

# A compact RBF-FD based meshless method for the incompressible Navier–Stokes equations

P P Chinchapatnam<sup>1</sup>, K Djidjeli<sup>2\*</sup>, P B Nair<sup>2</sup>, and M Tan<sup>3</sup>

<sup>1</sup>Centre for Medical Image Computing, University College London, UK

<sup>2</sup>Computational Engineering and Design Group, University of Southampton, UK

<sup>3</sup>Fluid Structure Interactions Group, University of Southampton, UK

*The manuscript was received on 2 February 2009 and was accepted after revision for publication on 16 April 2009.*

DOI: 10.1243/14750902JEME151

**Abstract:** Meshless methods for solving fluid and fluid-structure problems have become a promising alternative to the finite volume and finite element methods. In this paper, a mesh-free computational method based on radial basis functions in a finite difference mode (RBF-FD) has been developed for the incompressible Navier–Stokes (NS) equations in stream function vorticity form. This compact RBF-FD formulation generates sparse coefficient matrices, and hence advancing solutions will in time be of comparatively lower cost. The spatial discretization of the incompressible NS equations is done using the RBF-FD method and the temporal discretization is achieved by explicit Euler time-stepping and the Crank–Nicholson method. A novel ghost node strategy is used to incorporate the no-slip boundary conditions. The performance of the RBF-FD scheme with the ghost node strategy is validated against a variety of benchmark problems, including a model fluid–structure interaction problem, and is found to be in a good agreement with the existing results. In addition, a higher-order RBF-FD scheme (which uses ideas from Hermite interpolation) is then proposed for solving the NS equations.

**Keywords:** meshless method, radial basis functions, finite difference, incompressible Navier–Stokes equations, fluid–structure interaction, stream function

## 1 INTRODUCTION

In recent years, there has been an upsurge in interest in the development of so-called mesh-free methods as an alternative to the mesh-based methods, due to their potential in alleviating the mesh-generation complexities arising in traditional methods such as finite difference (FD), finite volume (FV), and finite element (FE) methods [1]. One common characteristic among all the mesh-free methods is that they can construct the functional approximation or interpolation entirely from the information at a set of scattered/random nodes or points. The earliest example of meshless methods is perhaps the generalized finite difference scheme [2]. Some of the well-known meshless methods are smooth particle

hydrodynamics (SPH) method [3], diffuse element method (DEM) [4], element free Galerkin method (EFGM) [5], reproducing Kernel particle method (RKPM) [6], partition of unity method (PUM) [7], finite point method (FPM) [8], and local Petrov–Galerkin method (MLPG) [9]. One of the main advantages of mesh-free methods is that it is computationally easy to add or remove nodes from a pre-existing set of nodes, which is not the case for mesh-based methods, where the addition or removal of a point/element would lead to heavy remeshing and hence computationally be difficult to implement.

For many years, radial basis functions (RBFs) have been synonymous with scattered data approximation, especially in higher dimensions. In recent years, there has also been an increased interest in their use in solving partial differential equations (PDEs) on irregular domains by a global collocation approach (see, for example, references [10] to [13]). Despite the excellent results of earlier works related

\*Corresponding author: School of Engineering Sciences, University of Southampton, University Road, SO17 1BJ, UK.  
email: kkd@soton.ac.uk

to the use of RBFs for the numerical solution of PDEs, the traditional RBFs are globally defined functions, which result in a dense linear system. This hinders the application of the RBFs to solve large-scale fluid dynamics problems (such as the Navier–Stokes (NS) equations), as they will be computationally intensive, owing to the resulting matrices being fully populated. Also for numerical problems, which necessitate large numbers of points, these fully populated matrices tend to be ill-conditioned. Some attempts have been made to resolve this problem [14–17] (and references therein). Among the recent proposed method is the local RBF-FD method, which is found to give spectral accuracy for a sparse, better-conditioned linear system and more flexibility for handling non-linearities when used to solve elliptic problems [18]. This idea of using RBFs in a finite-difference mode (RBF-FD), which can be seen as generalized finite differences with arbitrary/random points instead of a regular grid system, was proposed by Wright and Fornberg [18], and Tolstykh and Shriobokov [19] independently in the literature.

In this paper, the local RBF-FD collocation method is extended for solving incompressible flows in a stream function vorticity form. A novel ghost centre strategy is employed to satisfy the boundary conditions. The performance of the local RBF-FD scheme and its higher-order with the ghost node strategy is validated against a variety of benchmark problems, and compared with the existing results in the literature. In addition, a numerical study of near-bed submarine pipelines under current is presented to demonstrate the applicability of the RBF-FD scheme to model fluid–structure interactions.

## 2 RBF-FD FOR INCOMPRESSIBLE NAVIER–STOKES EQUATIONS

The non-dimensional governing equations for unsteady incompressible NS equations expressed in terms of vorticity ( $\omega$ ) and stream function ( $\psi$ ) are given by

$$\frac{\partial \omega}{\partial t} + u \frac{\partial \omega}{\partial x} + v \frac{\partial \omega}{\partial y} = \frac{1}{Re} \left( \frac{\partial^2 \omega}{\partial x^2} + \frac{\partial^2 \omega}{\partial y^2} \right) \quad (1)$$

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = -\omega \quad (2)$$

where  $Re$  is the Reynolds number and  $u$  and  $v$  denote the components of velocity in the  $x$  and  $y$  directions,

which can be expressed in terms of the stream function as

$$u = \frac{\partial \psi}{\partial y}, \quad v = -\frac{\partial \psi}{\partial x} \quad (3)$$

In the RBF-FD method, the complete domain is represented by a set of scattered nodes present in the interior and on the boundary. For each interior node, a supporting region/stencil is identified by choosing  $N$  nearest nodes. Then at each node, a local RBF interpolation problem is set up to determine the RBF-FD weights for each derivative.

The standard RBF interpolation is of the form

$$u(\mathbf{x}) \approx s(\mathbf{x}) = \sum_{i=1}^n \lambda_i \phi(\|\mathbf{x} - \mathbf{x}_i\|) + \beta \quad (4)$$

where  $\phi(\|\cdot\|)$  is the multiquadric RBF and  $\beta$  is a constant.

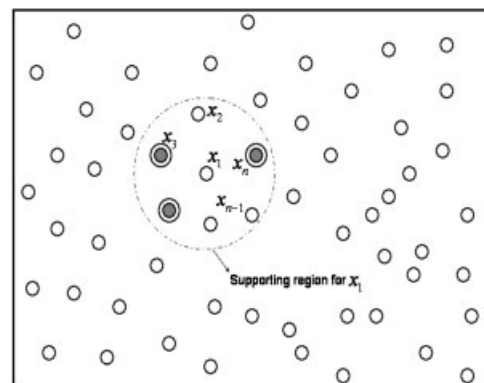
In Lagrangian form, equation (4) can be written as

$$\bar{s}(\mathbf{x}) = \sum_{i=1}^n \chi(\|\mathbf{x} - \mathbf{x}_i\|) u(\mathbf{x}_i) \quad (5)$$

where  $\chi(\|\mathbf{x} - \mathbf{x}_i\|)$  is of the form (4) and satisfies the usual cardinal conditions, i.e.

$$\chi(\|\mathbf{x}_k - \mathbf{x}_i\|) = \begin{cases} 1, & \text{if } k = i, \\ 0, & \text{if } k \neq i, \end{cases} \quad k = 1, \dots, n \quad (6)$$

To derive RBF-FD formula at the node, say  $\mathbf{x}_1$  (see Fig. 1), the differential operator is approximated using the Lagrangian form of the RBF interpolant, i.e.



**Fig. 1** Schematic diagram of a higher-order RBF-FD stencil. The circle indicates the supporting region/stencil for the node  $\mathbf{x}_1$

$$\mathcal{L}u(\mathbf{x}_1) \approx \mathcal{L}\tilde{s}(\mathbf{x}_1) = \sum_{i=1}^n \mathcal{L}\chi(\|\mathbf{x}_1 - \mathbf{x}_i\|)u(\mathbf{x}_i) \quad (7)$$

Equation (7) can be rewritten as a FD formula of the form

$$\mathcal{L}u(\mathbf{x}_1) \approx \sum_{i=1}^n w_{(1,i)} u(\mathbf{x}_i) \quad (8)$$

where the RBF-FD weights  $\{w_{(1,i)}\}_{i=1}^n$  are formally given by the operator  $\mathcal{L}$  applied on the Lagrange form of the basis functions, i.e.

$$w_{(1,i)} = \mathcal{L}\chi(\|\mathbf{x}_1 - \mathbf{x}_i\|) \quad (9)$$

In practise, the weights are computed by solving the linear system

$$\begin{bmatrix} \Phi & \mathbf{e} \\ \mathbf{e}^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \mu \end{bmatrix} = \begin{bmatrix} \mathcal{L}\Phi_1 \\ 0 \end{bmatrix} \quad (10)$$

where  $\mathcal{L}\Phi_1$  denotes the evaluation of the column vectors  $\mathcal{L}\Phi = [\mathcal{L}\phi(\|\mathbf{x} - \mathbf{x}_1\|) \mathcal{L}\phi(\|\mathbf{x} - \mathbf{x}_2\|) \dots \mathcal{L}\phi(\|\mathbf{x} - \mathbf{x}_n\|)]^T$  at the node  $\mathbf{x}_1$ . Here,  $\mu$  is a scalar value and enforces the condition

$$\sum_{i=1}^n w_{(1,i)} = 0$$

which ensures that the stencil is exact for all constants.

