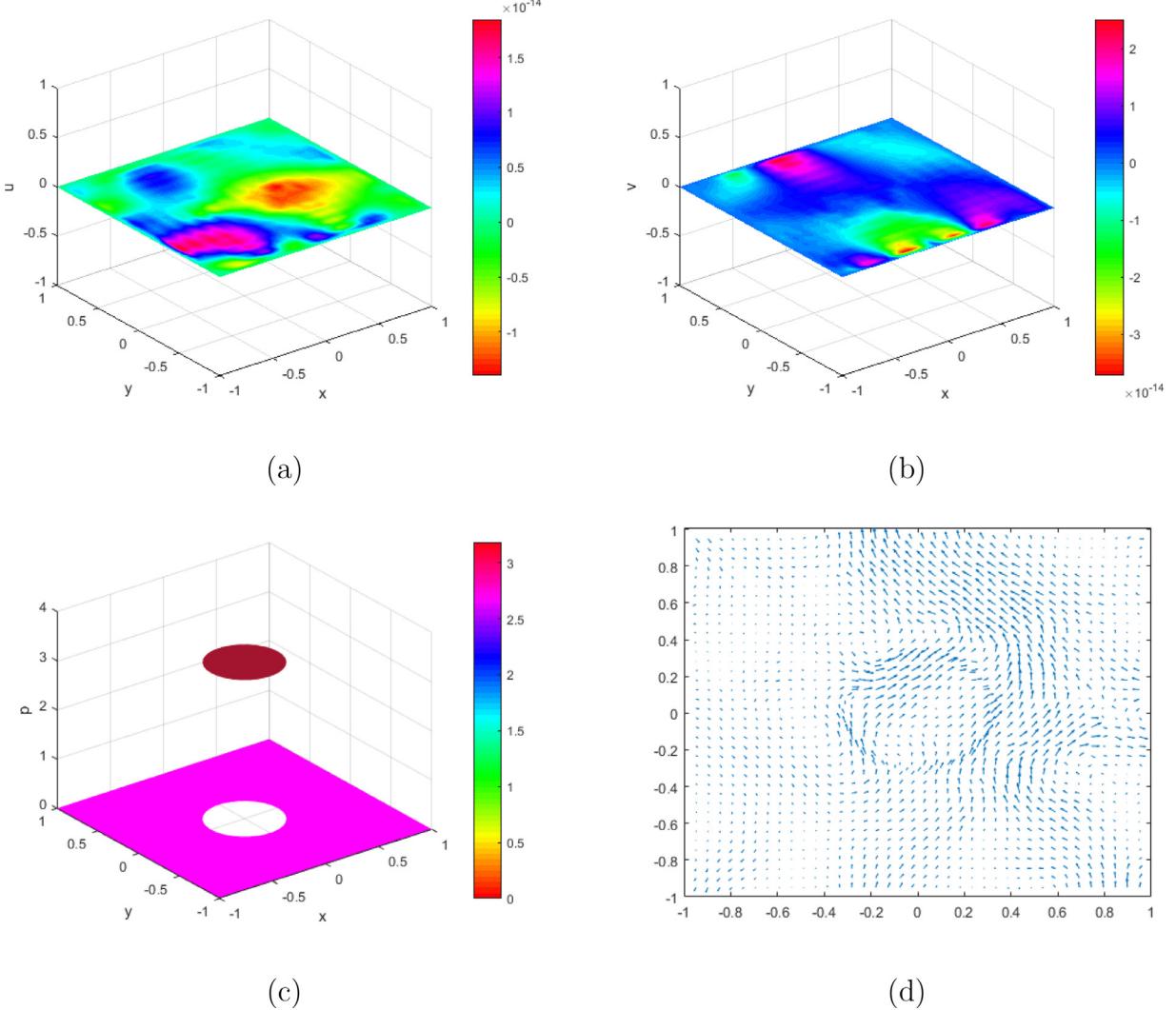
This equation can be used on the boundary  $\partial\Omega$  for the missing boundary conditions of the pressure term.

Eq. (12) can be solved for  $\alpha$  by any direct linear solver. Collecting the discretized form of the main equation along-with boundary and interface conditions, the matrix  $A$ , the vectors  $B$  and  $\alpha$  take the following forms

$$A = \left\{ \begin{array}{cccccc} A_2^+ & \mathbf{0} & \mathbf{0} & \mathbf{0} & -A_x^+ & \mathbf{0} \\ \mathbf{0} & A_2^- & \mathbf{0} & \mathbf{0} & \mathbf{0} & -A_x^- \\ \mathbf{0} & \mathbf{0} & A_2^+ & \mathbf{0} & -A_y^+ & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & A_2^- & \mathbf{0} & -A_y^- \\ A_x^+ & \mathbf{0} & A_y^+ & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & A_x^- & \mathbf{0} & A_y^- & \mathbf{0} & \mathbf{0} \\ \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ A_u^+ & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\mathbf{I} & \mathbf{0} \\ \mathbf{I}^+ & -\mathbf{I}^- & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ A_n^+ & -A_n^- & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I}^+ & -\mathbf{I}^- & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & A_n^+ & -A_n^- & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{I}^+ & -\mathbf{I}^- \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & A_n^+ & -A_n^- \end{array} \right\},$$

