Manuscript submitted to Int. J. Numer. Meth. Fluids

Local Moving Least Square - One-Dimensional IRBFN Technique for Incompressible Viscous Flows

D. Ngo-Cong[1,2], N. Mai-Duy[1], W. Karunasena[2] and T. Tran-Cong[1,3]

Abstract: This paper presents a local moving least square - one-dimensional integrated radial basis function networks (LMLS-1D-IRBFN) method for solving incompressible viscous flow problems using stream function-vorticity formulation. In this method, the partition of unity method is employed as a framework to incorporate the moving least square (MLS) and one-dimensional integrated radial basis function networks (1D-IRBFN) techniques. The major advantages of the proposed method include: (i) a banded sparse system matrix which helps reduce the computational cost; (ii) the Kronecker-δ property of the constructed shape function which helps impose the essential boundary condition in an exact manner; and (iii) high accuracy and fast convergence rate owing to the use of integration instead of conventional differentiation to construct the local RBF approximations. Several examples including two-dimensional Poisson problems, lid-driven cavity flow and flow past a circular cylinder are considered and the present results are compared with the exact solutions and numerical results from other methods in the literature to demonstrate the attractiveness of the proposed method.

Keywords: Incompressible viscous flow; Stream function-vorticity formulation; Integrated radial basis functions; Moving least square; Partition of unity; Cartesian grids; Numerical methods.

1 Introduction

Nowadays, numerical simulation has become an essential tool for the analysis of practical problems of engineering and physical sciences. Finite element method (FEM), finite difference method (FDM) and finite volume method (FVM) are meth-

[1] Computational Engineering and Science Research Centre, Faculty of Engineering and Surveying, The University of Southern Queensland, Toowoomba, QLD 4350, Australia.
[2] Centre of Excellence in Engineered Fibre Composites, Faculty of Engineering and Surveying, The University of Southern Queensland, Toowoomba, QLD 4350, Australia.
[3] Corresponding author, Email: trancong@usq.edu.au.

ods commonly employed to analyse those problems. In FEM, when solving structural problems with large deformation, element distortions happen, causing a deterioration of accuracy, thus requiring re-generation of the computational mesh to maintain accuracy. The FEM, FDM and FVM have difficulties in handling fluid-flow problems with free surface and moving boundary conditions. In the past decade, meshfree methods have become a very interesting research topics in computational mechanics because they possess a number of attractive properties. One of their most popular characteristics is that they require a set of nodes rather than a topological mesh to discretise the computational domain, thus computational cost associated with discretisation is highly reduced.

Two-dimensional (2D) incompressible Navier-Stokes flows have been extensively studied to verify new numerical methods. The main issues for a successful numerical solver for this kind of problems are the proper treatments of the nonlinear convection term and incompressibility. For the first issue, the presence of the convection term causes serious numerical difficulties in the form of oscillatory solutions or numerical divergence when Reynolds (Re) number or Peclet (Pe) number is high. To deal with this, schemes related to upwinding have been developed to stabilize the FDM, FEM, and FVM [1, 2, 3]. Brooks and Hughes [3] developed a Streamline Upwind/Petrov-Galerkin (SUPG) method for convection-dominated flows, which has the robustness of an upwind method and the accuracy associated with the wiggle-free Galerkin solutions. In their method, an additional stability term was added in the upwind direction and several different treatments of incompressibility are incorporated into the formulation. The upwind concept is also needed in the meshfree methods in order to obtain a good accuracy for convection-dominated flows. Lin and Atluri [4] proposed the meshless local Petrov-Galerkin (MLPG) method with two upwinding schemes for solving convection-diffusion problems. They skewed the weight function opposite to the streamline direction in the first scheme and shifted the local subdomain opposite to the streamline direction in the second scheme. Their numerical results indicated that the MLPG with the second scheme yielded better solutions than SUPG. This method was extended to solve the incompressible Navier-Stokes equations in [5].

For the second issue, i.e. treatment for incompressibility, incompressible flows can be solved through the stream function and vorticity formulation. This approach can satisfy the incompressibility condition automatically, and the pressure term is eliminated. However, this formulation experiences other type of difficulty arising from the computation of the vorticity boundary condition on the wall, especially the curved ones. For three-dimensional problems, the incompressible Navier-Stokes equations are usually based on primitive variables (pressure and velocity) as the stream function and vorticity formulation are not applicable. In order to

impose the incompressibility constraint, mixed formulations are considered by introducing another variable, the Lagrange multiplier. There are so-called inf-sup (or Ladyzenskaya-Babuŝka-Brezzi) stability conditions for this kind of formulations [6]. If these conditions are not satisfied, spurious pressure solutions may be obtained.

In 1990, Kansa proposed a collocation scheme based on multiquadric (MQ) radial basis functions for the numerical solution of partial differential equations (PDEs) [7]. Their numerical results showed that MQ scheme yielded an excellent interpolation and partial derivative estimates for a variety of two-dimensional functions over both gridded and scattered data. Since this original work, a number of meshfree methods have been developed and used to solve fluid-flow problems. Park and Youn [8] proposed the first-order least-squares method (LSMFM) to solve Laplace equations. Unlike the Galerkin method, the least-square formulation did not make use of the divergence theorem to convert the domain integral into a boundary integral. Therefore, the solution accuracy is less sensitive to the integration accuracy. However, the first-order least squares formulation requires more unknowns than the Galerkin formulation since the dual variables are employed as unknowns in addition to the primary variables, thereby increasing the computational cost. Zhang et al. [9] employed the LSMFM based on the first-order velocity-pressure-vorticity formulation to investigate the 2D steady incompressible viscous flow problems. Their numerical results showed that the least-squares method based on the minimization of the squared residuals can reduce oscillations and instability of the solutions in comparison with the behaviour of methods based on Galerkin formulation. In their approach, the penalty method was used to enforce the essential boundary conditions. It is well-known that the larger the penalty parameter, the more accurate the numerical solution will be, but large penalty parameters can affect the conditioning of the system matrix adversely [10]. Arzani and Afshar [11] developed discrete least-squares meshless (DLSM) method for the solution of convection-dominated problems. A fractional step method in conjunction with DLSM method was proposed to solve the steady-state incompressible Navier-Stokes equations in primitive form using large time steps without having to satisfy the inf-sup condition [12].

In contrast to the advantages of no mesh generation, most of the meshfree methods have difficulty in simulating large scale problems, because they produce very dense system matrices. Lee et al. [13] proposed the local multiquadric (LMQ) and the local inverse multiquadric (LIMQ) approximations for solving partial differential equations (PDEs). Their constructed shape functions strictly satisfied the Kronecker-δ condition which allows an imposition of the essential boundary condition in the same manner as in the standard FEM. Their numerical results showed that the LMQ and LIMQ often outperform their global counterparts, particularly

with regard to viability and stability. Šarler and Vertnik [14] presented an explicit local radial basis function (RBF) collocation method for diffusion problems. The method appeared efficient, because it does not require a solution of a large system of equations like the original RBF collocation method [7]. Babuŝka and Melenk [15] presented the partition of unity method (PUM) with attractive features. In the PUM, if analytic knowledge about the local behaviour of the problem solution is known, local approximation can be done with functions better suited than polynomials as in the classical FEM. The PU framework also provides a powerful approach to model mechanical problems with discontinuities and singularities. Krysl and Belytschko [16] proposed an approach to construct linear approximation basis functions for meshless method based on the concept of PU. In their work, the Shepard basis [17] is used as a PU function. The PUM was also employed by Chen et al. [18] to combine the reproducing kernel and RBF approximations in an approach that enjoys the exponential convergence of RBF and yields a banded and better-conditioned discrete system matrix. Le et al. [19] proposed a locally supported moving IRBFN-based meshless method for solving various problems including heat transfer, elasticity of both compressible and incompressible materials, and linear static crack problems.

In the past, lid-driven cavity flow and flow past a circular cylinder have been studied as benchmark problems by many researchers to verify their new numerical methods. In the first problem, the presence of singularities at two of the corners of the cavity, where the velocity is discontinuous, makes it difficult to predict the numerical results accurately. Ghia et al. [1] presented a FDM with a coupled strongly implicit multigrid method to obtain high-*Re* fine-mesh flow solutions. Botella and Peyret [20] introduced a third-order time-accurate Chebyshev projection method with an analytical treatment of the singularities for the lid-driven cavity flow. Their numerical results are widely considered as benchmark solutions in the literature. In the second problem, it is well-known that the flow has a stable pattern with a fixed pair of symmetric vortices behind the cylinder at *Re* up to 40. Ding et al. [21] presented a hybrid approach, which combines the conventional FDM and the meshfree least square-based finite difference (MLSFD) method for simulating the 2D steady and unsteady incompressible flows. In their works, the MLSFD method was adopted to deal with the spatial discretisation in the region with complex geometry and the conventional FDM was applied in the rest of the flow domain to take advantage of its high computational efficiency. Kim et al. [22] developed a meshfree point collocation method for the stream function-vorticity formulation of 2D incompressible Navier-Stokes equations. The MLS approximation was employed to construct shape functions in conjunction with a point collocation technique.

A one-dimensional integrated radial basis function networks (1D-IRBFN) colloca-

tion method for the solution of second- and fourth-order PDEs was presented by Mai-Duy and Tanner [23]. In this method, Cartesian grids were used to discretise both rectangular and non-rectangular problem domains. The computational cost associated with the Cartesian grid generation is negligible in comparison with that required for the body-fitted mesh. Along a grid line, IRBFNs are employed to represent the field variable and its relevant derivatives. Such networks are called 1D-IRBFNs. Through integration constants, one can impose derivative boundary conditions and the governing equations at the two end points of a grid line in an exact manner. The 1D-IRBFN method is much more efficient than the original IRBFN method reported in [24]. Ngo-Cong et al. [25] extended this method to investigate free vibration of composite laminated plates based on first-order shear deformation theory. The present work is concerned with the development of a new numerical method to handle 2D incompressible viscous flows at a high *Re* number and in large scale problems. The proposed method is based on the PU concept acting as a framework to incorporate MLS and 1D-IRBFN techniques, and from here on is named LMLS-1D-IRBFN, which is a local MLS-1D-IRBFN method. The approximation is locally supported, which leads to sparse system matrices and requires less computational effort than the case of using 1D-IRBFN method alone, while the order of accuracy remains high as in the case of 1D-IRBFN. Unlike conventional MLS-based methods, the LMLS-1D-IRBFN shape functions satisfy the Kronecker-δ property and thus the essential boundary conditions can be imposed in an exact manner.

The paper is organised as follows. Section 2 describes the notations. Section 3 briefly reproduces the MLS approximation technique. The LMLS-1D-IRBFN method is presented in Section 4. The governing equations for incompressible viscous flows are given in Section 5. The LMLS-1D-IRBFN discretisation of the governing equations is described in Section 6. Several numerical examples are investigated using the proposed method in Section 7. Section 8 concludes the paper.

2 Notations

In the remainder of the article, we use

- the notation $\overline{[\,]}$ for a vector/matrix $[\,]$ that is associated with a segment of a grid line;
- the notation $\widehat{[\,]}$ for a vector/matrix $[\,]$ that is associated with a grid line;
- the notation $\widetilde{[\,]}$ for a vector/matrix $[\,]$ that is associated with the whole set of grid lines;

- the notation $[\,]_{(\eta,\theta)}$ to denote selected rows η and columns θ of the matrix $[\,]$;
- the notation $[\,]_{(\eta)}$ to denote selected components η of the vector $[\,]$;
- the notation $[\,]_{(:,\theta)}$ to denote all rows and selected columns θ of the matrix $[\,]$; and
- the notation $[\,]_{(\eta,:)}$ to denote all columns and selected rows η of the matrix $[\,]$.

3 Moving least square approximation

The moving least square procedure [26] is briefly described in this section. The domain of interest is discretised using a Cartesian grid as shown in Fig 1. On an x-grid line, e.g. $[l]$, consider a nodal point x_i with its associated support domain, e.g. $[x_{i-1}, x_{i+1}]$ for the case of 3-node local support. Let $u^h(x)$ be the approximation of the field variable u along this support domain and given by

$$u^h(x) = \sum_{j=0}^{m} p_j(x) a_j(x) = \bar{p}^T(x)\bar{a}(x), \tag{1}$$

where m is the number of terms of monomials, $\bar{a}(x)$ a vector of coefficients and $\bar{p}^T(x)$ a complete polynomial basis, given by

$$\bar{a}(x) = \begin{pmatrix} a_0(x) & a_1(x) & \ldots & a_m(x) \end{pmatrix}^T, \tag{2}$$

$$\bar{p}(x) = \begin{pmatrix} p_0(x) & p_1(x) & \ldots & p_m(x) \end{pmatrix}^T = \begin{pmatrix} 1 & x & x^2 & \ldots & x^m \end{pmatrix}^T. \tag{3}$$

The expression for $\bar{a}(x)$ can be obtained at each point x by minimizing the following weighted residual

$$J = \sum_{I=1}^{n} W(x - x_I) \left[\bar{p}^T(x_I)\bar{a}(x) - u^{(I)} \right]^2, \tag{4}$$

where $u^{(I)}$ is the nodal value of the field variable u at $x = x_I$, and n the number of nodes in the support domain of x where the weight function $W(x - x_I) \neq 0$. In the present paper, the cubic spline weight function is used to construct MLS shape functions.

$$W(d) = \begin{cases} \frac{2}{3} - 4d^2 + 4d^3, & d \leq \frac{1}{2} \\ \frac{4}{3} - 4d + 4d^2 - \frac{4}{3}d, & \frac{1}{2} < d \leq 1 \\ 0, & d > 1 \end{cases} \tag{5}$$

where $d = |x - x_I| / d_w$ and d_w defines the size of the support domain. The minimization of the weighted residual J results in the following linear equation system

$$\mathbf{A}(x)\bar{a}(x) = \mathbf{B}(x)\bar{u}, \tag{6}$$

or

$$\bar{a}(x) = \mathbf{A}(x)^{-1}\mathbf{B}(x)\bar{u}, \tag{7}$$

where

$$\bar{u} = \begin{pmatrix} u^{(1)} & u^{(2)} & \dots & u^{(n)} \end{pmatrix}^T, \tag{8}$$

$$\mathbf{A}(x) = \sum_{I=1}^{n} W(x - x_I)\bar{p}(x_I)\bar{p}^T(x_I), \tag{9}$$

$$\mathbf{B}(x) = \begin{bmatrix} B_1 & B_2 & \dots & B_n \end{bmatrix}, \tag{10}$$

in which

$$B_I = W(x - x_I)\bar{p}(x_I). \tag{11}$$

Substituting (7) into (1), u^h can be expressed as

$$u^h(x) = \bar{\phi}^T(x)\bar{u}, \tag{12}$$

where $\bar{\phi}$ is the vector of MLS shape functions and given by

$$\bar{\phi}(x) = \begin{pmatrix} \bar{p}^T\mathbf{A}^{-1}B_1 & \bar{p}^T\mathbf{A}^{-1}B_2 & \dots & \bar{p}^T\mathbf{A}^{-1}B_n \end{pmatrix}^T. \tag{13}$$

It should be noted that the MLS shape functions do not satisfy the Kronecker-δ criterion, but possess a so-called partition of unity properties as follows.

$$\sum_{I=1}^{n} \bar{\phi}_I(x) = 1. \tag{14}$$

A new shape function possessing the Kronecker-δ function properties is created through a technique as described in the following section.

