



A high-order accurate meshless method for solution of incompressible fluid flow problems



Shantanu Shahane*, Anand Radhakrishnan, Surya Pratap Vanka

Department of Mechanical Science and Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801, United States of America

ARTICLE INFO

Article history:

Available online 10 August 2021

Keywords:

Meshless method
Radial basis function based finite difference
Polyharmonic spline
Incompressible Navier-Stokes equation

ABSTRACT

Meshless solution to differential equations using radial basis functions (RBF) is an alternative to grid based methods commonly used. Since the meshless method does not need an underlying connectivity in the form of control volumes or elements, issues such as grid skewness that adversely impact accuracy are eliminated. Gaussian, Multi-quadrics and inverse Multi-quadrics are some of the most popular RBFs used for the solutions of fluid flow and heat transfer problems. But they have additional shape parameters that have to be fine tuned for accuracy and stability. Moreover, they also face stagnation error when the point density is increased for accuracy. Recently, Polyharmonic splines (PHS) with appended polynomials have been shown to solve the above issues and give rapid convergence of discretization errors with the degree of appended polynomials. In this research, we extend the PHS-RBF method for the solution of incompressible Navier-Stokes equations. A fractional step method with explicit convection and explicit diffusion terms is combined with a pressure Poisson equation to satisfy the momentum and continuity equations. Systematic convergence tests have been performed for five model problems with two of them having analytical solutions. We demonstrate fast convergence both with refinement of number of points and degree of appended polynomials. The method is further applied to solve problems such as lid-driven cavity and vortex shedding over circular cylinder. We have also analyzed the performance of this approach for solution of Euler equations. The proposed method shows promise to solve fluid flow and heat transfer problems in complex domains with high accuracy.

© 2021 Elsevier Inc. All rights reserved.

1. Introduction

Various methodologies exist for the solution of Navier-Stokes equations in complex domains. Among these are finite volume and finite element methods (FVM, FEM) on unstructured hexahedral/tetrahedral grids and spectral element methods. Unstructured FVMs are widely used in commercial fluid flow software but can at best be second order accurate because of the inherent basis of the FVM concept, unless complex reconstruction schemes are devised. If grid skewness is significant, such methods can even degrade to first order accuracy. Thus, in problems where highly accurate solutions are required (for instance, direct numerical simulations of transition or turbulent flows), the mesh size has to be really small which adds to the computational cost. Moreover, problems involving moving fronts or interfaces such as solidification, shock fronts and multiphase flows can be solved more accurately using moving adaptive grids with refined elements near the interfaces

* Corresponding author.

E-mail addresses: shahaneshantanu@gmail.com, sshahan2@illinois.edu (S. Shahane).

and progressively coarser elements away. Since the unstructured FVM formulation involves geometric entities like faces and control volumes, it is cumbersome and expensive to adaptively refine and coarsen the grids at each timestep. Spectral domain and spectral element methods are alternatives to FVM but require placement of grid points at the pre-determined Gauss-Lobatto points, thus restricting local resolution and also impacting the time step based on the CFL criterion.

For the last few decades, there has been growing interest into meshless methods. Smoothed particle hydrodynamics [1–3], generalized finite difference method [4–6], reproducing kernel particle method [7–10], element-free Galerkin method [11–13], hp-clouds [14–16], partition of unity [17–19], finite point method [20–22] and radial basis function based finite difference (RBF-FD) method are some of the popular meshless methods. Meshless methods utilize only point clouds as a form of discretization for any domain. Connectivity in the form of edges, faces and control volumes is not required. This gives an elegant numerical formulation with lower processing and memory requirements. Hardy [23] proposed the RBF methodology for a cartography application which required a scattered node interpolation. Later, Kansa [24,25] showed that the RBF interpolants can be used as a tool to numerically solve parabolic, elliptic and hyperbolic partial differential equations with a globally connected multiquadric (MQ) function. The global scheme generated coefficient matrices that were full and became ill-conditioned as the number of points in the domain increased. Kansa and Hon [26] addressed this problem by block partitioning strategy with preconditioners which helped reduce the condition number compared to the global MQ matrix. Shu et al. [27] used a local MQ method in which the derivatives at any point of interest are approximated using a local cloud of points in the neighborhood of the point of interest. This strategy made the coefficient matrix sparsely connected and led to improvement of its condition number.

Some of the common RBFs are as follows:

$$\begin{aligned}
 \text{Multiquadratics (MQ): } \phi(r) &= (r^2 + \epsilon^2)^{1/2} \\
 \text{Inverse Multiquadratics (IMQ): } \phi(r) &= (r^2 + \epsilon^2)^{-1/2} \\
 \text{Gaussian: } \phi(r) &= \exp\left(\frac{-r^2}{\epsilon^2}\right) \\
 \text{Polyharmonic Splines (PHS): } \phi(r) &= r^{2a+1}, \quad a \in \mathbb{N} \\
 \text{Thin Plate Splines (TPS): } \phi(r) &= r^{2a} \log(r), \quad a \in \mathbb{N}
 \end{aligned} \tag{1}$$

where, r is the distance between the RBF central point and any other point in the domain and ϵ is known as the shape parameter. There is an extensive literature available which uses the Multiquadratics, inverse Multiquadratics or Gaussian RBFs for solving PDEs. Several researchers have used RBF cloud interpolations to solve a variety of fluid flow, heat transfer and solid mechanics problems [27–38].

A primary difficulty with the use of MQ, IMQ or Gaussian RBFs is the need to prescribe a shape parameter [27–29,31–35]. The shape parameter is an important quantity for accuracy as well as the condition number of the matrix. Yet, there is no theoretical basis for its prescription although, extensive investigations have been carried out to illustrate its effect [29,39–41]. In the limit of large shape parameter ($\epsilon \rightarrow \infty$), the RBFs become flat and are found to be highly accurate. The problem with this regime of ϵ is that the matrix becomes ill conditioned and hence, numerically, it is difficult to solve the linear system. On the other hand, in the small ϵ limit ($\epsilon \rightarrow 0$), the RBFs have sharp peaks and the linear system is well conditioned but the accuracy is poor. There has been research to stabilize the flat RBFs ($\epsilon \rightarrow \infty$) by techniques such as orthogonalization [42–44]. It is necessary to alter the shape parameter for each geometry and grid which is difficult for practical calculations. Further, these RBFs can result in discretization errors reaching a constant value when the mesh is refined beyond a certain extent. This is known as stagnation or saturation phenomenon.

In recent years, there has been an effort to use Polyharmonic Splines (PHS) which do not need any shape parameter. Fornberg and colleagues [45–49] have pursued this approach with a locally supported cloud based formulation. They showed that if polynomials are appended to the PHS, high accuracy can be achieved based on the degree of the polynomial. Further, adding polynomials to PHS also removes the saturation problem. They showed that this approach maintains high order accuracy at interior as well as boundary nodes by solving Poisson and scalar transport equations and low Mach number compressible flows. Santos et al. [50] compared the performance of stabilized flat Gaussian RBFs with PHS-RBF for two dimensional Poisson equation and observed that the PHS approach is more robust and computationally more efficient than the stabilized Gaussian RBFs. Bayona [51] compared a local weighted least squares approach with polynomial basis and the PHS-RBF method for interpolation and derivative approximation. The choice of weighting function is a difficulty in the least squares approach and was found to fail at high polynomial degrees. Hence they inferred that the PHS-RBF method is superior. Shankar [52] developed an overlapped PHS-RBF approach where, the stencil is shared by a portion of cloud points in contrast to the traditional cloud approach in which each point has a separate cloud around it. This method is efficient in computing the coefficients for discretized differential operators especially for large stencil sizes at higher polynomial degrees. They showed speedup factors up to 16 and 60 in two and three dimensions respectively. Shankar and Fogelson [53] further used the overlapped RBF approach to solve advection-diffusion equations at Peclet numbers upto 1000 by adding artificial hyper-viscosity. Jančić et al. [54] used the PHS-RBF method for solution of the Poisson equation in two and three dimensions and demonstrated the increasing order of accuracy with polynomial degree. Gunderman et al. [55] solved the advection equation on spherical geometries using the PHS-RBF method. They added a small artificial diffusion term to the hyperbolic equation to stabilize the method.

From the above literature survey, we see that there has been significant amount of research for the solution of fluid flow and heat transfer problems using RBFs such as Gaussian, Multiquadratics and Inverse Multiquadratics. The Polyharmonic splines have clear advantages since the shape factor tuning is not needed, saturation error is not observed and fast convergence is seen with higher degrees of appended polynomials. The PHS-RBFs have been applied for the solution of Poisson and scalar transport equations. However, to the best of our knowledge, a systematic analysis of the utility of PHS-RBFs for the solution of incompressible Navier-Stokes equation has not been reported. In this work, we present a time marching Navier-Stokes solver for incompressible flows and first demonstrate its accuracy in two problems with analytical solutions. Two other complex flow problems are solved for which numerical solutions are first generated by a very fine grid and used to evaluate the errors at coarser resolutions. We have later applied the algorithm to compute the errors in a initial field that decays to a null solution with time. Comparisons between results of several point sets are made to demonstrate spatial accuracy in a temporally varying velocity field using Richardson extrapolation. We further simulate the lid-driven cavity problem and flow over circular cylinder and show a comparison with various benchmark solutions available in the literature. We also assess the performance of this approach for solution of Euler equations and analyze the energy conservation properties. This method is developed in C++ with an object oriented framework under the title Meshless Multi-Physics Software (MeMPhyS) and is launched open source [56].