$$B = \left\{ \begin{array}{c} f_u^+(\xi_{\Omega^+}) \\ f_u^-(\xi_{\Omega^-}) \\ f_v^+(\xi_{\Omega^+}) \\ f_v^-(\xi_{\Omega^-}) \\ f_c^+(\xi_{\Omega^+}) \\ f_c^-(\xi_{\Omega^-}) \\ u_b^+(\xi_{\partial\Omega}) \\ u_b^+(\xi_{\partial\Omega}) \\ \int f_u^+ dx |_{\xi_{\partial\Omega}} \\ g_u(\xi_{\Gamma}) \\ h_u(\xi_{\Gamma}) \\ g_v(\xi_{\Gamma}) \\ h_v(\xi_{\Gamma}) \\ g_p(\xi_{\Gamma}) \\ h_p(\xi_{\Gamma}) \end{array} \right\}, \quad \alpha = \left\{ \begin{array}{c} \mathbf{u}^+ \\ \mathbf{u}^- \\ \mathbf{v}^+ \\ \mathbf{v}^- \\ \mathbf{p}^+ \\ \mathbf{p}^- \end{array} \right\}. \quad (14)$$

Note that the component matrices of the matrix  $A$  are themselves sparse. The explicit form of the component matrices of the matrix  $A$  in (14) can



**Fig. 4.** Computational results for test problem (1). (a): Surface plot of the numerical velocity  $u$ . (b): Surface plot of the numerical velocity  $v$ . (c): pressure field. (d): velocity filed with maximum magnitude  $10^{-14}$ .

be written as

$$\begin{aligned} A_2^+ &= \mu(D_x^{(2)} + D_y^{(2)}) = \mu(H_1^{(2)} B_1^{-1} + G_1^{(2)} B_1^{-1}), \quad x \in \Omega^+ \\ A_2^- &= \mu(D_x^{(2)} + D_y^{(2)}) = \mu(H_2^{(2)} B_2^{-1} + G_2^{(2)} B_2^{-1}), \quad x \in \Omega^- \\ A_x^+ &= D_x^{(1)} = H_1^{(1)} B_1^{-1}, \quad x \in \Omega^+, \\ A_x^- &= D_x^{(1)} = H_2^{(1)} B_2^{-1}, \quad x \in \Omega^-, \\ A_y^+ &= D_y^{(1)} = G_1^{(1)} B_1^{-1}, \quad x \in \Omega^+, \\ A_y^- &= D_y^{(1)} = G_2^{(1)} B_2^{-1}, \quad x \in \Omega^-, \\ A_n^+ &= (D_x^{(1)}, D_y^{(1)}). \vec{n} = (H_1^{(1)} B_1^{-1}, G_1^{(1)} B_1^{-1}). \vec{n}, \quad x \in \Omega^+, \\ A_n^- &= (D_x^{(1)}, D_y^{(1)}). \vec{n} = (H_2^{(1)} B_2^{-1}, G_2^{(1)} B_2^{-1}). \vec{n}, \quad x \in \Omega^-. \end{aligned}$$

To incorporate the Dirichlet boundary condition at the interface and outer boundary in the coefficient matrix  $A$ , we assume that  $I_1$  and  $I_2$  be the identity matrices, each of size  $(N^1 + N_\Gamma) \times (N^1 + N_\Gamma)$  and  $(N^2 + N_b + N_\Gamma) \times (N^2 + N_b + N_\Gamma)$  respectively. If the nodes arranged in the form  $[\xi_{\partial\Omega}; \xi_{\Omega^+}; \xi_\Gamma]$  in the outer domain, and  $[\xi_\Gamma; \xi_{\Omega^-}]$  in the inner domain in programming, then the matrix  $I$  is filled with the first  $N_b$  rows of matrix  $I_2$  and  $I^+$  is filled with the last  $N_\Gamma$  rows of  $I_2$ . Similarly,  $I^-$  is filled with first  $N_\Gamma$  rows of  $I_1$ .

#### 4. Numerical experiments

In this section, we discuss the meshless numerical solution of the Stokes equations, and compare its accuracy with the published literature