4 Moving least square - one dimensional integrated radial basis function networks technique

A schematic outline of the LMLS-1D-IRBFN method is depicted in Figure 2. The proposed method with 3-node support domains ($n = 3$) and 5-node local 1D-IRBF

networks ($n_s = 5$) is presented here. On an x-grid line $[l]$, a global interpolant for the field variable at a grid point x_i is sought in the form

$$u(x_i) = \sum_{j=1}^{n} \bar{\phi}_j(x_i) u^{[j]}(x_i), \tag{15}$$

where $\left\{\bar{\phi}_j\right\}_{j=1}^{n}$ is a set of the partition of unity functions constructed using MLS approximants, $u^{[j]}(x_i)$ is the nodal function value obtained from a local interpolant represented by a 1D-IRBF network $[j]$, n is the number of nodes in the support domain of x_i. In (15), MLS approximants are presently based on linear polynomials, which are defined in terms of 1 and x. Relevant derivatives of u at x_i can be obtained by differentiating (15)

$$\frac{\partial u(x_i)}{\partial x} = \sum_{j=1}^{n} \left(\frac{\partial \bar{\phi}_j(x_i)}{\partial x} u^{[j]}(x_i) + \bar{\phi}_j(x_i) \frac{\partial u^{[j]}(x_i)}{\partial x} \right), \tag{16}$$

$$\frac{\partial^2 u(x_i)}{\partial x^2} = \sum_{j=1}^{n} \left(\frac{\partial^2 \bar{\phi}_j(x_i)}{\partial x^2} u^{[j]}(x_i) + 2\frac{\partial \bar{\phi}_j(x_i)}{\partial x} \frac{\partial u^{[j]}(x_i)}{\partial x} + \bar{\phi}_j(x_i) \frac{\partial^2 u^{[j]}(x_i)}{\partial x^2} \right), \tag{17}$$

where the values $u^{[j]}(x_i), \partial u^{[j]}(x_i)/\partial x$ and $\partial^2 u^{[j]}(x_i)/\partial x^2$ are calculated from 1D-IRBFN networks with n_s nodes.

4.1 One-dimensional IRBFN

Consider a segment $[j]$ with n_s nodes on an x-grid line $[l]$ as shown in Figure 2. The variation of the nodal function $u^{[j]}$ along this segment is sought in the IRBF form. The second-order derivative of $u^{[j]}$ is decomposed into RBFs; the RBF network is then integrated once and twice to obtain the expressions for the first-order derivative of $u^{[j]}$ and the function $u^{[j]}$ itself as follows.

$$\frac{\partial^2 u^{[j]}(x)}{\partial x^2} = \sum_{k=1}^{n_s} w^{(k)} G^{(k)}(x) = \sum_{k=1}^{n_s} w^{(k)} H_{[2]}^{(k)}(x), \tag{18}$$

$$\frac{\partial u^{[j]}(x)}{\partial x} = \sum_{k=1}^{n_s} w^{(k)} H_{[1]}^{(k)}(x) + c_1, \tag{19}$$

$$u^{[j]}(x) = \sum_{k=1}^{n_s} w^{(k)} H_{[0]}^{(k)}(x) + c_1 x + c_2, \tag{20}$$

where $\left\{w^{(k)}\right\}_{k=1}^{n_s}$ are RBF weights to be determined; $\left\{G^{(k)}(x)\right\}_{k=1}^{n_s} = \left\{H_{[2]}^{(k)}(x)\right\}_{k=1}^{n_s}$ known RBFs; $H_{[1]}^{(k)}(x) = \int H_{[2]}^{(k)}(x)dx$; $H_{[0]}^{(k)}(x) = \int H_{[1]}^{(k)}(x)dx$; and c_1 and c_2 integration constants which are also unknown. An example of RBF, used in this work, is

the multiquadrics $G^{(k)}(x) = \sqrt{(x-x^{(k)})^2 + a^{(k)2}}$, $a^{(k)}$ - the RBF width determined as $a^{(k)} = \beta d^{(k)}$, β a positive factor, and $d^{(k)}$ the distance from the k^{th} center to its nearest neighbour.

It is more convenient to work in the physical space than in the network-weight space. The RBF coefficients including two integration constants can be transformed into the physically meaningful nodal variable values through the following relation

$$\vec{u}^{[j]} = \bar{\mathbf{H}} \begin{pmatrix} \bar{w} \\ \bar{c} \end{pmatrix}, \tag{21}$$

where $\bar{\mathbf{H}}$ is an $n_s \times (n_s+2)$ matrix and given by

$$\mathbf{H} = \begin{bmatrix} H_{[0]}^{(1)}(x_1) & H_{[0]}^{(2)}(x_1) & ... & H_{[0]}^{(n_s)}(x_1) & x_1 & 1 \\ H_{[0]}^{(1)}(x_2) & H_{[0]}^{(2)}(x_2) & ... & H_{[0]}^{(n_s)}(x_2) & x_2 & 1 \\ ... & ... & ... & ... & ... & ... \\ H_{[0]}^{(1)}(x_{n_s}) & H_{[0]}^{(2)}(x_{n_s}) & ... & H_{[0]}^{(n_s)}(x_{n_s}) & x_{n_s} & 1 \end{bmatrix}; \tag{22}$$

$\vec{u}^{[j]} = (u^{(1)}, u^{(2)}, ..., u^{(n_s)})^T$; $\bar{w} = (w^{(1)}, w^{(2)}, ..., w^{(n_s)})^T$ and $\bar{c} = (c_1, c_2)^T$. There are two possible transformation cases.

For a segment $[j]$ *with only interior points:* The direct use of (21) leads to an underdetermined system of equations

$$\vec{u}^{[j]} = \bar{\mathbf{H}} \begin{pmatrix} \bar{w} \\ \bar{c} \end{pmatrix} = \bar{\mathbf{C}} \begin{pmatrix} \bar{w} \\ \bar{c} \end{pmatrix}, \tag{23}$$

or

$$\begin{pmatrix} \bar{w} \\ \bar{c} \end{pmatrix} = \bar{\mathbf{C}}^{-1} \vec{u}^{[j]}, \tag{24}$$

where $\bar{\mathbf{C}} = \bar{\mathbf{H}}$ is the conversion matrix whose inverse can be found using the singular value decomposition (SVD) technique.

For a segment $[j]$ *with interior and boundary points:* Owing to the presence of c_1 and c_2, one can add an additional equation of the form

$$f = \mathbf{K} \begin{pmatrix} \bar{w} \\ \bar{c} \end{pmatrix} \tag{25}$$

to equation system (21). In the case of Neumann boundary conditions, this subsystem can be used to impose a derivative boundary value at $x = x_b$

$$f = \frac{\partial u(x_b)}{\partial x}, \tag{26}$$

$$\mathbf{K} = \begin{bmatrix} H_{[1]}^{(1)}(x_b) & H_{[1]}^{(2)}(x_b) & ... & H_{[1]}^{(n_s)}(x_b) & 1 & 0 \end{bmatrix}. \tag{27}$$

The conversion system can be written as

$$\begin{pmatrix} \bar{u}^{[j]} \\ f \end{pmatrix} = \begin{bmatrix} \bar{\mathbf{H}} \\ \mathbf{K} \end{bmatrix} \begin{pmatrix} \bar{w} \\ \bar{c} \end{pmatrix} = \bar{\mathbf{C}} \begin{pmatrix} \bar{w} \\ \bar{c} \end{pmatrix}, \tag{28}$$

or

$$\begin{pmatrix} \bar{w} \\ \bar{c} \end{pmatrix} = \bar{\mathbf{C}}^{-1} \begin{pmatrix} \bar{u}^{[j]} \\ f \end{pmatrix}. \tag{29}$$

It can be seen that (24) is a special case of (29), where f is simply set to null. By substituting Equation (29) into Equations (18)-(20), the second- and first-order derivatives and the function of the variable $u^{[j]}$ are expressed in terms of nodal variable values as

$$\frac{\partial^2 u^{[j]}(x)}{\partial x^2} = \left(H_{[2]}^{(1)}(x), H_{[2]}^{(2)}(x), ..., H_{[2]}^{(n_s)}(x), 0, 0\right) \bar{\mathbf{C}}^{-1} \begin{pmatrix} \bar{u}^{[j]} \\ f \end{pmatrix}, \tag{30}$$

$$\frac{\partial u^{[j]}(x)}{\partial x} = \left(H_{[1]}^{(1)}(x), H_{[1]}^{(2)}(x), ..., H_{[1]}^{(n_s)}(x), 1, 0\right) \bar{\mathbf{C}}^{-1} \begin{pmatrix} \bar{u}^{[j]} \\ f \end{pmatrix}, \tag{31}$$

$$u^{[j]}(x) = \left(H_{[0]}^{(1)}(x), H_{[0]}^{(2)}(x), ..., H_{[0]}^{(n_s)}(x), x, 1\right) \bar{\mathbf{C}}^{-1} \begin{pmatrix} \bar{u}^{[j]} \\ f \end{pmatrix}, \tag{32}$$

or

$$\frac{\partial^2 u^{[j]}(x)}{\partial x^2} = \bar{d}_{2x}^T \bar{u}^{[j]} + k_{2x}(x), \tag{33}$$

$$\frac{\partial u^{[j]}(x)}{\partial x} = \bar{d}_{1x}^T \bar{u}^{[j]} + k_{1x}(x), \tag{34}$$

$$u^{[j]}(x) = \bar{d}_{0x}^T \bar{u}^{[j]} + k_{0x}(x), \tag{35}$$

where k_{0x}, k_{1x} and k_{2x} are scalars whose values depend on x and a boundary value f; and d_{0x}, d_{1x} and d_{2x} are known vectors of length n_s.

By application of Equations (33) and (34) to n_s nodes on the segment $[j]$, the second- and first-order derivatives of $u^{[j]}$ at node x_i are determined as

$$\frac{\partial^2 u^{[j]}(x_i)}{\partial x^2} = \bar{\mathbf{D}}_{2x(idk,:)} \bar{u}^{[j]} + \bar{k}_{2x(idk)}, \tag{36}$$

$$\frac{\partial u^{[j]}(x_i)}{\partial x} = \bar{\mathbf{D}}_{1x(idk,:)} \bar{u}^{[j]} + \bar{k}_{1x(idk)}, \tag{37}$$

$$u^{[j]}(x_i) = \bar{\mathbf{D}}_{0x(idk,:)} \bar{u}^{[j]} + \bar{k}_{0x(idk)} = \bar{\mathbf{I}}_{(idk,:)} \bar{u}^{[j]}, \tag{38}$$

where $\bar{\mathbf{D}}_{1x}$ and $\bar{\mathbf{D}}_{2x}$ are known matrices of dimension $n_s \times n_s$; $\bar{k}_{1x}$ and $\bar{k}_{2x}$ are known vectors of length n_s; and idk is the index number indicating the location of node x_i

in the local network $[j]$. It is noted that $\bar{\mathbf{D}}_{0x} = \bar{\mathbf{I}}$, where $\bar{\mathbf{I}}$ is an identity matrix of dimension $n_s \times n_s$ and $k_{0x} = 0$. Therefore, the 1D-IRBFN shape function possesses the Kronecker-δ function properties.

4.2 Incorporation of MLS and 1D-IRBFN into the partition of unity framework

By substituting Equations (36)-(38) into Equations (15)-(17), the function $u(x_i)$ and its derivatives are expressed as

$$u(x_i) = \sum_{j=1}^{n} \bar{m}_{0x}^{[j]} \bar{u}^{[j]}, \tag{39}$$

$$\frac{\partial u(x_i)}{\partial x} = \sum_{j=1}^{n} \left(\bar{m}_{1x}^{[j]} \bar{u}^{[j]} + k_{1x}^{[j]} \right), \tag{40}$$

$$\frac{\partial^2 u(x_i)}{\partial x^2} = \sum_{j=1}^{n} \left(\bar{m}_{2x}^{[j]} \bar{u}^{[j]} + k_{2x}^{[j]} \right), \tag{41}$$

where

$$\bar{m}_{0x}^{[j]} = \bar{\phi}_j(x_i) \bar{\mathbf{I}}_{(idk,:)}, \tag{42}$$

$$\bar{m}_{1x}^{[j]} = \frac{\partial \bar{\phi}_j(x_i)}{\partial x} \bar{\mathbf{I}}_{(idk,:)} + \bar{\phi}_j(x_i) \bar{\mathbf{D}}_{1x(idk,:)}, \tag{43}$$

$$\bar{m}_{2x}^{[j]} = \frac{\partial^2 \bar{\phi}_j(x_i)}{\partial x^2} \bar{\mathbf{I}}_{(idk,:)} + 2 \frac{\partial \bar{\phi}_j(x_i)}{\partial x} \bar{\mathbf{D}}_{1x(idk,:)} + \bar{\phi}_j(x_i) \bar{\mathbf{D}}_{2x(idk,:)}, \tag{44}$$

$$k_{1x}^{[j]} = \bar{\phi}_j(x_i) \bar{k}_{1x(idk)}, \tag{45}$$

$$k_{2x}^{[j]} = 2 \frac{\partial \bar{\phi}_j(x_i)}{\partial x} \bar{k}_{1x(idk)} + \bar{\phi}_j(x_i) \bar{k}_{2x(idk)}. \tag{46}$$

From Equations (14), (39) and (42), one can see that the LMLS-1D-IRBFN shape function possesses the Kronecker-δ function properties.