2. The PHS-RBF method

In this research, we use a cloud based PHS-RBF approach with appended polynomials to obtain numerical estimates of the differential operators. For a problem with dimension d (which can be 1, 2 or 3) and maximum polynomial degree k , the number of appended monomials m is given by $\binom{k+d}{k}$. A scalar variable s is interpolated over q scattered points as

$$s(\mathbf{x}) = \sum_{i=1}^q \lambda_i \phi_i(||\mathbf{x} - \mathbf{x}_i||_2) + \sum_{i=1}^m \gamma_i P_i(\mathbf{x}) \quad (2)$$

In this work, we use the PHS-RBF given by $\phi(r) = r^{2a+1}$, $a \in \mathbb{N}$. The RBF (ϕ_i) is always a scalar function of the Euclidean distance between the points irrespective of the problem dimension. λ_i and γ_i are $q+m$ unknowns which have to be determined. Collocation at the q cloud points gives q conditions. Additional m constraints required to close the system of equations are given by

$$\sum_{i=1}^q \lambda_i P_j(\mathbf{x}_i) = 0 \quad \text{for } 1 \leq j \leq m \quad (3)$$

Writing in a matrix vector form,

$$\begin{bmatrix} \Phi & \mathbf{P} \\ \mathbf{P}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \lambda \\ \gamma \end{bmatrix} = [\mathbf{A}] \begin{bmatrix} \lambda \\ \gamma \end{bmatrix} = \begin{bmatrix} s \\ \mathbf{0} \end{bmatrix} \quad (4)$$

where, the superscript T denotes the transpose, $\lambda = [\lambda_1, \dots, \lambda_q]^T$, $\gamma = [\gamma_1, \dots, \gamma_m]^T$, $s = [s(\mathbf{x}_1), \dots, s(\mathbf{x}_q)]^T$ and $\mathbf{0}$ is the matrix of all zeros of appropriate size. Sizes of the submatrices Φ and \mathbf{P} are $q \times q$ and $q \times m$ respectively.

The submatrix Φ is given by:

$$\Phi = \begin{bmatrix} \phi(||\mathbf{x}_1 - \mathbf{x}_1||_2) & \dots & \phi(||\mathbf{x}_1 - \mathbf{x}_q||_2) \\ \vdots & \ddots & \vdots \\ \phi(||\mathbf{x}_q - \mathbf{x}_1||_2) & \dots & \phi(||\mathbf{x}_q - \mathbf{x}_q||_2) \end{bmatrix} \quad (5)$$

For two dimensional problem ($d = 2$) with maximum polynomial degree 2 ($k = 2$), there are $m = \binom{k+d}{k} = \binom{2+2}{2} = 6$ polynomial terms: $[1, x, y, x^2, xy, y^2]$. Thus, the submatrix \mathbf{P} is formed by evaluating these polynomial terms at the q cloud points.

$$\mathbf{P} = \begin{bmatrix} 1 & x_1 & y_1 & x_1^2 & x_1 y_1 & y_1^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_q & y_q & x_q^2 & x_q y_q & y_q^2 \end{bmatrix} \quad (6)$$

Solving the linear eq. (4) gives values of unknown coefficients λ_i and γ_i for interpolating any function:

$$\begin{bmatrix} \lambda \\ \gamma \end{bmatrix} = [\mathbf{A}]^{-1} \begin{bmatrix} s \\ \mathbf{0} \end{bmatrix} \quad (7)$$

Note that the inverse of matrix \mathbf{A} in eq. (7) is just used as a notation. Practically, explicit inverse is never computed to avoid numerical ill-conditioning.

The fluid flow equations involve differential operators such as gradient and Laplacian. The RBFs can be used to estimate these operators as a weighted linear combination of function values at the cloud points. Let \mathcal{L} denote any scalar linear operator such as $\frac{\partial}{\partial x}$ or the Laplacian ∇^2 . When \mathcal{L} is operated on eq. (2), using the linearity of \mathcal{L} gives:

$$\mathcal{L}[s(\mathbf{x})] = \sum_{i=1}^q \lambda_i \mathcal{L}[\phi_i(\mathbf{x})] + \sum_{i=1}^m \gamma_i \mathcal{L}[P_i(\mathbf{x})] \quad (8)$$

Collocating eq. (8) using the \mathcal{L} evaluated at the q cloud points gives a rectangular matrix vector system:

$$\mathcal{L}[\mathbf{s}] = [\mathcal{L}[\Phi] \quad \mathcal{L}[\mathbf{P}]] \begin{bmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\gamma} \end{bmatrix} \quad (9)$$

where, $\mathcal{L}[\Phi]$ and $\mathcal{L}[\mathbf{P}]$ are matrices of sizes $q \times q$ and $q \times m$ respectively.

$$\mathcal{L}[\mathbf{s}] = [\mathcal{L}[s(\mathbf{x})]_{x_1}, \dots, \mathcal{L}[s(\mathbf{x})]_{x_q}]^T \quad (10)$$

$$\mathcal{L}[\Phi] = \begin{bmatrix} \mathcal{L}[\phi(||\mathbf{x} - \mathbf{x}_1||_2)]_{x_1} & \dots & \mathcal{L}[\phi(||\mathbf{x} - \mathbf{x}_q||_2)]_{x_1} \\ \vdots & \ddots & \vdots \\ \mathcal{L}[\phi(||\mathbf{x} - \mathbf{x}_1||_2)]_{x_q} & \dots & \mathcal{L}[\phi(||\mathbf{x} - \mathbf{x}_q||_2)]_{x_q} \end{bmatrix} \quad (11)$$

For two dimensional problem ($d = 2$) with maximum polynomial degree 2 ($p = 2$), $\mathcal{L}[\mathbf{P}]$ is given by:

$$\mathcal{L}[\mathbf{P}] = \begin{bmatrix} \mathcal{L}[1]_{x_1} & \mathcal{L}[x]_{x_1} & \mathcal{L}[y]_{x_1} & \mathcal{L}[x^2]_{x_1} & \mathcal{L}[xy]_{x_1} & \mathcal{L}[y^2]_{x_1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \mathcal{L}[1]_{x_q} & \mathcal{L}[x]_{x_q} & \mathcal{L}[y]_{x_q} & \mathcal{L}[x^2]_{x_q} & \mathcal{L}[xy]_{x_q} & \mathcal{L}[y^2]_{x_q} \end{bmatrix} \quad (12)$$

The subscripts in eqs. (10) to (12) denote that the functions obtained by operating \mathcal{L} on the RBFs and the appended polynomials are evaluated at the cloud points (\mathbf{x}_i). Substituting eq. (7) in eq. (9) and simplifying, we get:

$$\begin{aligned} \mathcal{L}[\mathbf{s}] &= ([\mathcal{L}[\Phi] \quad \mathcal{L}[\mathbf{P}]] [\mathbf{A}]^{-1}) \begin{bmatrix} \mathbf{s} \\ \mathbf{0} \end{bmatrix} = [\mathbf{B}] \begin{bmatrix} \mathbf{s} \\ \mathbf{0} \end{bmatrix} \\ &= [\mathbf{B}_1 \quad \mathbf{B}_2] \begin{bmatrix} \mathbf{s} \\ \mathbf{0} \end{bmatrix} = [\mathbf{B}_1][\mathbf{s}] + [\mathbf{B}_2][\mathbf{0}] = [\mathbf{B}_1][\mathbf{s}] \end{aligned} \quad (13)$$

In eq. (13), the matrix $[\mathbf{B}]$ of size $q \times (q+m)$ is split along columns into two submatrices $[\mathbf{B}_1]$ and $[\mathbf{B}_2]$ of sizes $q \times q$ and $q \times m$ respectively. $[\mathbf{B}_1]$ is the matrix of weights in the linear combination to estimate values of the operator \mathcal{L} at the cloud points. $[\mathbf{B}_1]$ depends only on the co-ordinates of the cloud points and hence can be precomputed and stored.

3. Properties of PHS-RBF

3.1. Error in gradient and Laplacian

The PHS-RBF method discussed in section 2 is implemented here on a test function. The gradients and Laplacian at origin $(0, 0)$ are estimated numerically and compared with analytical values for error analysis. Summation of multiple sinusoids with different wave numbers is used as test function:

$$1 + \sin(4x) + \cos(3y) + \sin(2y) \quad (14)$$

A randomly generated set of points around the origin are chosen as neighboring cloud points. In Fig. 1, the central point (origin) where the differential operators are to be estimated is shown as a red square and the cloud points are shown as blue circles. As higher degree polynomials are appended, more points are chosen for interpolation. Typically, the total number of points in the cloud (including the center) should at least be equal to the number of terms of the appended polynomials. For example, in this case of two dimensional problem ($d = 2$) with maximum polynomial degree 5 ($k = 5$), there are $m = \binom{k+d}{k} = \binom{5+2}{2} = 21$ terms of appended polynomials. Random points are generated in a square domain $[-1, 1] \times [-1, 1]$ initially and the inter-point distance is decreased geometrically $[2^{-1}, 2^{-2}, \dots]$ until the numerical error reaches close to the roundoff error. Using the exact test function values at the neighbor points (blue circles in Fig. 1), the errors in gradients and Laplacian are computed at the origin (red square in Fig. 1) using different number of points in the neighborhood. Here, the number of discrete points is set equal to the number of appended polynomials. Figs. 2a and 2b show error in the gradients and Laplacian respectively at the origin for the test function. The circled lines indicate the absolute difference between analytical and numerical values. The dotted one is a best fit line indicating slope of the estimated error. It can be seen that for a polynomial degree k , the gradient and Laplacian estimations are $\mathcal{O}(k)$ and $\mathcal{O}(k-1)$ accurate respectively. Note that the order of accuracy increases with addition of points and higher polynomial degree (k), with error reaching the roundoff

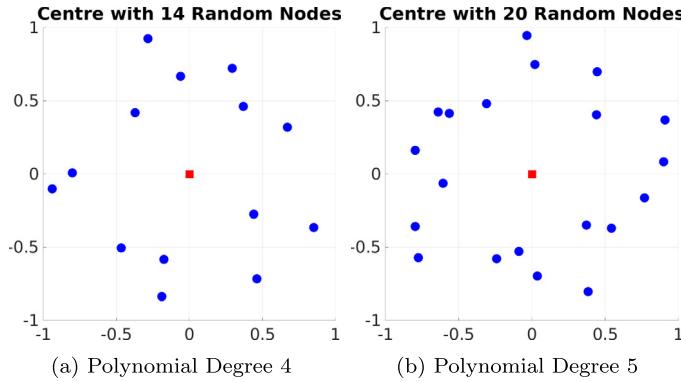


Fig. 1. Random points in two dimensions (red: central point, blue: cloud). (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