Once the RBF-FD method is applied to discretize the spatial derivatives in the governing equations, equations (1) to (2), at any interior node  $\mathbf{x}_i$  the following are obtained

$$\begin{aligned} \frac{d\omega_i}{dt} + u_i \sum_{j=1}^N w_{(i,j)}^{(x)} \omega_j + v_i \sum_{j=1}^N w_{(i,j)}^{(y)} \omega_j \\ = \frac{1}{Re} \left[ \sum_{j=1}^N \left( w_{(i,j)}^{(xx)} + w_{(i,j)}^{(yy)} \right) \omega_j \right] \end{aligned} \quad (11)$$

and

$$\sum_{j=1}^N \left( w_{(i,j)}^{(xx)} + w_{(i,j)}^{(yy)} \right) \psi_j = -\omega_i \quad (12)$$

where  $N$  is the total number of interior and boundary nodes which lie in the supporting region/stencil for the node  $\mathbf{x}_i$ , and  $w_{(i,j)}^{(x)}$ ,  $w_{(i,j)}^{(y)}$ ,  $w_{(i,j)}^{(xx)}$ ,  $w_{(i,j)}^{(yy)}$  are the RBF-

FD weights obtained from the system of equation (10) with the corresponding differential operator ( $\partial/\partial x$ ,  $\partial/\partial y$ ,  $\partial^2/\partial x^2$ ,  $\partial^2/\partial y^2$ ) applied to the basis functions on the right-hand side.

The system of ordinary differential equations obtained for vorticity after spatial discretization, equation (11), is advanced in time using the basic Euler time-stepping scheme. Denoting the value of any physical quantity at  $t = t^n$  with the superscript  $n$

$$\begin{aligned} \frac{\omega_i^{n+1} - \omega_i^n}{\delta t} + u_i^n \sum_{j=1}^N w_{(i,j)}^{(x)} \omega_j^n + v_i^n \sum_{j=1}^N w_{(i,j)}^{(y)} \omega_j^n \\ = \frac{1}{Re} \left[ \sum_{j=1}^N \left( w_{(i,j)}^{(xx)} + w_{(i,j)}^{(yy)} \right) \omega_j^n \right] \end{aligned} \quad (13)$$

where  $\delta t$  is the time-step. Similarly equation (11) is temporally discretized using a  $\theta$ -weighting scheme ( $0 \leq \theta \leq 1$ ), the discretised equation at the node  $\mathbf{x}_i$  reads as

$$\begin{aligned} \frac{\omega_i^{n+1} - \omega_i^n}{\delta t} = \theta \left[ \frac{1}{Re} \sum_{j=1}^N \left( w_{(i,j)}^{(xx)} + w_{(i,j)}^{(yy)} \right) \omega_j^{n+1} \right. \\ \left. - u_i^n \sum_{j=1}^N w_{(i,j)}^{(x)} \omega_j^{n+1} - v_i^n \sum_{j=1}^N w_{(i,j)}^{(y)} \omega_j^{n+1} \right] \\ + (1 - \theta) \left[ \frac{1}{Re} \sum_{j=1}^N \left( w_{(i,j)}^{(xx)} + w_{(i,j)}^{(yy)} \right) \omega_j^n \right. \\ \left. - u_i^n \sum_{j=1}^N w_{(i,j)}^{(x)} \omega_j^n - v_i^n \sum_{j=1}^N w_{(i,j)}^{(y)} \omega_j^n \right] \end{aligned} \quad (14)$$

Equations (13) and (14) need to be supplemented by the boundary condition for vorticity. The value of vorticity at the boundary is obtained by higher-order finite difference expressions [20]; see Table 1. Here, the subscript  $b$  refers to the value of the quantity on the boundary and subscript 1 refers to the interior node, which is locally orthogonal to the boundary and at a distance  $h$  from the boundary.

Once the value of vorticity in the whole domain is obtained, the governing equation for the stream function, equation (2), is solved with Dirichlet boundary conditions to update the stream function. This process is repeated until convergence

$$\frac{\|\omega_{\text{new}} - \omega_{\text{old}}\|_2}{\|\omega_{\text{new}}\|_2} \leq \varepsilon \quad (15)$$

**Table 1**  $\mathcal{O}(h^3)$  wall boundary conditions [20]

Left wall:	$\omega_b = -\frac{3}{h} \left[ u_b + \left( \frac{\psi_l - \psi_b}{h} \right) + \frac{\omega_l h}{6} \right]$
Right wall:	$\omega_b = \frac{3}{h} \left[ v_b + \left( \frac{\psi_b - \psi_l}{h} \right) - \frac{\omega_l h}{6} \right]$
Bottom wall:	$\omega_b = \frac{3}{h} \left[ u_b - \left( \frac{\psi_l - \psi_b}{h} \right) - \frac{\omega_l h}{6} \right]$
Top wall:	$\omega_b = -\frac{3}{h} \left[ u_b - \left( \frac{\psi_b - \psi_l}{h} \right) + \frac{\omega_l h}{6} \right]$

where  $\varepsilon$  is a predetermined convergence limit. The complete procedure is outlined in Table 2.

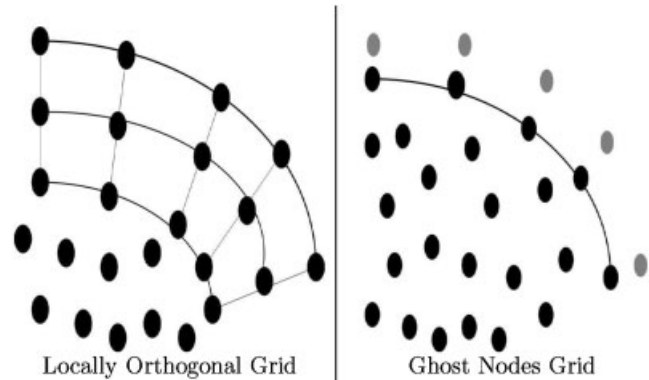
### 3 GHOST NODE STRATEGY FOR INCORPORATING BOUNDARY CONDITIONS

In the previous section, a locally orthogonal grid at the boundary was used to enforce the no-slip boundary conditions. This restriction on the nodes near the boundary makes the implementation of boundary conditions very straightforward. A considerable amount of work would be needed, however, to ensure a locally orthogonal grid near curved surfaces, and hence this approach would be cumbersome for complex geometries. In this section, a method is proposed for implementing the no-slip boundary conditions based on ghost nodes. This ghost node strategy enables randomly placed points near the boundary and is still able to satisfy the boundary conditions accurately. Sample point distributions used in the locally orthogonal grid and the ghost node strategies are shown in Fig. 2.

The no-slip boundary conditions at a boundary  $\Gamma$  are given by

$$\begin{aligned} \psi &= C_1 \quad \mathbf{x} \in \Gamma \\ \frac{\partial \psi}{\partial \vec{n}} &= C_2 \quad \mathbf{x} \in \Gamma \end{aligned} \quad (16)$$

where  $C_1$  and  $C_2$  are constants and  $\vec{n}$  is the outward normal direction from the boundary. In the proposed strategy, each boundary node is associated with a support region/stencil, which also includes a ghost node placed outside the computational do-



**Fig. 2** Schematic figure depicting the locally orthogonal boundary and the ghost nodes. Note that the ghost nodes are represented as grey shaded circles

main. The RBF-FD discretization is carried out to approximate the normal derivative at the boundary node  $\mathbf{x}_i$ , i.e.

$$\left. \frac{\partial \psi}{\partial \vec{n}} \right|_{\mathbf{x}_i} = \sum_{j=1}^N w_{(i,j)}^{(\vec{n})} \psi_j + w_{(i,\text{ghost})}^{(\vec{n})} \psi_{\text{ghost}} \quad (17)$$

where  $N$  is the number of supporting points inside and on the boundary. The value of the stream function at the ghost node is evaluated by substituting the no-slip boundary condition equation (16) and the stream function values of the interior nodes evaluated at the previous time step in equation (17). The value of vorticity on the boundaries can then be evaluated by RBF-FD discretization of equation (2) at the boundary node  $\mathbf{x}_i$ .

### 4 NUMERICAL STUDIES

This section presents numerical studies conducted on two test problems using the modified RBF-FD scheme with the ghost node strategy. Further, a numerical study of a near-bed submarine pipeline is presented to illustrate the applicability of the RBF-FD method to solve fluid–structure interaction problems.

**Table 2** RBF-FD algorithm for incompressible Navier–Stokes equations

<i>Given an initial guess <math>\psi^0</math> and <math>\omega^0</math> and a particular node configuration:</i>	
1.	For each interior node, determine the support/stencil size.
2.	Obtain the RBF-FD weights by solving the RBF interpolation problem.
3.	Advance the vorticity solution to the next step using a suitable time-stepping algorithm.
4.	Calculate the vorticity on the boundary using Table 1.
5.	Solve equation (2) with Dirichlet boundary conditions for stream function to obtain the new stream function values.
6.	Check for convergence. If converged, stop, else go to step 3.

#### 4.1 Square driven cavity flow

Numerical studies conducted on the lid-driven cavity flow problem in a square  $[0, 1] \times [0, 1]$  domain are first presented. The boundary conditions for this problem are given by

$$\begin{aligned} \psi &= 0, \quad \frac{\partial \psi}{\partial x} = 0 & \text{on } x=0 \text{ and } x=1 \\ \psi &= 0, \quad \frac{\partial \psi}{\partial y} = 0 & \text{on } y=0 \\ \psi &= 0, \quad \frac{\partial \psi}{\partial y} = 1 & \text{on } y=1 \end{aligned} \quad (18)$$

The results obtained using the presented RBF-FD formulation are validated against the benchmark multigrid finite difference results obtained in reference [21].

The time-dependent form of the governing equations is used in stream function-vorticity form. The spatial discretization is done using the RBF-FD scheme, while the temporal discretization is carried out using the Crank–Nicholson method (equation (14) with  $\theta = 0.5$ ), with a time step  $\delta t = 0.01$ . Both uniform and random point distributions are considered and the flow problem is solved for three different Reynolds numbers ( $Re = \{100, 400, 1000\}$ ). Nine supporting points are used in each RBF-FD stencil for discretizations of the function derivatives and the value of shape parameter is obtained using the leave-one-out optimization strategy [22], for each RBF-FD stencil. To apply the no-slip boundary conditions, the ghost node strategy proposed in section 3 is employed on the boundary RBF-FD stencils. This facilitates a complete random point distribution in the interior of the domain.