Equations (40) and (41) can be expressed as

$$\frac{\partial u(x_i)}{\partial x} = \bar{m}_{1x}^{[i]} \bar{u}^{[i]} + k_{1x}^{[i]}, \tag{47}$$

$$\frac{\partial^2 u(x_i)}{\partial x^2} = \bar{m}_{2x}^{[i]} \bar{u}^{[i]} + k_{2x}^{[i]}, \tag{48}$$

where $\bar{u}^{[i]} = \left(u^{(1)}, u^{(2)}, ..., u^{(n_r)} \right)^T$, n_r the number of nodes in the network $[i]$, $k_{1x}^{[i]}$ and $k_{2x}^{[i]}$ are known scalars, and $\bar{m}_{1x}^{[i]}$ and $\bar{m}_{2x}^{[i]}$ are known vectors of length n_r, defined by

$$\bar{m}_{1x(idj)}^{[i]} = \bar{m}_{1x(idj)}^{[i]} + \bar{m}_{1x}^{[j]}, \qquad j = 1, 2, ..., n \tag{49}$$

$$\bar{m}_{2x(idj)}^{[i]} = \bar{m}_{2x(idj)}^{[i]} + \bar{m}_{2x}^{[j]}, \qquad j = 1, 2, ..., n \tag{50}$$

in which idj is the index vector mapping the location of nodes of the local network $[j]$ to that in the LMLS-1D-IRBF network $[i]$.

The values of first- and second-order derivatives of u with respect to x at the nodal points on the grid line $[l]$ are given by

$$\frac{\partial \hat{u}}{\partial x} = \hat{\mathbf{M}}_{1x}^{[l]} \hat{u}^{[l]} + \hat{k}_{1x}^{[l]}, \tag{51}$$

$$\frac{\partial^2 \hat{u}}{\partial x^2} = \hat{\mathbf{M}}_{2x}^{[l]} \hat{u}^{[l]} + \hat{k}_{2x}^{[l]}, \tag{52}$$

where

$$\hat{u} = \left(u^{(1)}, u^{(2)}, ..., u^{(n_l)}\right)^T, \tag{53}$$

$$\hat{\mathbf{M}}_{1x(i,idi)}^{[l]} = \bar{m}_{1x}^{[i]}, \tag{54}$$

$$\hat{\mathbf{M}}_{2x(i,idi)}^{[l]} = \bar{m}_{2x}^{[i]}, \tag{55}$$

$$\hat{k}_{1x(i)}^{[l]} = k_{1x}^{[i]}, \tag{56}$$

$$\hat{k}_{2x(i)}^{[l]} = k_{2x}^{[i]}, \tag{57}$$

in which n_l is the number of nodes on the grid line $[l]$, and idi the index vector mapping the location of nodes of the local network $[i]$ to that in the grid line $[l]$.

The values of first- and second-order derivatives of u with respect to x at the nodal points over the problem domain are given by

$$\frac{\partial \tilde{u}}{\partial x} = \tilde{\mathbf{M}}_{1x} \tilde{u} + \tilde{k}_{1x}, \tag{58}$$

$$\frac{\partial^2 \tilde{u}}{\partial x^2} = \tilde{\mathbf{M}}_{2x} \tilde{u} + \tilde{k}_{2x}, \tag{59}$$

where

$$\tilde{u} = \left(u^{(1)}, u^{(2)}, ..., u^{(N_{ip})}\right)^T, \tag{60}$$

$$\frac{\partial \tilde{u}}{\partial x} = \left(\frac{\partial u^{(1)}}{\partial x}, \frac{\partial u^{(2)}}{\partial x}, ..., \frac{\partial u^{(N_{ip})}}{\partial x}\right)^T, \tag{61}$$

$$\frac{\partial^2 \tilde{u}}{\partial x^2} = \left(\frac{\partial^2 u^{(1)}}{\partial x^2}, \frac{\partial^2 u^{(2)}}{\partial x^2}, ..., \frac{\partial^2 u^{(N_{ip})}}{\partial x^2}\right)^T, \tag{62}$$

and $\tilde{\mathbf{M}}_{1x}$ and $\tilde{\mathbf{M}}_{2x}$ are known matrices of dimension $N_{ip} \times N_{ip}$; $\tilde{k}_{1x}$ and $\tilde{k}_{2x}$ are known vectors of length N_{ip}; and N_{ip} is the total number of interior nodal points. The

matrices $\tilde{\mathbf{M}}_{1x}$ and $\tilde{\mathbf{M}}_{2x}$ and the vectors $\tilde{k}_{1x}$ and $\tilde{k}_{2x}$ are formed as follows.

$$\tilde{\mathbf{M}}_{1x(idl,idl)} = \hat{\mathbf{M}}_{1x}^{[l]}, \tag{63}$$

$$\tilde{\mathbf{M}}_{2x(idl,idl)} = \hat{\mathbf{M}}_{2x}^{[l]}, \tag{64}$$

$$\tilde{k}_{1x(idl)} = \hat{k}_{1x}^{[l]}, \tag{65}$$

$$\tilde{k}_{2x(idl)} = \hat{k}_{2x}^{[l]}, \tag{66}$$

in which idl is the index vector mapping the location of nodes on the grid line $[l]$ to that in the whole grid.

Similarly, the values of the second- and first-order derivatives of u with respect to y at the nodal points over the problem domain are given by

$$\frac{\partial \tilde{u}}{\partial y} = \tilde{\mathbf{M}}_{1y}\tilde{u} + \tilde{k}_{1y}, \tag{67}$$

$$\frac{\partial^2 \tilde{u}}{\partial y^2} = \tilde{\mathbf{M}}_{2y}\tilde{u} + \tilde{k}_{2y}. \tag{68}$$

5 Governing equations for two-dimensional incompressible viscous flows

In this work we limit the analysis to two-dimensional problems and the governing equations for incompressible viscous flows can therefore be written in terms of stream function ψ and vorticity ω as [27]

$$\frac{\partial^2 \psi}{\partial x^2} + \frac{\partial^2 \psi}{\partial y^2} = -\omega, \tag{69}$$

$$\frac{1}{Re}\left(\frac{\partial^2 \omega}{\partial x^2} + \frac{\partial^2 \omega}{\partial y^2}\right) = \frac{\partial \omega}{\partial t} + \left(\frac{\partial \psi}{\partial y}\frac{\partial \omega}{\partial x} - \frac{\partial \psi}{\partial x}\frac{\partial \omega}{\partial y}\right), \tag{70}$$

where Re is the Reynolds number, t the time, and $(x,y)^T$ the position vector. The x and y components of the velocity vector can be defined in terms of the stream function as

$$u = \frac{\partial \psi}{\partial y}, \tag{71}$$

$$v = -\frac{\partial \psi}{\partial x}. \tag{72}$$

6 LMLS-1D-IRBFN discretisation of governing equations for incompressible viscous flows

The domain of interest is discretised using uniform Cartesian grids. With the backward Euler scheme for time discretisation, Equations (69) and (70) can be expressed as

$$\frac{\partial^2 \psi^{(n+1)}}{\partial x^2} + \frac{\partial^2 \psi^{(n+1)}}{\partial y^2} = -\omega^{(n)}, \tag{73}$$

$$\frac{\Delta t}{Re}\left(\frac{\partial^2 \omega^{(n+1)}}{\partial x^2} + \frac{\partial^2 \omega^{(n+1)}}{\partial y^2}\right) - \omega^{(n+1)} = -\omega^{(n)} + \Delta t\left(\frac{\partial \psi^{(n)}}{\partial y}\frac{\partial \omega^{(n)}}{\partial x} - \frac{\partial \psi^{(n)}}{\partial x}\frac{\partial \omega^{(n)}}{\partial y}\right), \tag{74}$$

where the superscripts (n) and $(n+1)$ denote the time levels and Δt the time discretisation step.

Making use of (58), (59), (67) and (68) and collocating the governing equations (73) and (74) at the interior points result in

$$\tilde{E}_1 \tilde{\psi}^{(n+1)} = RHS_1, \tag{75}$$

$$\tilde{E}_2 \tilde{\omega}^{(n+1)} = RHS_2, \tag{76}$$

where

$$\tilde{E}_1 = \tilde{\mathbf{M}}_{2x} + \tilde{\mathbf{M}}_{2y}, \tag{77}$$

$$RHS_1 = -\omega^{(n)} - \left(\tilde{k}_{2x\psi} + \tilde{k}_{2y\psi}\right), \tag{78}$$

$$\tilde{E}_2 = \frac{\Delta t}{Re}\left(\tilde{\mathbf{M}}_{2x} + \tilde{\mathbf{M}}_{2y} - \tilde{\mathbf{I}}\right), \tag{79}$$

$$\begin{aligned} RHS_2 &= -\omega^{(n)} - \tfrac{\Delta t}{Re}\left(\tilde{k}_{2x\omega} + \tilde{k}_{2y\omega}\right) \\ &+\Delta t\left[\left(\tilde{\mathbf{M}}_{1y}\tilde{\psi}^{(n)} + \tilde{k}^{(n)}_{1y\psi}\right).\left(\tilde{\mathbf{M}}_{1x}\tilde{\omega}^{(n)} + \tilde{k}^{(n)}_{1x\omega}\right) - \left(\tilde{\mathbf{M}}_{1x}\tilde{\psi}^{(n)} + \tilde{k}^{(n)}_{1x\psi}\right).\left(\tilde{\mathbf{M}}_{1y}\tilde{\omega}^{(n)} + \tilde{k}^{(n)}_{1y\omega}\right)\right], \end{aligned} \tag{80}$$

in which $\tilde{k}_{1x\psi}, \tilde{k}_{2x\psi}, \tilde{k}_{1y\psi}, \tilde{k}_{2y\psi}, \tilde{k}_{1x\omega}, \tilde{k}_{2x\omega}, \tilde{k}_{1y\omega}$ and $\tilde{k}_{2y\omega}$ are known vectors of length N_{ip}.

The nonlinear system of equations (75) and (76) is solved using the pseudo-time stepping procedure as follows:

- Step 1: Guess the initial solution of vorticity ω.
- Step 2: Solve (75) for ψ.

- Step 3: Compute the vorticity boundary conditions and the convection terms explicitly.
- Step 4: Solve (76) for ω.
- Step 5: Check convergence criterion for ω

$$\frac{\sqrt{\sum_{i=1}^{Nip}\left(\omega_i^{(t+1)}-\omega_i^{(t)}\right)^2}}{\sqrt{\sum_{i=1}^{Nip}\left(\omega_i^{(t+1)}\right)^2}} < TOL, \tag{81}$$

where TOL is a given tolerance and presently set to be 10^{-12}. If not converged, return to step 2. Otherwise, stop.

7 Numerical results and discussion

Several examples are investigated here to study the performance of the present method. The domains of interest are discretised using Cartesian grids. By using the LMLS-1D-IRBFN method to discretise governing equations and the LU decomposition technique to solve the resultant sparse system of simultaneous equations, the computational cost and data storage requirements are reduced. For the purpose of CPU times comparisons, all related computations are carried out on a single 2.4 *GHz* processor machine with 4 *GB* RAM.

7.1 Example 1: Two-dimensional Poisson equation in a square domain

The present method is first verified through the solution of the following 2D Poisson equation

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}=0, \tag{82}$$

defined on a square domain $0 \le x,y \le 1$ and subject to Dirichlet boundary conditions. The problem has the following exact solution

$$u_E = \frac{1}{\sinh(\pi)}\sin(\pi x)\sinh(\pi y). \tag{83}$$

A uniform grid of $N_x \times N_y$ is employed to discretise the problem domain. Two cases of boundary conditions are considered as follows.

- Case 1: Dirichlet boundary conditions are imposed along all four edges.

- Case 2: Dirichlet boundary conditions are imposed along two horizonal edges and Neumann boundary conditions are imposed along two vertical edges.

These boundary conditions can be derived from the exact solution. The proposed LMLS-1D-IRBFN method with the following two approaches is considered.

- Approach 1: $n = 3$ and $n_s = 3$, called LMLS-1D-IRBFN-3-node.
- Approach 2: $n = 3$ and $n_s = 5$, called LMLS-1D-IRBFN-5-node.

Figure 3 presents the grid convergence study for Case 1 for the two approaches in comparison with those of FDM with central-difference scheme and the 1D-IRBFN method. The convergence study for Case 2 for the two approaches in comparison with those of the 1D-IRBFN method is shown in Figure 4. The convergence behaviours of FDM, 1D-IRBFN, Approach 1 and Approach 2 for Case 1 are O($h^{2.05}$), O($h^{3.16}$), O($h^{1.78}$) and O($h^{2.69}$), respectively. The convergence behaviour of 1D-IRBFN, Approach 1 and Approach 2 for Case 2 are O($h^{1.98}$), O($h^{1.84}$) and O($h^{1.89}$), respectively. The numerical results show that the LMLS-1D-IRBFN-5-node is much more accurate than FDM and LMLS-1D-IRBFN-3-node, and slightly better than those of its global counterpart, i.e. 1D-IRBFN method.

Table 1 presents the comparison of the number of nonzero elements per row of the system matrix (N_{nzpr}) and condition number (*cond*) among the FDM, two present approaches and the 1D-IRBFN for Case 1, while Table 2 shows the comparison of CPU time and percentage of nonzero elements of the system matrix $\varepsilon = (N_{nz}/N_{total}) \times 100$ (N_{nz} and N_{total}: the number of nonzero elements and the total number of elements of the system matrix, respectively) among these methods. The comparison of condition number for Case 2 is given in Table 3. The condition numbers of 1D-IRBFN, Approach 1 and Approach 2 are of the same order of magnitude and at most one order of magnitude larger than those of FDM. The number of nonzero elements per row of the system matrix N_{nzpr} of the FDM with central-difference scheme, LMLS-1D-IRBF-3-node and LMLS-1D-IRBF-5-node methods are 5, 9, and 13, respectively and less than that of the 1D-IRBFN method. Therefore, for a given grid size, the CPU time and memory requirements of Approach 2 are larger than those of Approach 1 and FDM, and significantly less than those of the 1D-IRBFN method. For example, for a grid of 121×121, the CPU time and the ε of Approach 2 are 38.7 times and 2.6 times larger than those of the FDM, respectively, and 89.6 times and 18.5 times less than those of the 1D-IRBFN method, respectively. It is noted that for a given grid size the present Approach 2 is slower than the FDM. However, the present Approach 2 achieves a given level of accuracy with a coarser grid and hence more efficient. For example, as shown

in Figure 3 and Table 2, the present Approach 2 with grid=21×21 yields better accuracy ($Ne = 6.88e-6$) in 0.88 seconds than the FDM with grid=121×121 ($Ne = 3.49e-5$) in 1.74 seconds.