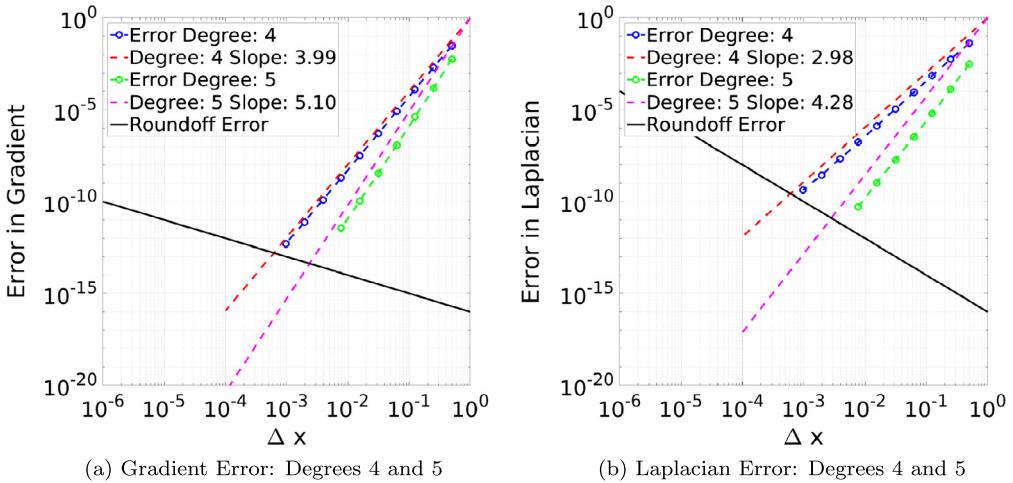


Fig. 2. Error for varying polynomial degrees and grid spacing (Δx).

error shown by the solid black line. For a machine precision of σ (assumed 10^{-16} here) and grid size Δx , the roundoff errors for numerical estimation of gradient and Laplacian are $\mathcal{O}(\sigma/\Delta x)$ and $\mathcal{O}(\sigma/\Delta x^2)$ respectively. The roundoff is a hard limit beyond which the error cannot be further reduced. Thus, the PHS-RBF method is seen to reach close to the roundoff error with the error dropping off exponentially. These results are similar to those previously reported [46,48,49].

3.2. Condition number of the RBF matrix

To compute the gradient and Laplacian coefficients, a linear system with RBF matrix \mathbf{A} has to be solved (eqs. (7) and (13)). The condition number of \mathbf{A} is important for stable coefficient estimation. In this section, a one dimensional problem is considered to demonstrate growth of the condition number with polynomial degree and PHS exponent.

The condition number for varying degrees of appended polynomial (k) and PHS degrees ($2a + 1$) are plotted in Fig. 3. Note that with increasing polynomial degree, more points are added to the cloud. For any PHS degree, it can be seen that the condition number increases with polynomial degree as the cloud becomes larger.

Another interesting property of PHS-RBF interpolations is that the condition number of the \mathbf{A} matrix can be improved by scaling and shifting of the cloud of points. For each discrete point and its cloud, we observe that the condition number of the RBF matrix decreases significantly if we define a local origin (at the point in consideration) and temporarily shift the coordinates of the points in the cloud with respect to this origin. Further, we observe that it is beneficial to scale the local distances in the cloud to lie in the range $[0, 1]$ in each direction.

To demonstrate this, we consider a uniform one dimensional cloud in the domain $[0, 1]$ as the base case. This base case is first modified by shifting the origin i.e., adding a constant to all the points in the cloud. For instance, shift of origin by 2 implies that the domain is shifted to $[2, 3]$ from $[0, 1]$. In the second analysis, the domain is scaled down. For example, a domain length scale of 0.1 implies that the domain is scaled to $[0, 0.1]$ from $[0, 1]$. Fig. 4 shows the condition number of the RBF matrix \mathbf{A} for both shifting and scaling separately. It can be seen that for all the polynomial degrees, the condition number is lowest if the origin is at the cloud center and the domain is scaled to unity. This implies that the coefficient

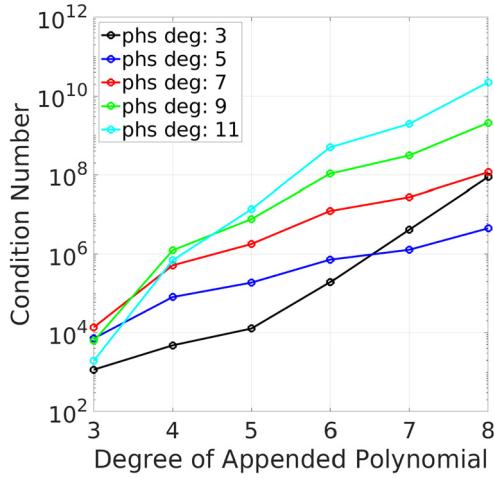


Fig. 3. Effect of polynomial degree and PHS degree (1D) on condition number of RBF matrix.

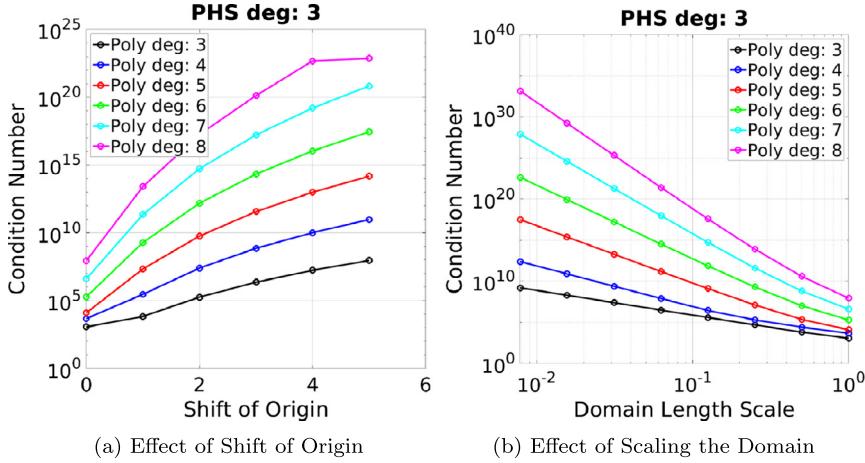


Fig. 4. Condition number of RBF matrix (1D).

estimation by eq. (13) is most stable when the cloud lies in the range [0, 1]. Thus, for practical fluid flow problems the overall domain and position of the local clouds can be modified as follows:

$$\mathbf{x}_t = \frac{\mathbf{x} - \min(\mathbf{x})}{\max(\mathbf{x}) - \min(\mathbf{x})} \quad \mathbf{y}_t = \frac{\mathbf{y} - \min(\mathbf{y})}{\max(\mathbf{y}) - \min(\mathbf{y})} \quad (15)$$

where, \mathbf{x} and \mathbf{y} are the original co-ordinates of points in the cloud and \mathbf{x}_t and \mathbf{y}_t are the transformed co-ordinates. The RBF matrix \mathbf{A} is first computed for the transformed co-ordinates and eq. (13) is solved. The transformation is then accounted for in the eqs. (11) and (12) by chain rule of derivatives. The derivatives are subsequently scaled back to reverse the local non-dimensionalization of the coordinate axes. The scaling and shifting do not impact the values of the derivatives.

4. Algorithm for incompressible flows

The high order of convergence and the flexibility in representing complex domains make the PHS-RBF interpolation method of solving partial differential equations attractive to simulate practical fluid flows. Previous works in this direction have considered the Poisson equation [46], scalar advection equation and the compressible Navier-Stokes equations [45]. The application of this method to incompressible flows has been however limited. In this work, we have developed a flow solver for incompressible flows using a time marching fractional step method [57]. The fractional step method integrates the time-dependent flow equations in two steps. First, an intermediate velocity field is computed by neglecting the pressure gradient term in the momentum equations. In the second step, a pressure-Poisson equation (PPE) is solved to project this intermediate velocity field to be divergence-free. The PPE is solved with Neumann boundary conditions obtained from the normal momentum equation at the boundary. The fractional step method can be written as follows:

$$\rho \frac{\hat{u} - u^n}{\Delta t} = -\rho \mathbf{u}^n \bullet (\nabla u^n) + \mu \nabla^2 u^n \quad (16)$$

$$\rho \frac{\hat{v} - v^n}{\Delta t} = -\rho \mathbf{u}^n \bullet (\nabla v^n) + \mu \nabla^2 v^n \quad (17)$$

$$\nabla \bullet (\nabla p) = \frac{\rho}{\Delta t} \left(\frac{\partial \hat{u}}{\partial x} + \frac{\partial \hat{v}}{\partial y} \right) \quad (18)$$

where, the superscript ‘n’ refers to the values at the previous time step. In the above discretization, we have shown first order accurate forward differencing of the time derivative for steady state problems. However, second order accurate Adams-Bashforth method is used for transient problems. The advection operator and the diffusion terms can be computed either explicitly or implicitly. Implicit formulation of the diffusion term will result in a Poisson equation for a potential function, whose gradient will project the intermediate velocity to a divergence-free field. The pressure and the potential can be shown to be related as

$$p = \phi + \nabla^2 \phi \quad (19)$$

Implicit treatment of the advection term will require iterations at a time step and will permit larger time steps. Currently, for simplicity, we have considered an explicit formulation for both advection and diffusion and used an appropriately small stable time step.

The advection and diffusion terms are evaluated with the cloud-based interpolation scheme described in section 2. Since the velocities are known at the previous time step, all advection and diffusion operators can be evaluated, and the intermediate velocities can be updated to the new time step. For u and v , Dirichlet conditions are currently prescribed at all the boundaries. The boundary values of \hat{u} and \hat{v} are estimated using the exact velocities and the numerically computed pressure gradient from the momentum equation at the boundaries. To satisfy the continuity equation, the pressure Poisson equation is solved with the source term as the local divergence in the \hat{u} and \hat{v} velocity fields. For given velocity boundary conditions, the boundary conditions on the pressure Poisson equation are all Neumann values, given by the normal momentum equation at the boundary points.

$$\nabla p \bullet \mathbf{N} = (-\rho(\mathbf{u} \bullet \nabla)\mathbf{u} + \mu \nabla^2 \mathbf{u}) \bullet \mathbf{N} \quad (20)$$

where, \mathbf{N} is the unit normal at the boundary points facing in the outward direction. The momentum equations at the boundary points are computed in the individual Cartesian directions by using the stencils for derivatives at the boundary clouds. The expression for the normal pressure gradient is then discretized and substituted in the equations of the interior nodes. The pressure Poisson equation with all Neumann boundary conditions is ill-conditioned. The pressure level is arbitrary because of incompressibility. In this work, we use the regularization approach in which sum of all the pressures is set to zero. This improves the condition number of the discrete system as well as fixes the pressure level [58,59]. Finally, the discrete equations are ordered by the RCM algorithm [60,61].