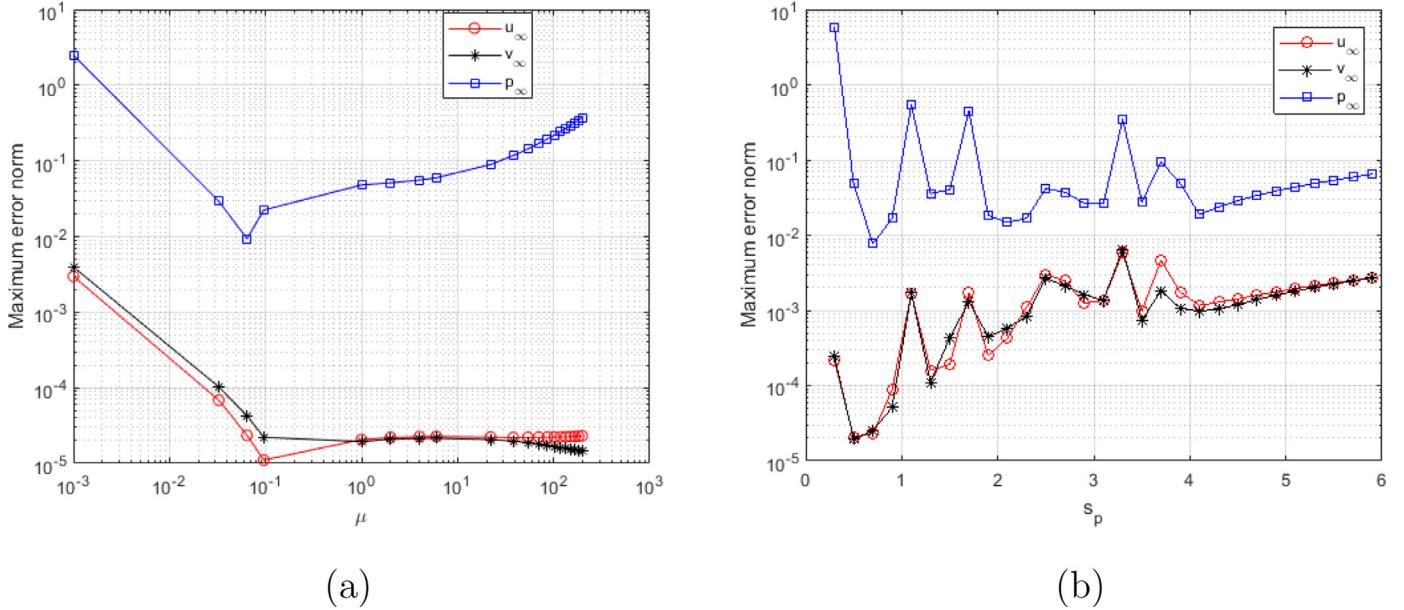
results. We take a square domain  $[-1, 1] \times [-1, 1]$  for each test problem with different types of interfaces as shown in Fig. 3. Value of the shape parameter is kept fixed at 0.5 unless mentioned otherwise and the stencil size is taken 30. We represent the total number of uniform nodes by  $N \times N$ . The total number of nodes in the interior domain  $\Omega^-$  and in the exterior domain  $\Omega^+$  is represented by  $N_1$  and  $N_2$ , respectively. To check accuracy of the meshless method, we use the following criteria

$$u_\infty = \max |\mathbf{u}_{app} - \mathbf{u}_{exact}|,$$

where  $\mathbf{u}_{exact}$  is the exact solution and  $\mathbf{u}_{app}$  is the approximate meshless numerical solution. We also investigate the condition number ( $Cond$ ) of the interpolation matrix in the last test problem.

**Test Problem 1.** We solve a static bubble problem by the proposed meshless method using the same geometry, interface and parameters as reported in [38]. Physically, a balance between the pressure and the surface tension force of a circular bubbles makes the bubble interface static in time. We consider (1) with  $f_u^+, f_u^-, f_v^+, f_v^-, f_c^+$  and  $f_c^-$  equal to zero. Both the velocity components at the boundary are set equal to zero and  $\mu$  is 1. The bubble is considered at the circular position having center at  $(0,0)$  and radius  $R = 0.31415926$ . The jump conditions are

$$[p]_\Gamma = -\tau\kappa, \quad \left[ \frac{\partial p}{\partial n} \right]_\Gamma = 0,$$



**Fig. 5.** Sensitivity of (a):  $\mu$  versus maximum error norm. (b):  $s_p$  versus maximum error norm: test problem (2).

**Table 1**  
Numerical results of LMM for test problem 1.

Uniform data				Scattered data				
$N \times N$	$\max(u)$	$\max(v)$	$[p].R$	$N_1$	$N_2$	$\max(u)$	$\max(v)$	$[p].R$
$20 \times 20$	$3.95e-14$	$4.82e-14$	1.000	44	531	$7.99e-15$	$6.79e-15$	1.000
$30 \times 30$	$1.33e-14$	$1.86e-14$	1.000	59	710	$1.22e-14$	$1.16e-14$	1.000
$40 \times 40$	$6.72e-15$	$5.33e-15$	1.000	76	980	$3.16e-14$	$2.73e-14$	1.000
$50 \times 50$	$7.18e-15$	$1.06e-14$	1.000	116	1391	$7.01e-13$	$9.04e-13$	1.000
$60 \times 60$	$8.50e-15$	$7.53e-15$	1.000	140	1767	$3.40e-15$	$3.02e-15$	1.000
$70 \times 70$	$4.11e-14$	$2.79e-14$	1.000	181	2192	$1.16e-14$	$1.70e-14$	1.000
$80 \times 80$	$2.64e-15$	$2.04e-15$	1.000	237	2871	$7.98e-15$	$5.35e-15$	1.000
$90 \times 90$	$4.54e-15$	$4.17e-15$	1.000	322	4019	$5.24e-15$	$3.53e-15$	1.000
$100 \times 100$	$3.40e-14$	$4.10e-14$	1.000	471	5631	$7.63e-15$	$4.12e-15$	1.000

Courtesy [38]			
Total mesh nodes	Smallest mesh size	$\max(u)$	$[p].R$
1129	0.01	$1.80e-09$	1.000
2036	0.005	$9.77e-09$	1.000
3330	0.0025	$4.02e-09$	1.000

$$\begin{aligned} [u]_\Gamma &= 0, \quad \left[ \frac{\partial u}{\partial n} \right]_\Gamma = 0, \\ [v]_\Gamma &= 0, \quad \left[ \frac{\partial v}{\partial n} \right]_\Gamma = 0, \end{aligned} \quad (15)$$

