Journal of Computational Science 79 (2024) 102311

Contents lists available at ScienceDirect

Journal of Computational Science

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

A non-boundary-fitted-grid method, based on compact integrated-RBF approximations, for solving differential problems in multiply-connected domains

T.T.V. Le [a,b], N. Mai-Duy [a,*], K. Le-Cao [c], S. Bordas [d], D.P. Vu [e], T. Tran-Cong [a]

[a] School of Engineering, University of Southern Queensland, Toowoomba, QLD 4350, Australia
[b] Department of Mathematics, Queensland University of Technology, Brisbane, QLD 4000, Australia
[c] Computational Engineering and Design Research Group, School of Engineering and Technology, Van Lang University, Ho Chi Minh City, Viet Nam
[d] Faculty of Computational and Data Sciences, University of Luxembourg, 4365 Esch-sur-Alzette, Luxembourg
[e] WorldQuant LLC, Old Greenwich, United States

ARTICLE INFO

Keywords:
Multiply-connected domains
Non-boundary-fitted grids
Integrated radial-basis-functions
Compact stencils
Viscous flows

ABSTRACT

This paper presents a new non-boundary-fitted-grid numerical technique for solving partial differential equations (PDEs) in multi-hole domains. A multiply-connected domain is converted into a simply-connected domain of rectangular or non-rectangular shape that is discretised using a Cartesian grid. Compact radial basis function (RBF) stencils, which are constructed through integration rather than the conventional differentiation, are used to discretise the field variables. The imposition of inner boundary conditions is conducted by means of body forces that are derived from satisfying the governing equations and prescribed boundary conditions in small subregions. Salient features of the proposed method include: (i) simple pre-processing (Cartesian grid), (ii) high rates of convergence of the solution accuracy with respect to grid refinement achieved with compact integrated-RBF stencils, where both nodal function and derivative values are included in the approximations, (iii) the system matrix kept unchanged for the case of moving holes, and (iv) no interpolation between Lagrange and Euler meshes required. Several linear and nonlinear problems, including rotating-cylinder flows and buoyancy-driven flows in eccentric and concentric annuli, are simulated to verify the proposed technique.

1. Introduction

In solving partial differential equations (PDEs), multiply-connected domains (Fig. 1) can be discretised using boundary-fitted and non-boundary-fitted grids/meshes. In the boundary-fitted-grid/mesh category, unstructured meshes are typically used. Boundary fitted grids/meshes can be used to accurately represent a geometrically-complex domain. Furthermore, an unstructured grid/mesh can locally be improved in selective regions to obtain refined information about the variable fields. However, generating an unstructured mesh is a time-consuming process. In the case of moving boundaries, the computational grid/mesh can be distorted. One needs to generate a new mesh; the variable fields are then projected onto it, which is a sophisticated task. Thus, the use of non-boundary-fitted grids/meshes to represent a multi-hole domain has received a great deal of attention (e.g., [1–14]). The basic idea of non-boundary-fitted-grid/mesh based methods is to extend the problem defined on a multiply-connected domain to that on a domain of simpler shape, where a regular grid/mesh and a fast algebraic solver can be used. The non-boundary-fitted-grid methods have the ability to work with any complex domains. The task of converting a multiply-connected domain into a simpler domain is relatively straightforward. In the case of moving interior holes, the grid/mesh may be kept unchanged and thus there is no need for carrying out the task of projecting the variable fields. Consequently, the system matrix needs to be determined once in the simulation process. In this category, special attention to the imposition of given inner boundary conditions is needed to match the solution on the extended domain, which is composed of the original/physical region and regions of the inner holes, with that on the original/physical domain. For this purpose, a body force is commonly introduced into the governing equations to describe the existence of the internal boundaries. Its main difficulty lies in a way of obtaining the body force field.

Many non-boundary-fitted-grid/mesh based methods have been reported in the literature. For example, Glowinski, Pan and Periaux [15, 16] proposed a class of fictitious-domain methods using Lagrange multipliers for simulating incompressible fluid flows. The methods were

* Corresponding author.
E-mail address: nam.mai-duy@usq.edu.au (N. Mai-Duy).

https://doi.org/10.1016/j.jocs.2024.102311
Received 15 April 2021; Received in revised form 26 July 2023; Accepted 23 April 2024
Available online 25 April 2024
1877-7503/© 2024 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

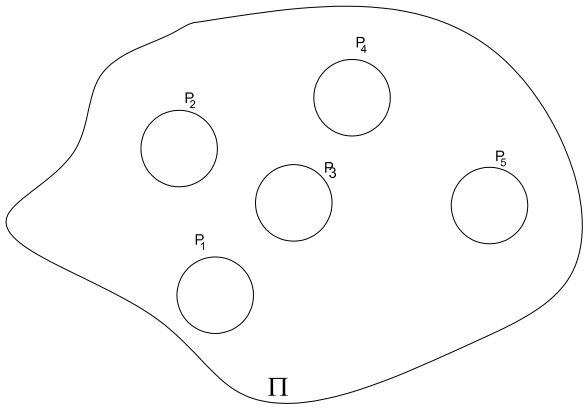

Fig. 1. A multiply-connected domain.

successfully implemented to solve practical problems such as rigid-body/fluid interactions (e.g., [17,18]), fluid/flexible-body interactions (e.g., [19–21]), and particulate suspension flows (e.g., [6,22–24]). Another approach is based on the immersed boundary method proposed in [25]. In this approach, the body force was generated by the elasticity of the material and then "spread out" to grid nodes using Dirac delta functions. In [26], the body force was calculated based on the desired velocities at the interfaces. In [27], the body force was first obtained on the immersed interfaces and was smoothly transferred to fixed grids by Dirac delta functions. In [28], an interpolation scheme for evaluating the velocities satisfying non-slip boundary conditions was proposed and the body force was then directly defined on grid nodes. Later on, Parvizian et al. [1] introduced a finite cell method for solving problems of solid mechanics. Duster et al. [29] extended the finite cell method to 3D-linear-elasticity problems. Maury [30] proposed a fat boundary method (FBM) for solving PDEs in multi-hole domains. Bertoluzza et al. [31] implemented a semi-discrete FBM in the framework of FEM. Vos et al. [32] coupled the classic fictitious domain method and FBM to constitute a method called the implicit FBM. Stein et al. [33,34] proposed a high-order immersed boundary smooth extension (IBSE) method, in which the solution is forced to be globally smooth on the computational domain to achieve high-order accuracy. The IBSE method needs to form and solve a high-order PDE outside the physical domain to smoothly extend the solution from the physical domain to the computational domain, from which the extension to the forcing is derived. The greater the level of regularity, the higher the order of the PDE will be. As discussed in [33], special care is needed in solving the high-order PDE as the system matrix can become ill-conditioned.

RBF approximation schemes have the ability to represent any smooth function to a prescribed degree of accuracy (e.g., [35–38]). Together with other meshless methods [39–41], they have rapidly been developed for the approximation of functions and numerical solution of PDEs. Several types of RBFs contain a free parameter. This class can exhibit an exponential rate of convergence as either the fill distance decreases (an increase in the number of RBFs) or the RBF's width increases. One of the most widely used RBFs is the multiquadric (MQ) function, which has a deep history of theoretical development. The utilisation of MQ-RBFs for solving PDEs was first introduced by Kansa [42], where the RBF approximations are constructed through differentiation (direct/differential approach, DRBFs). Later on, the utilisation of integrated RBFs (IRBFs) for solving PDEs was reported by Mai-Duy and Tran-Cong [43]. For the integral approach, firstly, the highest order derivatives in a given PDE are approximated by RBFs, and then lower order derivatives and the function itself are acquired by integrating RBFs. In this way, one can avoid a decrease in the convergence rate induced by differentiation and obtain a more stable numerical solution [43,44]. DRBFs and IRBFs were implemented with boundary-fitted grids for solution of the Navier–Stokes equation (e.g., [45–50]). We note some recent RBF studies (e.g. [51]) on the use of polyharmonic splines with augmented polynomials for solving PDEs. The use of polyharmonic splines helps overcome the issue of the RBF width and also provides simple ways to estimate convergence rates. Like the polyharmonic-spline-based methods, the integral approach also involves some polynomials. However, these polynomials are produced from integrating RBFs. Unlike the polyharmonic-spline-based methods, nodal values of derivatives of the field variable are included in the RBF approximations to form a determined interpolation system. It should be pointed out that, with global approximations, their interpolation matrices are entirely populated and become ill-conditioned when a large number of nodes are used. Recent RBF research has concentrated on resolving these shortcomings. An efficient way is to utilise local RBF stencils, where only a small subset of nodes are activated for the approximation of a function at a given point, and compact local RBF stencils, where the approximations involve not only function values but also derivative values at grid nodes. A sparse system matrix, which saves computer storage space and promotes the use of a much larger number of nodes, can be obtained with local schemes. Furthermore, like compact finite-difference-based stencils (see [52] and references therein), the inclusion of derivative values in the approximations here can also significantly improve the accuracy of local approximation schemes as shown in, e.g., [50,53]. Works reported in this research direction include [50–64].

In this study, the compact local IRBF stencils [50,54,59] are incorporated into the non-boundary-fitted-grid (NBFG) framework for solving PDEs. In our proposed method, the discretisation is based on only RBF approximations, in contrast with the augmented forcing method [12], where the discretisation is based on localised Hermite RBF approximations for regions near the inner boundaries and finite differences for regions far away from the inner boundaries. Unlike the IBSE method, the inner boundary conditions are incorporated into the body force by solving the same governing PDE on a small subregion that contains the inner boundary of a hole. Our proposed method does not require the use of regularised delta functions, and the system matrix is kept unchanged in the case of moving holes. The solution procedure for simulating fluid flows includes the following steps:

- Step 1: Estimate fluid velocities in local domains near inner boundaries, where the forcing term is omitted.
- Step 2: Derive the forcing terms from the difference between the desired velocities (i.e. velocities acquired taking into account the inner boundary conditions) and the estimated velocities obtained from Step 1. Note that the desired velocities are obtained by solving the governing equations subject to non-slip boundary conditions on local regions enclosing the inner boundaries.
- Step 3: Solve the governing equations in the extended domain with the obtained forcing terms.

Since the compact local IRBF stencils can work on irregular grids, the desired velocities and the forcing terms can be evaluated directly at grid nodes without interpolation.

The remainder of the paper is organised as follows. Section 2 presents the governing equations. Section 3 gives a brief review of integrated RBFs. In Section 4, the proposed IRBF-NBFG technique is described with emphasis placed on the formulation of forcing functions describing the influence of the interior interfaces on the solution. Details for the IRBF discretisation of the governing equations in the extended (rectangular) domain are also included here. Numerical solutions are reported in Section 5. Section 6 concludes the paper.

2. Governing equations

For an incompressible fluid, the conservation laws of momentum and mass are

$$\nabla \cdot \mathbf{u} = 0, \tag{1}$$

$$\rho\frac{D\mathbf{u}}{Dt} = \rho\boldsymbol{g} + \boldsymbol{\nabla}\cdot\boldsymbol{\sigma}, \tag{2}$$

where ρ is the fluid density, $\mathbf{u}$ the velocity vector, $\boldsymbol{g}$ a body acceleration vector, $\boldsymbol{\sigma}$ the total stress tensor, and $D[.]/Dt$ the derivative following the motion, which is defined as

$$\frac{D[\cdot]}{Dt} = \frac{\partial[\cdot]}{\partial t} + (\mathbf{u}\cdot\boldsymbol{\nabla})[\cdot]. \tag{3}$$

The stress tensor can be written as

$$\boldsymbol{\sigma} = -p\mathbf{1} + 2\eta\mathbf{D}, \tag{4}$$

for a Newtonian fluid, where p is the hydrodynamic pressure, $\mathbf{1}$ the unit tensor, $\mathbf{D}$ the strain rate tensor $\mathbf{D} = \frac{1}{2}[\boldsymbol{\nabla}\mathbf{u} + (\boldsymbol{\nabla}\mathbf{u})^T]$ and η the dynamic viscosity. In non-boundary-fitted-grid methods, it is allowed to use body force terms as a way to satisfy boundary conditions (i.e., as a way to model the presence of internal boundaries) [65]. Substitution of (4) into (2) yields the Navier–Stokes equation for isothermal fluid flows, and in two dimensions, its dimensionless form can be shown as

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0, \tag{5}$$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\frac{\partial p}{\partial x} + \frac{1}{Re}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) + f_x, \tag{6}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = -\frac{\partial p}{\partial y} + \frac{1}{Re}\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) + f_y, \tag{7}$$

where u and v are the x and y components of velocity vector $\mathbf{u}$, Re is the Reynolds number defined as $Re = U^{\check{}}/\nu$ in which U is a characteristic velocity, L a characteristic length and $\nu = \eta/\rho$ the kinematic viscosity, and $\boldsymbol{f} = (f_x, f_y)$ the body force vector involving the body accelerations $\boldsymbol{g}$ and fluid density ρ. The reader is referred to [20] for further details.

Since it is a lack of a transport equation for the pressure in (5)–(7), the velocity equations (6)–(7) need to be solved iteratively towards the satisfaction of the continuity condition (5). There are several treatments reported, including the semi-implicit scheme of pressure-linked equations (SIMPLE) (e.g., [66]), the projection method (e.g., [67]) and the implicit pressure with splitting of operators (PISO) (e.g., [68]).

For non-isothermal flows, with the Boussinesq approximation [69], their governing equations in two dimensions can be written as

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0, \tag{8}$$

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} = -\frac{\partial p}{\partial x} + \sqrt{\frac{Pr}{Ra}}\left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) + f_x, \tag{9}$$

$$\frac{\partial v}{\partial t} + u\frac{\partial v}{\partial x} + v\frac{\partial v}{\partial y} = -\frac{\partial p}{\partial y} + \sqrt{\frac{Pr}{Ra}}\left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2}\right) + T + f_y, \tag{10}$$

$$\frac{\partial T}{\partial t} + u\frac{\partial T}{\partial x} + v\frac{\partial T}{\partial y} = \frac{1}{\sqrt{RaPr}}\left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}\right), \tag{11}$$

where T is the temperature, and Ra and Pr the Rayleigh and Prandtl numbers defined, respectively, as $Ra = \kappa g\Delta T L^3/\alpha\nu$ and $Pr = \nu/\alpha$ in which κ is the thermal expansion coefficient, α the thermal diffusivity coefficient, g the gravity and ΔT and L the characteristic temperature difference and length, respectively. Here, the velocity scaling $U = \sqrt{gL\beta\Delta T}$ is used to balance the inertial and buoyancy forces. Both isothermal flows and non-isothermal flows are to be considered in our numerical verification.