In the present work, we have used a sparse LU factorization of the discrete pressure Poisson equation, and stored the factored matrices to be used repeatedly at every time step. Since the coefficients are not varying in time, the factorization is done only once and the time for factorization is amortized over the entire time integration. The back substitution step is much cheaper than the factorization. However, for cases where LU factorization is expensive (large number of points), we have developed a multilevel iterative algorithm that uses coarse sets of points to accelerate convergence. Iterative methods have the flexibility to terminate the convergence at arbitrary levels of accuracy, and hence can be more efficient if only steady state solution is desired. The multilevel algorithm and its assessment is reported separately.

The computational cost of an iterative solver as well as sparse matrix-vector product increases with the size and bandwidth (number of non-zeros per row) of the sparse matrix. Size of the matrix is equal to the total number of points (M) used to discretize the domain. Bandwidth is given by the cloud size at each point since all the points in a cloud are directly coupled with each other. As described in section 2, for two dimensional problems, the number of appended monomials $m = \binom{k+2}{k} = \frac{(k+1)(k+2)}{2}$ where, k is the maximum degree of appended polynomials. In this work, we set the cloud size to twice the number of monomials: $q = 2m$. Hence, the total computational cost is $\mathcal{O}(Mk^2)$. Overall, the use of a higher polynomial degree may be computationally efficient since the total number of points (M) can be lowered as the solution becomes higher order accurate.

5. Error analysis: steady state problems

In this section, we have applied the above calculation procedure to four fluid flow problems. The first problem is the Kovasznay flow [62] with exact solutions to the Navier-Stokes equations. The second problem considered here is the cylindrical Couette flow between two cylinders with the inner cylinder rotating at a constant angular speed. For these two problems, the numerical solutions are compared with exact analytical solutions that satisfy the governing equations. These two were chosen to verify the algorithm and demonstrate the discretization accuracy by systematic testing. After verification of the code, we applied it to two model flows: flow in an eccentric annulus with the inner cylinder rotating and flow of a rotating

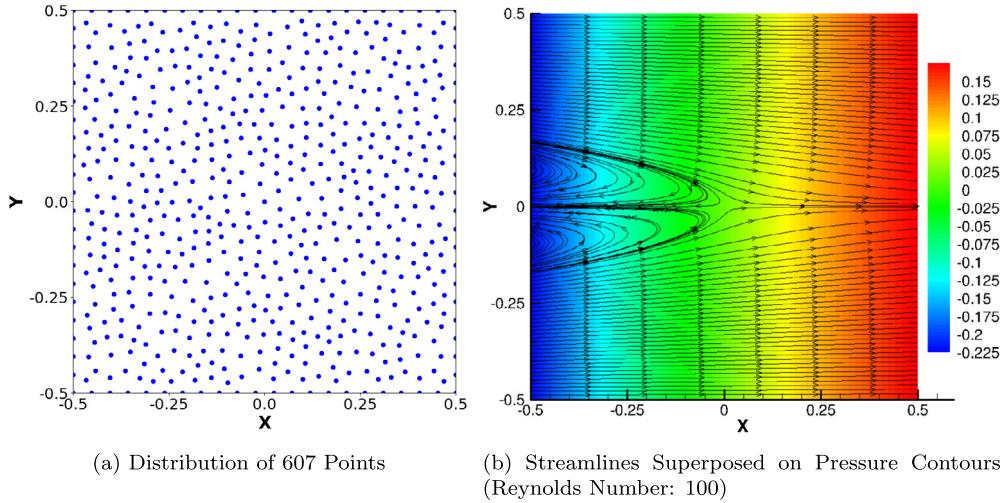


Fig. 5. Kovasznay flow.

cylinder inside an elliptic enclosure. For the next two problems, reference solutions are first generated with a large number of points and high order polynomials. The convergence of the discretization error is then calculated by comparing solutions of the varying number of points and degrees of appended polynomials. For each case, the vertices of an unstructured triangular element grid generated by the Gmsh [63] software are used as scattered points.

5.1. Kovasznay flow

The first problem used for demonstrating the algorithm is the flow behind a two-dimensional grid, known as the Kovasznay flow [62]. Kovasznay flow has an exact solution to the Navier-Stokes equations. The velocities and pressure in the Kovasznay flow are given in terms of a parameter λ which is a function of the Reynolds number:

$$\lambda = \frac{Re}{2} - \left(\frac{Re^2}{4} + 4\pi^2 \right)^{0.5} \quad (21)$$

The X and Y components of velocity and pressure denoted by u , v and p respectively, are given as [64]:

$$\begin{aligned} u &= 1 - \exp(\lambda x) \cos(2\pi y) \\ v &= \lambda \exp(\lambda x) \sin(2\pi y) / (2\pi) \\ p &= p_0 - (\exp(2\lambda x)/2) \end{aligned} \quad (22)$$

We consider a computational domain of a unit square with X and Y values ranging from -0.5 to 0.5, with X as the direction of the flow and Y being the periodic direction. The boundary conditions for u and v are prescribed from the exact solutions. In the fractional step procedure, we need the values of \hat{u} and \hat{v} . These boundary conditions are prescribed using the exact velocities and the numerically computed pressure gradient from the momentum equation at the boundaries. For the solution of the pressure Poisson equation, the Neumann boundary conditions are prescribed again by calculating the pressure gradient from the momentum equations at the boundary points and resolving the pressure gradients in the normal direction. The equations are solved to steady state from an initial distribution of zero velocities and pressure. Three sets of points are considered with 607, 2535 and 10023 points. As an example, the distribution with 607 points is plotted in Fig. 5a. The streamlines and contours of pressure are shown in Fig. 5b for a Reynolds Number of 100. Timestep value (Δt) of 1E-3 is used for the two coarser grids which needed around 10000 timesteps to reach a steady state. The finest grid reached steady state in 50000 timesteps with $\Delta t = 2E-4$.

The local differences of pressure and velocity components from the exact solutions are calculated at the point locations and their L1 norms are plotted as a function of grid spacings (Δx) in Fig. 6. The L1 norm of divergence of the velocity field is also plotted since it signifies the error in satisfying the continuity equation for an incompressible flow. Grid spacing is defined as: $\Delta x = \sqrt{(\text{flow area})/n_p}$ where, n_p is the total number of points. For each polynomial degree, a best fit line is plotted through the 12 errors. Slope of this line gives the order of convergence. It can be seen that the order of convergence increases by roughly one order of magnitude with polynomial degree.

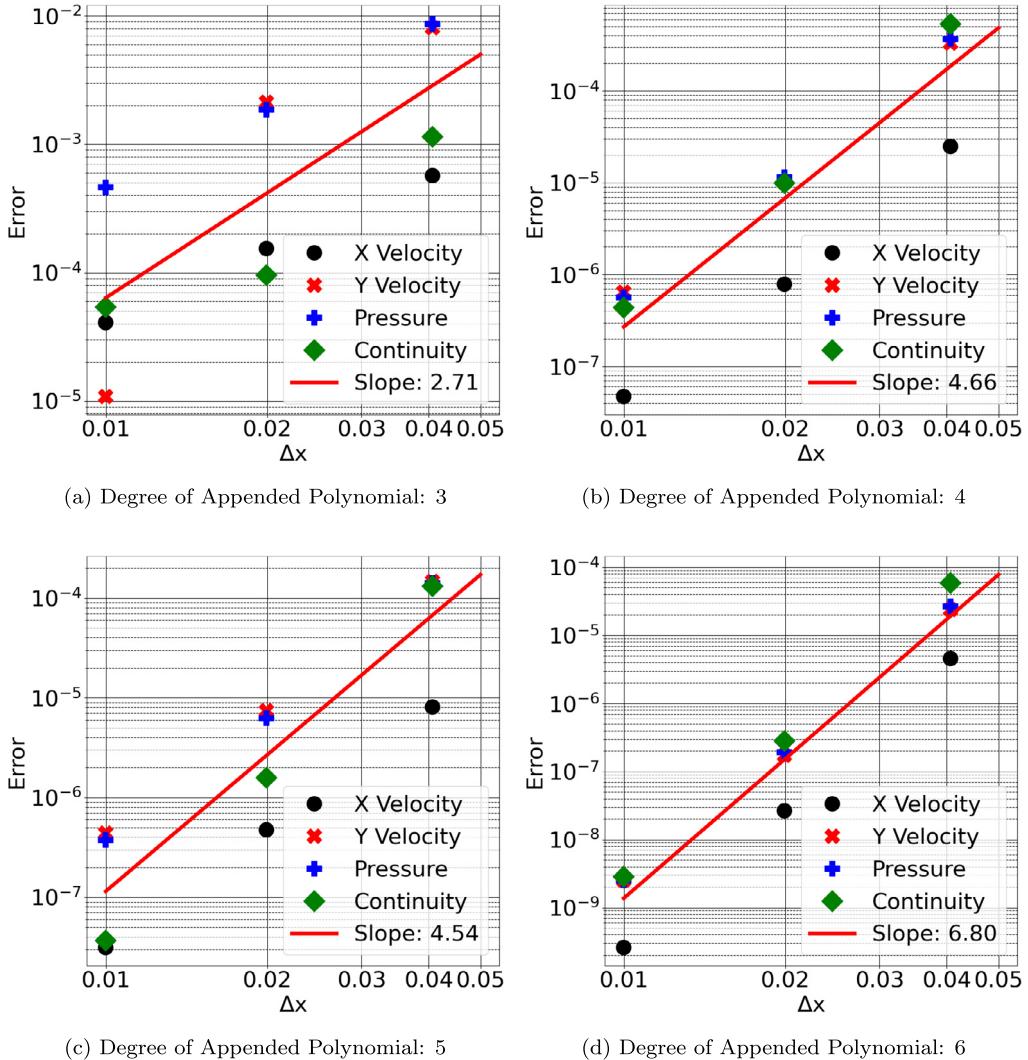


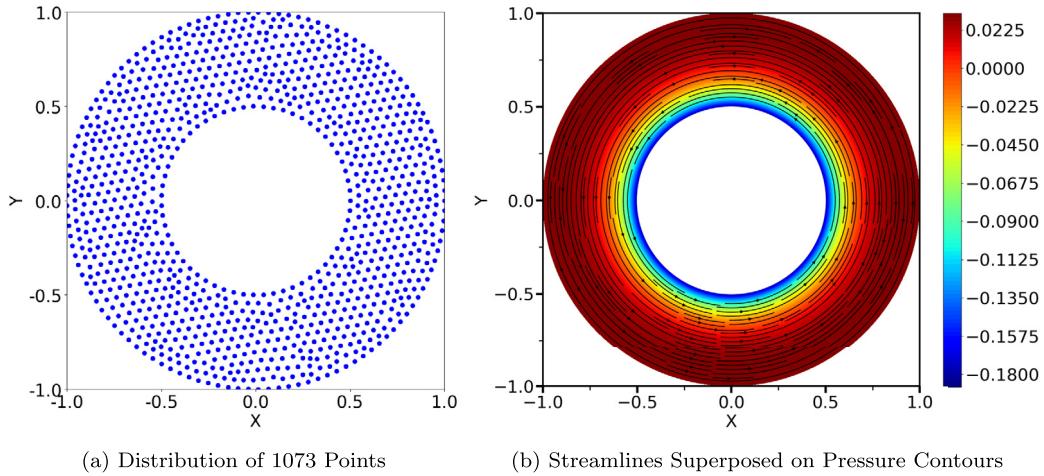
Fig. 6. Errors for Kovasznay flow.