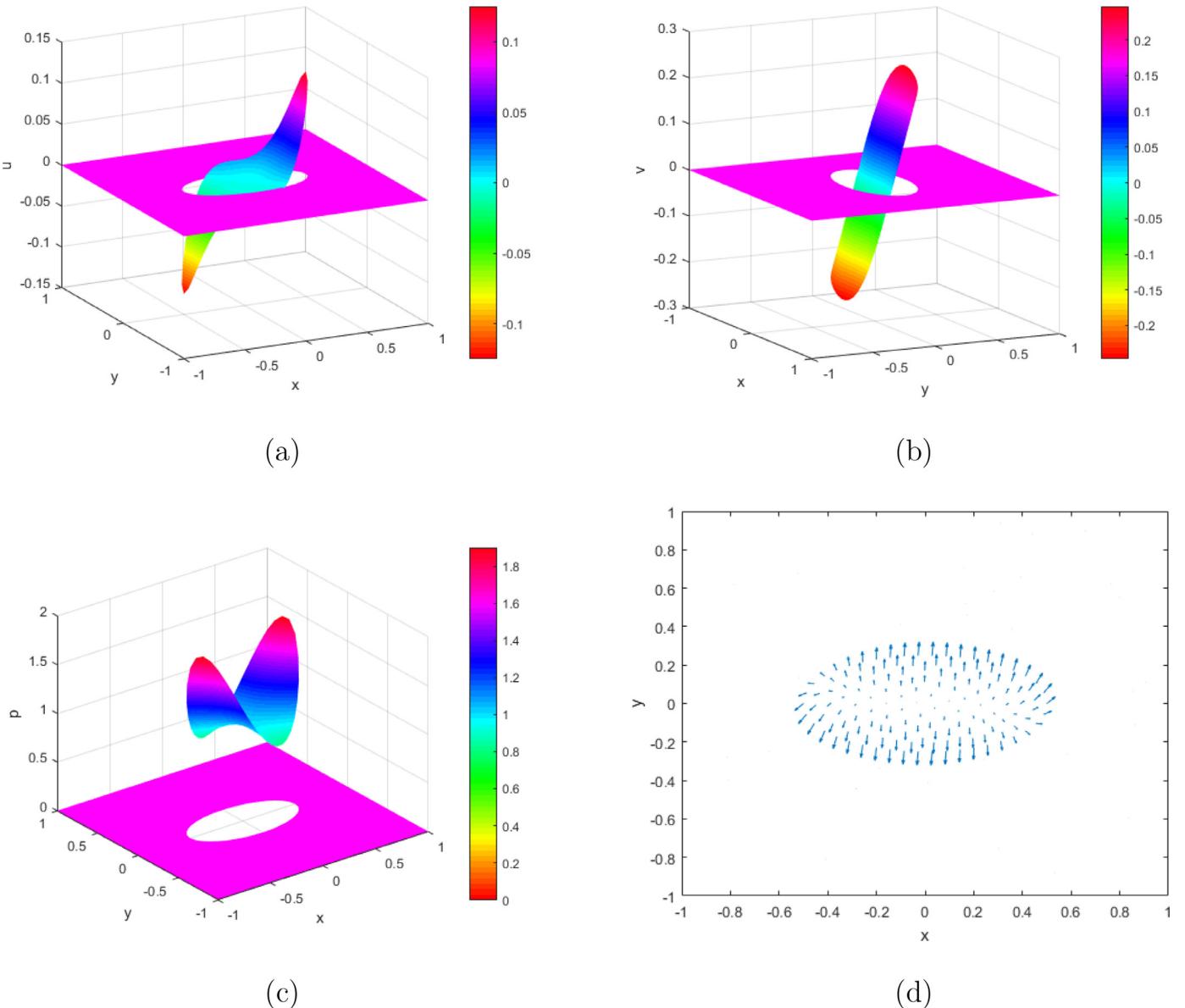
where  $\kappa$  is the mean curvature and  $\tau$  is the surface tension strength which is assumed to be constant. We solve numerically the pressure and the velocity for the circular bubble. The exact value of the velocity is zero, and the exact value of the pressure is a piecewise constant function having different values in the inner and outer sub-domains. The difference in the pressure values is equal to the product of surface tension and the curvature.

The computational results are shown in Table 1 alongwith comparison made with [38]. Like [38], two criteria are adopted to check accuracy of the proposed meshless method. We first find maximum values of  $|u|$  and  $|v|$  over the domain and then verify its maximum value is  $[p].R = 1$  at the interface. Note that, the exact solutions satisfy  $\max|u| = 0$  and  $\max|v| = 0$  and  $[p].R = 1$ . In the previous work [38], a two dimensional interface-fitted adaptive mesh method was used to solve the static bubble problem, i.e., the interface-fitting technique is combined with an existing adaptive mesh approach in the finite element framework to

solve the Stokes flow. In our implementation a meshless method based on radial basis functions is used without any adaptive mesh strategy, thus reducing the computational efforts due to non-adaptivity. The velocity profile for this test problem is shown in Fig. 4.

Comparison shown in Table 1 suggests that the proposed approach provides better accuracy, reaching  $10^{-14}$ , on both coarse and dense nodes. We can observe from Table 1 that accuracy of the method [38] is of the order  $10^{-9}$  on both coarser and dense meshes. The range of the mesh-size is taken  $0.01 - 0.0025$  for the method [38], whereas in the current implementation of the meshless method, the mesh size range is taken  $0.1 - 0.02$ , showing better performance of the current method versus method [38].

The dependencies of the viscosity parameter  $\mu$  and shape parameter  $s_p$  are shown in Fig. 5, for test problem 2. It can be observed from this figure that the numerical results of the present meshless method are stable for a range of values of  $\mu$  and the present method has a good agreement with the exact solution. Similarly, the meshless method has better accuracy for small value of shape parameter  $s_p$  as compared to that of large value of the shape parameter  $s_p$ .