3. Integrated RBFs

A system of RBFs enables a smooth function defined in a low-dimensional space (e.g., 1D, 2D, and 3D) to be represented in a higher-dimensional RBF space as follows

$$y(\mathbf{x}) \approx y_a(\mathbf{x}) = \sum_{i=1}^{m} w_i g_i(\mathbf{x}), \tag{12}$$

where y is the function, y_a the approximate function, $\mathbf{x}$ the position vector, m the number of RBFs, $\left(g_1(\mathbf{x}), g_2(\mathbf{x}), \ldots, g_m(\mathbf{x})\right)$ a set of RBFs, and $\left(w_1, w_2, \ldots, w_m\right)$ a set of weights to be determined. Function y is now expressed as a linear combination of m independent RBFs. In the RBF space, y is considered as a function in m dimensions [35]. Table 1 presents some common types of RBFs. It is noted that $\mathbf{c}_i$ and a_i are the centre and width of the ith RBF, respectively.

Table 1
Some common types of RBFs, where $\mathbf{c}_i$ and a_i are the centre and width of the ith RBF, respectively.

Multiquadric function	$g_i(\mathbf{x}) = \sqrt{\parallel \mathbf{x}-\mathbf{c}_i \parallel^2 + a_i^2}$
Inverse multiquadric function	$g_i(\mathbf{x}) = 1/\sqrt{\parallel \mathbf{x}-\mathbf{c}_i \parallel^2 + a_i^2}$
Gaussian function	$g_i(\mathbf{x}) = \exp\left(\frac{\parallel \mathbf{x}-\mathbf{c}_i \parallel^2}{a_i^2}\right)$

For several RBFs including those in Table 1, the interpolation matrices derived from (12) on a set of discrete data points are shown to be invertible (Micchelli's theorem [70]). Moreover, following Cover's theorem [35], the larger the number of RBFs used (i.e. higher dimensions of the hidden space), the more accurate the approximation will be, showing the characteristic of "mesh convergence" of RBFs.

In the integral approach [43], the highest order derivatives are first approximated by RBFs through (12), and lower-order derivatives and the function itself are then acquired by integrating (12). For example, considering a univariate function $\xi(x)$, the integral approach can be mathematically expressed as

$$\frac{d^\gamma \xi(x)}{dx^\gamma} = \sum_{i=1}^{m} w_i g_i(x) = \sum_{i=1}^{m} w_i \varphi_{[\gamma]i}(x), \tag{13}$$

$$\frac{d^{\gamma-1} \xi(x)}{dx^{\gamma-1}} = \sum_{i=1}^{m} w_i \varphi_{[\gamma-1]i}(x) + c_1, \tag{14}$$

$$\frac{d^{\gamma-2} \xi(x)}{dx^{\gamma-2}} = \sum_{i=1}^{m} w_i \varphi_{[\gamma-2]i}(x) + c_1 x + c_2, \tag{15}$$

$$\ldots \quad \ldots \quad \ldots \quad \ldots \quad \ldots$$

$$\frac{d\xi(x)}{dx} = \sum_{i=1}^{m} w_i \varphi_{[1]i}(x) + c_1\frac{x^{\gamma-2}}{(\gamma-2)!} + c_2\frac{x^{\gamma-3}}{(\gamma-3)!} + \cdots + c_{\gamma-2}x + c_{\gamma-1}, \tag{16}$$

$$\xi(x) = \sum_{i=1}^{m} w_i \varphi_{[0]i}(x) + c_1\frac{x^{\gamma-1}}{(\gamma-1)!} + c_2\frac{x^{\gamma-2}}{(\gamma-2)!} + \cdots + c_{\gamma-1}x + c_\gamma, \tag{17}$$

where γ is the highest-order of derivative terms in a PDE, $\varphi_{[\gamma-1]i}(x) = \int \varphi_{[\gamma]i}(x)dx$, $\varphi_{[\gamma-2]i}(x) = \int \varphi_{[\gamma-1]i}(x)dx$, $\cdots$, $\varphi_{[0]i}(x) = \int \varphi_{[1]i}(x)dx$, and $\{c_1, c_2, \ldots, c_\gamma\}$ the constants of integration. With the presence of integration constants, there are more unknown coefficients in the interpolation system and some extra equations can thus be added. This provides an effective way of constructing compact approximations.

4. Proposed IRBF-NBFG technique

Consider a rectangular domain containing holes such as the one shown in Fig. 2. The real domain $\mathbb{D}_R$ is extended to a regular domain $\mathbb{D}$ which is used for numerical simulation. The computational domain $\mathbb{D}$ thus comprises two sets of sub-regions: the holes and the multiply-connected domain (i.e. original domain). A Cartesian grid is employed to discretise the extended/computational domain, and compact IRBF stencils are then applied for approximating the field variables. It is straightforward to implement the outer boundary conditions. Nevertheless, appropriate schemes are needed to enforce the inner boundary conditions as, generally, grid nodes do not lie on the inner boundaries. Here, the momentum Eq. (2) is semi-discretised with respect to time by a first-order finite-difference scheme

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} + \boldsymbol{\nabla}p^n = -\boldsymbol{\nabla}\cdot(\mathbf{u}^n\cdot\mathbf{u}^n) + \frac{1}{Re}\boldsymbol{\nabla}^2\mathbf{u}^n + \boldsymbol{\Lambda}, \tag{18}$$

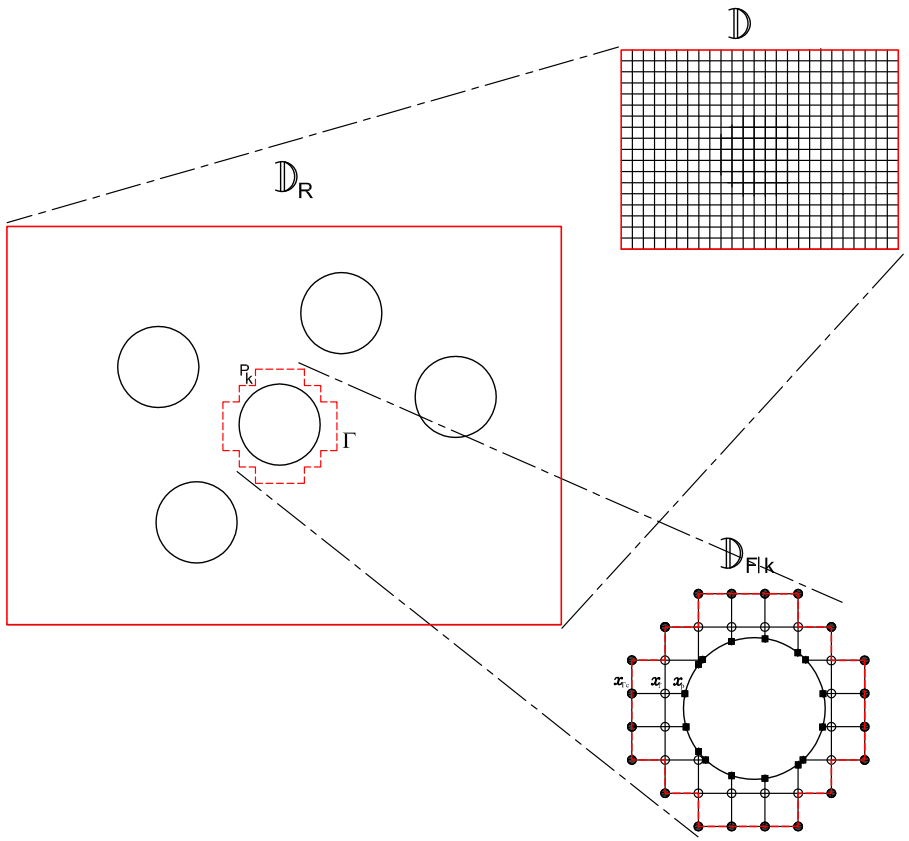

Fig. 2. A physical domain ($\mathbb{D}_R$), its extended domain ($\mathbb{D}$) and a forcing domain of kth hole ($\mathbb{D}_{F|k}$). The forcing domain is discretised using an irregular grid, where open circles are used to mark the forcing points of the kth hole, filled squares to indicate the inner boundary ∂P_k, and filled circles to indicate the boundary points of the frame $\Gamma_{|k}$.

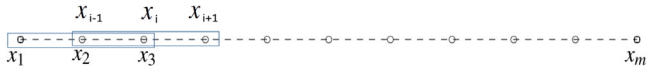

Fig. 3. Points on a grid line of the extended domain $\mathbb{D}$.

where the superscript denotes the time level and $\boldsymbol{\Lambda}$ is used to denote the body force $\mathbf{f}$ in discrete form. The body force $\boldsymbol{\Lambda}$ is included to enforce the flow field to satisfy non-slip conditions at the inner boundaries. The computation of $\boldsymbol{\Lambda}$ and the imposition of non-slip boundary conditions on the internal interfaces will be presented in Section 4.2. In the remaining parts, we will use the notation

- $\widetilde{[\,]}$ to represent a vector/matrix $[\,]$ which is associated with the whole computational domain $\mathbb{D}$,
- $\widehat{[\,]}$ to represent a vector/matrix $[\,]$ which is associated with a grid line of $\mathbb{D}$,
- $\overline{\overline{[\,]}}_{|k}$ to represent a vector/matrix $[\,]$ which is associated with the forcing domain $\mathbb{D}_{\mathbb{F}}$ of the kth hole,
- $\overline{[\,]}_{|k}$ to represent a vector/matrix $[\,]$ which is associated with a set of forcing points in a segment of $\mathbb{D}_{\mathbb{F}}$ of the kth hole,
- $([\,])_{if|k}$ to denote the selected indexes (of the extended computational domain) which are associated with the set of forcing points of the kth hole.

4.1. Compact IRBF stencils

Eqs. (6)–(7) involve the first- and second-order derivative terms. Consider an x-grid line (Fig. 3). Making use of (13)–(17) with the integer γ set to 2, $\partial^2 u/\partial x^2$ can be expressed as

$$\frac{\partial^2 u(x)}{\partial x^2} = \sum_{i=1}^{m} w_i g_i(x) = \sum_{i=1}^{m} w_i \varphi_{[2]i}(x). \tag{19}$$

Expressions for the first-order derivative and the function (field variable) are then found through integration

$$\frac{\partial u(x)}{\partial x} = \sum_{i=1}^{m} w_i \varphi_{[1]i(x)} + c_1, \tag{20}$$

$$u(x) = \sum_{i=1}^{m} w_i \varphi_{[0]i}(x) + c_1 x + c_2. \tag{21}$$

We implement the multiquadric (MQ) function and thus have

$$\varphi_{[2]i}(x) = \sqrt{(x-c_i)^2 + a_i^2},$$

$$\varphi_{[1]i}(x) = \frac{(x-c_i)}{2} A + \frac{a_i^2}{2} B,$$

$$\varphi_{[0]i}(x) = \left(\frac{-a_i^2}{3} + \frac{(x-c_i)^2}{6} \right) A + \frac{a_i^2 (x-c_i)}{2} B,$$

where c_i and a_i are the centre and the width of the ith MQ, respectively, and A and B are functions of x defined as $A = \sqrt{(x-c_i)^2 + a_i^2}$ and $B = \ln\left((x-c_i) + \sqrt{(x-c_i)^2 + a_i^2}\right)$. The width is chosen according to $a_i = \beta d_i$, where d_i is the shortest distance between c_i and its neighbours, and β is a scalar. For local approximations, large values of β can be employed. The reader is referred to [54] for a detailed study of the effect of β on the solution accuracy. A set of collocation points is chosen to coincide with the RBF centres. The influence domain used is a three-node stencil $[x_{i-1}, x_i, x_{i+1}]$ which is shifted along the grid line. The presence of two integration constants enables the addition of two extra equations to the conversion of the RBF coefficients into the physical space. We utilise these equations to impose derivatives at the two end-nodes. With compact approximations, we evaluate (21) at x_{i-1}, x_i and

x_{i+1}, and (19) at x_{i-1} and x_{i+1}, resulting in

$$\begin{pmatrix} u(x_{i-1}) \\ u(x_i) \\ u(x_{i+1}) \\ \frac{\partial^2 u(x_{i-1})}{\partial x^2} \\ \frac{\partial^2 u(x_{i+1})}{\partial x^2} \end{pmatrix} = \underbrace{\begin{bmatrix} \mathcal{I} \\ \mathcal{B} \end{bmatrix}}_{C} \begin{pmatrix} w_{i-1} \\ w_i \\ w_{i+1} \\ c_1 \\ c_2 \end{pmatrix}, \tag{22}$$

where the two nodal derivatives on LHS are known values, obtained from the previous time level,

$$\mathcal{I} = \begin{bmatrix} \varphi_{[0]i-1}(x_{i-1}) & \varphi_{[0]i}(x_{i-1}) & \varphi_{[0]i+1}(x_{i-1}) & x_{i-1} & 1 \\ \varphi_{[0]i-1}(x_i) & \varphi_{[0]i}(x_i) & \varphi_{[0]i+1}(x_i) & x_i & 1 \\ \varphi_{[0]i-1}(x_{i+1}) & \varphi_{[0]i}(x_{i+1}) & \varphi_{[0]i+1}(x_{i+1}) & x_{i+1} & 1 \end{bmatrix},$$

and

$$\mathcal{B} = \begin{bmatrix} \varphi_{[2]i-1}(x_{i-1}) & \varphi_{[2]i}(x_{i-1}) & \varphi_{[2]i+1}(x_{i-1}) & 0 & 0 \\ \varphi_{[2]i-1}(x_{i+1}) & \varphi_{[2]i}(x_{i+1}) & \varphi_{[2]i+1}(x_{i+1}) & 0 & 0 \end{bmatrix}.$$