5.2. Cylindrical Couette flow

The cylindrical Couette flow is a simple one dimensional test problem extensively used to verify a Navier-Stokes solver. It also serves to evaluate the order of convergence of the discretization error because of the availability of an exact analytical solution. Although the flow is one dimensional in cylindrical polar coordinates, in Cartesian coordinates the flow is two dimensional. In our method, the coordinate system is Cartesian with the two velocity components aligned with the Cartesian axes X and Y . The geometry with the distribution of points is shown in Fig. 7a. Because of the two dimensional computational domain and low rotational Reynolds number, the formation of longitudinal Taylor vortices is inhibited. We consider the flow to be steady and laminar with the inner cylinder rotating at an angular velocity ω . The outer cylinder is kept stationary. With no slip and no penetration at the boundaries, the analytical value of the tangential velocity (v_θ) as a function of the radial coordinate (r) is given as [65]:

$$v_\theta(r) = r_1 \omega \frac{r_1 r_2}{r_2^2 - r_1^2} \left(\frac{r_2}{r} - \frac{r}{r_2} \right) \quad (23)$$

where r_2 and r_1 denote the radii of the outer and inner cylinders respectively. The Reynolds number is based on inner cylinder's diameter and its tangential velocity. Currently, we have used a Reynolds number of 100 although, the velocity profile is independent of Reynolds number. We have computed this flow by solving the Cartesian form of the momentum equations with boundary conditions given by the rotating and stationary cylinder velocities. The solution is started from null fields and marched in time until a steady state is reached. The momentum equations are first solved for the intermediate

**Fig. 7.** Cylindrical Couette flow.

velocity fields (\hat{u} and \hat{v}). The pressure Poisson equation is then solved with Neumann boundary conditions given by the normal momentum equations at the boundaries.

We considered an aspect ratio $A = (r_2 - r_1)/r_1$ of unity. The streamlines and contours of pressure are shown in Fig. 7b. To investigate the convergence characteristics, three different sets of points are considered: 1073, 5630 and 10738. For each case, the polynomial degree is varied to investigate the convergence of error with the degree of polynomial. Timestep value (Δt) of 1E-3 is used for the two coarser grids which needed around 6000 timesteps to reach a steady state. The finest grid reached steady state in 30000 timesteps with $\Delta t = 2E-4$. The L_1 norm of the error between the analytical and numerical solutions is plotted in Fig. 8. It can be seen that the errors decrease rapidly with the mesh size. In this case as well, we see the improvement in convergence with increasing degree of appended polynomial.

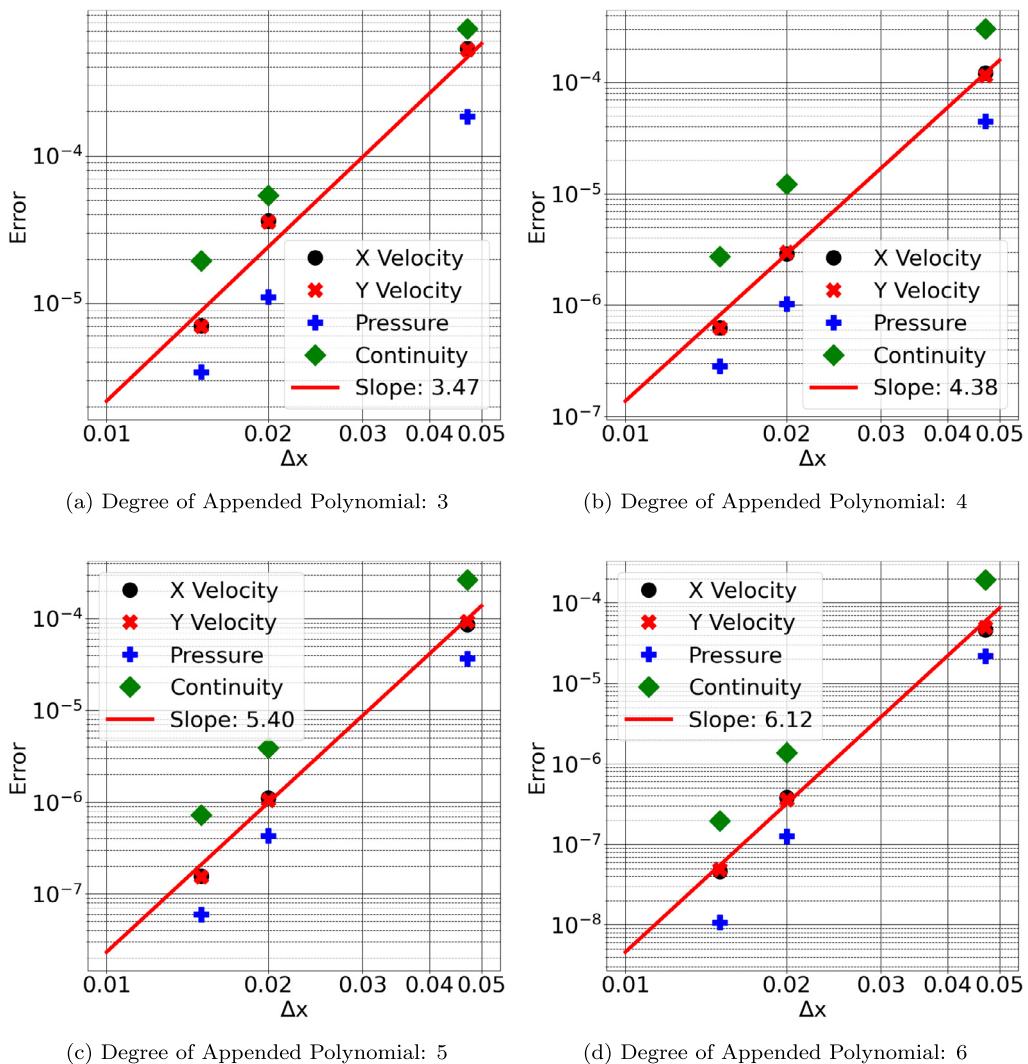
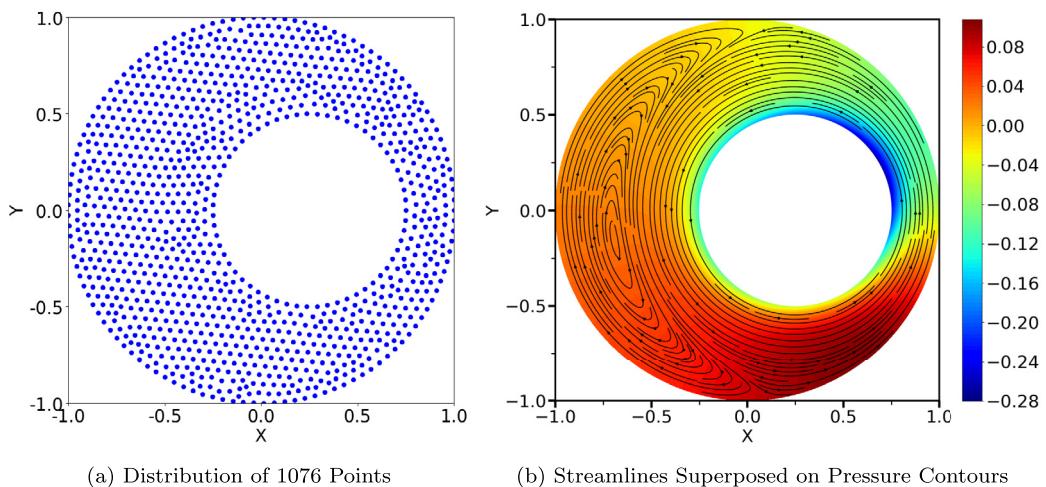
5.3. Eccentric cylindrical Couette flow

The third problem considered is the flow between two rotating cylinders with their axes of rotation parallel, but not coincident. The aspect ratio $A = (r_2 - r_1)/r_1$ and eccentricity $e = d/(r_2 - r_1)$ (where d is the perpendicular distance between axes of the cylinders) are fixed at 2 and 0.5 respectively. The Reynolds number based on tangential velocity and radius of the inner cylinder is set to be 50. Varying numbers of points, similar to the case of concentric cylinder, are considered (1076, 5014 and 10533). Fig. 9a shows an example of point distribution. The contours of the pressure superposed with streamlines are plotted in Fig. 9b. These are in agreement with a previous study performed with a ghost fluid Lattice Boltzmann method [66]. Timestep value and the number to reach steady state are similar to the case of cylindrical Couette flow mentioned in section 5.2.

The degree of the appended polynomial is varied from 3 to 6. A fine grid solution is first generated on a set of 55419 points with a polynomial degree of 6. The solution is further interpolated from the scattered points using the same PHS-RBF interpolation at 15 points along the vertical line: $X = -0.35$. These solutions are documented in Table 1. Differences between the results with different point sets and the most accurate estimates are then plotted as a function of grid spacings in Fig. 10. It is seen that the error decreases with refinement. Moreover, the rate of convergence (slope of the best fit line) increases with the polynomial degree.