**Fig. 6.** Computational results for test problem 2. (a): Surface plot of the numerical velocity  $u$ . (b): Surface plot of the numerical velocity  $v$ . (c): pressure field. (d): velocity filed with maximum magnitude 0.2772.

**Test Problem 2.** We consider a benchmark problem in which the computational domain, interface and parameters used in Stokes equations are taken similar to Li [39] for the purpose of comparison.

$$u = \begin{cases} x^3 + y^3 & \text{if } (x, y) \in \Omega^- \\ 0 & \text{if } (x, y) \in \Omega^+ \end{cases} \quad (16)$$

$$v = \begin{cases} x^3 + \sin(y) & \text{if } (x, y) \in \Omega^- \\ 0 & \text{if } (x, y) \in \Omega^+ \end{cases} \quad (17)$$

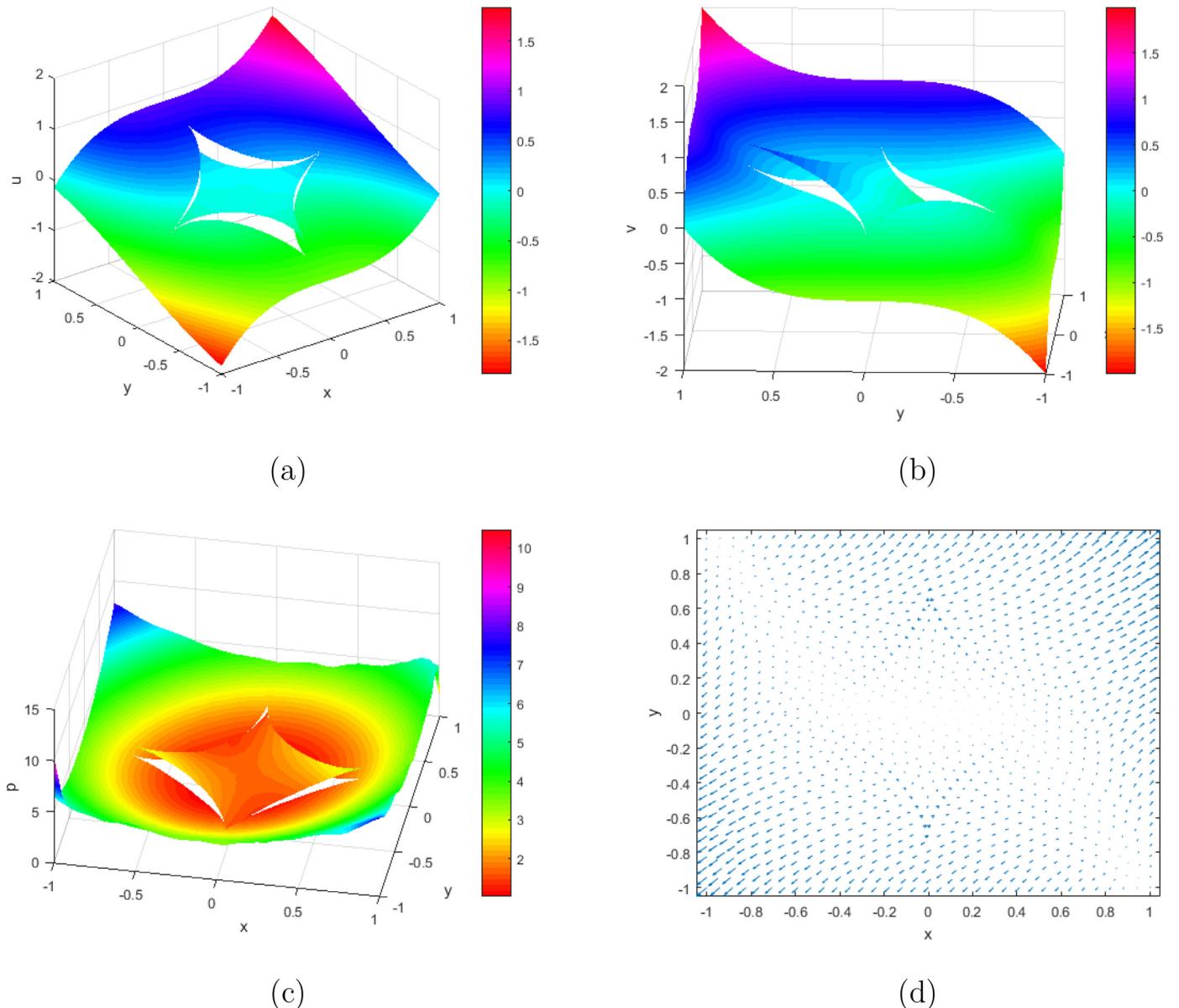
$$p = \begin{cases} 3x^2 + 6xy + \cos(y) & \text{if } (x, y) \in \Omega^- \\ 0 & \text{if } (x, y) \in \Omega^+ \end{cases} \quad (18)$$

These functions satisfy the Stokes Eqs. (1). In this case  $f_u^+$ ,  $f_u^-$ ,  $f_v^+$ ,  $f_v^-$ , and  $f_c^+$  are taken zero, but  $f_c^- = 3x^2 + \cos(y)$ . The difficulty associated with the problem is the discontinuities in the velocity and the source term in the continuity equation. The interface considered in this case is an ellipse  $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$  with semi major axis  $a = 0.5$  and semi minor axis  $b = 0.25$ , respectively. The velocity components at the boundaries are zero. Values of the jump conditions are not mentioned here as they can be easily found from the exact solution.