This system can be solved for the IRBF weights and two integration constants

$$\begin{pmatrix} w_{i-1} \\ w_i \\ w_{i+1} \\ c_1 \\ c_2 \end{pmatrix} = C^{-1} \begin{pmatrix} u(x_{i-1}) \\ u(x_i) \\ u(x_{i+1}) \\ \frac{\partial^2 u(x_{i-1})}{\partial x^2} \\ \frac{\partial^2 u(x_{i+1})}{\partial x^2} \end{pmatrix}, \tag{23}$$

where C^{-1} is the inverse of C. Making use of (23), (20) and (19), one can obtain the first and second derivatives of u at x_i as

$$\frac{\partial u(x_i)}{\partial x} = \underbrace{\begin{bmatrix} \varphi_{[1]i-1}(x_i) & \varphi_{[1]i}(x_i) & \varphi_{[1]i+1}(x_i) & 1 & 0 \end{bmatrix} C^{-1}}_{\mathcal{D}_{1x}(x_i)} \begin{pmatrix} u(x_{i-1}) \\ u(x_i) \\ u(x_{i+1}) \\ \frac{\partial^2 u(x_{i-1})}{\partial x^2} \\ \frac{\partial^2 u(x_{i+1})}{\partial x^2} \end{pmatrix}, \tag{24}$$

and

$$\frac{\partial^2 u(x_i)}{\partial x^2} = \underbrace{\begin{bmatrix} \varphi_{[2]i-1}(x_i) & \varphi_{[2]i}(x_i) & \varphi_{[1]i+1}(x_i) & 0 & 0 \end{bmatrix} C^{-1}}_{\mathcal{D}_{2x}(x_i)} \begin{pmatrix} u(x_{i-1}) \\ u(x_i) \\ u(x_{i+1}) \\ \frac{\partial^2 u(x_{i-1})}{\partial x^2} \\ \frac{\partial^2 u(x_{i+1})}{\partial x^2} \end{pmatrix}. \tag{25}$$

Similar to finite-difference and finite-element techniques, one will assemble these IRBF approximations to construct the global matrices $\widetilde{D_{2x}}$ and $\widetilde{D_{2y}}$. This task is fairly simple since the grid adopted here is regular. Expressions for computing derivatives of u at the interior grid nodes can be written as $\widetilde{\frac{\partial u}{\partial x}} = \widetilde{D}_{1x}\widetilde{u} + \widetilde{k}_{1x}$ and $\widetilde{\frac{\partial^2 u}{\partial x^2}} = \widetilde{D}_{2x}\widetilde{u} + \widetilde{k}_{2x}$. The two vectors $\widetilde{k}_{1x}$ and $\widetilde{k}_{2x}$ are related to the boundary conditions and the imposed second derivative values. In a similar manner, one can obtain compact IRBF discretisations for $\widetilde{\frac{\partial u}{\partial y}}$, $\widetilde{\frac{\partial^2 u}{\partial y^2}}$, $\widetilde{\frac{\partial v}{\partial x}}$, $\widetilde{\frac{\partial^2 v}{\partial x^2}}$, $\widetilde{\frac{\partial v}{\partial y}}$, $\widetilde{\frac{\partial^2 v}{\partial y^2}}$, $\widetilde{\frac{\partial p}{\partial x}}$ and $\widetilde{\frac{\partial p}{\partial y}}$.

4.2. Imposition of inner boundary conditions

To impose boundary conditions on the inner boundaries, we use some iteration steps which are similar to those in the direct forcing immersed boundary [26,65]. The inner boundary conditions are imposed by using the term $\boldsymbol{\Lambda}$ to force the solution in the extended domain $\mathbb{D}$ to match the solution in the real domain $\mathbb{D}_R$. At grid nodes that are close to an inner boundary, the forcing term is defined as

$$\boldsymbol{\Lambda} = \frac{\mathbf{u}_d - \mathbf{u}_e}{\Delta t}, \tag{26}$$

where $\mathbf{u}_d = (u_d, v_d)$ are the desired values of the velocity when the non-slip conditions at the inner boundaries are imposed, and $\mathbf{u}_e(u_e, v_e)$ the estimated values of the velocity, where the inner boundary conditions have not been taken into account

$$\mathbf{u}_e = \mathbf{u}^n - \Delta t \left(\nabla p^n + \nabla \cdot (\mathbf{u}^n \cdot \mathbf{u}^n) - \frac{1}{Re} \nabla^2 \mathbf{u}^n \right). \tag{27}$$

To calculate $\boldsymbol{\Lambda}$, the desired velocities $\mathbf{u}_d$ must be determined in advance. For boundary points that coincide with the grid nodes (regular boundary points), one can apply (26) directly with $\mathbf{u}_d = \mathbf{u}_b$, where $\mathbf{u}_b = (u_b, v_b)$ are given boundary values. Yet, in common, boundary points do not coincide with the grid nodes and thus $\mathbf{u}_d$ are unknowns. A new computational scheme to resolve this problem is suggested as follows.

Imagine that a virtual staircase-shaped frame Γ encloses a kth hole as shown in Fig. 2. A region lying between the kth hole and the frame Γ is considered to be a forcing domain, denoted by $\mathbb{D}_{F|k}$, which is employed to match the estimated solution (in the computational domain $\mathbb{D}$) with the real solution (in the physical domain $\mathbb{D}_R$). In Fig. 2, $\mathbf{x}_p(x_p, y_p)$ are boundary points of the kth hole ($\mathbf{x}_p \in \partial P_k$), $\mathbf{x}_{\Gamma c}(x_{\Gamma c}, y_{\Gamma c})$ boundary points of Γ, and $\mathbf{x}_f(x_f, y_f)$ interior nodes of the forcing zone. It can be seen that the forcing domain $\mathbb{D}_{F|k}$ has an irregular shape. Compact IRBF stencils can work with irregular grids and they are applied here. For $\mathbf{x}_p$, the IRBF centres lie on the real boundary of the kth hole, while for $\mathbf{x}_f$ and $\mathbf{x}_{\Gamma c}$, they are taken at the grid nodes of the Cartesian grid representing the extended domain $\mathbb{D}$.

Consider a segment in the x direction. A segment can be bounded by two faces of the frame (Fig. 4a) or by the boundary of the hole and the boundary of the frame (Fig. 4b). There are two sets of points on a segment. The first set consists of n_d interior points (regular nodes) which are forcing points. The desired values u_d at the forcing points ($\overline{x_f} = \{x_{f(i)}\}_{i=1}^{n_d}$) are unknown. The second set is constituted by the two boundary nodes x_{b1} and x_{b2}. Depending on how a segment is bounded, the boundary points x_{b1} and x_{b2} have specific locations. For example, one has ($x_{b1} \equiv x_{\Gamma a}$) and ($x_{b2} \equiv x_{\Gamma b}$) if a grid line is bounded by the two sides of the frame ($x_{b1} \in \Gamma$ and $x_{b2} \in \Gamma$), and ($x_{b1} \equiv x_{\Gamma c}$) and ($x_{b2} \equiv x_p$) if the bounding surfaces are the left side of Γ and the kth hole ($x_{b1} \in \Gamma$ and $x_{b2} \in \partial P_k$).

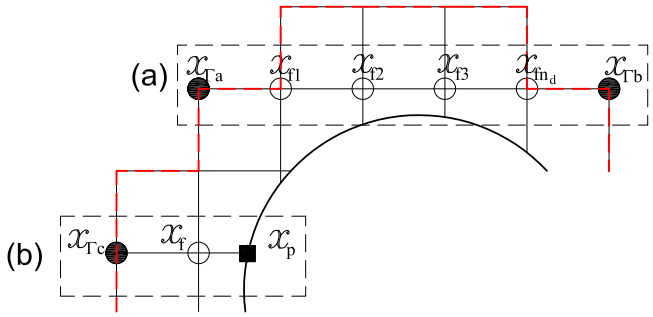

Fig. 4. Points on a segment of the forcing domain $\mathbb{D}_F$ of a hole.

In Fig. 4, for the segment (a), one can directly apply the IRBF approximations (24) and (25). For the segment (b), there are one interior grid node ($n_d = 1$) and one boundary point that is not located on the global grid nodes. In this case, we employ the following IRBF approximations.

Evaluation of (21) at nodal points $x_{\Gamma c}, x_f$ and x_p results in

$$\begin{bmatrix} u(x_{\Gamma c}) \\ u(x_f) \\ u(x_p) \end{bmatrix} = \overline{\varphi_{[0]}}_{|k} \begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ c_1 \\ c_2 \end{bmatrix}, \tag{28}$$

where

$$\overline{\varphi_{[0]}}_{|k} = \begin{bmatrix} \varphi_{[0]1}(x_\Gamma) & \varphi_{[0]2}(x_\Gamma) & \varphi_{[0]3}(x_\Gamma) & x_\Gamma & 1 \\ \varphi_{[0]1}(x_f) & \varphi_{[0]2}(x_f) & \varphi_{[0]3}(x_f) & x_f & 1 \\ \varphi_{[0]1}(x_p) & \varphi_{[0]2}(x_p) & \varphi_{[0]3}(x_p) & x_p & 1 \end{bmatrix}.$$

It should be emphasised that the function values at two boundary nodes are known. One has $u(x_{\Gamma c}) = u_e(x_{\Gamma c})$ using Eq. (27) and $u(x_p) = u_b$ using the given boundary condition.

The coefficients in (28) can then be computed in the form

$$\begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ c_1 \\ c_2 \end{bmatrix} = \left(\overline{\varphi_{[0]}}\right)^{-1}_{|k} \begin{bmatrix} u(x_{\Gamma c}) \\ u(x_f) \\ u(x_p) \end{bmatrix}, \tag{29}$$

where $\left(\overline{\varphi_{[0]}}\right)^{-1}_{|k}$ is the inverse of $\overline{\varphi_{[0]}}_{|k}$.

Taking (29) into account, the values of the second derivative of u in (19) at the forcing point x_f are given by

$$\frac{\partial^2 u(x_f)}{\partial x^2} = \underbrace{\begin{bmatrix} \varphi_{[2]1}(x_f) & \varphi_{[2]2}(x_f) & \varphi_{[2]3}(x_f) & 0 & 0 \end{bmatrix} \left(\overline{\varphi_{[0]}}\right)^{-1}_{|k}}_{\overline{D}_{2x|k} = \begin{bmatrix} D_1 & D_2 & D_3 \end{bmatrix}} \begin{bmatrix} u(x_{\Gamma c}) \\ u(x_f) \\ u(x_p) \end{bmatrix}. \tag{30}$$

or

$$\frac{\partial^2 u(x_f)}{\partial x^2} = D_2 u(x_f) + \underbrace{D_1 u(x_{\Gamma c}) + D_3 u(x_p)}_{\overline{d}_{2x|k}}, \tag{31}$$

where the last two terms, denoted by $\overline{d}_{2x|k}$, are known as $u(x_{\Gamma c})$, and $u(x_p)$ are given.

The IRBF approximations for the derivatives now take into account the boundary conditions u_b and they are expressed in terms of nodal variable values $u(x_f)$. Assembling the obtained matrices on each segment for the whole forcing domain $\mathbb{D}_{F|k}$ leads to the following form for the kth hole.

$$\overline{\overline{\frac{\partial^2 u_d}{\partial x^2}}}_{|k} = \overline{\overline{D}}_{2x|k} \overline{\overline{u}}_{d|k} + \overline{\overline{d}}_{2x|k}, \tag{32}$$

where $\overline{\overline{\frac{\partial^2 u_d}{\partial x^2}}}_{|k} = \frac{\partial^2 u(x_{(if|k)})}{\partial x^2}$.

The desired solution $\overline{\overline{u}}_{d|k}$ and $\overline{\overline{v}}_{d|k}$ around the kth hole is then determined by

$$\overline{\overline{u}}_{d|k} = \left(\overline{\overline{\mathcal{A}}}_{|k}\right)^{-1} \left(\left(\widetilde{u}^n\right)_{if|k} - \Delta t \left(\widetilde{u}^n \frac{\widetilde{\partial u}}{\partial x}^n + \widetilde{v}^n \frac{\widetilde{\partial u}}{\partial y}^n \right)_{if|k} - \Delta t \left(\frac{\widetilde{\partial p}}{\partial x}^n \right)_{if|k} - \frac{\Delta t}{Re} \left(\overline{\overline{d}}_{2x|k} + \overline{\overline{d}}_{2y|k} \right) \right), \tag{33}$$

and

$$\overline{\overline{v}}_{d|k} = \left(\overline{\overline{\mathcal{A}}}_{|k}\right)^{-1} \left(\left(\widetilde{v}^n\right)_{if|k} - \Delta t \left(\widetilde{u}^n \frac{\widetilde{\partial v}}{\partial x}^n + \widetilde{v}^n \frac{\widetilde{\partial v}}{\partial y}^n \right)_{if|k} - \Delta t \left(\frac{\widetilde{\partial p}}{\partial y}^n \right)_{if|k} - \frac{\Delta t}{Re} \left(\overline{\overline{d}}_{2x|k} + \overline{\overline{d}}_{2y|k} \right) \right), \tag{34}$$

where $\overline{\overline{\mathcal{A}}}_{|k} = \left(\mathbf{1} - \frac{\Delta t}{Re} \left(\overline{\overline{D}}_{2x|k} + \overline{\overline{D}}_{2y|k} \right) \right)$. The desired values $\overline{\overline{u}}_{d|k}$ and $\overline{\overline{v}}_{d|k}$ are thus derived from the satisfaction of the governing equation in a forcing domain and the imposition of non-slip conditions at the inner boundaries ∂P_k. In a similar manner, one can gather the desired values $\left\{\overline{\overline{u}}_{d|k}\right\}_{k=1}^{np}$ and $\left\{\overline{\overline{v}}_{d|k}\right\}_{k=1}^{np}$ for np holes.

4.3. Solution procedure

Step 1: Estimate the fluid velocities $\widetilde{u}_e$ and $\widetilde{v}_e$ using Eq. (27), where the inner boundary conditions are omitted.