Approach 2 yields much more accurate results than Approach 1 and FDM with central-difference scheme and is significantly more efficient than 1D-IRBFN in terms of computational cost, as grid density increases. Therefore, the remaining examples will be investigated using Approach 2, i.e. LMLS-1D-IRBFN-5-node.

7.2 Example 2: Two-dimensional Poisson equation in a square domain with a circular hole

This example is concerned with the following 2D Poisson equation

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = -8\pi^2 \sin(2\pi x)\sin(2\pi y), \tag{84}$$

defined on a square domain with a circular hole as shown in Figure 5 and subject to Dirichlet boundary conditions. The problem has the following exact solution

$$u_E = \sin(2\pi x)\sin(2\pi y), \tag{85}$$

from which the boundary values of u can be derived.

The grid convergence study for LMLS-1D-IRBFN and 1D-IRBFN methods is presented in Figure 6. Table 4 describes the relative error norms (Ne) and condition number ($cond$) of the present method in comparison with those of 1D-IRBFN method. The numerical results showed that the present method is not as accurate as the 1D-IRBFN method, but has a higher convergence rate (error norm of O($h^{3.70}$)) than the 1D-IRBFN method (error norm of O($h^{3.00}$)). Table 5 presents the comparison of CPU time and percentage of nonzero elements of the system matrix (ε) between the 1D-IRBFN and LMLS-1D-IRBFN methods. The present method is much more efficient than the 1D-IRBFN method in terms of CPU time (e.g. 101.3 times for a grid of 129×129) and memory requirements (e.g. 17.2 times for a grid of 129×129), thus the grid size can be refined to obtain more accurate solutions as shown in Figure 6.

In comparing the convergence behaviours in Example 1 (homogeneous Poisson equation on simply-connected domain) and Example 2 (non-homogeneous Poisson equation on multiply-connected domain), it is observed that the overall convergence rate of Approach 2 for the former is 2.69 and that for the latter is 3.70. At first glance, the results might seem strange. However, it is observed that to achieve similar accuracy (Ne of $O(10^{-5})$), the convergence rates are very similar, i.e. 3.72 for Example 1 and 3.70 for Example 2. In Example 1, the shape of solution is relatively simple and the method can achieve even higher accuracy (Ne of $O(10^{-7})$).

However, at this higher level of accuracy, the local convergence rate decreases, causing a lower overall convergence rate as described above.

7.3 Example 3: Lid-driven cavity flow

The cavity is taken to be a unit square with the lid sliding from left to right at a unit velocity as shown in Figure 7. The boundary conditions for stream function ψ are defined by

$$\psi = 0, \quad \text{on} \quad x = 0, x = 1, y = 0, y = 1, \tag{86}$$

$$\frac{\partial \psi}{\partial x} = 0, \ \text{on} \quad x = 0, x = 1, \tag{87}$$

$$\frac{\partial \psi}{\partial y} = 0, \ \text{on} \quad y = 0, \tag{88}$$

$$\frac{\partial \psi}{\partial y} = 1, \ \text{on} \quad y = 1. \tag{89}$$

It is noted that only the Dirichlet boundary conditions (86) are used for solving (69), while the Neumann boundary conditions (87)-(89) are used to derive computational vorticity boundary conditions for solving (70).

It is well-known that the major difficulties of lid-driven cavity flow simulation are: (i) the presence of singularities at two of the corners, which makes it difficult to predict the solution accurately; and (ii) the dominant convection terms, when dealing with high *Re*, which can cause oscillatory solutions if an improper scheme is used or computational grids are not sufficiently refined. The grid convergence study is first conducted for the lid-driven cavity flow problem with *Re* of 1000 using following two approaches.

- Approach 1: The convection terms are calculated using LMLS-1D-IRBFN technique.
- Approach 2: The convection terms are calculated using global 1D-IRBFN technique.

Table 6 shows the grid convergence study of the extrema of the horizontal and vertical velocity profiles along the center lines of the cavity for Approach 1 in comparison with FDMs [1, 28] and 1D-IRBFN [29]. The second-order accurate central finite-difference approximation was employed to approximate the linear terms in both FDMs mentioned above [1, 28], while the nonlinear convection terms were discretised by using a first-order accurate upwind difference scheme including its second-order accurate term as a deferred correction in FDM [1] and uncentered second-order differences in FDM [28]. In Table 6, the percentage errors

($\varepsilon = (V_m - V_s) \times 100/V_s$) of the extremal velocities (V_m) based on the corresponding spectral benchmark solutions (V_s) [20] are given. It can be seen that these errors reduce with increasing grid densities. The orders of convergence are 2.42, 2.61 and 2.92 for the minimum horizontal velocity u_{min}, the maximum vertical velocity v_{max} and the minimum vertical velocity v_{min} along the center lines, respectively. The present results for a grid of 101×101 are more accurate than those of FDMs with more refined grids [1, 28], but less than those of 1D-IRBFN [29]. Table 7 describes comparisons of the number of nonzero elements per row of the system matrix (N_{nzpr}), number of iterations ($N_{iteration}$) and total CPU time (T_{total}) required to obtain the converged solution with $TOL = 10^{-12}$. The time step Δt is set to be 5×10^{-3} for all cases. Note that for a given grid size the present approach is slower than the FDM. However, the present approach achieves a given level of accuracy with a coarser grid and hence more efficient. For example, as shown in Tables 6 and 7, the present approach with grid=81×81 yields better accuracy in 1559.77 seconds than the FDM with grid=129×129 in 1733.02 seconds.

The corresponding grid convergence study for Approach 2 is given in Table 8. The orders of convergence are 3.80, 3.26 and 4.26 for u_{min}, v_{max} and v_{min}, respectively. It is interesting to see that Approach 2 yields more accurate results than Approach 1 and the 1D-IRBFN method, and the convergence orders of Approach 2 are higher than those of Approach 1. Approach 2 is employed to study the cases with high Reynolds numbers ($Re = 3200$ and 7500). The contours of stream function and vorticity of the flow field inside the cavity at $Re = 1000, 3200$ and 7500 are shown in Figure 8. The vertical and horizontal velocities along the horizontal and vertical center lines at $Re = 1000, 3200$ and 7500 are given in Figure 9. These figures show that the current results are in good agreement with benchmark solutions of Ghia et al. [1] and Botella and Peyret [20].

7.4 Example 4: Flow past a circular cylinder

The steady flow past a circular cylinder at low Re numbers are considered in this section, where $Re = U_0 D/\nu$, U_0 is the far-field inlet velocity taken to be 1, D the diameter of the cylinder taken to be 1, ν the kinematic viscosity. The top, bottom, inlet and outlet boundaries are positioned at a distance of $20D, 20D, 10D$ and $30D$ away from the cylinder, respectively, as shown in Figure 10. These distances are large enough to assume that the far-field flow behaves as a potential flow [22] and the far-field stream function ψ^{far} can be defined by

$$\psi^{far} = U_0 y \left(1 - \frac{D^2}{4(x^2 + y^2)}\right). \tag{90}$$

The boundary conditions for stream function and vorticity are given by

$$\psi = \psi^{far},\ \omega = 0, \qquad \text{on } \Gamma_1, \Gamma_2, \Gamma_3, \tag{91}$$

$$\frac{\partial \psi}{\partial x} = 0,\ \frac{\partial \omega}{\partial x} = 0, \qquad \text{on } \Gamma_4, \tag{92}$$

$$\psi = 0, \frac{\partial \psi}{\partial n} = 0, \qquad \text{on } \Gamma_w. \tag{93}$$

where n is the direction normal to the cylinder surface as shown in Figure 11. The values of the vorticity on the circular boundary Γ_w can be computed as

$$\omega_w = -\left(\frac{\partial^2 \psi_w}{\partial x^2} + \frac{\partial^2 \psi_w}{\partial y^2}\right) \tag{94}$$

where the subscript w is used to denote quantities on the circular boundary. A formula of Le-Cao et al. [30] is employed here to derive the vorticity boundary conditions at boundary points on x- and y-grid lines as follows.

$$\omega_w^{(x)} = -\left[1 + \left(\frac{t_x}{t_y}\right)^2\right] \frac{\partial^2 \psi_w}{\partial x^2} - q_y, \tag{95}$$

$$\omega_w^{(y)} = -\left[1 + \left(\frac{t_y}{t_x}\right)^2\right] \frac{\partial^2 \psi_w}{\partial y^2} - q_x, \tag{96}$$

where q_x and q_y are known quantities defined by

$$q_x = -\frac{t_y}{t_x^2} \frac{\partial^2 \psi_w}{\partial y \partial s} + \frac{1}{t_x} \frac{\partial^2 \psi_w}{\partial x \partial s}, \tag{97}$$

$$q_y = -\frac{t_x}{t_y^2} \frac{\partial^2 \psi_w}{\partial x \partial s} + \frac{1}{t_y} \frac{\partial^2 \psi_w}{\partial y \partial s}, \tag{98}$$

in which $t_x = \partial x/\partial s, t_y = \partial y/\partial s$ and s is the direction tangential to the cylinder surface (Figure 11).

Calculation of drag and pressure coefficients

For viscous flow, the forces acting on the body come from two sources including pressure and friction. For the case of flow past a circular cylinder, the drag F_D and its coefficient C_D can be defined by

$$F_D = R \int_0^{2\pi} \left(\mu R \frac{\partial \omega}{\partial n} - \mu \omega\right) \sin\theta d\theta, \tag{99}$$

$$C_D = \frac{F_D}{\rho U_0^2 R}, \tag{100}$$

where R is the radius of the cylinder, ρ fluid density and μ the dynamic viscosity. The dimensionless pressure coefficient is given by

$$C_p(\theta) = \frac{p(\theta) - p_0}{1/2\rho U_0^2}, \tag{101}$$

where p_0 is the far-field inlet pressure, and $p(\theta)$ is the pressure on the cylinder surface at angle θ, evaluated as [31]

$$p(\theta) = (p_0 + 1/2\rho U_0^2) - \int_R^{d_0} \left(\frac{\mu}{r} \frac{\partial \omega}{\partial \theta} \right) \Big|_{\theta=0} dr - R \int_0^{\theta} \mu \frac{\partial \omega}{\partial n} \Big|_{r=R} d\theta, \tag{102}$$

in which d_0 is the distance from the cylinder center to the inlet boundary.

Non-overlapping domain decomposition technique

As described in Section 5, the relevant governing equations are of Poisson type. Thus, consider the following Poisson problem in a domain Ω with Dirichlet boundary condition on the boundary $\partial\Omega$

$$\Delta u = f(x,y) \qquad \text{in } \Omega \tag{103}$$

$$u = b \qquad \text{on } \partial\Omega \tag{104}$$

It is noted that the Neumann boundary conditions (92) can be imposed directly through the conversion process (26-29). Therefore, we just need to consider the Poisson problem with Dirichlet boundary condition here.

Without loss of generality, the domain of interest Ω is partitioned into just two non-overlapping subdomains Ω_1 and Ω_2 as shown in Figure 12. The Poisson problem can be reformulated in the equivalent multi-domain form as follows [33].

$$\Delta u^{[1]} = f^{[1]} \qquad \text{in } \Omega_1 \tag{105}$$

$$u = b^{[1]} \qquad \text{on } \partial\Omega_1 \cap \partial\Omega \tag{106}$$

$$\Delta u^{[2]} = f^{[2]} \qquad \text{in } \Omega_2 \tag{107}$$

$$u = b^{[2]} \qquad \text{on } \partial\Omega_2 \cap \partial\Omega \tag{108}$$

$$u^{[1]} = u^{[2]} \qquad \text{on } \Gamma \tag{109}$$

$$\frac{\partial u^{[1]}}{\partial n} = \frac{\partial u^{[2]}}{\partial n} \qquad \text{on } \Gamma \tag{110}$$

where Γ is the interface between Ω_1 and Ω_2, $\partial\Omega_1$ and $\partial\Omega_1$ are the boundaries of the subdomains Ω_1 and Ω_2, respectively, and the superscript $[\cdot]$ denotes a subdomain. Equations (109) and (110) are the transmission conditions for $u^{[1]}$ and $u^{[2]}$ on the interface Γ. By solving the system of Equations (105-110), one can obtain the interface values u_Γ, and the subdomain solutions $u^{[1]}$ and $u^{[2]}$.

We now describe an algorithm for solving the system of Equations (105-110) as follows. Let the subscripts ip, bp and fb represent the location indices of interior points, known boundary points and interface points over a subdomain, respectively; N, N_{ip}, N_{bp} and N_{fp} are the total number of points, the number of interior points, known boundary points and interface points of a subdomain, respectively.

System of Equations (105-110) are written in matrix form as follows.

$$\tilde{E}^{[1]}\tilde{u}^{[1]} = RHS^{[1]}, \tag{111}$$

$$\tilde{u}^{[1]}_{(bp)} = u^{[1]}_b, \tag{112}$$

$$\tilde{E}^{[2]}\tilde{u}^{[2]} = RHS^{[2]}, \tag{113}$$

$$\tilde{u}^{[2]}_{(bp)} = u^{[2]}_b, \tag{114}$$

$$\tilde{u}^{[1]}_{(fp)} = \tilde{u}^{[2]}_{(fp)} = u_\Gamma, \tag{115}$$

$$D^{[1]}\tilde{u}^{[1]} = D^{[2]}\tilde{u}^{[2]}, \tag{116}$$

where $\tilde{E}^{[1]}$ and $\tilde{E}^{[2]}$ are the known matrices of dimension $(N^{[1]}_{ip} \times N^{[1]})$ and $(N^{[2]}_{ip} \times N^{[2]})$, respectively; $\tilde{u}^{[1]}$ and $\tilde{u}^{[2]}$ are field variable vectors of length $N^{[1]}$ and $N^{[2]}$, respectively; $RHS^{[1]}$, $RHS^{[2]}$, $u^{[1]}_b$ and $u^{[2]}_b$ are the known vectors of length $N^{[1]}_{ip}$, $N^{[2]}_{ip}$, $N^{[1]}_{bp}$ and $N^{[2]}_{bp}$, respectively; u_Γ unknown vector of length N_{fp}; and $D^{[1]}$ and $D^{[2]}$ the known matrices of dimension $(N_{fp} \times N^{[1]})$ and $(N_{fp} \times N^{[2]})$, respectively.