In Fig. 6, we have shown the surface plots velocity components, pressure field and velocity field. Table 2 shows a comparison of the present meshless method with the Immersed interface method [39] as well as mutual comparison of the meshless method for uniform and scattered data nodes. Table 2 reveals that the velocity computed through meshless method is in good agreement with both exact velocity components for both type of nodes. However, as compared to the velocity field, approximate value of the pressure field through meshless method is comparatively of low accuracy. This is due to the lack of information of the pressure term at the boundary and the mechanism we have adopted for its evaluation. Numerical results of the meshless method and immersed finite element method are comparable. As can be seen in this test problem, the local meshless method provides better accuracy for the problems having different types of interfaces and provides a viable alternative to the existing class of methods.

**Test Problem 3.** We define the following functions:

$$u = \begin{cases} x^3 + \sin(y) & \text{if } (x, y) \in \Omega^- \\ x^3 + y^3 & \text{if } (x, y) \in \Omega^+ \end{cases} \quad (19)$$



**Fig. 7.** Computational results for test problem (3). (a): Surface plot of the numerical velocity  $u$ . (b): Surface plot of the numerical velocity  $v$ . (c): pressure field. (d): velocity filed with maximum magnitude 2.7186.

$$v = \begin{cases} x^3 + y^3 & \text{if } (x, y) \in \Omega^- \\ x^3 + \sin(y) & \text{if } (x, y) \in \Omega^+ \end{cases} \quad (20)$$

$$p = \begin{cases} 3x^2 + 3y^2 - x \sin(y) & \text{if } (x, y) \in \Omega^- \\ 3x^2 + 6xy + \cos(y) & \text{if } (x, y) \in \Omega^+ \end{cases} \quad (21)$$

which satisfy the Stokes Eqs. (1). Values of  $f_u^-$ ,  $f_u^+$ , and  $f_v^-$  are zero, while  $f_v^+ = 6x + x \cos(y)$ ,  $f_c^+ = 3x^2 + 3y^2$  and  $f_c^- = 3x^2 + \cos(y)$ . Value of the velocity at the boundaries and other related jumps condition are derived from the exact solution, and  $\mu$  is set equal to 1. This problem is similar to the previous one except the velocity, which is not zero in the sub-domain  $\Omega^+$ . The source term is again discontinuous as in the previous problem. The interface is defined for the current problem as

$$\begin{aligned} x &= 0.7(2 \cos(\theta) + \cos(2\theta)), \\ y &= 0.7(2 \sin(\theta) - \sin(2\theta)), \quad 0 \leq \theta \leq 2\pi. \end{aligned} \quad (22)$$

In Fig. 7 we have shown the surface plots velocity components, pressure field and velocity field. The numerical results are shown in Table 3,

where a comparison is made for two types of boundary conditions for the pressure, that is to say the boundary condition (23) and boundary condition (13). In [38], Stokes Eq. (1) is converted into three Poisson equations and solved them iteratively. A unit normal vector is multiplied with vectorized form of Stokes Eq. (1) to get Neumann's boundary conditions for pressure term  $p$  at the boundary  $\partial\Omega$  i.e.,

$$\frac{\partial p}{\partial \mathbf{n}}|_{\partial\Omega} = \mu (\nabla^2 \mathbf{u} \cdot \mathbf{n})|_{\partial\Omega}. \quad (23)$$

It can be seen from the numerical results that the meshless method has good approximation to the exact velocity in both the cases. However, meshless method provides good approximation of the pressure term is more accurate using boundary condition (13) as compared to boundary condition (23). The meshless solution of the Stokes equation with Neumann's boundary condition (23) has a high condition number as compared to the boundary condition (13).

**Table 2**  
Numerical results of LMM for test problem 2.

Uniform data				Scattered data				
N × N	$u_\infty$	$v_\infty$	$p_\infty$	$N_1$	$N_2$	$u_\infty$	$v_\infty$	$p_\infty$
20 × 20	3.63e - 04	3.72e - 04	1.77e + 00	100	689	1.78e - 05	2.46e - 05	1.75e - 01
30 × 30	8.64e - 05	1.32e - 04	5.12e + 00	145	957	4.78e - 05	4.89e - 05	2.50e + 00
40 × 40	1.12e - 05	1.34e - 05	9.55e - 02	175	1359	1.78e - 04	1.54e - 04	1.71e + 00
50 × 50	6.51e - 06	6.76e - 06	2.18e - 01	232	1772	1.86e - 05	2.04e - 05	1.39e - 02
60 × 60	4.70e - 06	4.23e - 06	9.52e - 02	296	2144	1.61e - 05	1.06e - 05	1.16e - 01
70 × 70	6.80e - 06	8.24e - 06	1.16e - 01	415	2809	1.49e - 05	1.79e - 05	3.32e - 02
80 × 80	6.12e - 06	4.33e - 06	1.51e - 01	595	3931	1.65e - 05	1.30e - 05	5.43e - 03
90 × 90	4.75e - 05	3.66e - 05	2.95e - 01	595	5505	8.79e - 06	1.02e - 05	6.37e - 02
100 × 100	1.07e - 05	1.65e - 05	6.24e - 02	925	8658	7.75e - 05	4.04e - 05	1.91e - 02