Step 2: Compute the forcing term $\boldsymbol{\Lambda}$

$$\widetilde{\Lambda_x} = \frac{\widetilde{u_d} - \widetilde{u_e}}{\Delta t}, \tag{35}$$

$$\widetilde{\Lambda_y} = \frac{\widetilde{v_d} - \widetilde{v_e}}{\Delta t}, \tag{36}$$

where $\widetilde{u_d}$ and $\widetilde{v_d}$ are obtained by assembling $\left\{\overline{\overline{u}}_{d|k}\right\}_{k=1}^{np}$ and $\left\{\overline{\overline{v}}_{d|k}\right\}_{k=1}^{np}$ in Eqs. (33) and (34). It is noted that the forcing term is zeros in $\mathbb{D} \backslash \mathbb{D}_F$.

Step 3: Calculate the velocity fields $\widetilde{u}^*$ and $\widetilde{v}^*$ by solving the momentum equations (18) with the obtained forcing term $\boldsymbol{\Lambda}$ in $\mathbb{D}$. To improve the stability, the diffusion term $\boldsymbol{\nabla}^2\mathbf{u}$ is treated implicitly. It is noted that $\widetilde{u}^*$ and $\widetilde{v}^*$ are velocity components which have not satisfied Eq. (1) yet. The pseudo pressure variable is obtained by solving the following Poisson's equation

$$\left(\frac{\partial^2 \widetilde{\phi}}{\partial x^2} + \frac{\partial^2 \widetilde{\phi}}{\partial y^2} \right) = \frac{1}{\Delta t} \left(\widetilde{\frac{\partial u}{\partial x}}^* + \widetilde{\frac{\partial v}{\partial y}}^* \right). \tag{37}$$

The velocity variables are corrected by the pseudo pressure gradient term ϕ to satisfy the incompressibility constraint

$$\widetilde{u}^{n+1} = \widetilde{u}^* - \Delta t \widetilde{\frac{\partial \phi}{\partial x}}, \tag{38}$$

$$\widetilde{v}^{n+1} = \widetilde{v}^* - \Delta t \widetilde{\frac{\partial \phi}{\partial y}}. \tag{39}$$

For problems with moving inner boundaries, the system matrices can remain unchanged during the solving process. The above solution procedure is for the Navier–Stokes equation. For parabolic PDEs (e.g., the diffusion equation), the proposed solution procedure becomes simpler as there are no pressure terms involved. Notice that, with one dependent variable, the term $\boldsymbol{\Lambda}$ in (26) has only one component. For elliptic PDEs (e.g. the Poisson equation), the elliptic equation is transformed into a parabolic PDE by adding a pseudo time derivative. For this modified form, one is interested in the solution that becomes independent with respect to time.

5. Numerical examples

For all examples studied here, compact IRBF stencils are implemented with the MQ function. The solution accuracy is estimated through the discrete relative L_2 norm

$$Ne = \frac{\sqrt{\sum_{i=1}^{n_{ip}} (u_i^{(e)} - u_i)^2}}{\sqrt{\sum_{i=1}^{n_{ip}} (u_i^{(e)})^2}}, \tag{40}$$

where n_{ip} is the number of interior points in the real domain, and $u^{(e)}$ and u are the exact and numerical solutions, respectively.

For grid refinement study, the convergence rate of the solution is measured in the form

$$Ne(h) \approx \beta h^\alpha = O(h^\alpha), \tag{41}$$

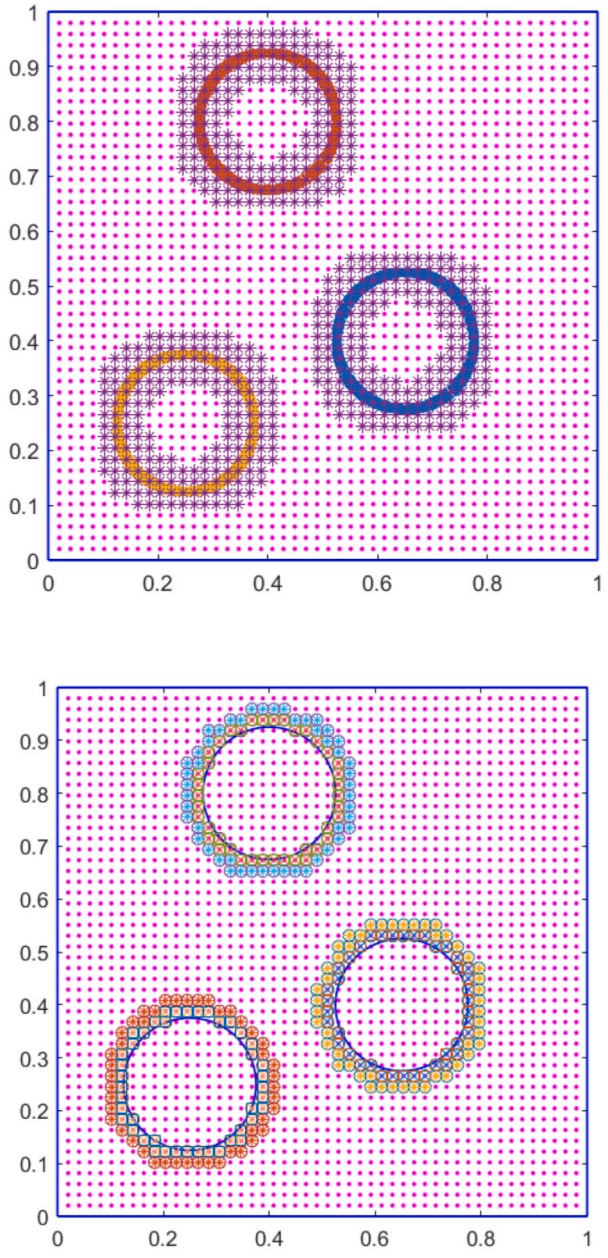

Fig. 5. Example 1 (elliptic PDE in three-hole domain): Forcing point areas used in the present IRBF-NBFG technique method (bottom) and Peskin interpolation (top).

where h is the average grid size, and β and α are exponential model's parameters that can be found by means of the linear least-squares method.

5.1. Example 1 - elliptic PDE

The proposed method is first tested with Poisson's equation

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = b(x, y), \tag{42}$$

where $b(x, y)$ is the driving function. Two domains are considered. **A domain with three holes**

Here, we are interested in a square domain, $[0, 1]\times[0, 1]$, with three circular holes of radius $R = 0.2$ and their centres located at positions (0.65,0.4), (0.4,0.8) and (0.25,0.25) as shown in Fig. 5. The exact solution of this example is

$$u^{(e)}(x, y) = \sin(2\pi x)\sin(2\pi y), \tag{43}$$

from which the driving function $b(x, y)$ and Dirichlet boundary conditions can be derived in an exact manner. The problem domain can now be embedded in a square one which is simply discretised using a uniform Cartesian grid. Both Dirac Delta functions (Fig. 5 - Top) and RBFs (Fig. 5 - Bottom) are employed here for coupling the inner boundaries. Table 2 displays the solution accuracy obtained by the IRBF-NBFG method and Dirac Delta interpolations. The proposed scheme outperforms the Dirac Delta interpolations with respect to both convergence rate and accuracy. For example, to reach the accuracy of 8.50×10^{-4}, the Dirac Delta interpolations need a grid of 90×90 while only 20×20 with the IRBF-NBFG method. The proposed scheme yields a fast rate of convergence, $O(h^{3.40})$.

Table 2
Example 1 (elliptic PDE in three-hole domain): Solution accuracy obtained by the Peskin interpolation method and the proposed IRBF-NBFG method for different grid sizes. It is noted that $a(b)$ represents $a\times10^b$.

$n_x = n_y$	Peskin interpolation	Proposed scheme
10	2.56(−1)	7.32(−3)
20	4.42(−2)	8.50(−4)
30	1.76(−2)	1.43(−4)
40	9.43(−3)	5.80(−5)
50	5.86(−3)	2.89(−5)
60	3.99(−3)	1.63(−5)
70	2.08(−3)	1.00(−5)
80	1.72(−3)	6.60(−6)
90	9.09(−4)	4.53(−6)
100	8.09(−4)	1.22(−6)

A domain with more than 3 holes

In this case, the driving function is taken to be $b(x, y) = -1$, and homogeneous Dirichlet boundary conditions are imposed. The interested domain is a unit square with 9 holes of radius 0.0625. This example provides a good means of testing the ability of the IRBF-NBFG method in dealing with problems with multi-hole domains. It is known that these geometrically-complex-domains can be found in numerous practical situations such as the thermal conductivity of composite materials, fluid flows in a porous medium and particulate suspensions. As the exact solution to this problem is unavailable, the FEM is also employed here. Fig. 6 displays discretisations used in the IRBF-NBFG technique and FEM. It can be seen that the pre-processing process of the present technique is much simpler than that of FEM. We plot a visual comparison of the contours of the solution u between the IRBF-NBFG technique (a grid of 100×100) and FEM (the MATLAB PDE Toolbox). Figs. 7–8 show the two solutions. Overall, they have comparable variations. However, the display of the solution near inner boundaries by the proposed method can be seen to be slightly different from that by the FEM. The reason for it could be that the IRBF solution is directly obtained at only the grid nodes that generally do not lie on the boundaries of the internal holes. In general, for a given number of points, grid resolutions near inner boundaries by the non-boundary-fitted-grid methods are typically lower than those by the boundary-fitted-grid methods.

5.2. Example 2 - Parabolic equation

Here, we are interested in the following parabolic PDE

$$\frac{\partial u}{\partial t} - \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2}\right) = \sin(\pi x)\sin(\pi y)\left(1000e^{-1000t} + 2k^2\pi^2\left(1 - e^{-1000t}\right)\right), \tag{44}$$

in which k is a given value. Fig. 9 shows the problem domain which is the region between a circle with radius of 1 and three smaller circles of radius 0.125.

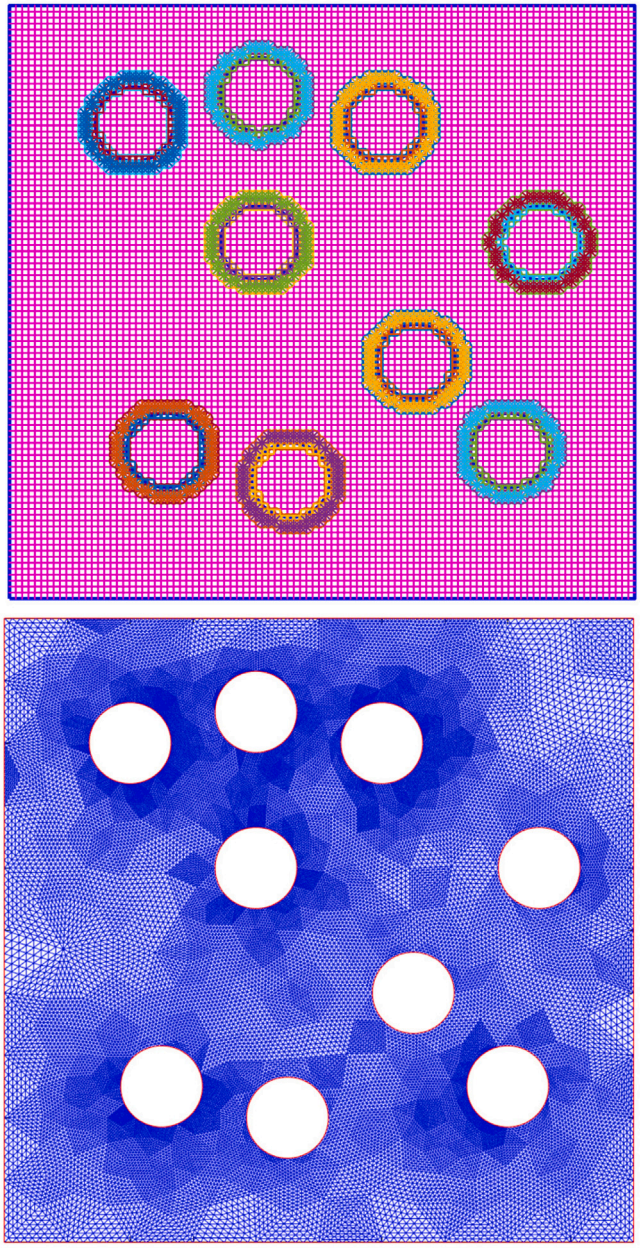

Fig. 6. Example 1 (elliptic PDE in multi-hole domain): Discretisation by the present IRBF-NBFG technique (top) and FEM (bottom).

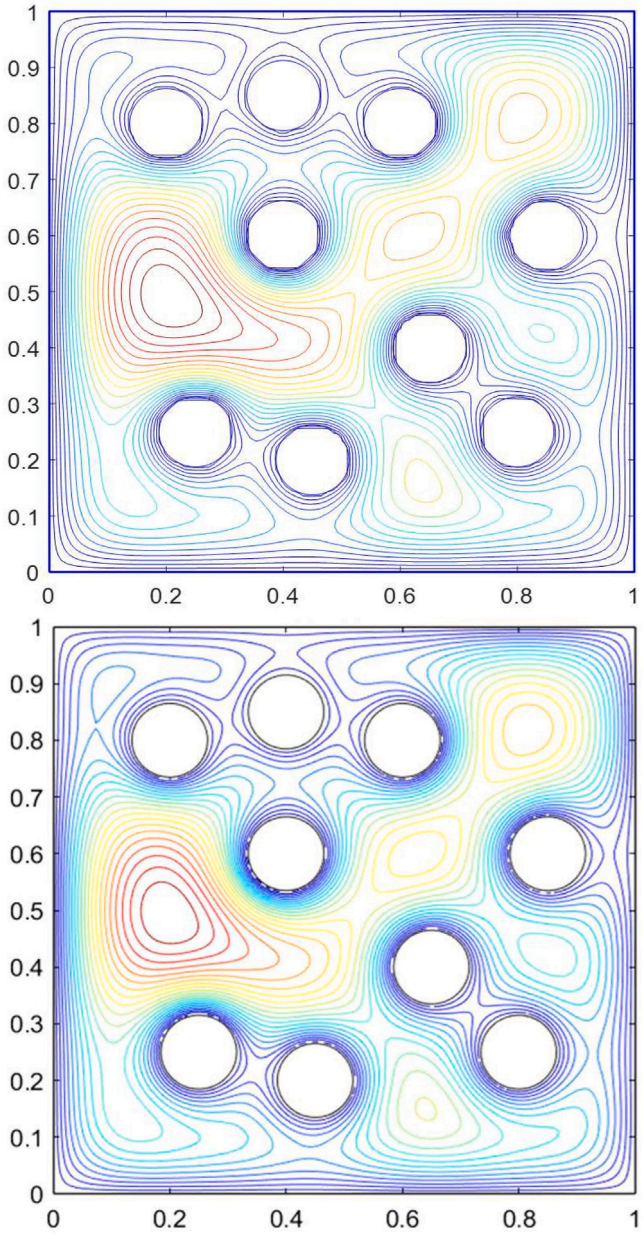

Fig. 7. Example 1 (elliptic PDE in multi-hole domain): A contour plot of u by the present IRBF-NBFG technique using a grid of 100 × 100 (top) and FEM (the MATLAB PDE Toolbox, bottom).