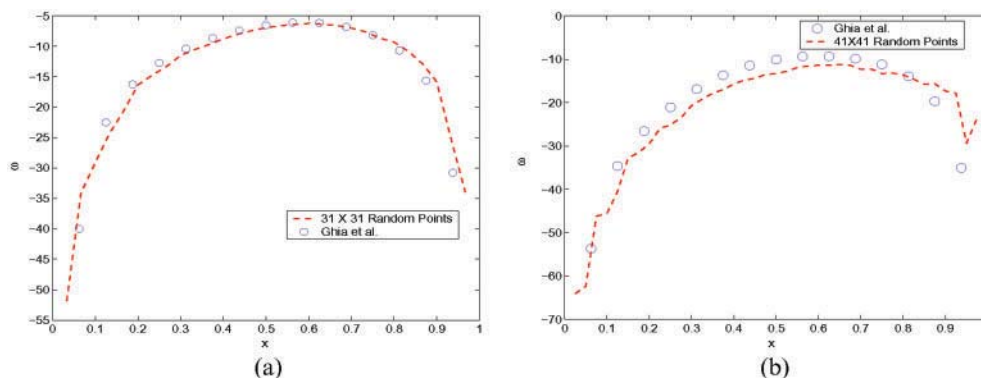
The accuracy of the proposed ghost node strategy is first examined. Figure 3 shows the wall vorticity distribution obtained on the moving lid for the

square lid-driven cavity flow problem at two different Reynolds numbers. A complete random distribution of points without any restriction at the nodes near the boundary was considered for obtaining the results. The results are compared with the wall vorticity values obtained by Ghia *et al.* [21] for the purpose of validation. From the figures, it can be seen that the obtained vorticity distribution agrees well with the benchmark results.

In Fig. 4, the stream function and vorticity contours obtained using the RBF-FD method for  $Re = 100$  are shown. The results displayed are generated using  $41 \times 41$  randomly spaced points. From the plot of the stream function contours it can be seen that the secondary and tertiary vortices near the bottom wall are also captured. It is worth noting that the global features of the flow were captured with relatively small  $21 \times 21$  distribution of points.

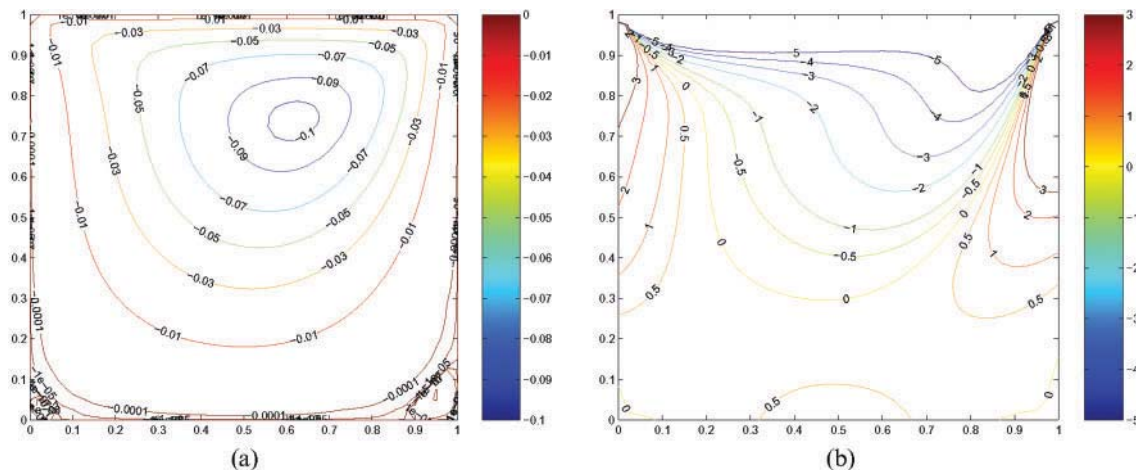
The comparison of velocity components at the horizontal and vertical centres of the cavity with those obtained by Ghia *et al.* [21] are displayed in Fig. 5. The velocity profiles obtained using  $31 \times 31$  and  $41 \times 41$  uniform and random distribution of points are presented in Fig. 5. It can clearly be seen that the velocity profiles are captured accurately as the number of points in the domain is increased.

Figure 6 shows the stream function and vorticity contours obtained for  $Re = 400$  and  $51 \times 51$  random point distribution. The comparison of velocity profiles is presented in Fig. 7. To capture the velocity profiles for  $Re = 400$  accurately, a larger number of points ( $51 \times 51$ ) were needed as compared with those required for  $Re = 100$  ( $41 \times 41$ ). However, the points required were much less than that of the second-order finite difference method, which required about  $129 \times 129$  points in order to capture the velocity profiles [21]. This may be owing to RBF basis with an optimized shape parameter employed in the RBF-FD



**Fig. 3** Comparison of wall vorticity obtained using ghost nodes on the moving boundary of the square driven cavity flow with reference [21]. (a)  $Re = 100$ , (b)  $Re = 400$





**Fig. 4** Square driven cavity:  $Re = 100$ , contours of stream function and vorticity obtained using  $41 \times 41$  random point distribution. (a) Stream function, (b) vorticity

method as compared with the polynomial basis used in reference [21].

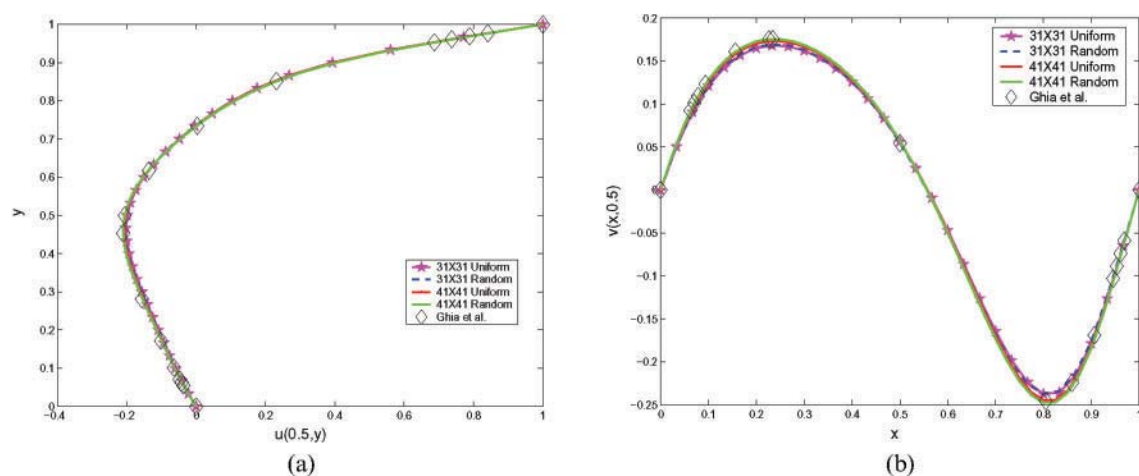
Similar results for  $Re = 1000$  are shown in Figs 8 and 9. The primary, secondary, and tertiary vortices are captured satisfactorily. The velocity profiles obtained using  $51 \times 51$  and  $61 \times 61$  uniform and random point distributions are displayed in Fig. 9.

In comparison with the global RBF collocation method developed by the authors [13], the RBF-FD method is able to provide similar accuracy, but at much lower computational cost. This reduction in computational cost is mainly attributable to the sparse structure of the coefficient matrices. It is also observed that, although the sensitivity of the shape parameter is reduced for the RBF-FD method, the shape parameter still influences the accuracy of the obtained solution, particularly when random node

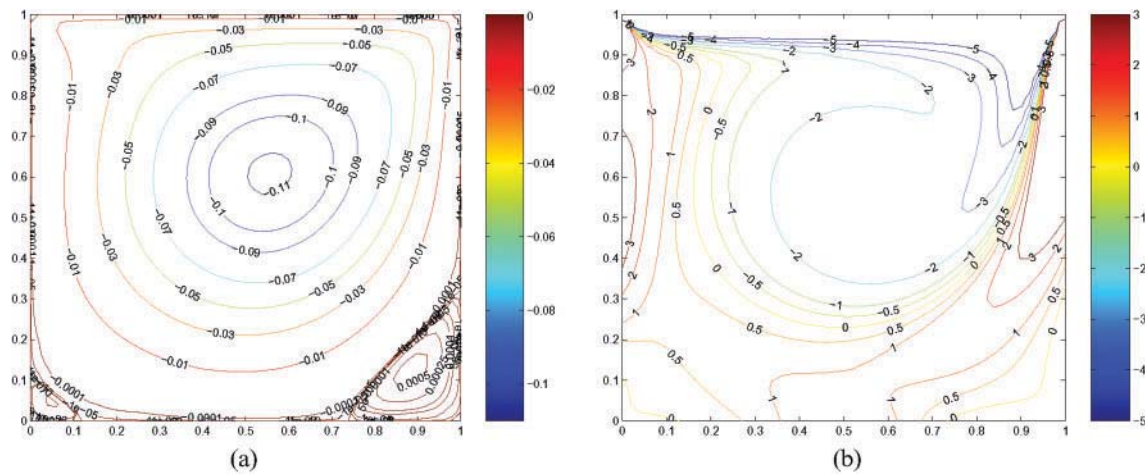
stencils are used for spatial discretization. Finally, additional computational cost incurred by this method with respect to a second-order finite difference method is the calculation of the weights for each stencil, which involves an optimization strategy involving typically around 20 inversions of a  $9 \times 9$  matrix in the present paper.