From (111), (112) and (115), one is able to obtain the following expression

$$\begin{bmatrix} \tilde{E}^{[1]}_{(:,ip)} & \tilde{E}^{[1]}_{(:,bp)} & \tilde{E}^{[1]}_{(:,fp)} \end{bmatrix} \begin{pmatrix} \tilde{u}^{[1]}_{(ip)} \\ u^{[1]}_b \\ u_\Gamma \end{pmatrix} = RHS^{[1]} \tag{117}$$

or

$$\tilde{u}^{[1]}_{(ip)} = A^{[1]} + B^{[1]}u_\Gamma \tag{118}$$

where

$$A^{[1]} = \left(\tilde{E}^{[1]}_{(:,ip)}\right)^{-1}\left(RHS^{[1]} - \tilde{E}^{[1]}_{(:,bp)}u^{[1]}_b\right), \tag{119}$$

$$B^{[1]} = -\left(\tilde{E}^{[1]}_{(:,ip)}\right)^{-1}\tilde{E}^{[1]}_{(:,fp)}. \tag{120}$$

Similarly, from (113), (114) and (115), the interior values of the subdomain Ω_2 is given by

$$\tilde{u}^{[2]}_{(ip)} = A^{[2]} + B^{[2]} u_\Gamma \tag{121}$$

where

$$A^{[2]} = \left(\tilde{E}^{[2]}_{(:,ip)}\right)^{-1} \left(RHS^{[2]} - \tilde{E}^{[2]}_{(:,bp)} u^{[2]}_b\right), \tag{122}$$

$$B^{[2]} = -\left(\tilde{E}^{[2]}_{(:,ip)}\right)^{-1} \tilde{E}^{[2]}_{(:,fp)}. \tag{123}$$

Equation (116) can be expressed as

$$\begin{bmatrix} D^{[1]}_{(:,ip)} & D^{[1]}_{(:,bp)} & D^{[1]}_{(:,fp)} \end{bmatrix} \begin{pmatrix} \tilde{u}^{[1]}_{(ip)} \\ u^{[1]}_b \\ u_\Gamma \end{pmatrix} = \begin{bmatrix} D^{[2]}_{(:,ip)} & D^{[2]}_{(:,bp)} & D^{[2]}_{(:,fp)} \end{bmatrix} \begin{pmatrix} \tilde{u}^{[2]}_{(ip)} \\ u^{[2]}_b \\ u_\Gamma \end{pmatrix}. \tag{124}$$

By substituting Equations (118) and (121) into (124), the interface values u_Γ are determined as

$$u_\Gamma = \frac{-D^{[1]}_{(:,ip)} A^{[1]} - D^{[1]}_{(:,bp)} u^{[1]}_b + D^{[2]}_{(:,ip)} A^{[2]} + D^{[2]}_{(:,bp)} u^{[2]}_b}{D^{[1]}_{(:,ip)} B^{[1]} + D^{[1]}_{(:,fp)} - D^{[2]}_{(:,ip)} B^{[2]} - D^{[2]}_{(:,fp)}}. \tag{125}$$

By substituting (125) into (118) and (121), one can obtain the subdomain solutions $\tilde{u}^{[1]}_{(ip)}$ and $\tilde{u}^{[2]}_{(ip)}$.

The combination of LMLS-1D-IRBFN and domain decomposition technique is developed to handle this large scale problem using a PC with 2.99 *GHz* CPU and 3.25 *GB* of RAM. The computational domain is discretised using Cartesian grids as shown in Figure 13. The grid convergence study of vorticity distribution on the cylinder surface for *Re* number of 40 is presented in Figure 14. It can be seen that the current simulations converge with increasing grid densities. The results obtained for the grids of 151×151 and 167×167 are in good agreement with those of Kim et al. [22] and Dennis and Chang [32]. Therefore, the grid of 151×151 is then used to investigate the flow field with the other values of *Re* numbers (i.e. 5, 10 and 20). It is noted that when the flow reaches the steady state, a pair of

vortices and the separated region behind the cylinder are formed. The length of the wake is measured from the rear of the cylinder to the end of the separated region, while the angle of separation is defined at the point where the vorticity vanishes. Table 9 presents the length of the wake behind the cylinder (L_{sep}), the separation angle (θ_{sep}) and the drag coefficient (C_D) for *Re* numbers of 5, 10, 20 and 40. The comparison of vorticity and pressure coefficient distribution on the cylinder surface in the case of *Re* numbers of 5, 10, 20 and 40 are given in Figure 15 and 16, respectively. It can be seen that the present numerical results are in good agreement with other published results. The contours of stream function and vorticity of the flow field around the cylinder are shown in Figure 17.

8 Conclusions

A local MLS-1D-IRBFN method is proposed for solving incompressible viscous flow problems in terms of stream function and vorticity. The present approach is based on the PU concept and incorporates the MLS and 1D-IRBFN methods. The LMLS-1D-IRBFN approach offers the same order of accuracy as the 1D-IRBFN method, while the system matrix is more sparse than that of the 1D-IRBFN, which helps reduce the computational cost significantly as discussed earlier. The LMLS-1D-IRBFN shape function possesses the Kronecker-δ property which allows an exact imposition of the essential boundary condition. Cartesian grids are used to discretise both rectangular and irregular problem domains. The numerical results for the lid-driven cavity flows at high *Re* numbers showed that the calculation of convection terms using the global 1D-IRBFN technique are more accurate than the one using the LMLS-1D-IRBFN technique. The combination of the LMLS-1D-IRBFN method and a domain decomposition technique is successfully developed for solving a larger problem. The obtained numerical results for both cases of lid-driven cavity flow and flow past a circular cylinder are in good agreement with other published results available in the literature. The present method can be used to handle problems with irregular domains, while the standard finite different method cannot be applied directly at the grid points near the boundary of irregular domains. Owing to the use of integrated RBFN for local approximation, the present method appears to be more accurate than the FDM with central-difference scheme. Owing to the use of a fixed Cartesian grid, the present method is expected to be more efficient than the conventional FDM, FVM and FEM when solving problems with moving boundary.

Acknowledgement: This research is supported by the University of Southern Queensland, Australia through a USQ Postgraduate Research Scholarship awarded to D. Ngo-Cong. The authors would like to thank the reviewers for their helpful

comments.

References

[1]Ghia U, Ghia KN, Shin CT. High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method. *Journal of Computational Physics* 1982; 48:387-411.

[2]Leonard BP. A stable and accurate convective modelling procedure based on quadratic upstream interpolation. *Computer methods in applied mechanics and engineering* 1979; 19:59-98.

[3]Brooks AN, Hughes TJR. Streamline upwind/Petrov-Galerkin formulations for convection domainted flows with particular emphasis on the incompressible Navier-Stokes equations. *Computer methods in applied mechanics and engineering* 1982; 32:199-259.

[4]Lin H, Atluri SN. Meshless local Petrov-Galerkin (MLPG) method for convection-diffusion problems. *Computer Modeling in Engineering & Sciences* 2000; 1(2):45-60.

[5]Lin H, Atluri SN. The meshless local Petrov-Galerkin (MLPG) method for solving incompressible Navier-Stokes equations. *Computer Modeling in Engineering & Sciences* 2001; 2(2):117-142.

[6]Babuŝka I. Error-bounds for finite element method. *Numerische Mathematik* 1971; 16(4):322-333.

[7]Kansa EJ. Multiquadrics - A Scattered Data Approximation Scheme with Applications to Computational Fulid-Dynamics - II: Solutions to parabolic, Hyperbolic and Elliptic Partial Differential Equations. *Computers & Mathematics with Applications* 1990; 19(8/9):147-61.

[8]Park SH, Youn SK. The least-squares meshfree method. *International Journal for Numerical Methods in Engineering* 2001; 52:997-1012.

[9]Zhang XK, Kwon KC, Youn SK. The least-squares meshfree method for the steady incompressible viscous flow. *Journal of Computational Physics* 2005; 206:182-207.

[10]Hetherington J, Askes H. Penalty methods for time domain computational dynamics based on positive and negative inertia. *Computers & Structures* 2009; 87:1474-1482.

[11]Arzani H, Afshar MH. Solving Poisson's equations by the discrete least squares meshless method. In *Boundary Elements and Other Mesh Reduction Methods XXVIII*. WIT Pres: Skiathos, Greece, 2006; 23-32.

[12]Firoozjaee AR, Afshar MH. Steady-state solution of incompressible Navier-Stokes equations using discrete least-squares meshless method. *Intenational Journal for Numerical Methods in Fluids* 2011; 67:369-382.

[13]Lee CK, Liu X, Fan SC. Local multiquadric approximation for solving boundary value problems. *Computational Mechanics* 2003; 30:396-409.

[14]Šarler B, Vertnik R. Meshfree explicit local radial basis function collocation method for diffusion problems. *Computers and Mathematics with Applications* 2006; 51:1269-1282.

[15]Babuŝka I, Melenk JM. The partition of unity method. *International Journal for Numerical Methods in Engineering* 1997; 40:727-758.

[16]Krysl P, Belytschko T. An efficient linear-precision partition of unity basis for unstructured meshless methods. *Communications in Numerical Methods in Engineering* 2000; 16:239-255.

[17]Shepard D. A two dimensional interpolation function for irregularly spaced data. In *Proceedings of the 23rd National Conference ACM*. ACM: New York 1968; 517-523.

[18]Chen JS, Hu W, Hu HY. Reproducing kernel enhanced local radial basis collocation method. *International Journal for Numerical Methods in Engineering* 2008; 75:600-627.

[19]Le PBH, Rabczuk T, Mai-Duy N, Tran-Cong T. A moving IRBFN-based integration-free meshless method. *Computer Modeling in Engineering & Sciences* 2010; 61(1):63-109.

[20]Botella O, Peyret R. Benchmark spectral results on the lid-driven cavity flow. *Computers & Fluids* 1998; 27(4):421-433.

[21]Ding H, Shu C, Yeo KS, D. Xu. Simulation of incompressible viscous flows past a circular cylinder by hybrid FD scheme and meshless least square-based finite difference method. *Computer Methods in Applied Mechanics and Engineering* 2004; 193:727-744.

[22]Kim Y, Kim DW, Jun S, Lee JH. Meshfree point collocation method for the stream-vorticity formulation of 2D incompressible Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering* 2007; 196:3095-3109.

[23]Mai-Duy N, Tanner RI. A Collocation Method based on One-Dimensional RBF Interpolation Scheme for Solving PDEs. *International Journal of Numerical Methods for Heat & Fluid Flow* 2007; 17(2):165-186.

[24]Mai-Duy N, Tran-Cong T. Numerical solution of differential equations using multiquadric radial basis function networks. *Neural Networks* 2001; 14:185-199.

[25]Ngo-Cong D, Mai-Duy N, Karunasena W, Tran-Cong T. Free vibration analysis of laminated composite plates based on FSDT using one-dimensional IRBFN method. *Computers & Structures* 2011; 89:1-13.

[26]Liu GR. *Meshfree Methods: Moving beyond the Finite Element Method*. CRC Press LLC: Boca Raton, Florida, 2003.

[27]Glowinski R. Finite Element Methods for Incompressible Viscous Flows. In *Handbook of Numerical Analysis*, Vol. 9, Ciarlet PG and Lions JL (eds). North-Holland: Amsterdam, 2003.

[28]Bruneau CH, Jouron C. An efficient scheme for solving steady incompressible Navier-Stokes equations. *Journal of Computational Physics* 1990; 89(2):389-413.

[29]Mai-Duy N, Tran-Cong T. Integrated radial-basis-function networks for computing Newtonian and non-Newtonian fluid flows. *Computers & Structures* 2009; 87:642-650.

[30]Le-Cao K, Mai-Duy N, Tran-Cong T. An effective integrated-RBFN Cartesian-grid discretization for the stream function-vorticity-temperature formulation in nonrectangular domains. *Numerical Heat Transfer, Part B* 2009; 55:480-502.

[31]Muralidhar K, Sundararajan T. Computational fluid flow and heat transfer. In *IIT Kanpur Series of Advanced Texts*. Narosa Publishing House: New Delhi, 1995 pages 219-220.

[32]Dennis SCR, Chang GZ. Numerical solutions for steady flow past a circular cylinder at Reynolds number up to 100. *Journal of Fluid Mechanics* 1970; 42(3):471-489.

[33]Quarteroni A, Valli A. *Domain Decomposition Methods for Partial Differential Equations*. Clarendon Press: Oxford, 1999.

Table 1: Poisson equation in a square domain subject to Dirichlet boundary conditions: comparisons (with FDM and 1D-IRBFN) of the number of nonzero elements per row of the system matrix (N_{nzpr}) and condition number (*cond*). The system matrix is stored in a sparse matrix format.