Courtesy ([39])			
$N \times N$			
	$u_\infty$	$v_\infty$	$p_\infty$
20 × 20	7.31e - 04	5.31e - 04	6.94e - 02
40 × 40	1.99e - 04	1.59e - 04	4.99e - 02
80 × 80	5.02e - 05	4.18e - 05	2.54e - 02
160 × 160	1.22e - 05	1.07e - 05	1.61e - 02
320 × 320	3.04e - 06	2.72e - 06	1.02e - 02

**Table 3**  
Numerical results of LMM for test problem 3.

$N_1$	$N_2$	Neumann's boundary condition Eq. (23)			Cond	Boundary condition Eq. (13)		
		$u_\infty$	$v_\infty$	$p_\infty$		$u_\infty$	$v_\infty$	$p_\infty$
85	492	5.02e - 03	7.60e - 03	1.98e + 11	1.14e + 20	4.17e - 03	4.60e - 03	1.85e + 01
107	649	1.08e - 04	2.79e - 04	8.50e + 07	3.10e + 21	3.72e - 04	4.75e - 04	2.34e + 01
145	905	2.01e - 04	2.87e - 04	2.16e + 09	5.57e + 20	2.92e - 04	3.47e - 04	4.26e - 01
214	1279	9.06e - 05	7.68e - 05	1.89e + 08	3.25e + 21	4.18e - 04	5.73e - 04	1.16e + 01
266	1630	2.82e - 05	4.56e - 05	1.93e + 06	2.21e + 21	6.07e - 05	6.49e - 05	8.49e - 01
331	2024	7.11e - 05	7.65e - 05	1.11e + 09	7.30e + 22	2.18e - 04	1.61e - 04	1.65e + 01
436	2654	4.20e - 05	3.38e - 05	4.13e + 08	6.57e + 22	1.38e - 04	7.37e - 05	1.22e + 00
604	3724	4.39e - 04	4.42e - 04	7.21e + 08	3.62e + 22	3.62e - 04	2.17e - 04	2.58e + 00
872	5216	6.39e - 04	4.15e - 04	2.77e + 09	1.87e + 22	1.77e - 04	2.34e - 04	1.06e + 00

## 5. Conclusions

We have obtained accurate solution of the Stokes equation by radial basis functions collocation method. Three different cases are considered of the Stokes Eqs. (1), that is to say

- (a) In the first benchmark problem velocity components are zeros in the entire domain and pressure term is unknown piece-wise constant having different linear values in the inner and outer sub-domains.
- (b) In the second benchmark problem, velocity components and pressure are zero in  $\Omega^+$  but have non-zero values in  $\Omega^-$  and the right side function of the continuity Eq. (1c) is discontinuous. In this problem, we dealt with the discontinuities of all field variables in the computational domain.
- (c) The third problem resembles with previous two cases except the fact the field variables velocity and pressure are non-zeros in inner and outer sub-domains.

It can be seen that the present method is capable to provide high accuracy with lower computational cost in comparison with the literature methods [38]. In addition, the flexibility of the local meshless method for different nodal distribution, i.e., uniform and scattered points is also unveiled and evaluated with better and accurate results.

## Declaration of Competing Interest

None.

## Acknowledgments

The authors greatly acknowledge the Higher Education Commission of Pakistan for their support in the present work under the project No. 7870/KPK/NRPU/R&D/HEC/2017.

## References

- [1] Ahmad M, Siraj-ul-Islam, Larsson E. Local meshless methods for second order elliptic interface problems with sharp corners. *J Comp Phys* 2020;416:109500.
- [2] Siraj-ul-Islam, Aziz I, Ahmad M. Numerical solution of two-dimensional elliptic PDEs with nonlocal boundary conditions. *Comput Math Applic* 2015;69:180–205.
- [3] Aziz I, Siraj-ul-Islam, Haider N. Meshless and multi-resolution collocation techniques for steady state interface models. *Int J Comput Methods* 2018;15:1750073.
- [4] Siraj-ul-Islam, Ahmad M. Meshless analysis of elliptic interface boundary value problems. *Eng Anal Bound Elem* 2017;92:38–49.
- [5] Ahmad M, Siraj-ul-Islam. Meshless analysis of parabolic interface problems. *Eng Anal Bound Elem* 2018;94:134–52.
- [6] Esfahani H, Roohani GH, Etesami SK. A meshless method for the investigation of electromagnetic scattering from arbitrary shaped anisotropic cylindrical objects. *J Electromag Waves Appl* 2017;31:477–94.
- [7] Yang J, Ferry MP, Hu H. Solving the stationary Navier–Stokes equations by using Taylor meshless method. *Eng Anal Bound Elem* 2019;98:8–16.
- [8] Gong J, Zou D, Kong X, Qu Y, Liu J. An extended meshless method for 3D interface simulating soil-structure interaction with flexibly distributed nodes. *Soil Dynamics and Earthquake Engineering* 2019;125: 1056–88.
- [9] Gordon KR, Hutchcraft WE. Two approaches for the numerical determination of electromagnetic fields near material interfaces using radial basis functions in a meshless method. *Num Method Part Differ Equ* 2010;26:945–56.
- [10] Reuther K, Rettenmayr M. Simulating phase transformations by a meshless method with front tracking. *Acta Mater* 2012;60:2128–34.
- [11] B FR, Rizaner A. Approximate solutions of initial value problems for ordinary differential equations using radial basis function networks. *Neur Process Lett* 2018;48:1063–71.
- [12] Siraj-ul-Islam, Zaheer-ud-Din. Meshless methods for two-dimensional oscillatory Fredholm integral equations. *J Comput Appl Math* 2018;335:33–50.