## 4.2 Rectangular driven cavity flow

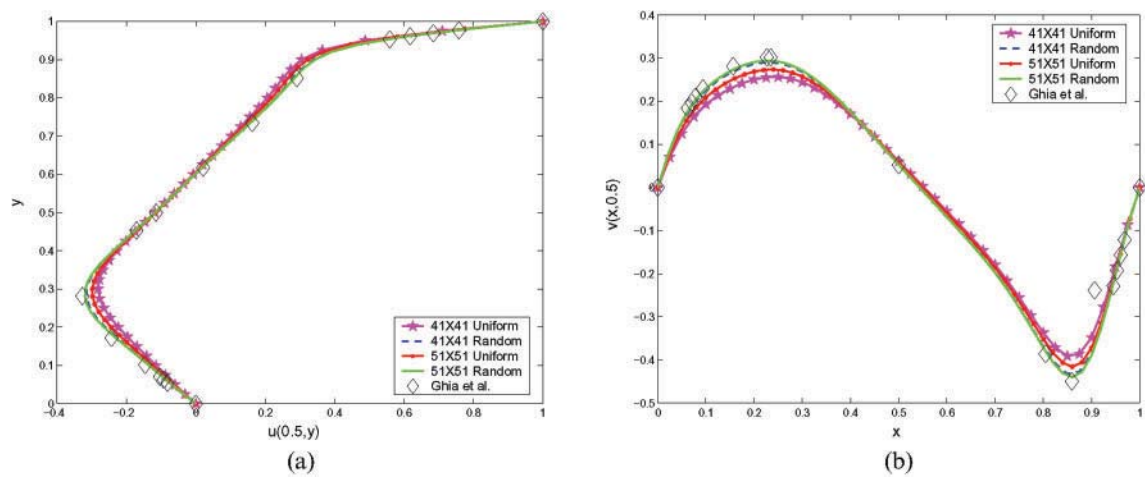
In this subsection, the RBF-FD approach is applied with the ghost node strategy for solving the driven cavity flow in a rectangular cavity with aspect ratio 2. The problem is defined and solved in the rectangle  $0 \leq x \leq 1$ ,  $0 \leq y \leq 2$ . This problem is solved for three different Reynolds numbers of 100, 400, and 1000.



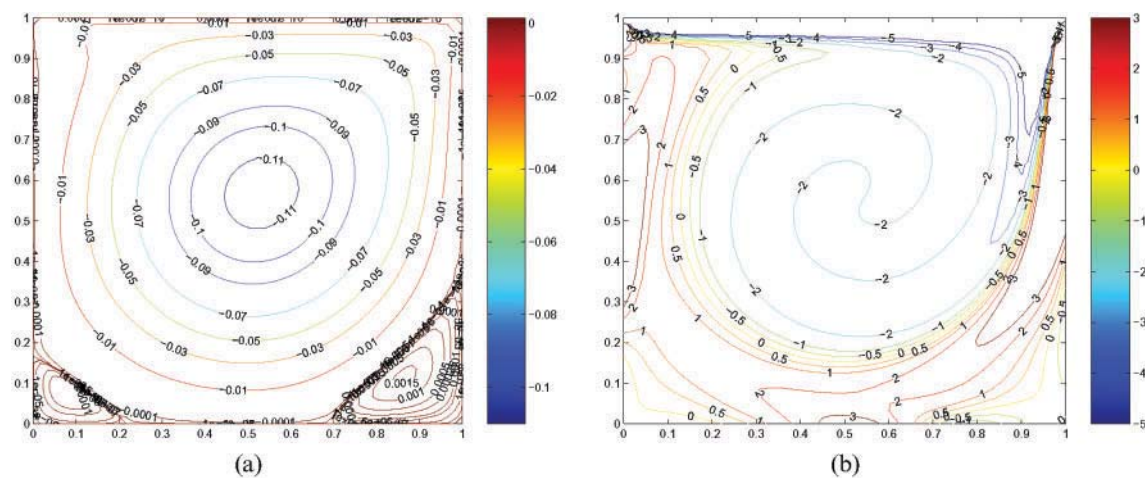
**Fig. 5** Square driven cavity: comparison of velocity profiles obtained on the vertical and horizontal centre-lines using RBF method with reference [21] for  $Re = 100$ . (a) Vertical centre-line, (b) horizontal centre-line



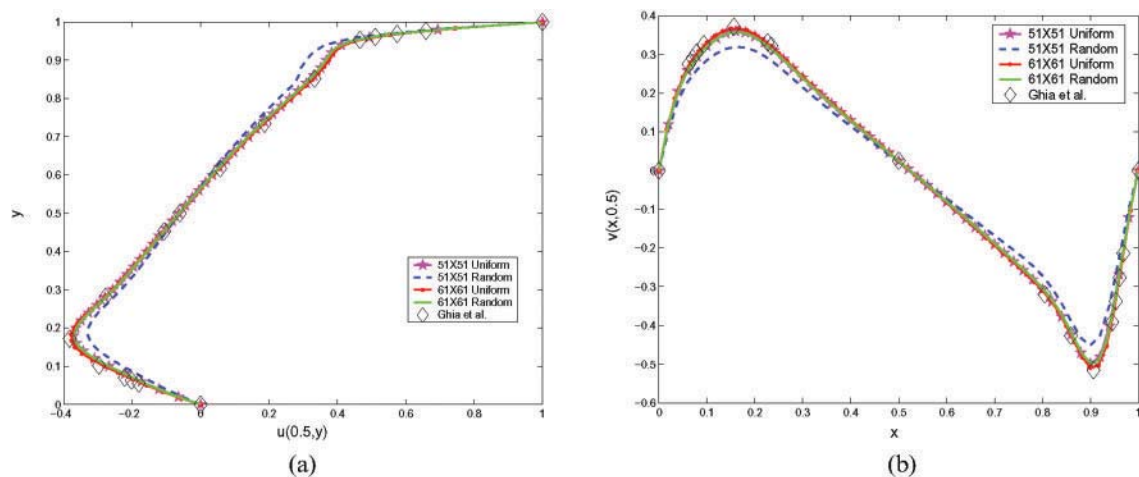
**Fig. 6** Square driven cavity:  $Re = 400$ , contours of stream function and vorticity obtained using  $51 \times 51$  random point distribution. (a) Stream function, (b) vorticity



**Fig. 7** Square driven cavity: comparison of velocity profiles obtained on the vertical and horizontal centre-lines using RBF method with reference [21] for  $Re = 400$ . (a) Vertical centre-line, (b) horizontal centre-line



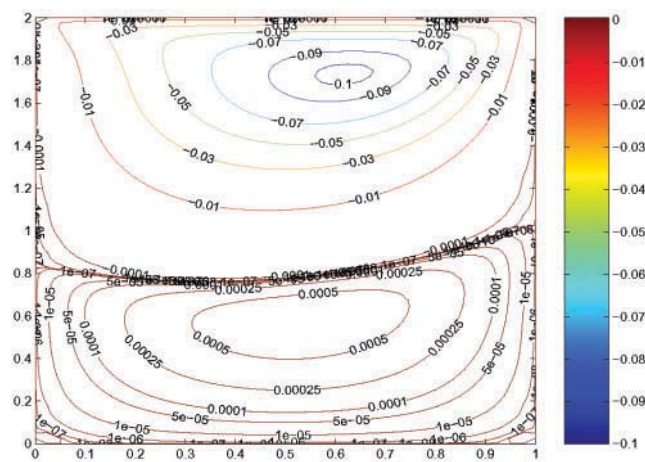
**Fig. 8** Square driven cavity:  $Re = 1000$ , contours of stream function and vorticity obtained using  $61 \times 61$  random point distribution. (a) Stream function, (b) vorticity



**Fig. 9** Square driven cavity: comparison of velocity profiles obtained on the vertical and horizontal centre-lines using RBF method with reference [21] for  $Re = 1000$ . (a) Vertical centre-line, (b) horizontal centre-line

The results are validated against those obtained by Gupta and Kalita [23].

The stream function contours obtained for the three Reynolds numbers are shown in Figs 10 and 11. From the figures, it can be observed that there are two rotating primary vortices as well as secondary vortices in the bottom corners of the rectangular cavity. The top primary vortex properties are reported in Table 3, and are compared with those obtained by Bruneau and Jouron [24]. It can be seen that the RBF-FD method results are in close agreement with the benchmark results.



**Fig. 10** Rectangular driven cavity: streamline patterns obtained for  $Re = 100$  using  $41 \times 81$  uniform point distribution

### 4.3 Near-bed pipeline under current

In this section, the RBF-FD scheme is applied to conduct a stability analysis of a near-bed pipeline under current. When the pipeline is placed near the seabed the fluid current across the pipeline is asymmetrical, thus causing a fluid force to exert on the structure, resulting in the deformation of the pipeline. This problem can be simplified as a beam with different boundary conditions and a non-linear fluid forcing term (see Fig. 12). Based on the Timoshenko beam theory, the general governing equations of the pipeline can be written as

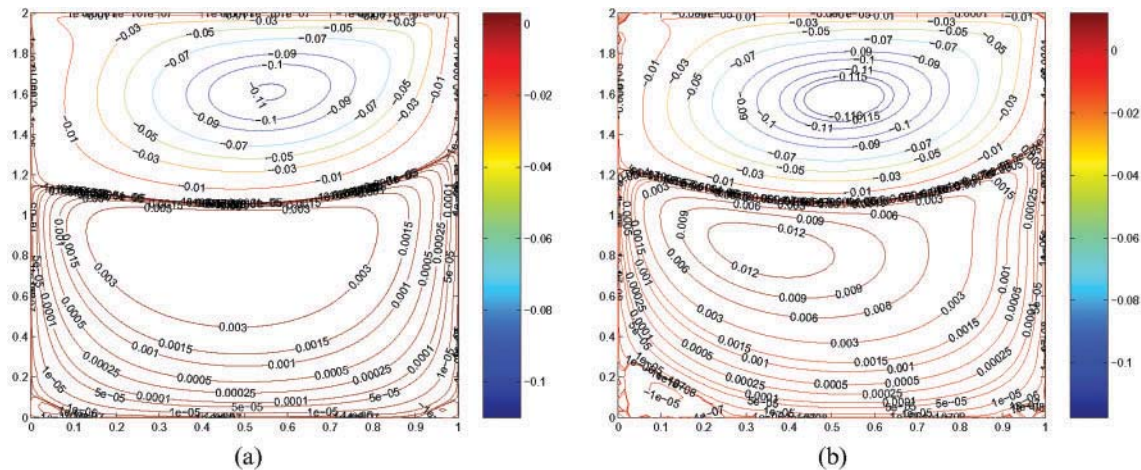
$$\begin{aligned} \frac{\partial}{\partial x} \left[ G A k_s \left( \frac{\partial w}{\partial x} + \theta \right) \right] + f &= 0 \\ \frac{\partial}{\partial x} \left( E I \frac{\partial \theta}{\partial x} \right) - G A k_s \left( \frac{\partial w}{\partial x} + \theta \right) &= 0 \end{aligned} \quad (19)$$

where  $w$  is the deflection of the pipeline,  $\theta$  the rotation,  $G$  the shear modulus,  $A$  cross-section area,  $k_s$  the shear correction coefficient,  $E$  the elasticity modulus,  $I$  the moment of inertia.  $f(x)$  is the fluid force caused by the current pressure difference. Although it has an analytical solution, it converges slowly. Lam *et al.* [25] proposed an approximate rational expression for this fluid force in order to speed up computations and is given by

$$f(x) = \frac{1}{2} \rho A U_0^2 c(d) \quad (20)$$

where  $\rho$  is the mass density,  $U_0$  the current velocity,  $c$





**Fig. 11** Rectangular driven cavity: streamline patterns obtained for  $Re = 400$  using  $81 \times 161$  uniform points and  $Re = 1000$  using  $101 \times 201$  uniform points. (a)  $Re = 400$ , (b)  $Re = 1000$