Grid	System matrix	N_{nzpr}				*cond*			
		FDM	1D-IRBFN	App. 1	App. 2	FDM	1D-IRBFN	App. 1	App. 2
11×11	81×81	5	21	9	13	5.85E+01	1.59E+02	1.66E+02	1.55E+02
21×21	361×361	5	41	9	13	2.35E+02	5.85E+02	6.93E+02	6.24E+02
31×31	841×841	5	61	9	13	5.30E+02	1.32E+03	1.57E+03	1.41E+03
41×41	1521×1521	5	81	9	13	9.43E+02	2.35E+03	2.80E+03	2.50E+03
51×51	2401×2401	5	101	9	13	1.47E+03	3.67E+03	4.37E+03	3.91E+03
61×61	3481×3481	5	121	9	13	2.12E+03	5.28E+03	6.30E+03	5.63E+03
71×71	4761×4761	5	141	9	13	2.89E+03	7.19E+03	8.58E+03	7.66E+03
81×81	6241×6241	5	161	9	13	3.77E+03	9.39E+03	1.12E+04	1.00E+04
91×91	7921×7921	5	181	9	13	4.77E+03	1.19E+04	1.42E+04	1.27E+04
101×101	9801×9801	5	201	9	13	5.89E+03	1.47E+04	1.75E+04	1.56E+04
111×111	11881×11881	5	221	9	13	7.13E+03	1.78E+04	2.12E+04	1.89E+04
121×121	14161×14161	5	241	9	13	8.49E+03	2.11E+04	2.52E+04	2.25E+04

Table 2: Poisson equation in a square domain subject to Dirichlet boundary conditions: comparison (with FDM and 1D-IRBFN) of CPU time and percentage of nonzero elements of the system matrix (ε). Note that for a given grid size the present Approach 2 is slower than the FDM. However, the present Approach 2 achieves a given level of accuracy with a coarser grid and hence more efficient. For example, as shown in Figure 3, the present Approach 2 with grid=21×21 yields better accuracy ($Ne = 6.88e-6$) in 0.88 seconds than the FDM with grid=121×121 ($Ne = 3.49e-5$) in 1.74 seconds.

Grid	CPU time (*seconds*) for all shape functions				Total CPU time (*seconds*)				$\varepsilon(\%)$			
	FDM	1D-IRBFN	App. 1	App. 2	FDM	1D-IRBFN	App. 1	App. 2	FDM	1D-IRBFN	App. 1	App. 2
11×11	0.00	0.03	0.12	0.22	0.00	0.05	0.18	0.23	5.624	20.988	9.465	12.757
21×21	0.00	0.06	0.53	0.86	0.01	0.10	0.54	0.88	1.327	10.249	2.318	3.251
31×31	0.01	0.39	1.31	2.05	0.02	0.58	1.33	2.08	0.578	6.778	1.021	1.447
41×41	0.03	2.12	2.54	3.83	0.05	2.85	2.62	3.97	0.322	5.062	0.571	0.814
51×51	0.05	8.58	4.35	6.34	0.09	10.71	4.46	6.58	0.205	4.040	0.365	0.521
61×61	0.10	25.98	6.82	9.71	0.16	30.86	6.99	10.00	0.142	3.361	0.253	0.362
71×71	0.18	66.68	10.01	14.02	0.25	77.73	10.24	14.49	0.104	2.878	0.185	0.266
81×81	0.30	169.14	14.08	19.44	0.40	190.49	14.40	20.14	0.079	2.516	0.142	0.203
91×91	0.46	462.68	19.15	26.11	0.62	502.23	19.67	26.90	0.063	2.235	0.112	0.161
101×101	0.69	1073.42	25.36	34.27	0.88	1139.92	26.04	35.58	0.051	2.010	0.091	0.130
111×111	1.00	2202.37	32.84	43.85	1.25	2308.72	33.61	45.60	0.042	1.826	0.075	0.108
121×121	1.43	4959.75	41.74	55.33	1.74	5123.18	42.58	57.37	0.035	1.674	0.063	0.090

Table 3: Poisson equation in a square domain subject to Dirichlet and Neumann boundary conditions: comparison condition number (*cond*).

Grid	*cond*		
	1D-IRBFN	App. 1	App. 2
11×11	3.87E+02	3.70E+02	4.55E+02
21×21	1.41E+03	1.36E+03	1.63E+03
31×31	3.10E+03	2.94E+03	3.53E+03
41×41	5.45E+03	5.12E+03	6.14E+03
51×51	8.45E+03	7.89E+03	9.47E+03
61×61	1.21E+04	1.13E+04	1.35E+04
71×71	1.64E+04	1.52E+04	1.83E+04
81×81	2.14E+04	1.98E+04	2.37E+04
91×91	2.70E+04	2.49E+04	2.99E+04
101×101	3.33E+04	3.07E+04	3.68E+04
111×111	4.02E+04	3.70E+04	4.45E+04
121×121	4.78E+04	4.40E+04	5.28E+04

Table 4: Poisson equation in a square domain with a circular hole subject to Dirichlet boundary conditions: comparison of relative error norm (Ne) and condition number (*cond*).

Grid	*Ne*		*cond*	
	1D-IRBFN	Present	1D-IRBFN	Present
25×25	8.62E-03	4.86E-02	4.15E+02	4.28E+02
33×33	3.43E-03	2.01E-02	6.20E+02	6.17E+02
41×41	1.72E-03	9.12E-03	9.60E+02	8.75E+02
49×49	9.95E-04	4.61E-03	1.35E+03	1.21E+03
57×57	6.29E-04	2.57E-03	2.19E+03	2.21E+03
65×65	4.27E-04	1.54E-03	2.15E+03	2.20E+03
73×73	2.98E-04	9.80E-04	3.55E+03	3.58E+03
81×81	2.19E-04	6.55E-04	3.58E+03	3.72E+03
89×89	1.65E-04	4.55E-04	6.15E+03	6.29E+03
97×97	1.28E-04	3.27E-04	6.65E+03	6.74E+03
105×105	1.00E-04	2.41E-04	7.83E+03	8.28E+03
113×113	8.02E-05	1.83E-04	1.35E+04	1.41E+04
121×121	6.53E-05	1.41E-04	1.20E+04	1.28E+04
129×129	5.39E-05	1.11E-04	1.47E+04	1.52E+04

Table 5: Poisson equation in a square domain with a circular hole subject to Dirichlet boundary conditions: comparison of CPU time and percentage of nonzero elements of the system matrix (ε).

Grid	CPU time (*seconds*)		$\varepsilon(\%)$	
	1D-IRBFN	Present	1D-IRBFN	Present
25×25	0.39	1.71	7.579	2.323
33×33	0.89	2.71	5.739	1.325
41×41	3.18	4.41	4.600	0.851
49×49	9.68	6.65	3.824	0.591
57×57	24.72	9.49	3.276	0.434
65×65	55.88	13.02	2.871	0.333
73×73	115.22	17.27	2.548	0.263
81×81	222.04	22.56	2.295	0.213
89×89	464.42	28.76	2.085	0.176
97×97	946.15	35.96	1.913	0.148
105×105	1793.67	44.39	1.766	0.126
113×113	3153.54	54.16	1.639	0.109
121×121	5140.94	65.55	1.530	0.095
129×129	7937.85	78.39	1.435	0.083

Table 6: Lid-driven cavity flow, $Re = 1000$: the grid convergence study and comparison of extrema of velocity profiles along the center lines. The convection terms are calculated using LMLS-1D-IRBFN technique. Note that “Error" is relative to a Benchmark solution.

	Grid	u_{min}	Error (%)	y	v_{max}	Error (%)	x	v_{min}	Error (%)	x
Present	21×21	-0.33342	14.193	0.333	0.27403	27.301	0.220	-0.34690	34.184	0.827
	31×31	-0.33043	14.962	0.202	0.32097	14.848	0.172	-0.43390	17.678	0.888
	41×41	-0.35408	8.876	0.183	0.34304	8.993	0.165	-0.47541	9.804	0.901
	51×51	-0.36903	5.029	0.177	0.35730	5.211	0.162	-0.49891	5.345	0.906
	61×61	-0.37750	2.848	0.174	0.36561	3.006	0.160	-0.51164	2.930	0.907
	71×71	-0.38218	1.644	0.173	0.37027	1.769	0.159	-0.51846	1.635	0.908
	81×81	-0.38478	0.976	0.172	0.37290	1.072	0.159	-0.52215	0.935	0.909
	91×91	-0.38626	0.595	0.172	0.37441	0.670	0.158	-0.52420	0.547	0.909
	101×101	-0.38712	0.373	0.172	0.37531	0.432	0.158	-0.52536	0.326	0.909
1D-IRBFN [29]	101×101	-0.38772	0.218	0.172	0.37601	0.247	0.158	-0.52598	0.208	0.909
FDM ($\psi - \omega$) [1]	129×129	-0.38289	1.462	0.172	0.37095	1.589	0.156	-0.51550	2.197	0.906
FDM ($\mathbf{u} - p$) [28]	256×256	-0.37640	3.132	0.160	0.36650	2.770	0.152	-0.52080	1.192	0.910
Benchmark [20]		-0.38857		0.172	0.37694		0.158	-0.52708		0.909
Present	Order of convergence		2.42			2.61			2.92	

Table 7: Lid-driven cavity flow, $Re = 1000$: comparisons of the number of nonzero elements per row of the system matrix (N_{nzpr}), number of iterations ($N_{iteration}$) and total CPU time (T_{total}) required to obtain the converged solution with $TOL = 10^{-12}$. The time step Δt is set to be 5×10^{-3} for all cases. Note that for a given grid size the present approach is slower than the FDM. However, the present approach achieves a given level of accuracy with a coarser grid and hence more efficient. For example, as shown in Table 6, the present approach with grid=81×81 yields better accuracy in 1559.77 seconds than the FDM with grid=129×129 in 1733.02 seconds.

		FDM			Present		
Grid	System matrix	N_{nzpr}	$N_{iteration}$	$T_{total}(seconds)$	N_{nzpr}	$N_{iteration}$	$T_{total}(seconds)$
21×21	361×361	5	52207	36.46	13	51088	43.66
31×31	841×841	5	44914	45.67	13	40590	63.48
41×41	1521×1521	5	41703	68.59	13	43047	220.69
51×51	2401×2401	5	36467	148.10	13	44513	452.89
61×61	3481×3481	5	39591	250.27	13	45239	781.17
71×71	4761×4761	5	41803	354.93	13	45569	884.19
81×81	6241×6241	5	42893	482.21	13	45714	1559.77
91×91	7921×7921	5	43568	679.77	13	45779	2356.15
101×101	9801×9801	5	44028	898.53	13	45807	2964.84
111×111	9801×9801	5	44360	1207.33	13	-	-
121×121	9801×9801	5	44608	1433.07	13	-	-
129×129	9801×9801	5	44764	1733.02	13	-	-

Table 8: Lid-driven cavity flow, $Re = 1000$: the grid convergence study and comparison of extrema of horizontal and vertical velocity profiles along the center lines. The convection terms are calculated using global 1D-IRBFN technique. Note that “Error" is relative to a Benchmark solution.

	Grid	u_{min}	Error (%)	y	v_{max}	Error (%)	x	v_{min}	Error (%)	x
Present	21×21	-0.30543	21.397	0.223	0.29460	21.844	0.181	-0.39550	24.963	0.866
	31×31	-0.35522	8.583	0.179	0.34326	8.936	0.166	-0.47452	9.971	0.900
	41×41	-0.37207	4.245	0.173	0.35938	4.660	0.162	-0.50276	4.615	0.906
	51×51	-0.38005	2.193	0.172	0.36744	2.519	0.160	-0.51576	2.147	0.908
	61×61	-0.38423	1.117	0.171	0.37183	1.356	0.159	-0.52208	0.949	0.909
	71×71	-0.38642	0.552	0.171	0.37421	0.725	0.158	-0.52512	0.371	0.909
	81×81	-0.38756	0.259	0.171	0.37549	0.385	0.158	-0.52655	0.100	0.909
	91×91	-0.38815	0.108	0.171	0.37618	0.203	0.158	-0.52720	0.022	0.909
	101×101	-0.38845	0.032	0.171	0.37655	0.104	0.158	-0.52746	0.073	0.909
1D-IRBFN [29]	101×101	-0.38772	0.218	0.172	0.37601	0.247	0.158	-0.52598	0.208	0.909
FDM ($\psi - \omega$) [1]	129×129	-0.38289	1.462	0.172	0.37095	1.589	0.156	-0.51550	2.197	0.906
FDM ($\mathbf{u} - p$) [28]	256×256	-0.37640	3.132	0.160	0.36650	2.770	0.152	-0.52080	1.192	0.910
Benchmark [20]		-0.38857		0.172	0.37694		0.158	-0.52708		0.909
Present	Order of convergence		3.80			3.26			4.26	

Table 9: Flow past a circular cylinder: comparison of the wake length (L_{sep}), the separation angle (θ_{sep}) and the drag coefficient (C_D) for $Re = 5, 10, 20$ and 40, using a grid of 151×151.

Re	Source	L_{sep}	θ_{sep}	C_D
5	Dennis and Chang [32]	-	-	4.116
	Kim et al. [22]	-	-	4.282
	Present	-	-	4.108
10	Dennis and Chang [32]	0.265	29.6	2.846
	Ding et al. [21]	0.252	30.0	3.070
	Kim et al. [22]	0.281	29.5	2.920
	Present	0.27	30.1	2.829
20	Dennis and Chang [32]	0.94	43.7	2.045
	Ding et al. [21]	0.93	44.1	2.180
	Kim et al. [22]	0.91	43.7	2.017
	Present	0.92	43.6	2.010
40	Dennis and Chang [32]	2.345	53.8	1.522
	Ding et al. [21]	2.20	53.5	1.713
	Kim et al. [22]	2.187	55.1	1.640
	Present	2.31	53.7	1.542

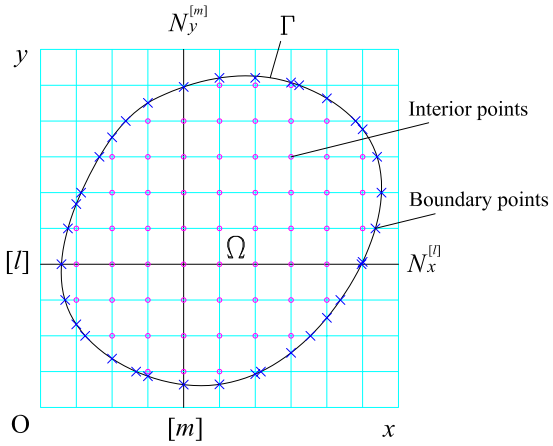


Figure 1: Cartesian grid discretisation.

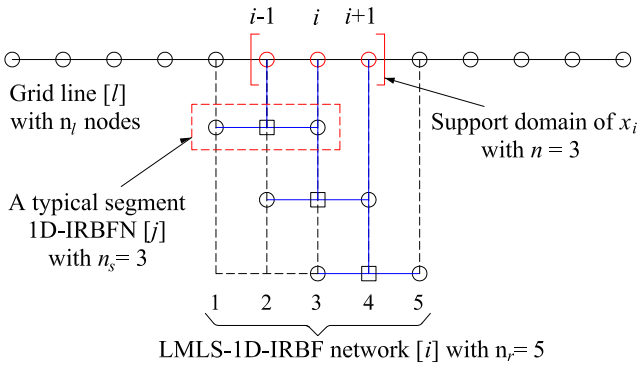


Figure 2: LMLS-1D-IRBFN-3-node scheme.