- [13] Parand K, Rad JA. Numerical solution of nonlinear volterra–fredholm–hammerstein integral equations via collocation method based on radial basis functions. *Appl Math Comput* 2012;218:5292–309.
- [14] Assari P, Dehghan M. The approximate solution of nonlinear volterra integral equations of the second kind using radial basis functions. *Appl Numer Math* 2018;131:140–57.
- [15] Siraj-ul-Islam, Aziz I, Zaheer-ud-din. Meshless methods for multivariate highly oscillatory fredholm integral equations. *Eng Anal Bound Elem* 2015;53:100–12.
- [16] Mai-Duy N, Thai-Quang N, Hoang-Trieu T-T, Tran-Cong T. A compact 9 point stencil based on integrated rbf's for the convection–diffusion equation. *Appl Math Model* 2014;38(4):1495–510.
- [17] Rashidinia J, Khani M, Fasshauer GE. A stable gaussian radial basis function method for solving nonlinear unsteady convection–diffusion–reaction equations. *Comput Math Applic* 2018;75:1831–50.
- [18] Xie H, Li D. A meshless method for burgers?? equation using MQ-RBF and high-order temporal approximation. *Appl Math Model* 2013;37:9215–22.
- [19] Fan C, Yang C, Gu M. Applications of the local RBF collocation method and the fictitious time integration method for burgers?? equations. *Procedia Eng* 2014;79:569–74.
- [20] Soleymani F, Akgül A. Improved numerical solution of multi-asset option pricing problem: a localized RBF-FD approach. *Chaos, Solitons & Fractals* 2019;119:298–309.
- [21] Ullah MZ. An RBF-FD sparse scheme to simulate high-dimensional black-Scholes partial differential equations. *Comput Math Applic* 2020;79(2):426–39.
- [22] Siraj-ul-Islam, Ahmad I. A comparative analysis of local meshless formulation for multi-asset option models. *Eng Anal Bound Elem* 2016;65:159–76.
- [23] Radmanesh M, Ebadi M. A local mesh-less collocation method for solving a class of time-dependent fractional integral equations: 2d fractional evolution equation. *Eng Anal Bound Elem* 2020;113:372–81.
- [24] Kumar A, Bhardwaj A, Kumar BVR. A meshless local collocation method for time fractional diffusion wave equation. *Comput Math Applic* 2019;78:1851–61.
- [25] Fu Z, Zhang J, Li P, Zheng J-H. A semi-Lagrangian meshless framework for numerical solutions of two-dimensional sloshing phenomenon. *Eng Anal Bound Elem* 2020;112:58–67.
- [26] Mramor K, Vertnik R, Šarler B. Application of the local RBF collocation method to natural convection in a 3d cavity influenced by a magnetic field. *Eng Anal Bound Elem* 2020;116:1–13.
- [27] Fu Z, Reutskiy S, Sun H, Ma J, Khan MA. A robust kernel-based solver for variable-order time fractional pdes under 2d/3d irregular domains. *Appl Math Lett* 2019;94:105–11.
- [28] Cavoretto R, Rossi AD. A two-stage adaptive scheme based on RBF collocation for solving elliptic PDEs. *Comput Math Applic* 2020;79:3206–22.
- [29] Abbaszadeh M, Dehghan M. An upwind local radial basis functions-differential quadrature (RBFs-DQ) technique to simulate some models arising in water sciences. *Ocean Eng* 2020;197:106844.
- [30] Chaabasri E, Jeyar M, Borthwick G. Explicit radial basis function collocation method for computing shallow water flows. *Procedia Comput Sci* 2019;148:361–70.
- [31] Zhou Y. A sharp-interface treatment technique for two-phase flows in meshless methods. *Comput Fluid* 2017;147:90–101.
- [32] Li X. Meshless analysis of two-dimensional stokes flows with the galerkin boundary node method. *Eng Anal Bound Elem* 2010;34:79–91.
- [33] Tan F, Zhang Y, Li Y. Development of a meshless hybrid boundary node method for stokes flows. *Eng Anal Bound Elem* 2013;37:899–908.
- [34] Waters J, Pepper DW. Global versus localized RBF meshless methods for solving incompressible fluid flow with heat transfer. *Num Heat Transf* 2015;68:185–203.
- [35] Lin H, Atluri SN. The meshless local Petrov-Galerkin method for solving incompressible Navier-Stokes equations. Center for Aerospace Research and Education, 7704 Boelter Hall University of California, Los Angeles 2001;2:117–42.
- [36] Abbaszadeh M, Dehghan M. Reduced order modeling of time-dependent incompressible Navier-Stokes equation with variable density based on a local radial basis functions-finite difference (LRBF-FD) technique and the POD/DEIM method. *Comput Methods Appl Mech Eng* 2020;364:112914.
- [37] Fornberg B, Flyer N. Fast generation of 2-D node distributions for mesh-free PDE discretizations. *Comput Math Appl* 2015;69:531–44.
- [38] Zheng X, Lowengrub J. An interface-fitted adaptive mesh method for elliptic problems and its application in free interface problems with surface tension. *Adv Comput Math* 2016;42:1225–57.
- [39] Li Z. The immersed interface method: a numerical approach for partial differential equations with interfaces; University of Washington, 1994.
- [40] Kansa EJ. Multiquadratics scattered data approximation scheme with applications to computational fluid-dynamics surface approximations and partial derivative estimates. *Comput Math Applic* 1990;19:127–45.