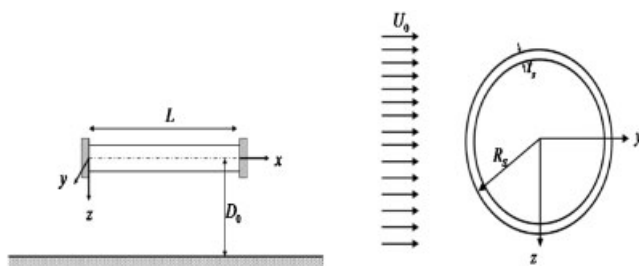
**Table 3** Rectangular driven cavity: top primary vortex strength and location and comparison with reference [24]

Reynolds number		$\psi_{\min}$	$\psi_{\min}$ location
100	[24]	-0.1033	(0.6172, 1.7344)
	Present method	-0.1030	(0.625, 1.721)
400	[24]	-0.1124	(0.5547, 1.5938)
	Present method	-0.1120	(0.555, 1.6125)
1000	[24]	-0.1169	(0.5273, 1.5625)
	Present method	-0.1165	(0.525, 1.57)

and  $d$  are dimensionless coefficients and have the relation

$$c(d) = \frac{2.23d^2 + 12.54d + 0.02}{0.77d^3 + 0.44d^2 + 0.02d} \quad (21)$$

where  $d$  is defined as  $d = (D_0 - w(x) - R_s)/(2R_s)$ , and  $D_0$  is the distance between the central line of pipeline at initial status and the seabed,  $R_s$  the outer pipe radius. For the present numerical study, fixed boundary conditions are employed. Several physical and material parameters employed are obtained from Li *et al.* [26]. The spatial derivatives are approximated using RBF-FD scheme and the discretized non-linear problem is solved using fixed-point iteration. Figure 13 presents the results obtained by conducting the stability analysis. The



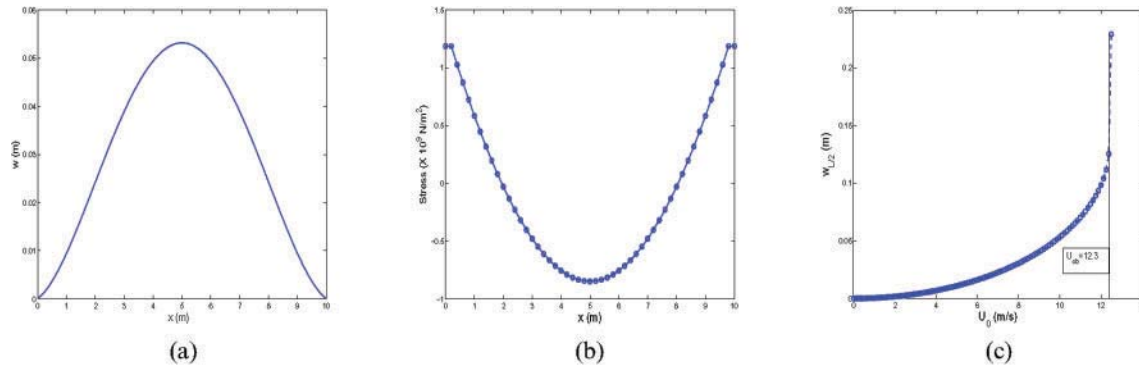
**Fig. 12** Schematic diagram for model fluid-structure interaction problem

critical velocity ( $U_{cb} = 12.3$  m/s) of the current obtained using RBF-FD for instability failure analysis is very near to the one obtained in reference [26] ( $U_{cb} = 12.2$  m/s), thus demonstrating the ability of the RBF-FD method to be applicable for problems involving fluid-structure interaction.

## 5 HIGHER-ORDER RBF-FD SCHEMES

Previous sections have demonstrated the applicability of RBF-FD schemes for the incompressible NS equations. A higher-order version of the RBF-FD scheme is now explored using ideas from Hermite interpolation. This higher-order discretization method using RBFs can be regarded as a generalization of the Mehrstellenverfahren introduced by Collatz [27] and later developed into compact FD formulae by Lele [28]. In the compact FD methodology, for example, the partial derivative of an unknown function with respect to the  $x$ -coordinate at any grid point  $(i, j)$  is given by

$$\left. \frac{\partial u}{\partial x} \right|_{(i,j)} \approx \sum_{k \in \{i-1, i+1\}} w_{(k,j)} u_{(k,j)} + \sum_{k \in \{i-1, i+1\}} \tilde{w}_{(k,j)} \left. \frac{\partial u}{\partial x} \right|_{(k,j)} \quad (22)$$



**Fig. 13** (a) Deflection and (b) stress distribution along the pipeline (when  $U_0 = 10 \text{ m/s}$  and  $D_0 = 0.7 \text{ m}$ ); and (c) instability failure analysis carried out using RBF-FD method

The accuracy of the FD approximation is increased by adding the second term (derivative information), as shown in equation (22). Note that this additional term does not change the stencil size at the grid point  $(i, j)$  in the compact finite difference methodology. Higher-order RBF-FD methods were earlier used for the solution of linear and non-linear Poisson problems [29]. In this work, this approach is extended to develop higher-order schemes for the incompressible NS equations. The higher-order accuracy of the presented formulation is demonstrated by solving the steady-state incompressible Navier–Stokes equations.

### 5.1 Basic formulation

The RBF-FD method generates a local RBF interpolant for expressing the function derivatives at a node as a linear combination of the function values on the nodes present in the support region of the considered node. In the spirit of compact FD schemes, the accuracy of the RBF-FD discretization can be increased by considering not only the function values but also the derivative values on the nodes present in the supporting region. The weights of the higher-order stencil are computed using the Hermite interpolation technique.

First a brief introduction to the Hermite interpolation method is given. Let  $\mathcal{L}$  be an arbitrary linear differential operator and let  $\boldsymbol{\eta}$  be a vector containing some combination of  $m \leq n$  distinct numbers from the set  $\{1, 2, \dots, n\}$ . The function values  $u(\mathbf{x}_i)$  are specified at each of the  $n$  distinct data points  $\{\mathbf{x}_i\}_{i=1}^n$ . In addition, data corresponding to the differential operator operating on the function,  $\mathcal{L}u(\mathbf{x}_{\eta_l})$ , are specified at  $m$  points  $\{\mathbf{x}_{\eta_l}\}_{l=1}^m$ . Note that the point set  $\{\mathbf{x}_{\eta_l}\}_{l=1}^m$  is a subset of the set  $\{\mathbf{x}_i\}_{i=1}^n$ . Then, the interpolant passing through all the data can be written as

$$u(\mathbf{x}) \approx s(\mathbf{x}) = \sum_{i=1}^n \lambda_i \phi(\|\mathbf{x} - \mathbf{x}_i\|) + \sum_{l=1}^m \tilde{\lambda}_l \mathcal{L}_2 \phi(\|\mathbf{x} - \mathbf{x}_{\eta_l}\|) + \beta \quad (23)$$

where  $\mathcal{L}_2 \phi(\|\cdot\|)$  is a basis function derived by the functional  $\mathcal{L}$  acting on the multiquadric basis  $\phi(\|\cdot\|)$  as a function of the second variable (centre) and  $\beta$  is a constant. The unknown coefficients are obtained by enforcing the conditions  $s(\mathbf{x}_i) = u(\mathbf{x}_i)$ ,  $i = 1, \dots, n$ ;  $\mathcal{L}s(\mathbf{x}_{\eta_l}) = \mathcal{L}u(\mathbf{x}_{\eta_l})$ ,  $l = 1, \dots, m$ ; and  $\sum_{i=1}^n \lambda_i = 0$ . Imposing these conditions leads to the following block linear system of equations

$$\begin{bmatrix} \Phi & \mathcal{L}_2 \Phi & \mathbf{e} \\ \mathcal{L} \Phi & \mathcal{L} \mathcal{L}_2 \Phi & \mathbf{0} \\ \mathbf{e}^T & \mathbf{0}^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\lambda} \\ \tilde{\boldsymbol{\lambda}} \\ \beta \end{bmatrix} = \begin{bmatrix} \mathbf{u} \\ \mathcal{L} \mathbf{u} \\ 0 \end{bmatrix} \quad (24)$$

where

$$\begin{aligned} \Phi_{i,j} &= \phi(\|\mathbf{x}_i - \mathbf{x}_j\|), \quad i, j = 1, \dots, n \\ \mathcal{L}_2 \Phi_{i,j} &= \mathcal{L}_2 \phi(\|\mathbf{x}_i - \mathbf{x}_{\eta_j}\|), \quad i = 1, \dots, n, \quad j = 1, \dots, m \\ \mathcal{L} \Phi_{i,j} &= \mathcal{L} \phi(\|\mathbf{x}_{\eta_i} - \mathbf{x}_j\|), \quad i = 1, \dots, m, \quad j = 1, \dots, n \\ \mathcal{L} \mathcal{L}_2 \Phi_{i,j} &= \mathcal{L} \mathcal{L}_2 \phi(\|\mathbf{x}_{\eta_i} - \mathbf{x}_{\eta_j}\|), \quad i = 1, \dots, m, \quad j = 1, \dots, m \end{aligned}$$

and  $\mathbf{e}_i = 1$ ,  $i = 1, \dots, n$ . Equation (24) is solved using a backwards substitution routine.