5.4. Flow in an elliptical annulus with rotating inner cylinder

The fourth problem considered is the flow in the annular space formed between an elliptical outer enclosure and an inner circular cylinder with the two axes coincident (Fig. 11a). The flow is generated by the rotation of the inner cylinder at an angular velocity ω . The flow is similar to that in section 5.2, except that the outer enclosure is an ellipse, which demonstrates another application to a complex geometry. Fig. 11b shows the contours of the pressure and streamlines from the finest set of points. The most refined point set consisting 51412 points with a polynomial degree of 6, is used to generate the reference solution. Three other sets of points are considered (1036, 5057 and 10440), and the degree of appended polynomial is varied from 3 to 6 for each point set. Timestep value and the number to reach steady state are similar to the case of cylindrical Couette flow mentioned in section 5.2. The differences between the reference solution, and the other calculations are evaluated at 15 points along the vertical axis of the elliptical annulus ($X = 0$) by interpolating the solution from the scattered points. The interpolated values are documented in Table 2. Fig. 12 plots the L_1 norm of the errors for the various cases. As before, it can be seen that the order of convergence improves by approximately unity with increasing degree of appended polynomial.

**Fig. 8.** Errors for cylindrical Couette flow.**Fig. 9.** Eccentric cylindrical Couette flow.

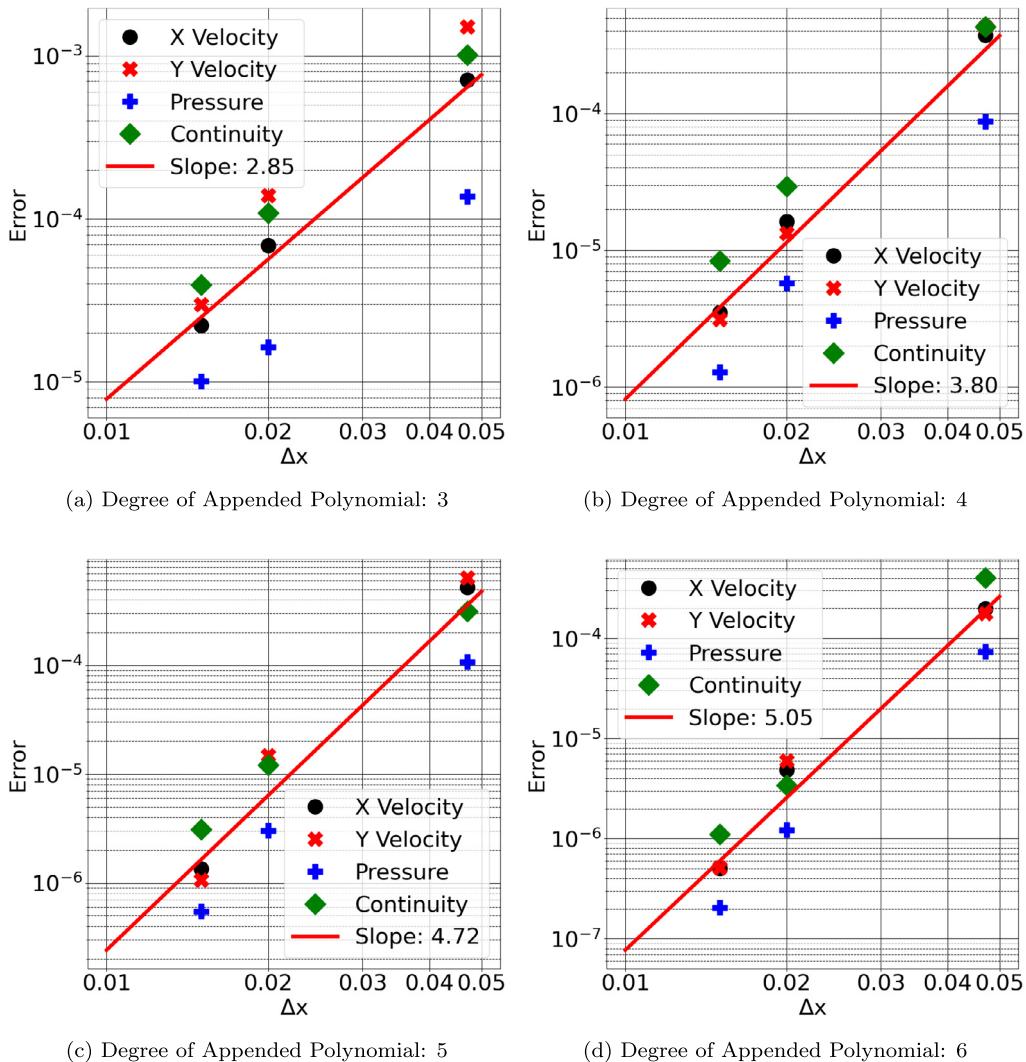


Fig. 10. Errors for eccentric cylindrical Couette flow.

Table 1
Reference values for eccentric cylindrical Couette flow.

X	Y	u	v	p
-0.35	-8.000000E-01	-1.9079592E-02	4.7313446E-03	4.9115286E-02
-0.35	-6.8571429E-01	8.1661301E-03	-1.9183109E-02	4.5560404E-02
-0.35	-5.7142857E-01	6.1684959E-02	-7.5648774E-02	4.0809071E-02
-0.35	-4.5714286E-01	1.2197777E-01	-1.6810504E-01	3.3351733E-02
-0.35	-3.4285714E-01	1.6431517E-01	-2.9251677E-01	2.0851835E-02
-0.35	-2.2857143E-01	1.6299774E-01	-4.3025881E-01	2.5791952E-03
-0.35	-1.1428571E-01	1.0359107E-01	-5.4349940E-01	-1.7126709E-02
-0.35	2.7755576E-17	6.7240764E-04	-5.8731376E-01	-2.9612243E-02
-0.35	1.1428571E-01	-1.0148266E-01	-5.4042011E-01	-3.0062418E-02
-0.35	2.2857143E-01	-1.5882169E-01	-4.2423074E-01	-2.3165751E-02
-0.35	3.4285714E-01	-1.5830580E-01	-2.8515341E-01	-1.7072929E-02
-0.35	4.5714286E-01	-1.1702532E-01	-1.6228223E-01	-1.5428133E-02
-0.35	5.7142857E-01	-6.1749566E-02	-7.3639191E-02	-1.7157194E-02
-0.35	6.8571429E-01	-1.4637270E-02	-2.1045138E-02	-2.0009898E-02
-0.35	8.000000E-01	1.0178600E-02	1.4208480E-03	-2.2632250E-02

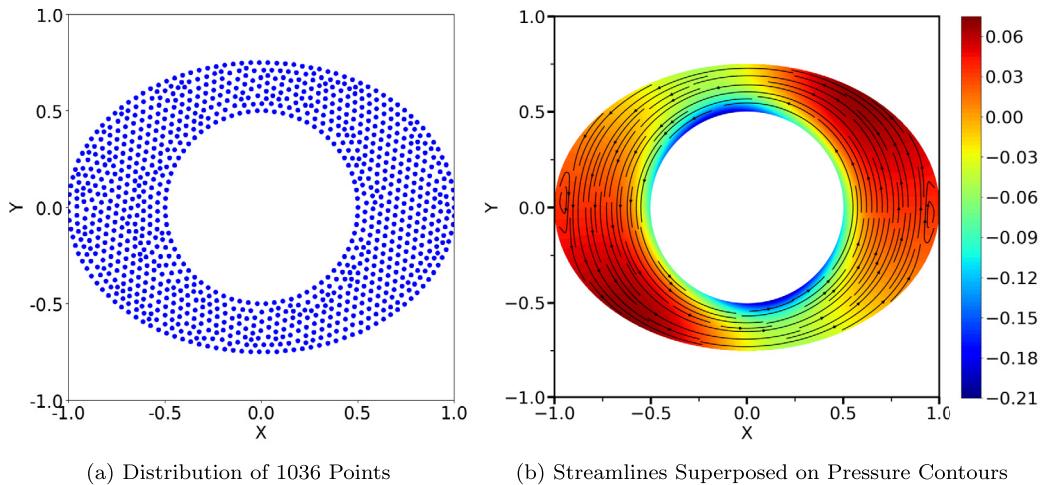
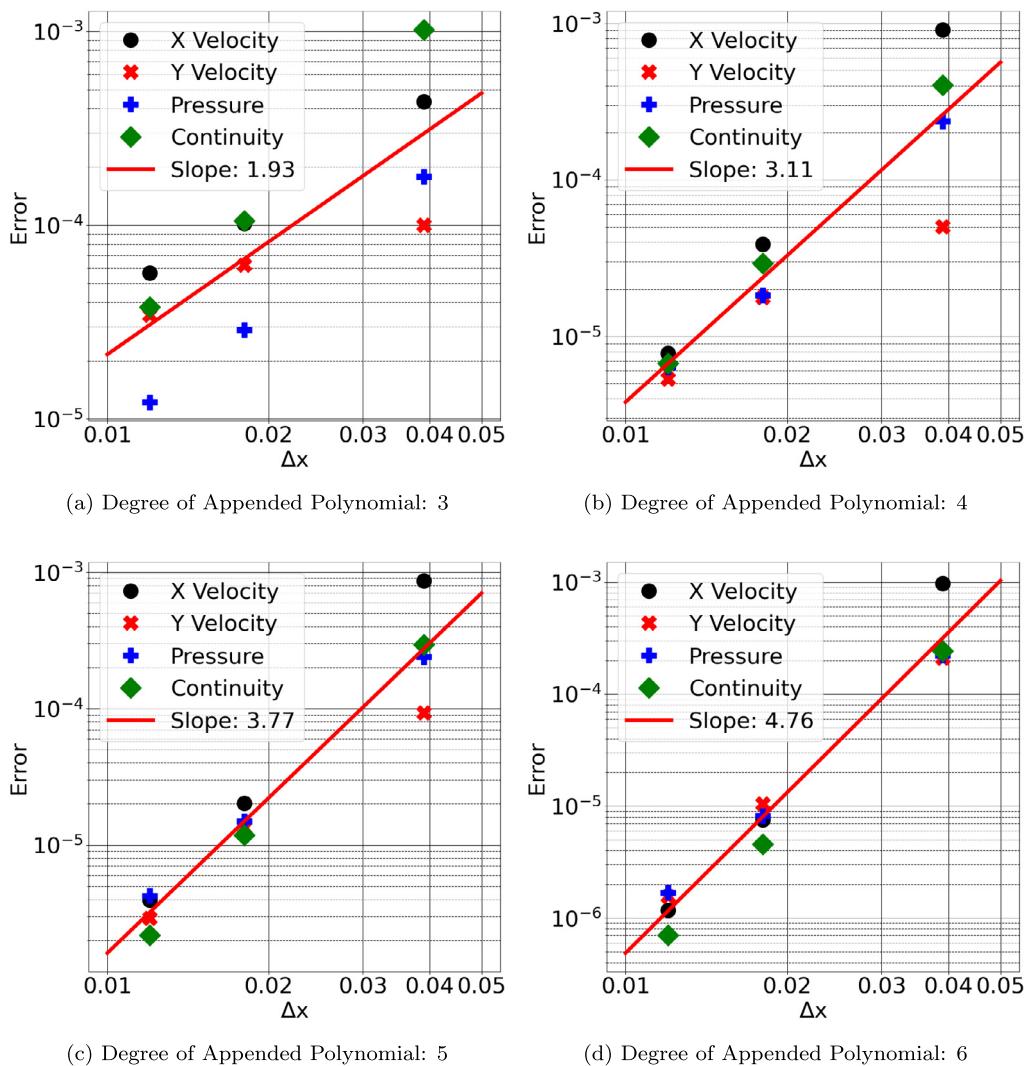
**Fig. 11.** Flow in an elliptical annulus with rotating inner cylinder.**Fig. 12.** Errors for flow in an elliptical annulus with rotating inner cylinder.