The initial solution, Dirichlet boundary conditions on the inner circular boundaries and Neumann boundary conditions on the external boundary can be computed exactly from the following analytical solution

$$u^{(e)}(x, y, t) = \sin(k\pi x)\sin(k\pi y)t. \tag{45}$$

Results concerning Ne for several time steps and grid densities with $k = 3$ are listed in Table 3. The condition number of the system matrix is 3.50×10^3 for a grid of 40 × 40 and 5.40×10^3 for 80 × 80. It can be seen that the proposed IRBF-NBFG technique can accurately capture the problem solution.

5.3. Example 3: Cylinder-driven flows

Case 1: Rotating cylinder

This test problem is concerned with the simulation of a steady incompressible viscous flow in an annulus between two concentric cylinders. The inner and outer cylinders are of circular and square shapes, respectively. The domain and its discretisation are shown in Figs. 10 and 11, respectively. The flow is induced by assigning a constant angular velocity Ω to the inner cylinder. The values of u and v on the outer wall are simply fixed to zero, while the value of u and v on the inner wall are set as $u = -\Omega y$ and $v = \Omega x$.

The problem domain is extended to a rectangular one which is discretised by uniform Cartesian grids. We treat the diffusion term implicitly and the convection term explicitly. Furthermore, the first-order finite-difference scheme is used to discretise the solution with respect to time. At the initial time ($t = 0$), one is required to choose initial values for all the field variables. An effective way is to use a lower-Re solution. For $Re = 100$, the initial values can be set to zero. Then, the problem solution is computed and updated till a steady-state is reached.

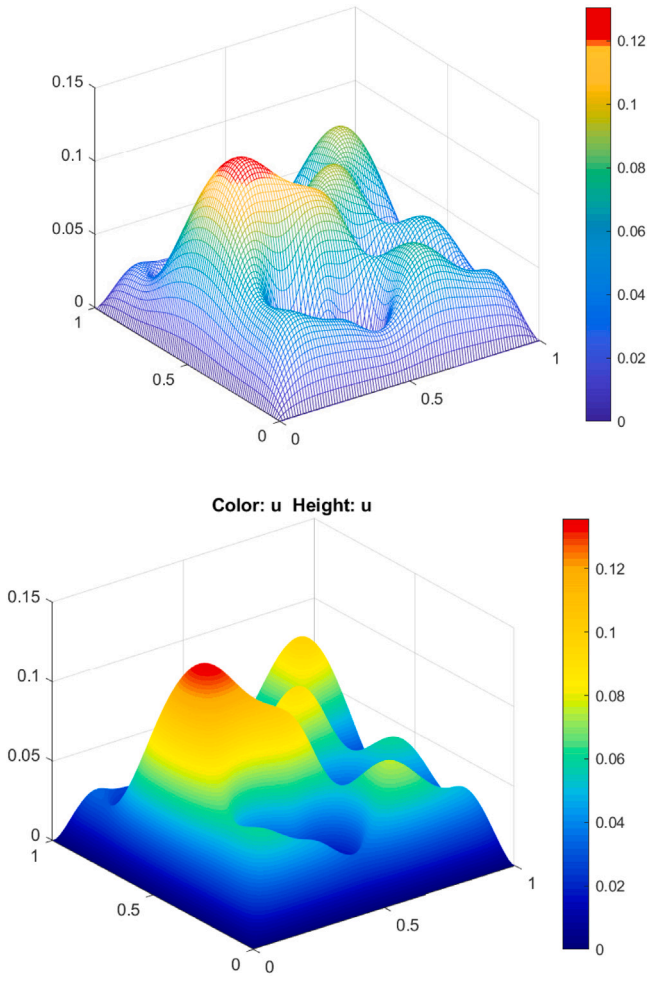

Fig. 8. Example 1 (elliptic PDE in multi-hole domain): The variation of u by the present IRBF-NBFG technique using a grid of 100 × 100 (top) and FEM (the MATLAB PDE Toolbox, bottom).

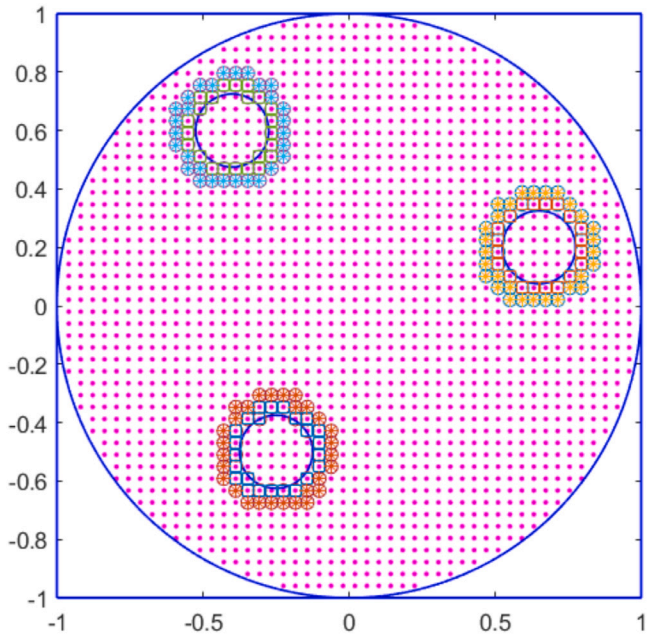

Fig. 9. Example 2 (parabolic PDE in three-hole domain): A discretisation using a grid of 50 × 50.

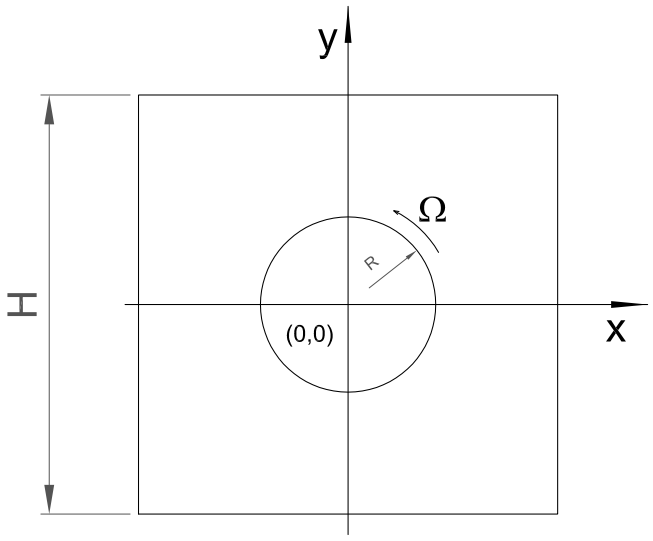

Fig. 10. Example 3 (rotating cylinder): geometry.

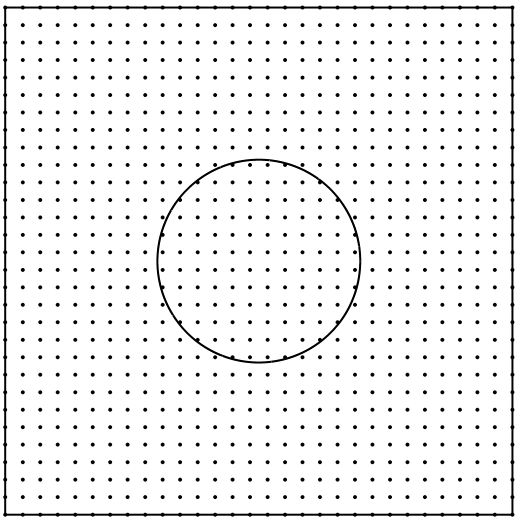

Fig. 11. Computational domain and its discretisation. It is noted that a real domain is the region between an inner circular cylinder and an outer square cylinder.

The governing equations (5)–(7) need be solved simultaneously to determine values of the two components of the velocity and pressure at grid nodes within the domain. First derivatives of the pseudo pressure on boundaries are utilised to derive Dirichlet boundary conditions for Poisson equation (37) [57]. Consequently, all the boundary conditions are of Dirichlet type. The projection method is employed for solving fluid variables. The solution procedure involves the following main steps:

a. Guessing the initial values for the field variables u, v and p.
b. Discretising equations (5)–(7) with respect to time using a finite-difference scheme.
c. Discretising equations (5)–(7) with respect to space using compact IRBF stencils. Because the interpolation matrices are identical for all the field variables, the matrix establishment process only need to be carried out once. The system matrices, which include the IRBF approximations for the first and second derivative terms in the governing equations, are kept unchanged during the computational loop.

Table 3
Example 2 (parabolic PDE in three-hole domain): Solution accuracy obtained by proposed IRBF-NBFG for different time steps and grid sizes. It is noted that $a(b)$ represents $a \times 10^b$.

t	10 × 10	20 × 20	30 × 30	40 × 40	50 × 50	60 × 60	70 × 70	80 × 80
0.01	1.84(−3)	3.16(−4)	9.99(−5)	3.90(−5)	1.92(−5)	1.77(−5)	1.03(−5)	6.01(−6)
0.11	6.35(−4)	7.34(−5)	2.14(−5)	8.88(−6)	5.67(−6)	3.39(−6)	2.00(−6)	1.25(−6)
0.21	6.28(−4)	4.72(−5)	1.36(−5)	6.73(−6)	5.93(−6)	1.98(−6)	1.00(−6)	7.12(−7)
0.31	6.55(−4)	5.19(−5)	1.46(−5)	7.29(−6)	6.36(−6)	1.90(−6)	9.09(−7)	6.84(−7)
0.41	6.73(−4)	5.33(−5)	1.51(−5)	7.52(−6)	6.57(−6)	1.94(−6)	8.89(−7)	7.00(−7)
0.51	6.74(−4)	5.44(−5)	1.53(−5)	7.58(−6)	6.59(−6)	1.89(−6)	8.68(−7)	6.83(−7)
0.61	6.78(−4)	5.43(−5)	1.54(−5)	7.62(−6)	6.63(−6)	1.91(−6)	8.69(−7)	6.92(−7)
0.71	6.76(−4)	5.46(−5)	1.54(−5)	7.62(−6)	6.62(−6)	1.88(−6)	8.64(−7)	6.84(−7)
0.81	6.78(−4)	5.45(−5)	1.54(−5)	7.63(−6)	6.63(−6)	1.89(−6)	8.63(−7)	6.89(−7)
0.91	6.77(−4)	5.46(−5)	1.54(−5)	7.62(−6)	6.63(−6)	1.88(−6)	8.64(−7)	6.86(−7)

d. Computing the desired values u_d and v_d using Eqs. (33) and (34), respectively.
e. Obtaining the forcing terms $\boldsymbol{\Lambda}$ by Eqs. (35) and (36).
f. Solving the momentum equations with the obtained forcing term $\boldsymbol{\Lambda}$ to get $\widetilde{u}^*$ and $\widetilde{v}^*$.
g. Deriving the boundary conditions for ϕ and solving the pressure Poisson's Eq. (37) for ϕ.
h. Correcting the velocity fields in Eqs. (38)–(39) to satisfy the incompressibility constraint (1).
i. Checking the steady state through the convergence measure (CM) defined as

$$CM[.] = \frac{\sqrt{\sum_{i=1}^{n_{ip}} \left([.]_i^{(l+1)} - [.]_i^{(l)}\right)^2}}{\sqrt{\sum_{i=1}^{n_{ip}} \left([.]_i^{(l+1)}\right)^2}} < \epsilon, \tag{46}$$

where [.] can be u, v, or p, n is the number of interior points in the real domain, l the time level, and ϵ the tolerance. In this study, ϵ is chosen to be 10^{-10}.

The flow is simulated with $\Omega = 1.0$ using a uniform grid of 100 × 100. Several values of the Reynolds number, including $\{100, 200, 500, 700\}$, are studied. For comparison purposes, the stream function and vorticity variables can also be derived from solving the following equation

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}. \tag{47}$$

Results concerning the maximum value of stream function (ψ_{max}) and vorticity (ω_{max}) by the IRBF-NBFG scheme and the finite-difference scheme [71] are displayed in Tables 4–5, showing a good agreement. In Fig. 12, the variation of the convergence measure CM is plotted against the total number of time steps. It can be observed that the calculation of high-Re number flows needs a higher number of steps. Plots of the velocity vector field and pressure field for the cases of $Re = \{200, 700\}$ are presented in Fig. 13.

Case 2: Moving cylinder

The second case is the same as the first one, except that the cylinder is repositioned after a certain number of time steps. The flow geometry is similar to that in Fig. 10. The cylinder's radius is 0.1 and the angular velocity Ω is set to 1. Fig. 14 displays the velocity vector fields for four y−positions of the cylinder, $y_e = \{0.15, 0.1, 0.05, -0.02\}$, for $Re = 10$.