In Lagrange form, the Hermite interpolant can also be written as

$$\bar{s}(\mathbf{x}) = \sum_{i=1}^n \chi(\|\mathbf{x} - \mathbf{x}_i\|) u(\mathbf{x}_i) + \sum_{l=1}^m \tilde{\chi}(\|\mathbf{x} - \mathbf{x}_{\eta_l}\|) \mathcal{L}u(\mathbf{x}_{\eta_l}) \quad (25)$$

where  $\chi(\|\mathbf{x} - \mathbf{x}_i\|)$  and  $\tilde{\chi}(\|\mathbf{x} - \mathbf{x}_{\eta_i}\|)$  are of the form equation (23) and satisfy the cardinal conditions, i.e.

$$\chi(\|\mathbf{x}_k - \mathbf{x}_i\|) = \begin{cases} 1, & \text{if } k=i, \\ 0, & \text{if } k \neq i, \end{cases} \quad k=1, \dots, n \quad (26)$$

$$\mathcal{L}\chi(\|\mathbf{x}_{\eta_k} - \mathbf{x}_i\|) = 0, \quad k=1, \dots, m \quad (27)$$

and

$$\tilde{\chi}(\|\mathbf{x}_k - \mathbf{x}_{\eta_l}\|) = 0, \quad k=1, \dots, n \quad (28)$$

$$\mathcal{L}\tilde{\chi}(\|\mathbf{x}_{\eta_k} - \mathbf{x}_{\eta_l}\|) = \begin{cases} 1, & \text{if } k=l, \\ 0, & \text{if } k \neq l, \end{cases} \quad k=1, \dots, m \quad (29)$$

Equation (25) is the basis for deriving higher-order RBF-FD stencils. Consider the node  $\mathbf{x}_1$  with its support region containing  $n$  points, denoted by a dashed circle around  $\mathbf{x}_1$ , as in Fig. 1. The goal is to obtain a higher-order RBF-FD discretization of  $\mathcal{L}u(\mathbf{x}_1)$ . The nodes in the support region, which are shaded grey, are those nodes where both the function values ( $u(\mathbf{x})$ ) and the functional values ( $\mathcal{L}u(\mathbf{x})$ ) are used, i.e. the set  $\eta$  of cardinality, say,  $m \leq n$ .

The higher-order RBF-FD discretization for  $\mathcal{L}u(\mathbf{x}_1)$  is given by the Lagrange form of the interpolant, i.e.

$$\begin{aligned} \mathcal{L}u(\mathbf{x}_1) \approx \mathcal{L}\bar{s}(\mathbf{x}_1) &= \sum_{i=1}^n \mathcal{L}\chi(\|\mathbf{x}_1 - \mathbf{x}_i\|)u(\mathbf{x}_i) \\ &+ \sum_{l=1}^m \mathcal{L}\tilde{\chi}(\|\mathbf{x}_1 - \mathbf{x}_{\eta_l}\|)\mathcal{L}u(\mathbf{x}_{\eta_l}) \end{aligned} \quad (30)$$

Equation (30) can be rewritten as a compact FD formula of the form

$$\mathcal{L}u(\mathbf{x}_1) \approx \sum_{i=1}^n w_{(1,i)}^{\mathcal{L}} u(\mathbf{x}_i) + \sum_{l=1}^m \tilde{w}_{(1,l)}^{\mathcal{L}} \mathcal{L}u(\mathbf{x}_{\eta_l}) \quad (31)$$

where the weights for higher-order RBF-FD  $\{w_{(1,i)}^{\mathcal{L}}\}_{i=1}^n$  and  $\{\tilde{w}_{(1,l)}^{\mathcal{L}}\}_{l=1}^m$  are now given by

$$w_{(1,i)}^{\mathcal{L}} = \mathcal{L}\chi(\|\mathbf{x}_1 - \mathbf{x}_i\|), \quad \tilde{w}_{(1,l)}^{\mathcal{L}} = \mathcal{L}\tilde{\chi}(\|\mathbf{x}_1 - \mathbf{x}_{\eta_l}\|) \quad (32)$$

where the superscript  $\mathcal{L}$  on the weights denote that the higher-order RBF-FD weights are computed for that particular operator.

In practise, the weights are computed by solving the linear system

$$\begin{bmatrix} \Phi & \mathcal{L}_2\Phi & \mathbf{e} \\ \mathcal{L}\Phi & \mathcal{L}\mathcal{L}_2\Phi & 0 \\ \mathbf{e}^T & 0^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \tilde{\mathbf{w}} \\ \mu \end{bmatrix} = \begin{bmatrix} \mathcal{L}^*\Phi_1 \\ \mathcal{L}^*\tilde{\Phi}_1 \\ 0 \end{bmatrix} \quad (33)$$

where  $\mathcal{L}^*\Phi_1$  and  $\mathcal{L}^*\tilde{\Phi}_1$  denote the evaluation of the column vectors  $\mathcal{L}^*\Phi = [\mathcal{L}\phi(\|\mathbf{x} - \mathbf{x}_1\|) \mathcal{L}\phi(\|\mathbf{x} - \mathbf{x}_2\|) \dots \mathcal{L}\phi(\|\mathbf{x} - \mathbf{x}_n\|)]^T$ , and  $\mathcal{L}^*\tilde{\Phi} = [\mathcal{L}\mathcal{L}_2\phi(\|\mathbf{x} - \mathbf{x}_{\eta_1}\|) \mathcal{L}\mathcal{L}_2\phi(\|\mathbf{x} - \mathbf{x}_{\eta_2}\|) \dots \mathcal{L}\mathcal{L}_2\phi(\|\mathbf{x} - \mathbf{x}_{\eta_m}\|)]^T$  at the node  $\mathbf{x}_1$ . Here,  $\mu$  is a scalar value related to the constant  $\beta$  in equation (23) and enforces the condition

$$\sum_{i=1}^n w_{(1,i)} = 0$$

which ensures that the stencil is exact for all constants.

Once the weights are computed by solving equation (33) for each node, they can be stored and used to discretize the partial differential equation in a similar manner as in the compact FD schemes.

## 6 HIGHER ORDER RBF-FD FOR THE INCOMPRESSIBLE NS EQUATIONS

The previous section outlined the idea of using the Hermite interpolation technique to obtain a higher-order RBF-FD discretisation for each RBF-FD stencil. It can be observed that a family of higher-order schemes can be derived by defining which operator  $\mathcal{L}$  one is using for the higher-order RBF-FD stencil. This section presents one such formulation for the steady-state incompressible NS equations. This formulation has the advantage of easier implementation of the no-slip boundary conditions. First the governing equations of steady incompressible flows in streamfunction ( $\psi$ ) - vorticity ( $\omega$ ) formulation are recalled

$$\nabla^2 \psi = -\omega \quad (34)$$

$$\nabla^2 \omega = Re \left( u \frac{\partial \omega}{\partial x} + v \frac{\partial \omega}{\partial y} \right) \quad (35)$$

where  $Re$  is the Reynolds number and  $(u, v)$  are the Cartesian velocity components of the flow.

As usual, the entire domain is discretized into a set of interior and boundary nodes and determine the stencil at each node. Each higher-order RBF-FD stencil contains  $n$  nodes and the vector  $\boldsymbol{\eta}$  of cardinality  $m \leq n$ . The stencil information consists

of the function values ( $\psi(\mathbf{x})$  or  $\omega(\mathbf{x})$ ) at each of the  $n$  nodes and the functional information ( $\mathcal{L}\psi(\mathbf{x})$  or  $\mathcal{L}\omega(\mathbf{x})$ ) on each of the  $m$  nodes. Note that the operator  $\mathcal{L}$  is arbitrary.

The higher-order RBF-FD discretizations for the Laplacian of stream function at each interior node  $\mathbf{x}_i$  is given by

$$\nabla^2 \psi(\mathbf{x}_i) \approx \sum_{j=1}^n w_{(i,j)}^{\nabla^2} \psi(\mathbf{x}_j) + \sum_{l=1}^m \tilde{w}_{(i,l)}^{\nabla^2} \{\nabla^2 \psi(\mathbf{x}_{\eta_l})\} \quad (36)$$

where the higher-order RBF-FD weights  $\{w_{(i,j)}^{\nabla^2}\}_{j=1}^n$  and  $\{\tilde{w}_{(i,l)}^{\nabla^2}\}_{l=1}^m$  are obtained using equation (24) with the operator  $\mathcal{L} \equiv \nabla^2$ .

Similarly the discretization of Laplacian of vorticity can be obtained as

$$\nabla^2 \omega(\mathbf{x}_i) \approx \sum_{j=1}^n w_{(i,j)}^{\nabla^2} \omega(\mathbf{x}_j) + \sum_{l=1}^m \tilde{w}_{(i,l)}^{\nabla^2} \{\nabla^2 \omega(\mathbf{x}_{\eta_l})\} \quad (37)$$

For the sake of brevity, the value of any physical quantity at node  $\mathbf{x}_j$  is denoted by the subscript  $j$ . In equation (36), the second term (quantity in curly brackets) can be replaced by the right-hand side of equation (34). Similarly, in equation (37), the second term can be replaced by the right-hand side of equation (35).