Table 2
Reference values for flow in an elliptical annulus with rotating inner cylinder.

X	Y	u	v	p
0.00	5.0000000E-01	-1.0000000E+00	1.2833611E-14	-1.1921286E-01
0.00	5.1785714E-01	-9.4873251E-01	1.5998650E-05	-8.5927834E-02
0.00	5.3571429E-01	-8.9293824E-01	-2.1156478E-04	-5.7615108E-02
0.00	5.5357143E-01	-8.3368481E-01	-8.0766410E-04	-3.3924495E-02
0.00	5.7142857E-01	-7.7184033E-01	-1.6753742E-03	-1.4429810E-02
0.00	5.8928571E-01	-7.0790039E-01	-2.6278077E-03	1.3266544E-03
0.00	6.0714286E-01	-6.4198095E-01	-3.4644281E-03	1.3800200E-02
0.00	6.2500000E-01	-5.7389110E-01	-4.0154090E-03	2.3424082E-02
0.00	6.4285714E-01	-5.0323668E-01	-4.1680020E-03	3.0601974E-02
0.00	6.6071429E-01	-4.2952646E-01	-3.8821536E-03	3.5708587E-02
0.00	6.7857143E-01	-3.5226288E-01	-3.1984837E-03	3.9094469E-02
0.00	6.9642857E-01	-2.7100522E-01	-2.2395268E-03	4.1091713E-02
0.00	7.1428571E-01	-1.8539674E-01	-1.2041977E-03	4.2018226E-02
0.00	7.3214286E-01	-9.5151425E-02	-3.5528821E-04	4.2179153E-02
0.00	7.5000000E-01	-9.5963561E-13	1.4881676E-13	4.1865046E-02

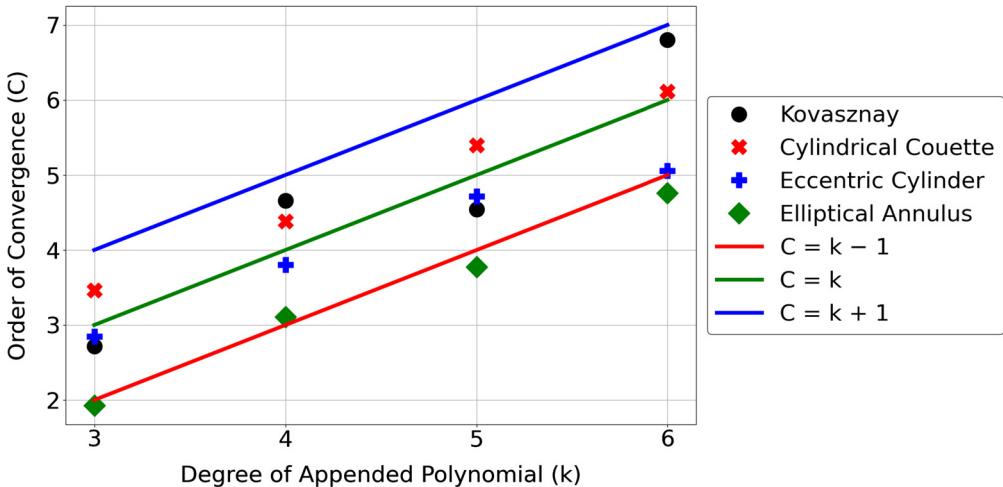


Fig. 13. Order of convergence with polynomial degree for all four problems.

5.5. Composite plot of observed convergence

Section 3.1 demonstrates that when polynomials with maximum degree k are appended to the PHS-RBF, the gradient and Laplacian estimations are $\mathcal{O}(k)$ and $\mathcal{O}(k-1)$ accurate respectively. Navier-Stokes equations have terms with both gradient and Laplacian operators. The Reynolds number determines the relative strengths of the convection and diffusion terms which have the gradient and Laplacian operators respectively. Hence, order of accuracy of the fractional step method is expected to lie in between $\mathcal{O}(k-1)$ and $\mathcal{O}(k)$. In this section, the order of convergence (slope of best fit line) for polynomial degrees of 3 to 6 for all the 4 cases (sections 5.1 to 5.4) is combined in a single scatter plot. Lines $k-1$, k and $k+1$ are also plotted for reference in Fig. 13. It can be seen that most of the points follow the expected trend. All the cases display exponential convergence i.e., monotonous increase in the order of convergence with the degree of appended polynomial.

6. Simulation of transient problem

In this section, we use the problem described by Bell et al. [67] in order to analyze the rate of convergence of spatial errors for a flow problem with temporal evolution. The flow is initialized with the following stream function on a unit square domain:

$$\Psi(x, y) = \frac{\sin^2(\pi x)\sin^2(\pi y)}{\pi} \quad (24)$$

Thus, the initial velocities are given by:

$$u(x, y) = \frac{\partial \Psi}{\partial y} = \sin^2(\pi x)\sin(2\pi y) \quad (25)$$

$$v(x, y) = -\frac{\partial \Psi}{\partial x} = -\sin(2\pi x)\sin^2(\pi y)$$

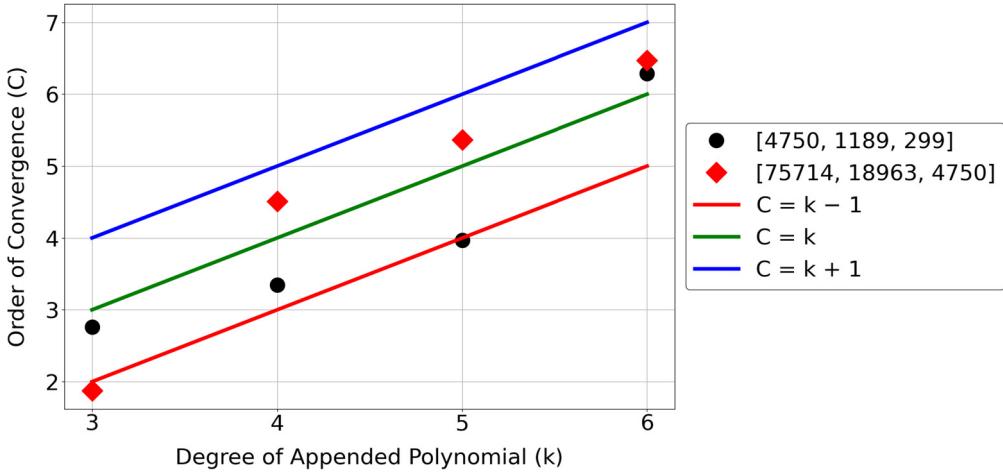


Fig. 14. Order of convergence with polynomial degree for transient simulation.

This velocity field is divergence free and has homogeneous Dirichlet boundary conditions on all the sides of the unit square. The maximum value of initial velocity is unity. Reynolds number is defined as: $Re = \rho u_0 L / \mu$ where, density (ρ), characteristic velocity (u_0) and length of the square cavity (L) are all set to unity. The dynamic viscosity (μ) is calculated based on the prescribed Reynolds number of 100. We have integrated the velocity field to time 0.5. Five different point distributions are chosen with [299, 1189, 4750, 18963, 75714] nodes corresponding to an average Δx of [0.0578, 0.0290, 0.0148, 0.0073, 0.0036] respectively. These grids give successive refinements approximately by a factor of two in each direction. Since the points are not located on a Cartesian grid, it is difficult to get grids with precise refinements. For each grid, degrees of appended polynomials are varied from 3 to 6 thus, giving four simulations. All the simulations are performed with $\Delta t = 5E-5$ which satisfies the stability limit for the finest point distribution. At the final time of 0.5, the X and Y components of the velocities are interpolated using the PHS-RBF function to 100 points along vertical and horizontal center lines. Let $[f^h, f^{2h}, f^{4h}]$ denote the interpolated velocities along the center lines for point distributions with three successive refinements. In this case, two such triplets of point distributions are considered: [4750, 1189, 299] and [75714, 18963, 4750]. Thus, the numerical values of f can be expressed in terms of the truncation errors as follows:

$$\begin{aligned} f^h &= \alpha + \beta(h)^C + \mathcal{O}(\Delta t) + \text{H.O.T.} \\ f^{2h} &= \alpha + \beta(2h)^C + \mathcal{O}(\Delta t) + \text{H.O.T.} \\ f^{4h} &= \alpha + \beta(4h)^C + \mathcal{O}(\Delta t) + \text{H.O.T.} \end{aligned} \quad (26)$$

where, C is the order of convergence, h is the average value of Δx and H.O.T. denote the higher order terms. Since the explicit Euler method is used for integrating in time, the temporal error is shown by first order accuracy: $\mathcal{O}(\Delta t)$. Subtracting the successive equations eliminates α :

$$\begin{aligned} f^{2h} - f^h &= \beta((2h)^C - (h)^C) \\ f^{4h} - f^{2h} &= \beta((4h)^C - (2h)^C) \end{aligned} \quad (27)$$

As mentioned before, since a fixed Δt of $5E-5$ is used for all the cases, we neglect the differences in the temporal errors in the above equations. Taking ratio followed by log to the base 2 gives an expression for the order of convergence C (Richardson extrapolation):