Case 3: Multiple cylinders

This case is studied to verify the proposed technique in dealing with fluid flows in geometrically-complex domains. The rotating cylinder problem is extended to the case of many cylinders which are fixed and rotated at the same angular velocity. Here, the domain of interest is a unit square containing 9 circular cylinders of radius 0.02. The cylinders are located at (0.15,−0.1), (−0.1,0.35), (−0.25,-0.25), (−0.1,+0.1), (−0.3,0.3), (−0.05,-0.3), (0.35,+0.1), (0.1,+0.3) and (0.3,−0.25). The discretisation of the IRBF-NBFG technique is similar to that of Example 1 (Fig. 6). Since these radii are uniform, one just needs to update the location of forcing points by an amount of eccentricity (x_e, y_e). The pre-processing for this case is thus quite convenient. We present a visual distribution of u and ψ using a grid of 100 × 100 in Figs. 15 and 16, respectively.

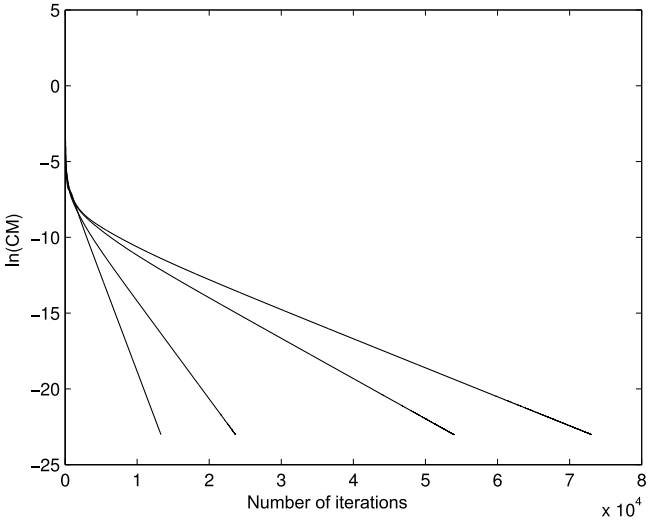

Fig. 12. Example 3 (rotating cylinders): Iterative convergence. The value of CM that is defined in (46) becomes less than 10^{-10} (or the natural logarithm of CM is less than about 23) when the number of iterations reaches 13308, 23659, 53974, and 73086 for $Re = \{100, 200, 500, 700\}$, respectively. Using the last point on the curves as a positional indicator, from left to right, the curves correspond to $Re = \{100, 200, 500, 700\}$.

5.4. Example 4: Buoyancy-driven flows in doubly-connected domains

For this example, buoyancy-driven flows between a heated internal circular cylinder and a cooled external square enclosure are studied (Fig. 17). These flows have been widely investigated by both experimental and numerical simulations. For the latter, many numerical methods were carried out such as FDM (e.g., [72,73]), FEM (e.g., [74, 75]), FVM (e.g., [76,77]), RBF (e.g., [78,79]) and spectral techniques (e.g., [80,81]).

The governing equations (8)–(11) include one Poisson equation for the pressure field and three parabolic equations for the velocity and temperature fields. Similar to Example 3, a Cartesian grid (Fig. 11) is utilised to discretise the annulus domain. Parameters for the simulation include an aspect ratio of $H/D_i = 2.5$ (D_i: the diameter of the internal hole and H: the length of the external square) and $Pr = 0.71$. For comparison purposes, the stream function can be derived by solving the following equation

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}. \tag{48}$$

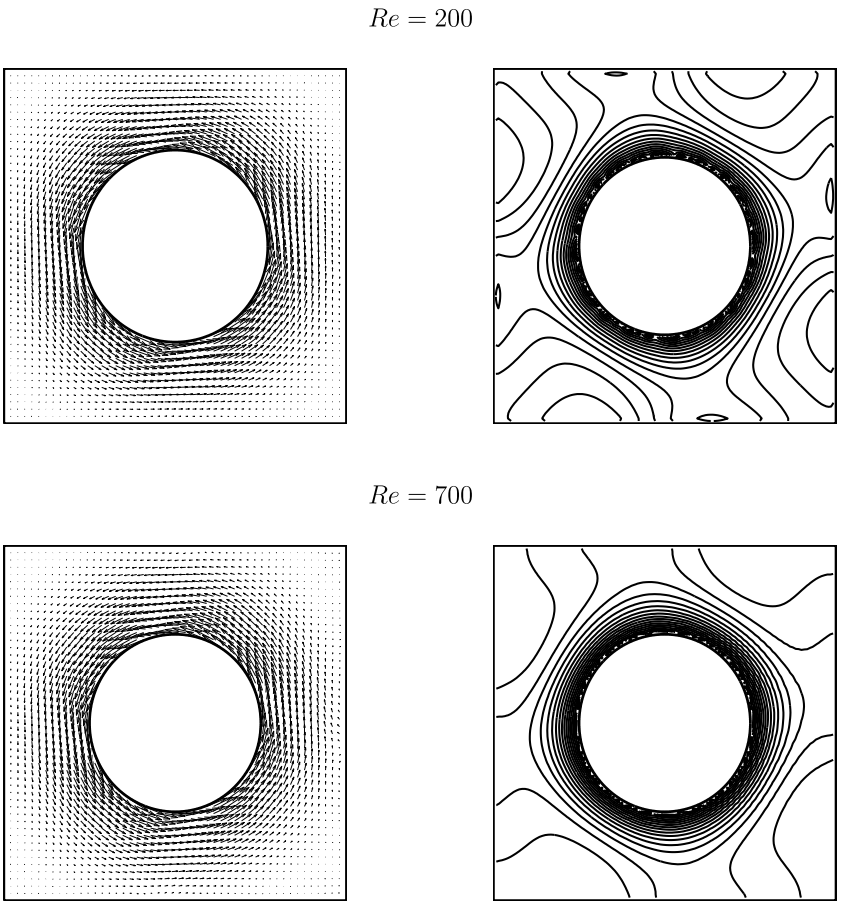

Fig. 13. Example 3 (rotating cylinder): Velocity vector field (left) and pseudo pressure field (right) for the flow at $Re = \{200, 700\}$.

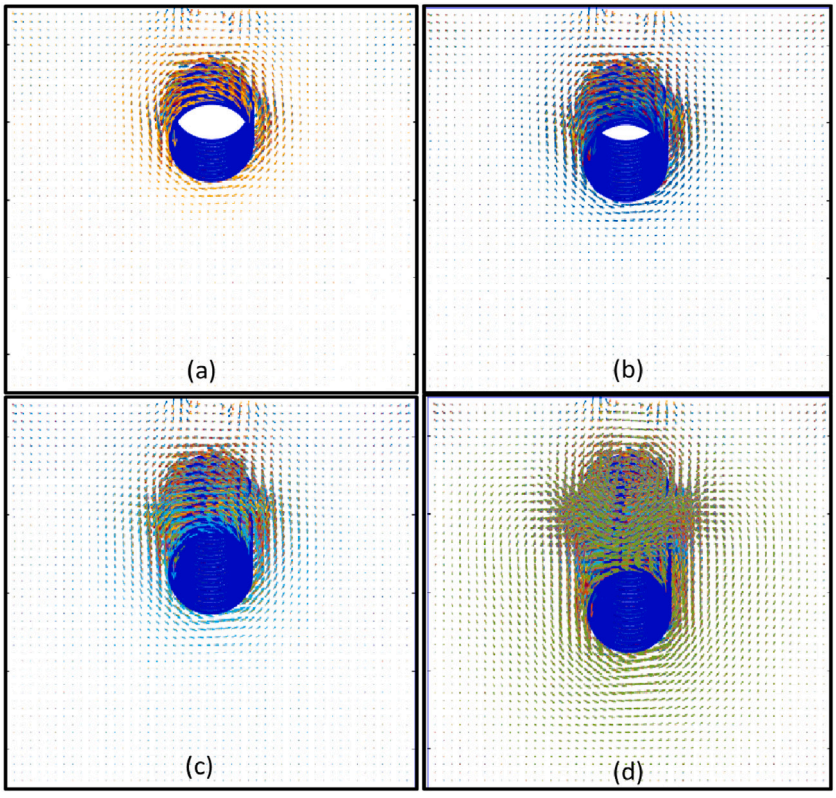

Fig. 14. Example 3 (rotating cylinder): Velocity vector field of the viscous flow for the moving cylinder with predefined velocity.

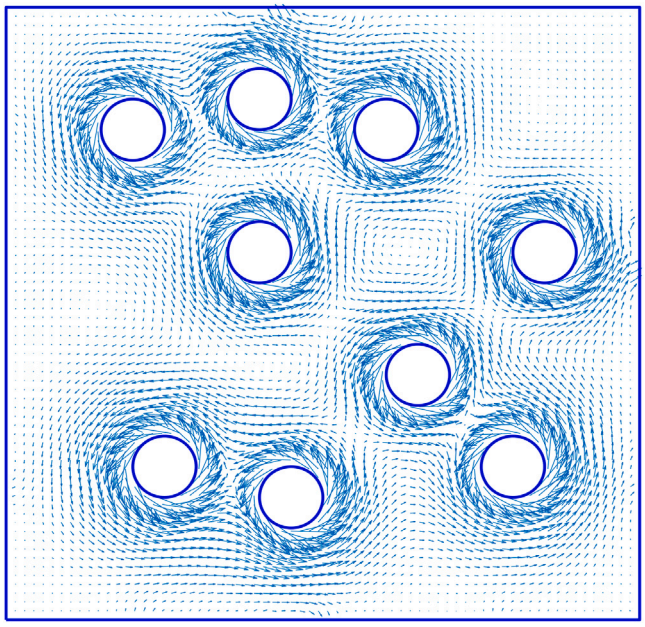

Fig. 15. Example 3 with 9 rotating cylinders: Velocity vector field.

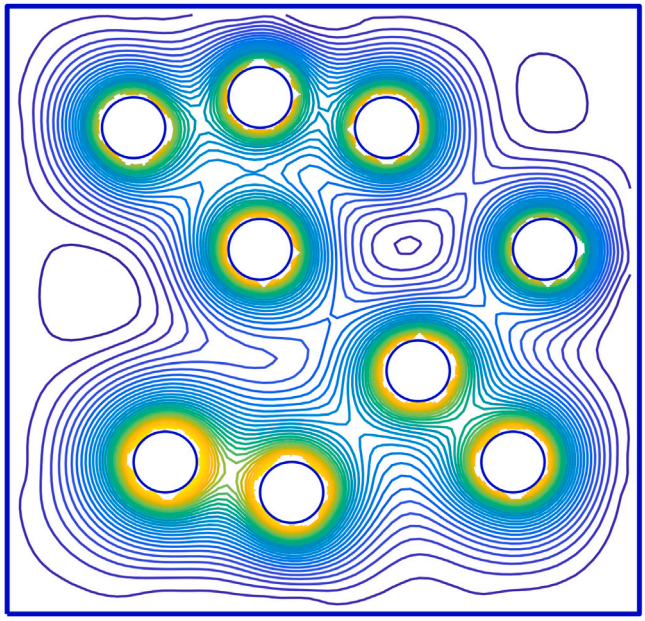

Fig. 16. Example 3 with 9 rotating cylinders: Stream-function contours.

The local heat transfer coefficient and the average Nusselt number are determined by ([77])

$$\theta = -k\frac{\partial T}{\partial n}, \tag{49}$$

where k the thermal conductivity, and

$$Nu = \frac{\overline{\theta}}{k}, \tag{50}$$

where $\overline{\theta} = -\oint \frac{\partial T}{\partial n} ds$.

Results concerning the maximum value of the stream function (ψ_{max}) and the average Nusselt number for five values of Ra, namely $\{10^4, 5\times10^4\}$ (uniform grid of 60×60) and $\{10^5, 5\times10^5, 10^6\}$ (uniform grids of 84×84), with a time step of 0.0001 are presented in Table 6 and Table 7, respectively. For $Ra = 10^4$, the initial solution is set to

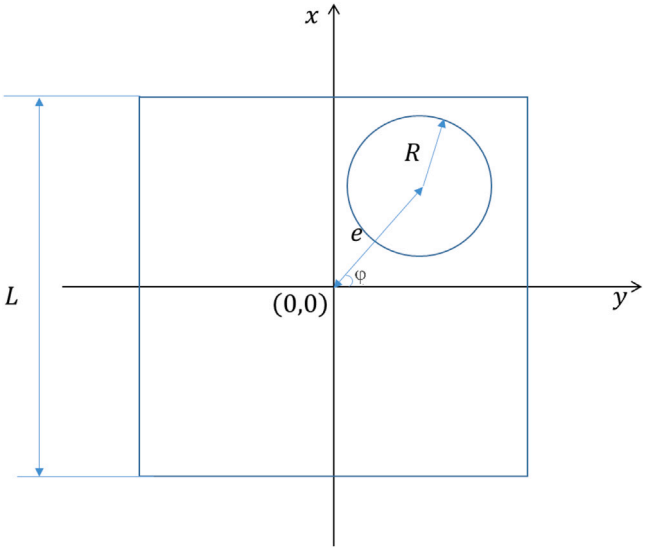

Fig. 17. Example 4 (buoyancy-driven flows in square-circular annulus): geometry.

Table 4
Example 3 (rotating cylinder): Comparison of the maximum values of the stream function ψ between the present IRBF-NBFG technique (a grid of 100×100) and finite difference technique for several values of Re.

Re	100	200	500	700
		ψ		
Present	0.4520	0.4546	0.4553	0.4550
FDM ([71])	0.4656	0.4539	0.4465	0.4423

Table 5
Example 3 (rotating cylinder): Comparison of the maximum values of the vorticity ω between the present IRBF-NBFG technique (a grid of 100×100) and finite difference technique for several values of Re.

Re	100	200	500	700
		ω		
Present	1.1154	1.2660	1.3717	1.3937
FDM ([71])	1.0186	1.2559	1.3430	1.3693

Table 6
Example 4 (buoyancy flows in square-circular annulus): Comparison of the maximum value of the stream function ψ for Ra from 10^4 to 10^6 between the present technique and some other techniques.

Ra	10^4	5×10^4	10^5	5×10^5	1×10^6
			ψ_{max}		
Present method	1.04	5.13	8.34	19.94	24.29
1D-IRBFN ([46])	1.00	5.03	8.37	20.00	24.34
MQ-DQ ([82])	1.00		8.32		24.13
FVM ([77])	1.02		8.38		24.07

Table 7
Example 4 (buoyancy flows in square-circular annulus): Comparison of the average Nusselt number, Nu, for Ra from 10^4 to 10^6 between the present technique and some other techniques.