The modified discretizations now become

$$\nabla^2 \psi_i \approx \sum_{j=1}^n w_{(i,j)}^{\nabla^2} \psi_j + \sum_{l=1}^m \tilde{w}_{(i,l)}^{\nabla^2} (-\omega_{\eta_l}) \quad (38)$$

and

$$\nabla^2 \omega_i \approx \sum_{j=1}^n w_{(i,j)}^{\nabla^2} \omega_j + Re \sum_{l=1}^m \tilde{w}_{(i,l)}^{\nabla^2} \left( u \frac{\partial \omega}{\partial x} + v \frac{\partial \omega}{\partial y} \right)_{\eta_l} \quad (39)$$

The solution of the governing NS equations via a fixed-point iteration scheme is now returned to. Denoting the iteration number  $k$  with a superscript  $k$  on the physical variable, the governing equations at iteration  $k+1$  for the node  $\mathbf{x}_i$  are given by

$$\begin{aligned} \nabla^2 \psi_i^{k+1} &= -\omega_i^k \\ \nabla^2 \omega_i^{k+1} &= Re \left[ \bar{u}_i \left( \frac{\partial \omega}{\partial x} \right)_i^{k+1} + \bar{v}_i \left( \frac{\partial \omega}{\partial y} \right)_i^{k+1} \right] \end{aligned} \quad (40)$$

where  $\bar{u}_i$  and  $\bar{v}_i$  are the current estimates of components of the velocity vector. Substituting the derived higher-order RBF-FD discretizations for Laplacian of streamfunction (equation (38)) and vorticity (equation (39)) into equation (40)

$$\sum_{j=1}^n w_{(i,j)}^{\nabla^2} \psi_j^{k+1} + \sum_{l=1}^m \tilde{w}_{(i,l)}^{\nabla^2} (-\omega_{\eta_l}^k) = -\omega_i^k \quad (41)$$

and

$$\begin{aligned} &\sum_{j=1}^n w_{(i,j)}^{\nabla^2} \omega_j^{k+1} \\ &+ Re \sum_{l=1}^m \tilde{w}_{(i,l)}^{\nabla^2} \left[ \bar{u}_{\eta_l} \left( \frac{\partial \omega}{\partial x} \right)_{\eta_l}^k + \bar{v}_{\eta_l} \left( \frac{\partial \omega}{\partial y} \right)_{\eta_l}^k \right] \\ &- Re \left[ \bar{u}_i \sum_{j=1}^n w_{(i,j)}^{(x)} \omega_j^{k+1} + \bar{v}_i \sum_{j=1}^n w_{(i,j)}^{(y)} \omega_j^{k+1} \right] = 0 \end{aligned} \quad (42)$$

Note that in equation (42), the vorticity gradients  $\left( \frac{\partial \omega}{\partial x}, \frac{\partial \omega}{\partial y} \right)$  were discretized using the RBF-FD method.

The velocity components ( $\bar{u}$ ,  $\bar{v}$ ) in equation (42) are obtained using the higher-order RBF-FD discretizations given by

$$\bar{u}_i \equiv \frac{\partial \bar{\psi}}{\partial y} \Big|_i \approx \sum_{j=1}^n w_{(i,j)}^y \bar{\psi}_j + \sum_{l=1}^m \tilde{w}_{(i,l)}^y \frac{\partial \bar{\psi}}{\partial y} \Big|_{\eta_l} \quad (43)$$

$$\bar{v}_i \equiv -\frac{\partial \bar{\psi}}{\partial x} \Big|_i \approx - \left[ \sum_{j=1}^n w_{(i,j)}^x \bar{\psi}_j + \sum_{l=1}^m \tilde{w}_{(i,l)}^x \frac{\partial \bar{\psi}}{\partial x} \Big|_{\eta_l} \right] \quad (44)$$

where  $\bar{\psi}$  is the current estimate of the stream function.

The iteration procedure is explained for a problem with no-slip boundary conditions. Recall that the no-slip boundary condition consists of a Dirichlet and a Neumann condition for stream function at each boundary point, see equation (16). Now, given an initial guess for stream function and vorticity, it is possible to solve the system of equations arising from satisfying equation (41) at all interior nodes along with the Dirichlet boundary conditions for the stream function to obtain the new estimate for stream function ( $\bar{\psi}$ ). To obtain the new velocity vector estimate, the system of equations arising from satisfying equations (43) and (44) at all interior nodes is solved. Note that, whenever the support point  $\mathbf{x}_{\eta_l}$  for a node is on the boundary, the Neumann condition for stream



function is used thus facilitating an easier implementation of no-slip boundary conditions. Next, the new estimate of vorticity on the boundary is obtained using the ghost node strategy proposed in section 3. Once the velocity vector estimate is known, the linear system of equations arising from satisfying equation (42) is solved with the Dirichlet vorticity conditions obtained using the ghost node strategy. Once the physical quantities are obtained, advance to the next iteration. This procedure is repeated until convergence.

### 6.1 Numerical results

Numerical results obtained for the higher-order RBF-FD method for the steady convection diffusion equation and the incompressible NS equations are now presented.

First the higher-order RBF-FD approach is illustrated for the steady-state convection–diffusion equation of the form

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} - P_e \frac{\partial u}{\partial x} = 0 \quad (45)$$

in the domain  $[0, 1] \times [0, 0.6]$  with the boundary conditions

$$u = 1 \quad \text{on } x = 0, \quad u = 2 \quad \text{on } x = 1 \quad (46)$$

$$\frac{\partial u}{\partial y} = 0 \quad \text{on } y = 0, \quad \frac{\partial u}{\partial y} = 0 \quad \text{on } y = 1 \quad (47)$$

The exact solution for this problem is given by

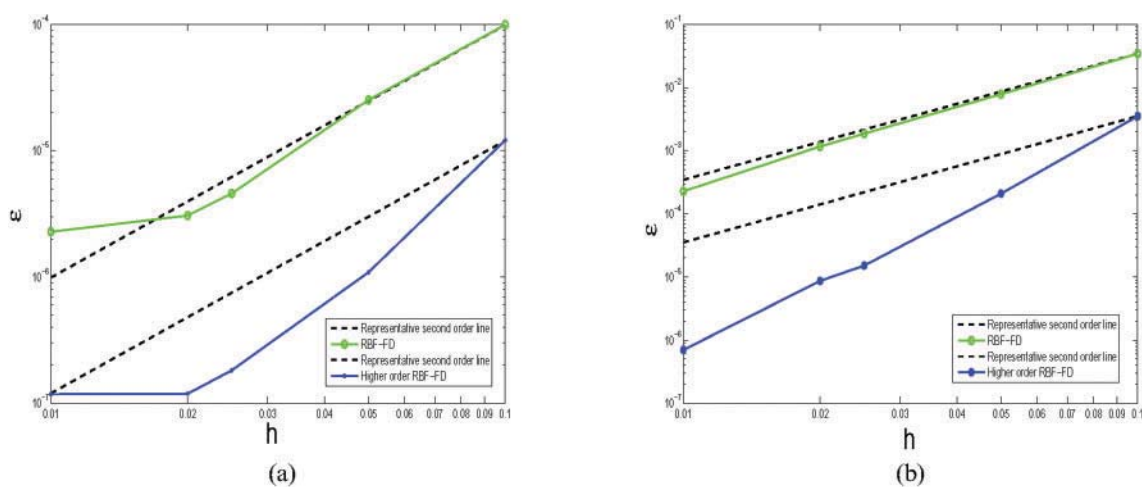
$$u_{\text{exact}} = 2 - \frac{1 - \exp[P_e(x-1)]}{1 - \exp(-P_e)} \quad (48)$$

Figure 14 presents the convergence plots of the RBF-FD and higher-order RBF-FD for the convection diffusion problem, equation (45), for two Peclet numbers 1.0 and 10.0. The operator  $\mathcal{L}$  is taken as

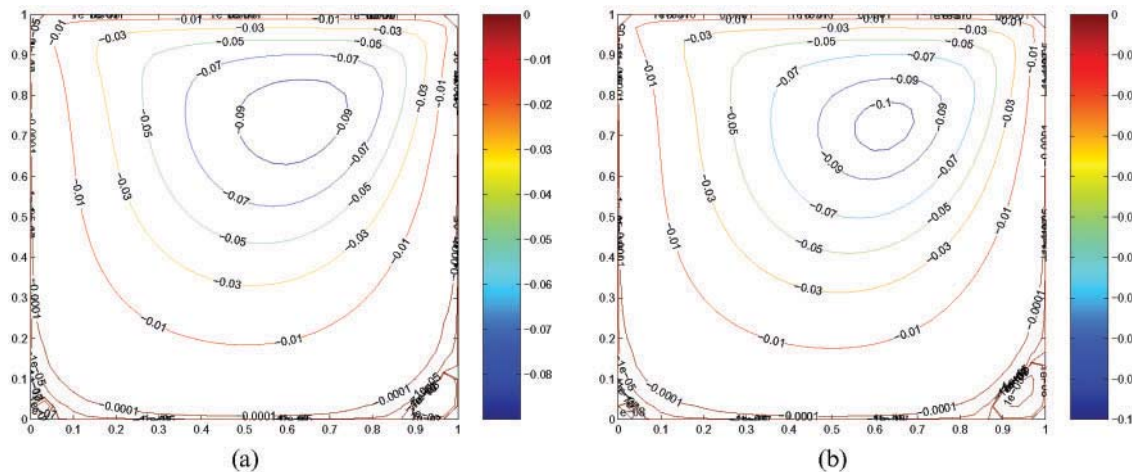
$$\mathcal{L} \equiv \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} - P_e \frac{\partial}{\partial x}$$

and the higher-order RBF-FD weights are obtained using equation (33). From Fig. 14, it can be seen that the results obtained using the higher-order method are at least two orders more accurate than that of RBF-FD method. It is also worth mentioning that for  $P_e = 100.0$ , with a uniform discretization of  $201 \times 201$  nodes, the error norm observed for the computed solution using higher-order RBF-FD was  $\varepsilon = \mathcal{O}(10^{-4})$ .

Next, turning to the incompressible NS, the formulation outlined in section 6 is used to solve the square lid-driven cavity flow. Figure 15 presents the stream function contours obtained for the cavity flow at  $Re = 100$  with a uniform distribution of  $31 \times 31$  nodes. The left subfigure solution is obtained using the RBF-FD method and the right subfigure is obtained using the higher-order RBF-FD method. From Fig. 15, it can be clearly seen that the higher-order method captures the solution more accurately with a small number of  $31 \times 31$  points.