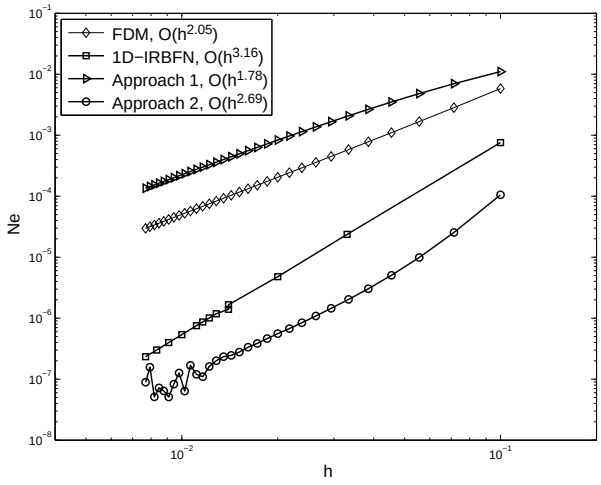


Figure 3: Poisson equation in a square domain subject to Dirichlet boundary conditions: convergence study for 1D-IRBFN, Approach 1 with $\beta = 10$ and Approach 2 with $\beta = 15$. FDM (central difference) results are included for comparison.

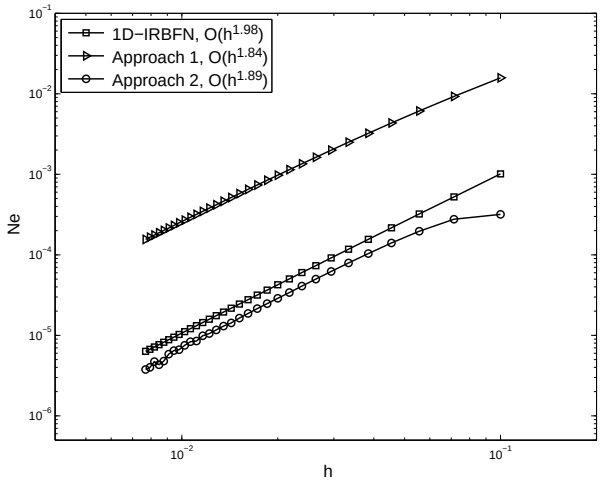


Figure 4: Poisson equation in a square domain subject to Dirichlet and Neumann boundary conditions: convergence study for 1D-IRBFN, Approach 1 with $\beta = 10$ and Approach 2 with $\beta = 5$.

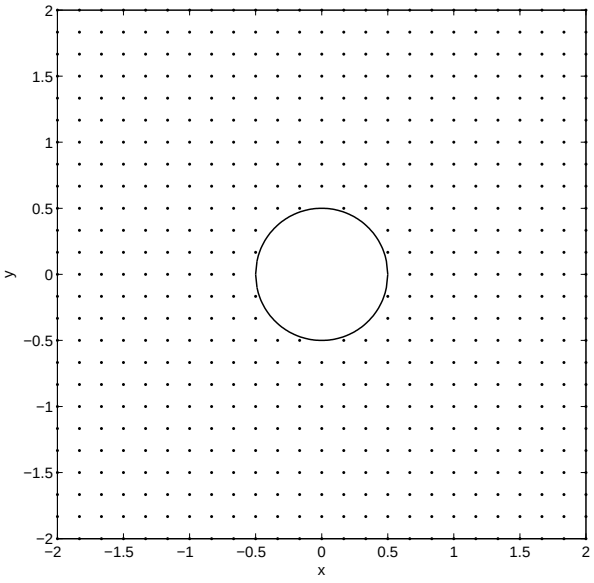


Figure 5: A square domain with a circular hole.

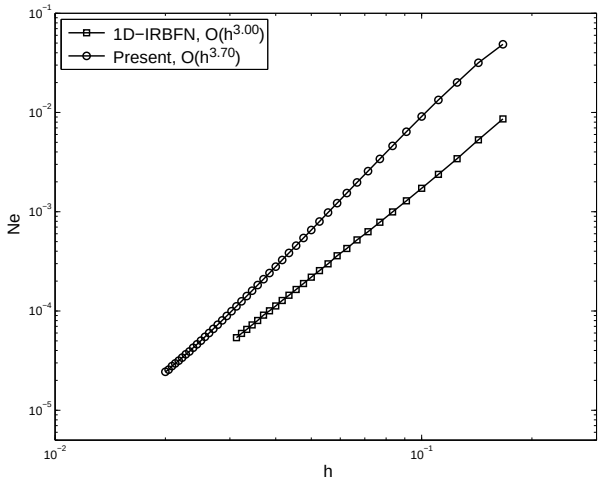


Figure 6: Poisson equation in a square domain with a circular hole subject to Dirichlet boundary conditions: convergence study for 1D-IRBFN and the present method (LMLS-1D-IRBFN-5-node) with $\beta = 15$.

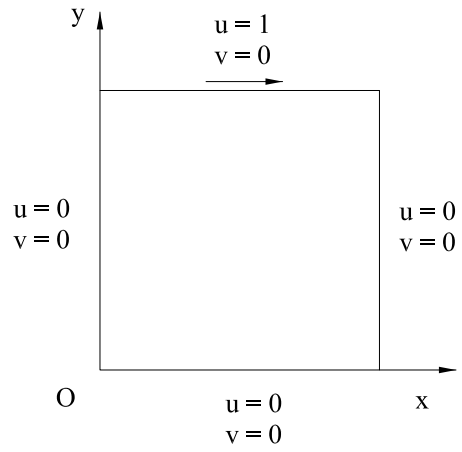
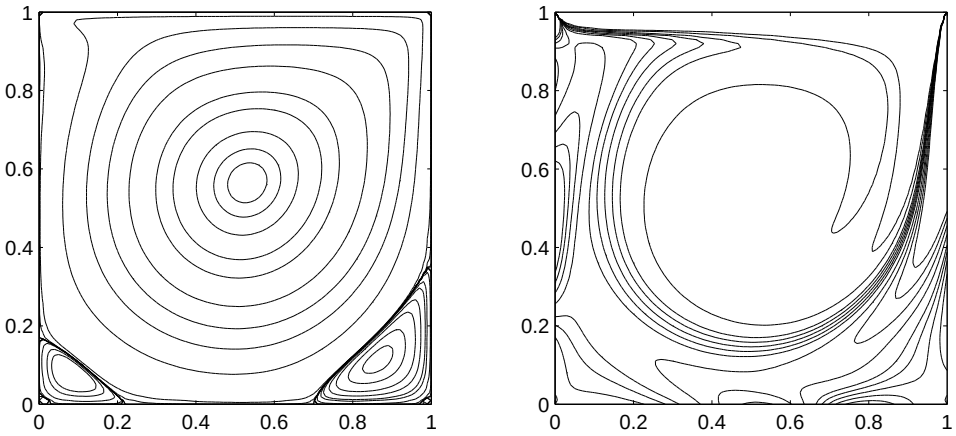
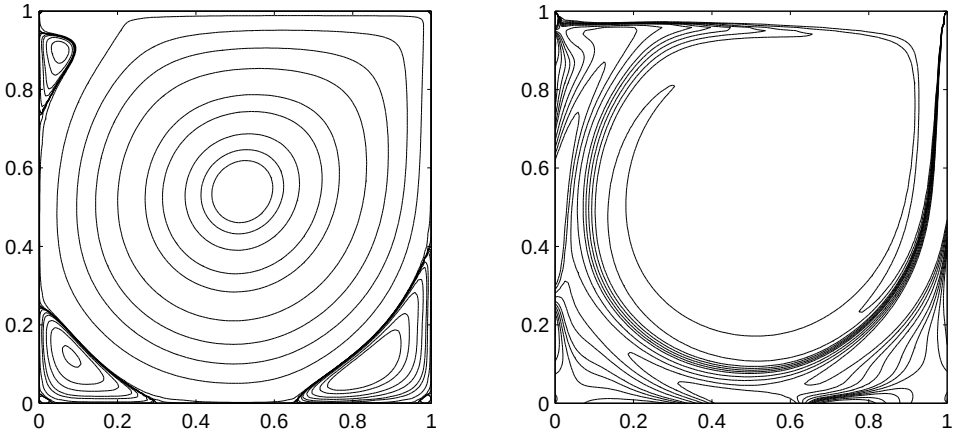


Figure 7: Lid-driven cavity flow: problem geometry and boundary conditions.

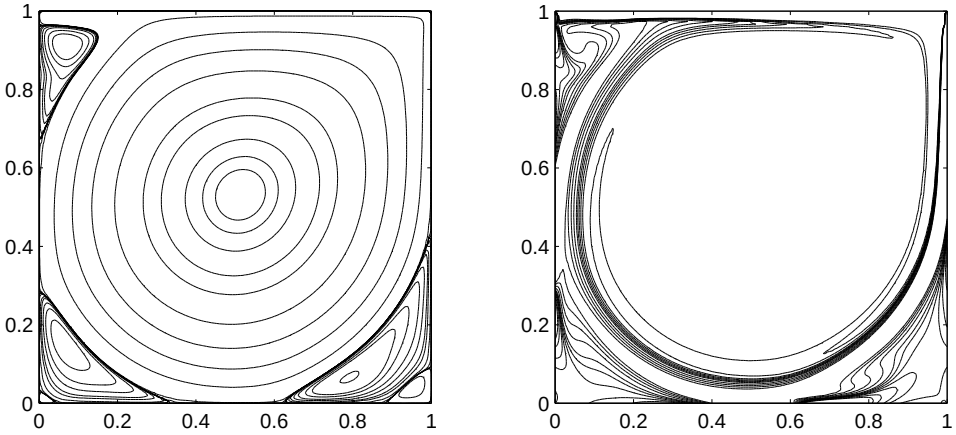
$Re = 1000$



$Re = 3200$



$Re = 7500$



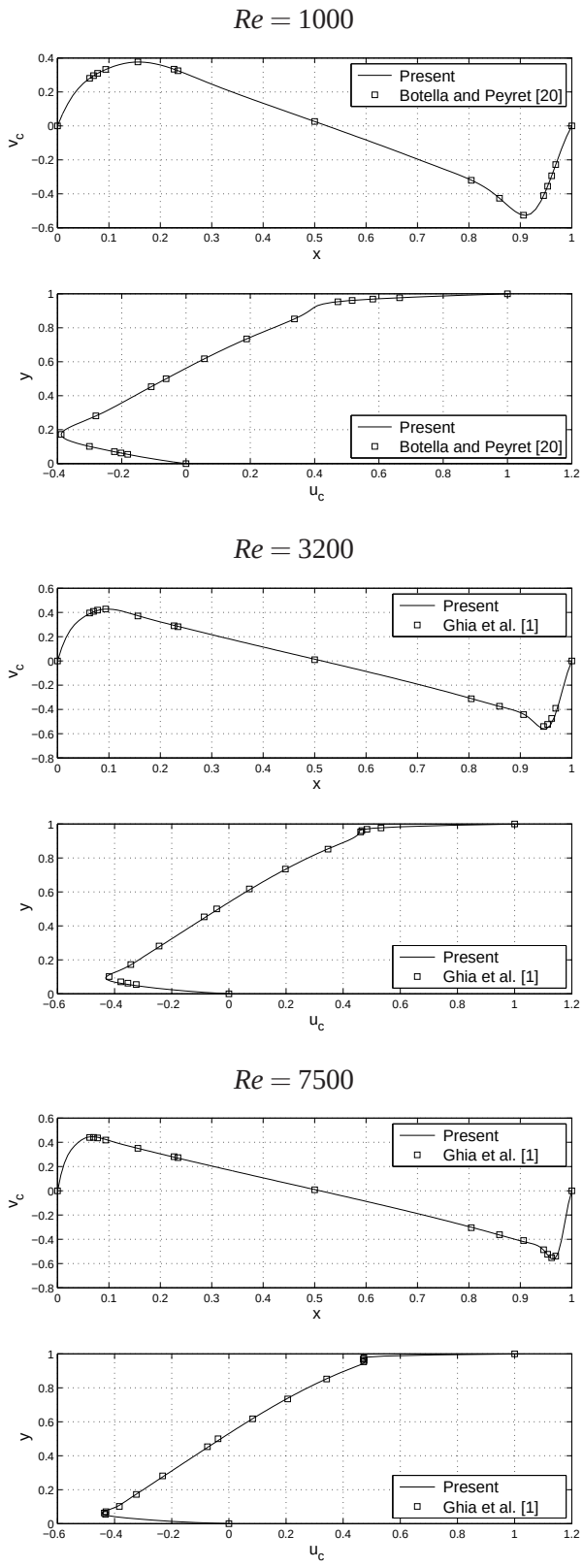


Figure 9: Lid-driven cavity flow: comparison of profiles of vertical and horizontal

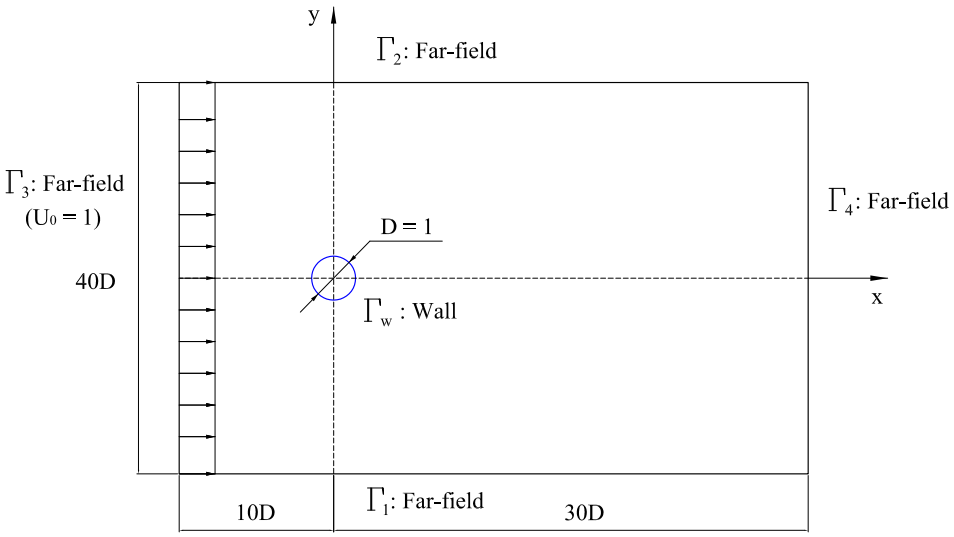


Figure 10: Flow past a circular cylinder: problem geometry and boundary conditions.