Ra	10^4	5×10^4	10^5	5×10^5	10^6
			Nu		
Present method	3.13	4.23	5.35	7.11	9.30
1D-IRBFN ([46])	3.22	4.04	4.89	7.43	8.70
DQM [81]	3.24	4.02	4.86	7.53	8.90
FDM [72]	3.33		5.08		9.37

Table 8
Example 4 (buoyancy flows in square-circular annulus): Comparison of the maximum stream-function values, ψ_{max}, for special cases $\varphi = \{-90^0, 90^0\}$ between the present technique and MQ-DQ technique.

e	0.25	0.5	0.75	0.95
φ		-90^0		
		ψ_{max}		
Present method	17.8	20.75	22.0	22.97
MQ-DQ [82]	18.64	21.29	23.52	
φ		90^0		
		ψ_{max}		
Present method	12.7	11.06	10.90	9.57
MQ-DQ [82]	12.39	11.38	10.09	

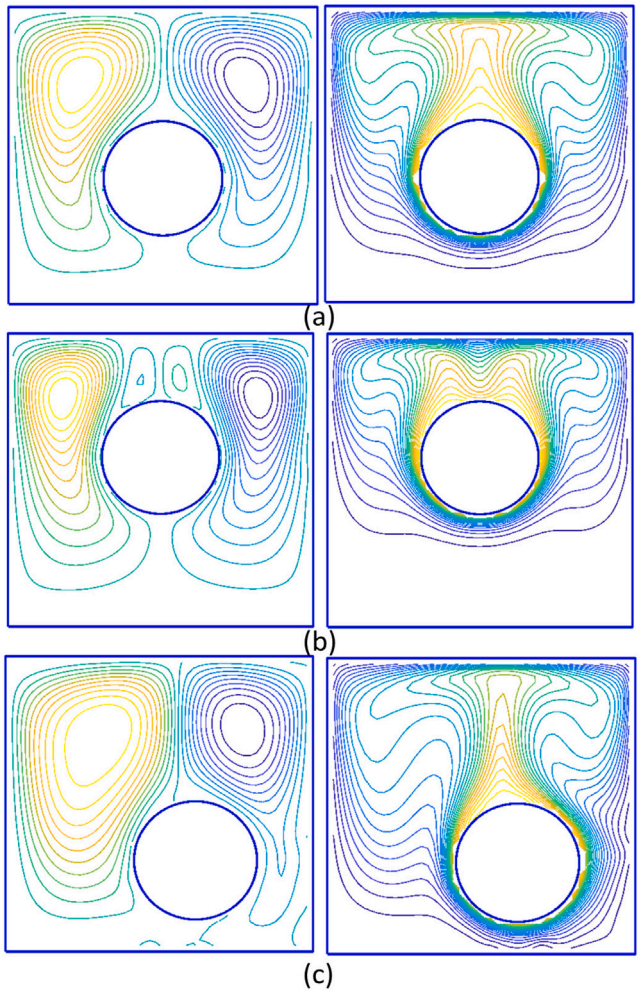

Fig. 18. Example 4 (buoyancy-driven flows): streamlines and isotherms for three different eccentric configurations.

zero and for higher values of Ra, the initial solution is taken from the solution at the most next lower Ra. These results agree well with those in [46,72,77,81,82]. We also consider the shifting circular boundary, where the centre of the internal cylinder moves inside the external square. Varying amounts of position of the cylinder centre (e), $\{0.25, 0.5, 0.75$ and $0.95\}$, are considered. Results concerning ψ_{max} together with those of [82] for $Ra = 3 \times 10^5$ are displayed in Table 8. A good agreement between the results obtained by IRBF-NBFG scheme and those of the boundary fitted grid methods can be observed. The streamlines and isotherms of the flow for $Ra = 3 \times 10^5$ using a grid of 60×60 are plotted in Fig. 18. Each plot comprises 24 contour lines which have levels varying linearly from the lowest to highest values. We also study flow fields for different sizes of the inner cylinder. Fig. 19 shows the distribution of the temperature and velocity vector fields for $H/D_i = \{5, 2.5, 1.67\}$ at $Ra = 10^6$. It can be seen that the present IRBF-NBFG results are in good agreement with those presented in [82].

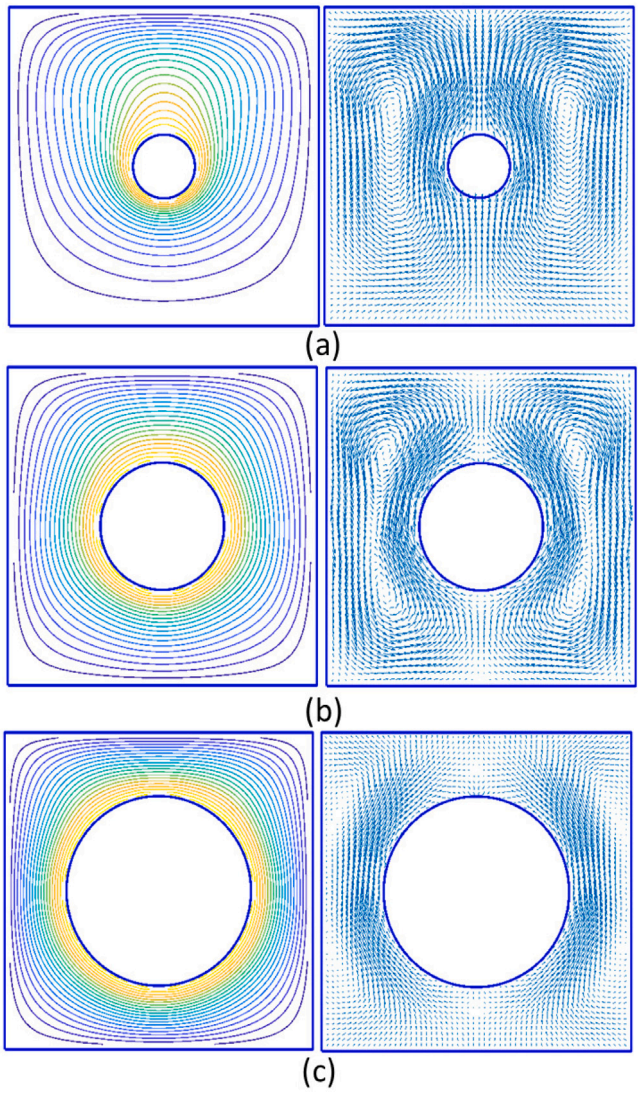

Fig. 19. Example 4 (buoyancy-driven flows): Isotherms and velocity vector fields for three different sizes of the inner cylinder.

6. Concluding remarks

In this work, a new non-boundary-fitted-grid method is reported. Compact integrated RBF stencils are utilised to discretise the field variables on the computational domain, and the forcing terms are directly estimated from the local satisfaction of the governing equations. Unlike other immersed boundary methods, no interpolation between Lagrange and Euler meshes is required here. The proposed method is successfully verified in several practical problems. Numerical results show that a high convergence rate is achieved and the matrix condition number is relatively low. These attractive features together with the advantages of using non-boundary-fitted grids allow an efficient scheme to be developed for simulating 2D and 3D flows of complex structure fluids such as particulate flows.

CRediT authorship contribution statement

T.T.V. Le: Conceptualization, Methodology, Writing – original draft, Writing – review & editing, Software, Validation. **N. Mai-Duy:** Conceptualization, Methodology, Writing – review & editing, Supervision, Validation, Resources. **K. Le-Cao:** Conceptualization, Methodology, Supervision, Software, Validation. **S. Bordas:** Supervision, Resources. **D.P. Vu:** Supervision. **T. Tran-Cong:** Conceptualization, Methodology, Supervision, Validation, Resources.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgements

This research is supported by the CESRC, School of Mechanical and Electrical Engineering, Faculty of Health, Engineering and Science, University of Southern Queensland, and by the Institute of Applied Mechanics and Informatics (IAMI), HCMC Vietnam Academy of Science and Technology (VAST).

References

[1] J. Parvizian, A. Duster, E. Rank, Finite cell method, Comput. Mech. 41 (1) (2007) 121–133.
[2] S.Z. Husain, J.M. Floryan, J. Szumbarski, Over-determined formulation of the immersed boundary conditions method, Comput. Methods Appl. Mech. Eng. 199 (1–4) (2009) 94–112.
[3] N. Mai-Duy, H. See, T. Tran-Cong, A spectral collocation technique based on integrated Chebyshev polynomials for biharmonic problems in irregular domains, Appl. Math. Model. 33 (1) (2009) 284–299.
[4] M. Buffat, L. Le Penven, A spectral fictitious domain method with internal forcing for solving elliptic PDEs, J. Comput. Phys. 230 (7) (2011) 2433–2450.
[5] D. Devendran, C.S. Peskin, An immersed boundary energy-based method for incompressible viscoelasticity, J. Comput. Phys. 231 (14) (2012) 4613–4642.
[6] A. Dechaume, W.H. Finlay, P.D. Minev, A two-grid fictitious domain method for direct simulation of flows involving non-interacting particles of a very small size, Int. J. Numer. Methods Fluids. 63 (11) (2010) 1241–1255.
[7] S. Kang, Y.K. Suh, An immersed-boundary finite-volume method for direct simulation of flows with suspended paramagnetic particles, Int. J. Numer. Methods Fluids. 67 (1) (2011) 58–73.
[8] L. Shi, T.W. Pan, R. Glowinski, Numerical simulation of lateral migration of red blood cells in Poiseuille flows, Int. J. Numer. Methods Fluids. 68 (11) (2012) 1393–1408.
[9] Y. Wang, C. Shu, L.M. Yang, Y. Sun, On the immersed boundary-lattice Boltzmann simulations of incompressible flows with freely moving objects, Int. J. Numer. Methods Fluids. 83 (4) (2017) 331–350.
[10] H. Mo, F.S. Lien, F. Zhang, D.S. Cronin, An immersed boundary method for solving compressible flow with arbitrarily irregular and moving geometry, Int. J. Numer. Methods Fluids. 88 (5) (2018) 239–263.
[11] M. Haji Mohammadi, F. Sotiropoulos, J. Brinkerhoff, Moving least squares reconstruction for sharp interface immersed boundary methods, Int. J. Numer. Methods Fluids. 90 (2) (2019) 57–80.
[12] V. Shankar, G.B. Wright, A.L. Fogelson, R.M. Kirby, A radial basis function (RBF) finite difference method for the simulation of reaction–diffusion equations on stationary platelets within the augmented forcing method, Int. J. Numer. Methods Fluids. 75 (1) (2014) 1–22.
[13] J. Ames, D.F. Puleri, P. Balogh, J. Gounley, E.W. Draeger, A. Randles, Multi-GPU immersed boundary method hemodynamics simulations, J. Comput. Sci. 44 (2020) 101–153.
[14] A. Dadvand, M. Baghalnezhad, I. Mirzaee, B.C. Khoo, S. Ghoreishi, An immersed boundary–lattice Boltzmann approach to study the dynamics of elastic membranes in viscous shear flows, J. Comput. Sci. 5 (5) (2014) 709–718.
[15] R. Glowinski, T.W. Pan, J. Periaux, Distributed Lagrange multiplier methods for incompressible viscous flow around moving rigid bodies, Comput. Methods Appl. Mech. Eng. 151 (1–2) (1998) 181–194.
[16] R. Glowinski, T.W. Pan, J. Periaux, A fictitious domain method for external incompressible viscous flow modeled by Navier–Stokes equations, Comput. Methods Appl. Mech. Eng. 112 (1–4) (1994) 133–148.
[17] N.A. Patankar, P. Singh, D.D. Joseph, R. Glowinski, T.W. Pan, A new formulation of the distributed Lagrange multiplier/fictitious domain method for particulate flows, Int. J. Multiphase Flow. 26 (9) (2000) 1509–1524.
[18] B. Coesnon, M. Heniche, C. Devals, F. Bertr, P.A. Tanguy, A fast and robust fictitious domain method for modelling viscous flows in complex mixers: The example of propellant make-down, Int. J. Numer. Methods Fluids. 58 (4) (2008) 427–449.
[19] R. Van Loon, P.D. Anderson, J. De Hart, F.P. Baaijens, A combined fictitious domain/adaptive meshing method for fluid structure interaction in heart valves, Int. J. Numer. Methods Fluids. 46 (5) (2004) 533–544.
[20] Z. Yu, A DLM/FD method for fluid/flexible-body interactions, J. Comput. Phys. 207 (1) (2005) 1–27.
[21] X. Shi, G. Lin, J. Zou, D.A. Fedosov, A lattice Boltzmann fictitious domain method for modeling red blood cell deformation and multiple-cell hydrodynamic interactions in flow, Int. J. Numer. Methods Fluids. 72 (8) (2013) 895–911.
[22] R. Glowinski, T.W. Pan, T.I. Hesla, D.D. Joseph, A distributed Lagrange multiplier/fictitious domain method for particulate flows, Int. J. Multiphase Flow. 25 (5) (1999) 75.
[23] D. Wan, S. Turek, Direct numerical simulation of particulate flow via multigrid FEM techniques and the fictitious boundary method, Int. J. Numer. Methods Fluids. 51 (5) (2006) 531–566.
[24] V.P. Nguyen, T. Rabczuk, S. Bordas, M. Duflot, Meshless methods: a review and computer implementation aspects, Math. Comput. Simul. 79 (3) (2008) 763–813.
[25] C.S. Peskin, Flow patterns around heart valves: a numerical method, J. Comput. Phys. 10 (2) (1972) 252–271.
[26] E.A. Fadlun, R. Verzicco, P. Orlandi, J. Mohd-Yusof, Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations, J. Comput. Phys. 161 (1) (2000) 35–60.
[27] M. Uhlmann, An immersed boundary method with direct forcing for the simulation of particulate flows, J. Comput. Phys. 209 (2) (2005) 448–476.
[28] J. Kim, D. Kim, H. Choi, An immersed-boundary finite-volume method for simulations of flow in complex geometries, J. Comput. Phys. 171 (1) (2001) 132–150.
[29] A. Duster, J. Parvizian, Z. Yang, E. Rank, The finite cell method for three-dimensional problems of solid mechanics, Comput. Methods Appl. Mech. Eng. 197 (45–48) (2008) 3768–3782.
[30] B. Maury, A fat boundary method for the Poisson problem in a domain with holes, J. Sci. Comput. 16 (3) (2001) 319–339.
[31] S. Bertoluzza, M. Ismail, B. Maury, The fat boundary method: semi-discrete scheme and some numerical experiments, in: Domain Decomposition Methods in Science and Engineering, Springer, Berlin, Heidelberg, 2005, pp. 513–520.
[32] P.E. Vos, R. Van Loon, S.J. Sherwin, A comparison of fictitious domain methods appropriate for spectral/hp element discretisations, Comput. Methods Appl. Mech. Eng. 197 (25–28) (2008) 2275–2289.
[33] D.B. Stein, R.D. Guy, B. Thomases, Immersed boundary smooth extension: A high-order method for solving PDE on arbitrary smooth domains using Fourier spectral methods, J. Comput. Phys. 304 (2016) 252–274.
[34] D.B. Stein, R.D. Guy, B. Thomases, Immersed boundary smooth extension (IBSE): A high-order method for solving incompressible flows in arbitrary smooth domains, J. Comput. Phys. 335 (2017) 155–178.
[35] S. Haykin, Neural Networks: A Comprehensive Foundation, Prentice Hall PTR, Upper Saddle River, NJ, USA, 1998.
[36] T. Poggio, F. Girosi, Networks for approximation and learning, Proc. IEEE 78 (9) (1990) 1481–1497.
[37] J. Park, I.W. Sandberg, Universal approximation using radial-basis-function networks, Neural Comput. 3 (2) (1991) 246–257.
[38] D.F. Trevisan, J.P. Gois, H.C. Batagelo, A low-cost-memory CUDA implementation of the conjugate gradient method applied to globally supported radial basis functions implicits, J. Comput. Sci. 5 (5) (2014) 701–708.
[39] T. Jacquemin, S. Tomar, K. Agathos, S. Mohseni-Mofidi, S.P. Bordas, Taylor-series expansion based numerical methods: A primer, performance benchmarking and new approaches for problems with non-smooth solutions, Arch. Comput. Methods Eng. 27 (5) (2020) 1465–1513.
[40] A. Obeidat, S.P. Bordas, An implicit boundary approach for viscous compressible high Reynolds flows using a hybrid remeshed particle hydrodynamics method, J. Comput. Phys. 391 (2019) 347–364.
[41] C. Introini, D. Baroli, S. Lorenzi, A. Cammi, S. Bordas, B. Peters, A mass conservative Kalman filter algorithm for thermo-computational fluid dynamics. Materials (ISSN 1996-1944).
[42] E.J. Kansa, Multiquadrics - a scattered data approximation scheme with applications to computational fluid-dynamics-II solutions to parabolic, hyperbolic and elliptic partial differential equations, Comput. Math. Appl. 19 (8–9) (1990) 147–161.
[43] N. Mai-Duy, T. Tran-Cong, Numerical solution of differential equations using multiquadric radial basis function networks, Neural Netw. 14 (2) (2001) 185–199.
[44] N. Mai-Duy, T. Tran-Cong, Approximation of function and its derivatives using radial basis function networks, Appl. Math. Model. 27 (3) (2003) 197–220.