**Fig. 14** Comparison of convergence behaviours of RBF-FD and higher-order RBF-FD for a model steady state convection–diffusion equation,  $h$  is the mesh spacing and  $\varepsilon$  denotes the  $L_\infty$  norm between exact and computed solutions. (a)  $P_e = 1.0$ , (b)  $P_e = 10.0$

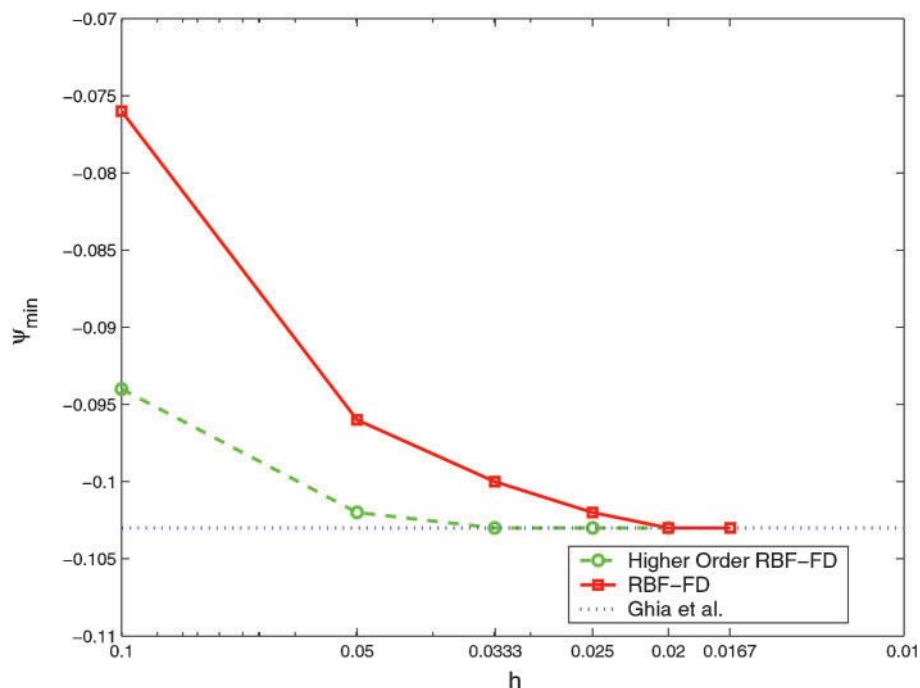


**Fig. 15** Comparison of convergence behaviours of (a) RBF-FD and (b) higher-order RBF-FD for a model steady-state driven cavity problem

Figure 16 estimates the performance of higher-order RBF-FD and RBF-FD in terms of accuracy. On the  $x$ -axis, the mesh spacing  $h$  is plotted on a log scale in the reverse direction. On the  $y$ -axis, the minimum value of stream function  $\psi_{\min}$  in the whole domain (strength of the primary vortex) is plotted. The benchmark value obtained by Ghia *et al.* [21] is shown as a horizontal dotted line in the figure. From Fig. 16, it can clearly be observed that the higher-order method captures the true solution at considerably less points ( $h \approx 0.033$ ) as compared with the original RBF-FD method.

## 7 CONCLUDING REMARKS

The RBF-FD method is presented for solving incompressible Navier–Stokes equations. This method approximates the function derivatives at a node in terms of the function values on a scattered set of points present in support region of the node. The RBF-FD method uses local interpolation problems and hence generates sparse and well-conditioned matrices. It also has the property of decreased sensitivity with respect to shape parameter value in comparison with the RBF collocation method. A



**Fig. 16** Convergence of  $\psi_{\min}$  for higher-order RBF-FD and RBF-FD at Reynolds number 100

ghost node strategy employed for incorporating no-slip boundary conditions removes the limitations of having a locally orthogonal grid near the boundary and thus makes the method more suitable for complete random node discretisations. Numerical studies conducted on the driven cavity flow problems and the fluid–structure interaction of near-bed submarine pipelines using the RBF-FD method show that this method achieves accurate results which are in good agreement with the benchmark results.

A higher-order RBF-FD method is explored for solving partial differential equations. The higher-order method is obtained by using Hermite RBF interpolation method to construct the function approximation at each node in the domain. A higher-order formulation for steady incompressible Navier–Stokes equations is presented. The accuracy of the higher-order method is investigated by solving for a model convection diffusion equation and square lid-driven cavity flow. Numerical results obtained indicate that this method indeed is a higher-order method with a higher capability of spatial resolution with respect to the RBF-FD method.

© Authors 2009

## REFERENCES

- 1 Belytschko, T., Krongauz, Y., Organ, D., Fleming, M., and Krysl, P. Meshless methods. An overview and recent developments. *Computer Methods Appl. Mech. Engng*, 1996, **139**, 49–74.
- 2 Perrone, N. and Kao, R. A general finite difference method for arbitrary meshes. *Computers and Structs*, 1975, **5**, 45–47.
- 3 Lucy, L. B. A numerical approach to the testing of fission hypothesis. *Astronomical J.*, 1977, **8**, 1013–1024.
- 4 Nayroles, B., Touzot, G., and Villon, P. Generalizing the finite element method: diffuse approximation and diffuse elements. *Comput. Mech.*, 1992, **10**, 307–318.
- 5 Belytschko, T., Lu, Y., and Gu, L. Element free Galerkin methods. *Int. J. Numer. Methods Engng*, 1994, **37**, 229–256.
- 6 Liu, W. K., Jun, S., and Belytschko, T. Reproducing Kernel particle methods. *Int. J. Numer. Methods in Fluids*, 1995, **20**, 1081–1106.
- 7 Melenk, J. M. and Babuska, I. The partition of unity finite element method: basic theory and applications. Technical Report, Texas Institute for Computational and Applied Mathematics, 1996.
- 8 Onate, E., Idelsohn, S., Zienkiewicz, O. C., Taylor, R. L., and Sacco, C. A stabilized finite point method for analysis of fluid mechanics problems. *Computer Methods Appl. Mech. Engng*, 1996, **139**, 315–346.
- 9 Atluri, S. N. and Zhu, T. New meshless local Petrov–Galerkin approach in computational mechanics. *Comput. Mech.*, 1998, **22**, 117–127.
- 10 Kansa, E. J. Multiquadrics—a scattered data approximation scheme with applications to computation fluid dynamics-I. Surface approximations and partial derivatives estimates. *Computers Math. with Applic.*, 1990, **19**(8/9), 127–145.
- 11 Wang, J. G. and Liu, G. R. On optimal shape parameters of radial basis functions used for 2D meshless methods. *Computer Methods in Appl. Mech. Engng*, 2002, **191**, 2611–2630.
- 12 Shokri, M. A numerical method for KdV equation using collocation and radial basis functions. *Non-linear Dynamics*, 2007, **50**, 111–120.
- 13 Chinchapatnam, P. P., Djidjeli, K., and Nair, P. B. Radial basis function meshless method for the steady incompressible Navier–Stokes equations. *Int. J. Computer Math.*, 2007, **84**, 1509–1521.
- 14 Fasshauer, G. E. Solving partial differential equations with radial basis functions: multilevel methods and smoothing. *Adv. in Comput. Math.*, 1999, **11**, 139–159.
- 15 Kansa, E. J. and Hon, Y. C. Circumventing the ill-conditioning problem with multiquadric radial basis functions: applications to elliptic partial differential equations. *Comput. Math. Appl.*, 2002, **39**, 123–137.
- 16 Ling, L. and Kansa, E. J. A least-squares preconditioner for radial basis functions collocation methods. *Adv. Comput. Math.*, 2005, **23**, 31–54.
- 17 Chinchapatnam, P. P., Djidjeli, K., and Nair, P. B. Domain decomposition for time-dependent problems using radial based meshless methods. *Numer. Methods for Partial Differential Equations*, 2006, **23**, 38–59.
- 18 Wright, B. W. and Fornberg, B. Scattered node compact finite difference-type formulas generated from radial basis functions. *J. Comput. Phys.*, 2006, **212**, 99–123.
- 19 Tolstykh, A. I. and Shirobokov, D. A. On using radial basis functions in a finite difference mode with applications to elasticity problems. *Comput. Mech.*, 2003, **33**, 68–79.
- 20 Spotz, W. F. *High-order compact finite difference schemes for computational mechanics*, PhD Thesis, University of Texas at Austin, 1995.
- 21 Ghia, U., Ghia, K. N., and Shin, C. T. High-Re solutions for incompressible flow using the Navier–Stokes equations and a multigrid method. *J. Comput. Phys.*, 1982, **48**, 387–411.
- 22 Chinchapatnam, P. P. *Radial basis function based meshless methods for fluid flow problems*, PhD Thesis, Southampton University, 2006.
- 23 Gupta, M. M. and Kalita, J. C. A new paradigm for solving Navier–Stokes equations: stream function-velocity formulation. *J. Comput. Phys.*, 2005, **207**, 52–68.

- 24 **Bruneau, C. H.** and **Jouron, C.** An efficient scheme for solving steady incompressible Navier–Stokes equations. *J. Comput. Phys.*, 1990, **89**, 389–413.
- 25 **Lam, K. Y., Wang, Q. X., and Zong, Z.** A nonlinear fluid-structure interaction analysis of a near-bed submarine pipeline in a current. *J. Fluids and Structs*, 2002, **16**, 1177–1191.
- 26 **Li, H., Cheng, J. Q., Ng, T. Y., Chen, J., and Lam, K. Y.** A meshless Hermite-cloud method for nonlinear fluid-structure analysis of near-bed submarine pipelines under current. *Engng Structs*, 2004, **26**, 531–542.
- 27 **Collatz, L.** *The numerical treatment of differential equations*, 1960 (Springer Verlag, Berlin).
- 28 **Lele, K.** Compact finite difference schemes with spectral-like resolution. *J. Comput. Phys.*, 1992, **103**, 16–42.
- 29 **Wright, G. B. and Fornberg, B.** Scattered node compact finite difference-type formulas generated from radial basis functions. *J. Comput. Phys.*, 2006, **212**, 99–123.