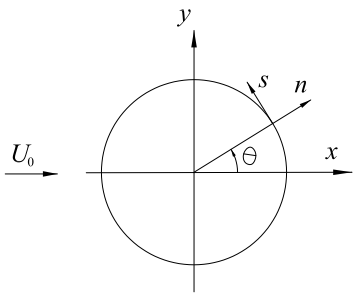


Figure 11: Circular cylinder and associated coordinate systems.

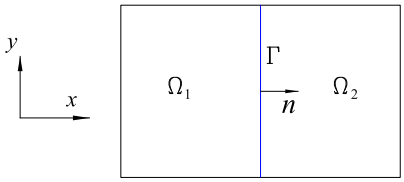


Figure 12: Non-overlapping partition of the domain Ω into two subdomains Ω_1 and Ω_2.

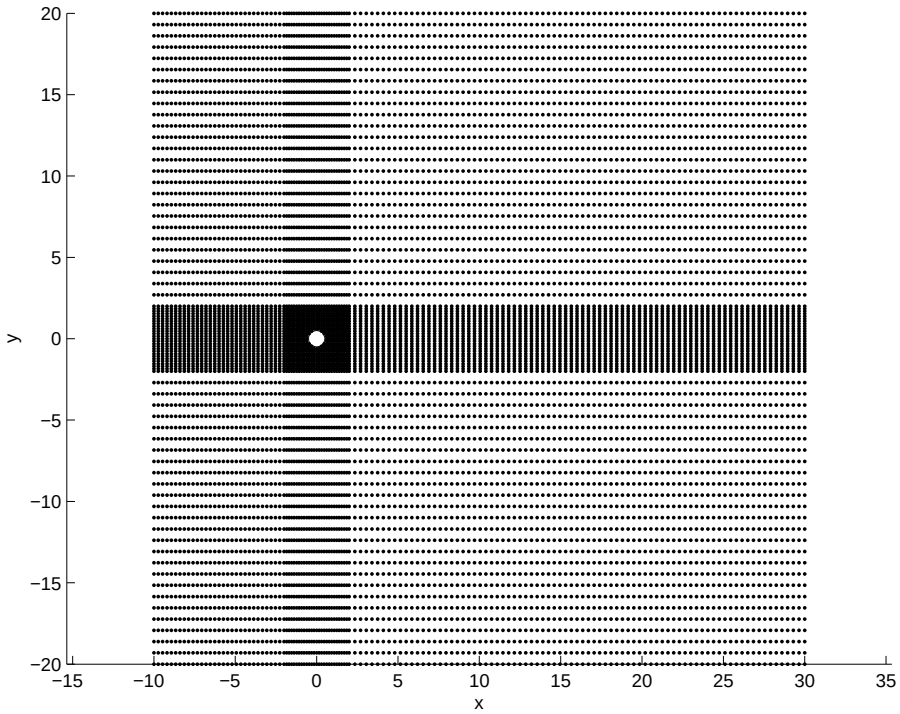


Figure 13: Flow past a circular cylinder: grid configuration.

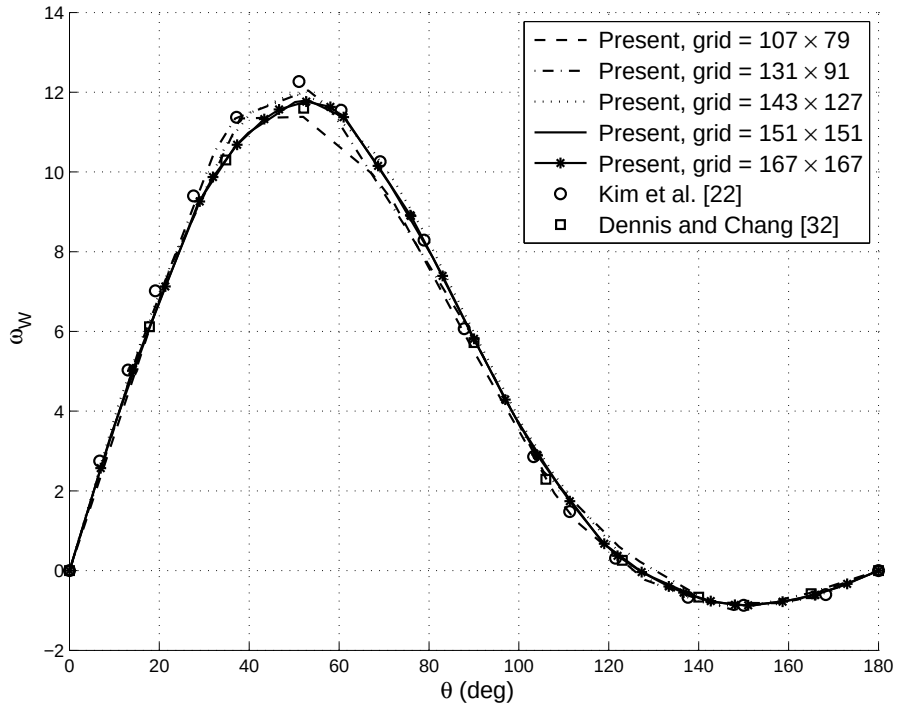


Figure 14: Flow past a circular cylinder: grid convergence study, Re = 40.

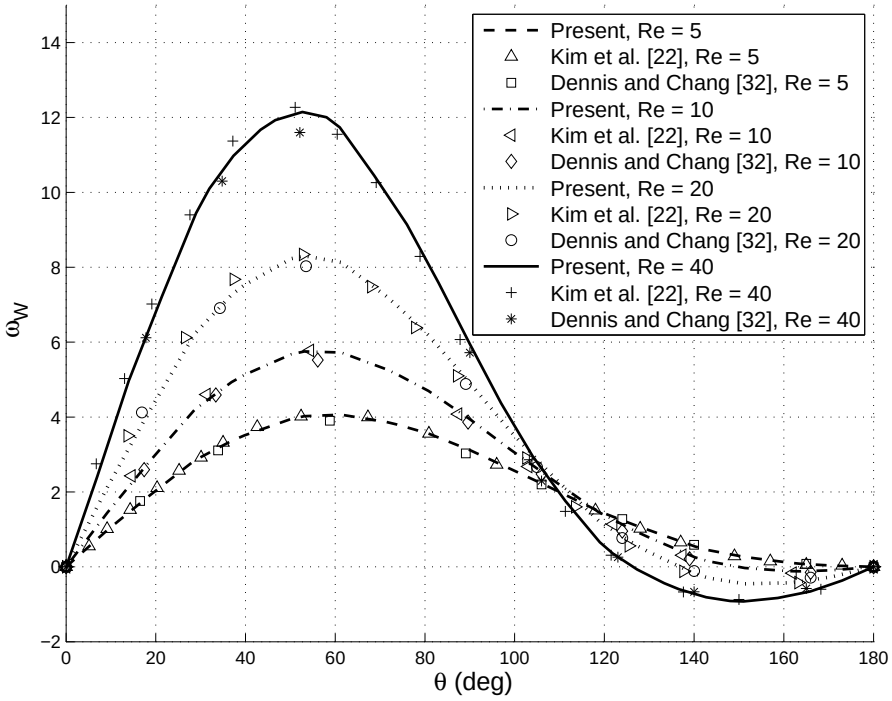


Figure 15: Flow past a circular cylinder: comparison of vorticity on the circular cylinder in the cases of $Re = 5, 10, 20$ and 40, using a grid of 151×151.

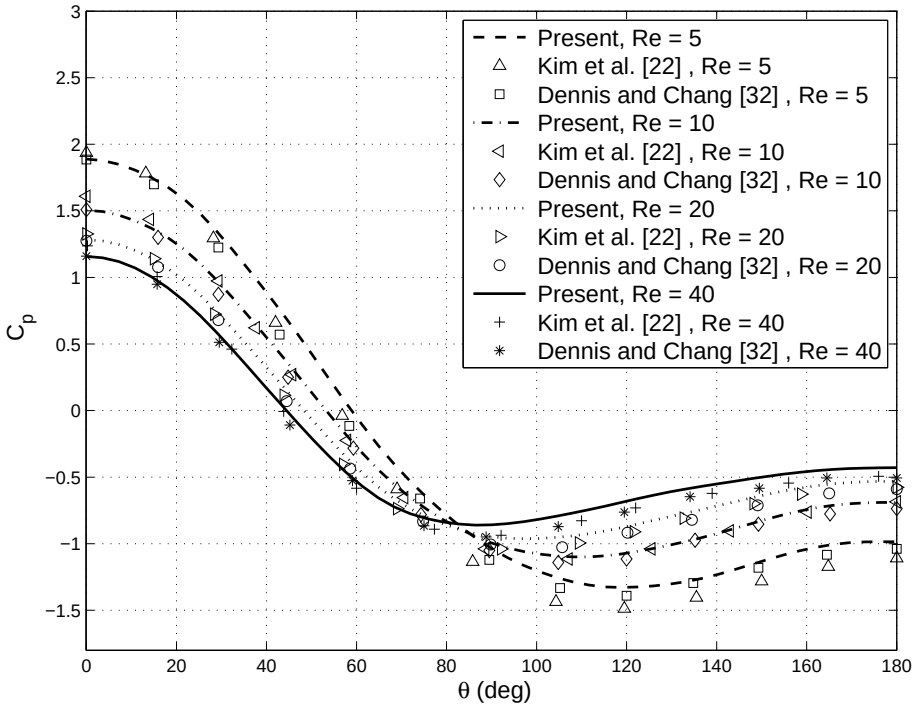


Figure 16: Flow past a circular cylinder: comparison of pressure coefficient on the circular cylinder in the cases of $Re = 5, 10, 20$ and 40, using a grid of 151×151.

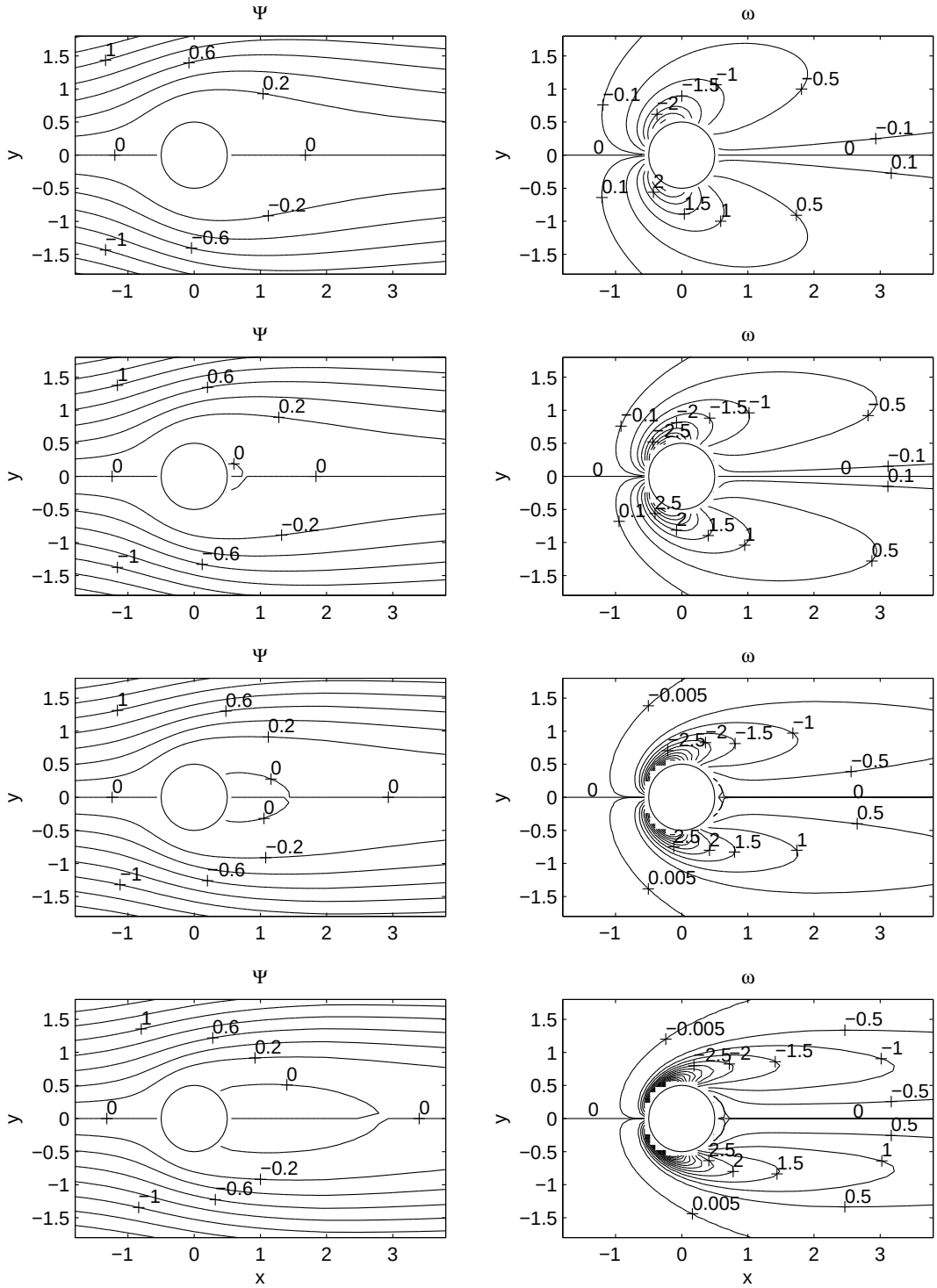


Figure 17: Flow past a circular cylinder: contours of stream function (left) and vorticity (right) for the cases of $Re = 5, 10, 20$ and 40, from top to bottom, using a grid of 151×151.