[45] N. Mai-Duy, K. Le-Cao, T. Tran-Cong, A Cartesian grid technique based on one-dimensional integrated radial basis function networks for natural convection in concentric annuli, Int. J. Numer. Methods Fluids. 57 (12) (2008) 1709–1730.
[46] K. Le-Cao, N. Mai-Duy, T. Tran-Cong, An effective integrated-RBFN cartesian-grid discretization for the stream function-vorticity-temperature formulation in nonrectangular domains, Numer. Heat Transf. B: Fund. 55 (6) (2009) 480–502.
[47] K. Le-Cao, N. Mai-Duy, C.D. Tran, T. Tran-Cong, Numerical study of stream-function formulation governing flows in multiply-connected domains by integrated RBFs and Cartesian grids, Comput. Fluids 44 (1) (2011) 32–42.
[48] Z. Zhao, M. Li, L. He, S. Shao, L. Zhang, High-order curvilinear mesh generation technique based on an improved radius basic function approach, Int. J. Numer. Methods Fluids. 91 (2019) 97–111.
[49] D. Xiao, F. Fang, C. Pain, G. Hu, Non-intrusive reduced order modelling of the Navier–Stokes equations based on RBF interpolation, Int. J. Numer. Methods Fluids. 79 (11) (2015) 580–595.
[50] T.T.V. Le, N. Mai-Duy, K. Le-Cao, T. Tran-Cong, A time discretization scheme based on integrated radial basis functions for heat transfer and fluid flow problems, Numer. Heat Transf., B: Fund. 74 (2) (2018) 498–518.
[51] V. Bayona, N. Flyer, B. Fornberg, G.A. Barnett, On the role of polynomials in RBF-FD approximations: II. Numerical solution of elliptic PDEs, J. Comput. Phys. 332 (2017) 257–273.
[52] E. Lehto, V. Shankar, G.B. Wright, A radial basis function (RBF) compact finite difference (FD) scheme for reaction–diffusion equations on surfaces, SIAM J. Sci. Comput. 39 (5) (2017) A2129–51.
[53] N. Mai-Duy, D. Strunin, New approximations for one-dimensional 3-point and two-dimensional 5-point compact integrated RBF stencils, Eng. Anal. Bound. Elements. 125 (2021) 12–22.
[54] N. Mai-Duy, T.T.V. Le, C.M.T. Tien, D. Ngo-Cong, T. Tran-Cong, Compact approximation stencils based on integrated flat radial basis functions, Eng. Anal. Bound. Elem. 74 (2017) 79–87.
[55] N. Mai-Duy, D. Dalal, T.T.V. Le, D. Ngo-Cong, T. Tran-Cong, A symmetric integrated radial basis function method for solving differential equations, Numer. Methods Partial Differential Equations. 34 (3) (2018) 959–981.
[56] C.M.T. Tien, N. Thai-Quang, N. Mai-Duy, C.D. Tran, T. Tran-Cong, A three-point coupled compact integrated RBF scheme for second-order differential problems, Comput. Model Eng. Sci. 104 (6) (2015) 425–469.
[57] N. Thai-Quang, K. Le-Cao, N. Mai-Duy, C.D. Tran, T. Tran-Cong, A high-order compact local integrated-RBF scheme for steady-state incompressible viscous flows in the primitive variables, CMES: Comput. Model. Eng. Sci. 84 (6) (2012) 528–557.
[58] N. Thai-Quang, K. Le-Cao, N. Mai-Duy, C.D. Tran, T. Tran-Cong, A numerical scheme based on compact integrated-RBFs and Adams–Bashforth/Crank–Nicolson algorithms for diffusion and unsteady fluid flow problems, Eng. Anal. Bound. Elem. 37 (12) (2013) 1653–1667.
[59] N. Mai-Duy, T. Tran-Cong, A compact five-point stencil based on integrated RBFs for 2D second-order differential problems, J. Comput. Phys. 235 (2013) 302–321.
[60] I. Ahmad, Siraj-ul Islam, A.Q.M. Khaliq, Local RBF method for multi-dimensional partial differential equations, Comput. Math. Appl. 74 (2) (2017) 292–324.
[61] M. Dehghan, M. Abbaszadeh, The use of proper orthogonal decomposition (POD) meshless RBF-FD technique to simulate the shallow water equations, J. Comput. Phys. 351 (2017) 478–510.
[62] M. Dehghan, M. Abbaszadeh, An upwind local radial basis functions-differential quadrature (RBF-DQ) method with proper orthogonal decomposition (POD) approach for solving compressible Euler equation, Eng. Anal. Bound. Elem. 92 (2018) 244–256.
[63] H. Pourbashash, M.K. Oshagh, Local RBF-FD technique for solving the two-dimensional modified anomalous sub-diffusion equation, Appl. Math. Comput. 339 (2018) 144–152.
[64] C. Shu, H. Ding, K.S. Yeo, Local radial basis function-based differential quadrature method and its application to solve two-dimensional incompressible Navier–Stokes equations, Comput. Methods Appl. Mech. Eng. 192 (7–8) (2003) 941–954.
[65] J. Mohd-Yusof, Combined immersed boundaries/b-splines methods for simulations of flows in complex geometries, in: CTR Annual Research Briefs, NASA Ames., 1997, pp. 317–327.
[66] S.V. Patankar, D.B. Spalding, A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows, Int. J. Heat Mass Transf. 15 (1972) 1787–1806.
[67] A. Chorin, Numerical solution of the Navier–Stokes equations, Math. Comput. 22 (104) (1968) 745–762.
[68] R.I. Issa, Solution of the implicitly discretised fluid flow equations by operator-splitting, J. Comput. Phys. 62 (1) (1986) 40–65.
[69] S. Ostrach, Natural convection in enclosures, J. Heat Transf. 110 (4-B) (1988) 1175–1190.
[70] C.A. Micchelli, Interpolation of scattered data: distance matrices and conditionally positive definite functions, Constr. Approx. 2 (1) (1986) 11–22.
[71] E. Lewis, Steady flow between a rotating circular cylinder and fixed square cylinder, J. Fluid Mech. 95 (3) (1979) 497–513.
[72] T.H. Kuehn, R.J. Goldstein, An experimental and theoretical study of natural convection in the annulus between horizontal concentric cylinders, J. Fluid Mech. 74 (4) (1976) 695–719.
[73] G. De Vahl Davis, Natural convection of air in a square cavity: a bench mark numerical solution, Int. J. Numer. Methods Fluids. 3 (3) (1983) 249–264.
[74] M.T. Manzari, An explicit finite element algorithm for convection heat transfer problems, Int. J. Numer. Methods Heat Fluid Flow 9 (8) (1999) 860–877.
[75] H. Sammouda, A. Belghith, C. Surry, Finite element simulation of transient natural convection of low-Prandtl-number fluids in heated cavity, Int. J. Numer. Methods Heat Fluid Flow 9 (5) (1999) 612–624.
[76] E.K. Glakpe, C.B. Watkins Jr., J.N. Cannon, Constant heat flux solutions for natural convection between concentric and eccentric horizontal cylinders, Numer. Heat Transf. A: Appl. 10 (3) (1986) 279–295.
[77] F. Moukalled, S. Acharya, Natural convection in the annulus between concentric horizontal circular and square cylinders, J. Thermophys. Heat Transf. 10 (3) (1996) 524–531.
[78] B. Sarler, J. Perko, C.S. Chen, Radial basis function collocation method solution of natural convection in porous media, Int. J. Numer. Methods Heat Fluid Flow 14 (2) (2004) 187–212.
[79] D. Ho-Minh, N. Mai-Duy, T. Tran-Cong, A Galerkin-RBF approach for the stream function-vorticity-temperature formulation of natural convection in 2D enclosured domains, CMES: Comput. Model. Eng. Sci. 44 (3) (2009) 219–248.
[80] P. Le Quere, Accurate solutions to the square thermally driven cavity at high Rayleigh number, Comput. Fluids 20 (1) (1991) 29–41.
[81] C. Shu, Y.D. Zhu, Efficient computation of natural convection in a concentric annulus between an outer square cylinder and an inner circular cylinder, Int. J. Numer. Methods Fluids. 38 (5) (2002) 429–445.
[82] H. Ding, C. Shu, K.S. Yeo, Z.L. Lu, Simulation of natural convection in eccentric annuli between a square outer cylinder and a circular inner cylinder using local MQ-DQ method, Numer. Heat Transf. A: Appl. 47 (2005) 291–313.

Thi Thuy Van Le earned her Ph.D. Degree Award in 2020 from the University of Southern Queensland, Australia. In 2013, she obtained the Master of Applied Mathematics at the University of Technology, Vietnam. She has many years of working as a researcher in the Department of Computational Technology and Knowledge Technology - Institute of Applied Mechanics and Informatics, Viet Nam Academy of Science and Technology. Her research interests are radial basis function methods, numerical methods for solving PDEs/SDEs, mathematical modelling, stochastic simulations.

Nam Mai-Duy is a professor of Computational Engineering at the University of Southern Queensland, Australia. His recent research includes contributions in developing novel methods (meshless, radial basis function, particle-based) and their applications in simulating fluid–structure interactions, complex-structured fluids (particulate fluids, viscoelastic fluids) and multiscale calculations.

Khoa Le-Cao studied at the University of Southern Queensland (Australia) and got his Ph.D. degree in 2011. Afterwards, he did his postdoc research at the University of Southern Queensland from 2011 to 2012. Dr Le-Cao went on to work as a Research Fellow at the National University of Singapore (NUS) in 2012–2020. His current research interests focus on computational methods including Dissipative Particle Dynamics, Smoothed Particle Hydrodynamics and Radial Basis Functions methods and numerical modelling of non-Newtonian fluids (e.g. viscoelasticity, yield stress, thixotropy).

Stéphane Bordas graduated from Northwestern University in 2003 with a Ph.D. in theoretical and applied mechanics and have been a Professor of Computational Mechanics at Cardiff University's School of Engineering since 2009 and at the University of Luxembourg since 2013. He is now the Head of the Computational and Data Science Research Priority at the University of Luxembourg and the Founding Director of the Ph.D. programme in Data and Computational Sciences. He published over 300 international journal papers, 17500+ citations. His team's work led to the ISI Highly Cited Award in Computer Science (2015–2020) and Engineering (2017–2020).

Duc Phu Vu achieved his Ph.D. degree in Numerical Analysis at the University of Buffalo, New York, USA, in 2009. Before that, he was a lecturer in the Computational Engineering Department at Vietnamese-German University. His research areas are Real-life data analysis, Computational Algorithms and Modelling, Numerical methods, Optimisation. Currently, he is the Executive Research Director, Worldquant, LLC, New York.

Thanh Tran-Cong holds a Ph.D. in Computational Rheology from The University of Sydney, Australia. He was a Professor of Mechanical Engineering and Executive Director of the Computational Engineering and Science Research Centre at the University of Southern Queensland where he is now an Emeritus Professor. He has contributed to research in computational methodology with applications in rheology.