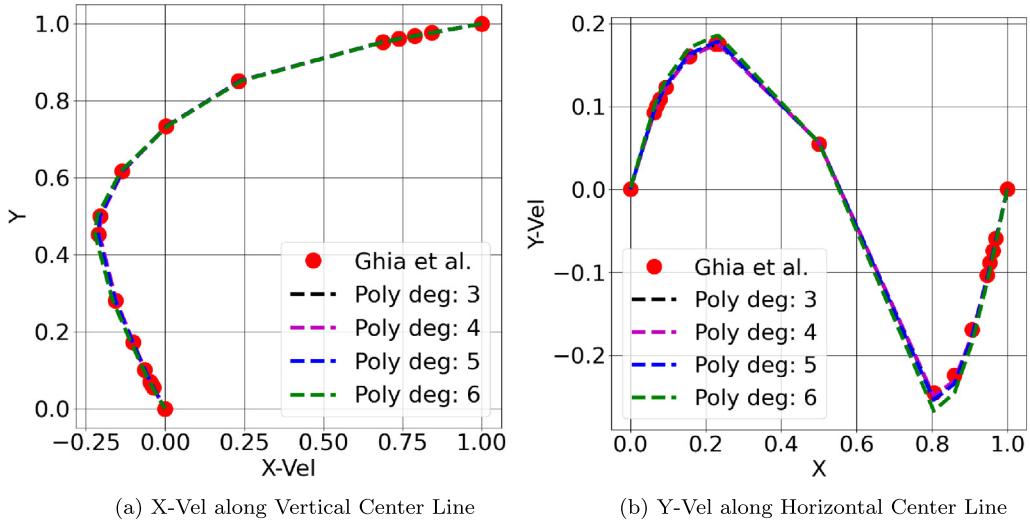
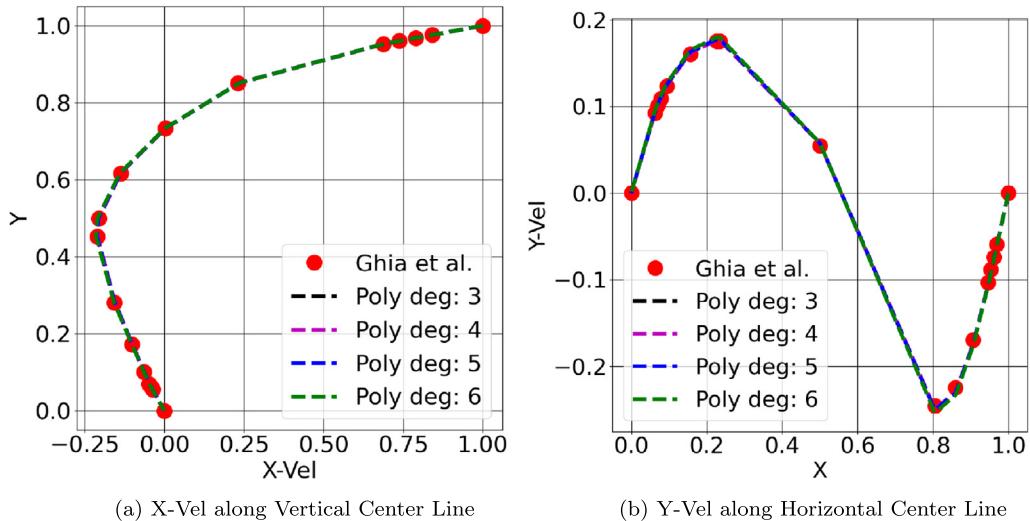
$$C = \log_2 \left(\frac{f^{4h} - f^{2h}}{f^{2h} - f^h} \right) \quad (28)$$

Equation (28) is applied to the L1 norm of the differences between the interpolated velocity values along the center lines. The orders of convergence are plotted in Fig. 14 for both the subsets of point distributions. All the estimated orders of convergence lie inside the expected range.

7. Applications

7.1. Flow in a driven cavity

Flow in a square cavity with top wall moving is a popular benchmark problem used for verification of numerical algorithms. We have applied the fractional step method described in this paper to simulate the fluid flow in the driven cavity

Fig. 15. $Re = 100$, $\Delta x = 0.0093$, 11515 nodes: comparison with Ghia et al. [68].Fig. 16. $Re = 100$, $\Delta x = 0.0047$, 45437 nodes: comparison with Ghia et al. [68].

and the velocities along the center lines are compared with the solution reported by Ghia et al. [68]. Reynolds number is defined as: $Re = \rho u_t L / \mu$ where, density (ρ), top wall velocity (u_t) and length of the square cavity (L) are all set to unity. The dynamic viscosity (μ) is calculated based on the prescribed Reynolds number. We have simulated for Reynolds numbers of 100 and 400 with two grid resolutions (average Δx of 0.0093 and 0.0047) and four degrees of appended polynomials (3, 4, 5 and 6) for each case. The velocities obtained at the discrete points are interpolated using the PHS-RBF functions to the coordinates tabulated in the paper of Ghia et al. [68] for comparison. Figs. 15 to 18 show that the present numerical estimates plotted by dashed lines are in good agreement with the reported solutions.

7.2. Flow over circular cylinder

Vortex shedding over bluff bodies is a popular problem in fluid mechanics with multiple applications in engineering. Here, we apply the present algorithm to simulate laminar unsteady flow over cylinder for two Reynolds numbers and compare the Strouhal number, lift and drag coefficients with previously published values. Fig. 19 shows a schematic of the flow domain. The left boundary is set to uniform inlet flow with unit velocity. Pressure boundary condition is applied at the outlet on the right side. The top and bottom boundaries are set to the symmetry boundary condition. The dynamic viscosity μ is computed based on the Reynolds number $Re = \rho U_i D / \mu$ where, density $\rho = 1$, inlet velocity $U_i = 1$ and diameter $D = 1$. We simulate for two Reynolds numbers (100 and 200) which are known to give unsteady flow. Two different point resolutions are generated using Gmsh [63] with 80 and 100 points on the cylinder. To reduce the computational cost,

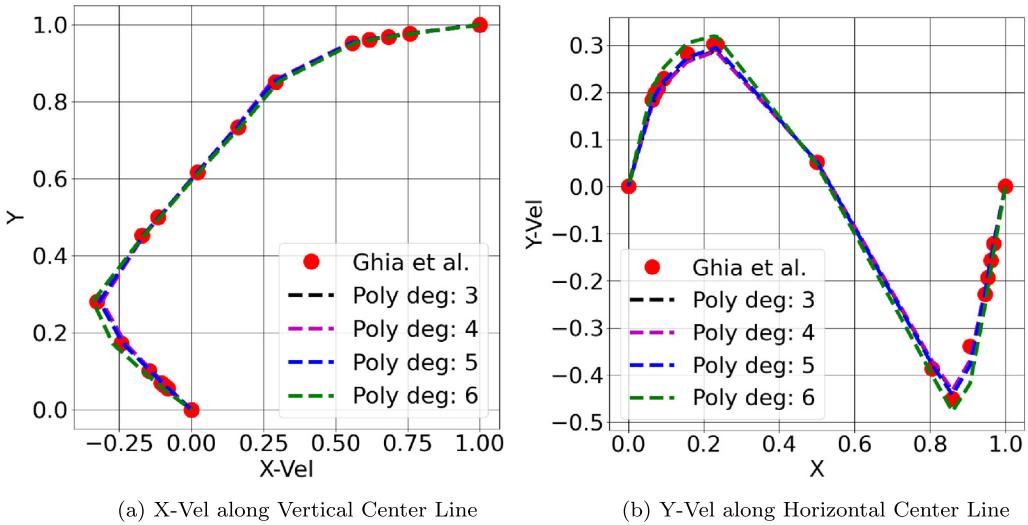
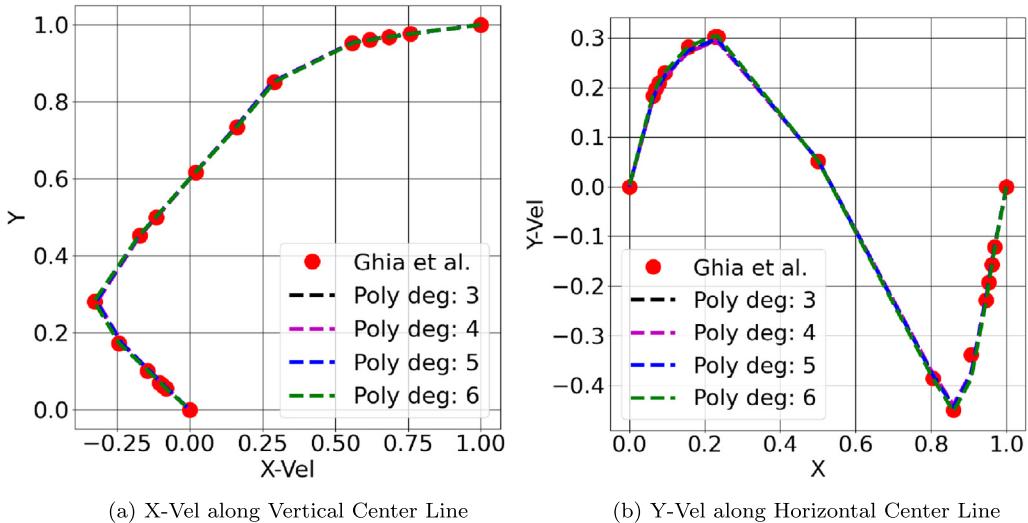
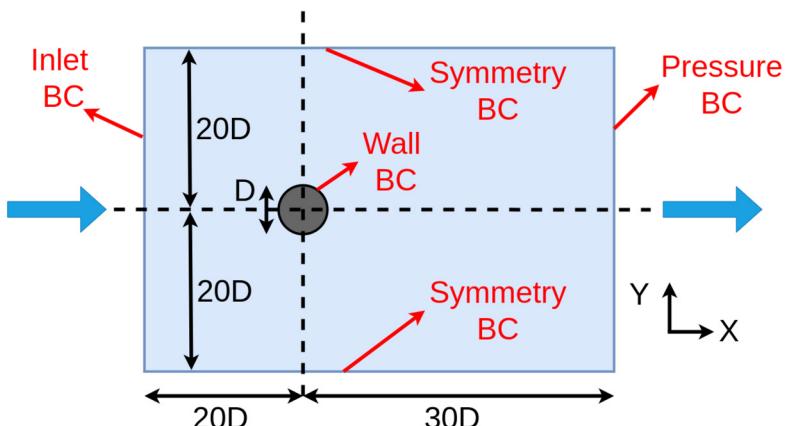
Fig. 17. $Re = 400$, $\Delta x = 0.0093$, 11515 nodes: comparison with Ghia et al. [68].Fig. 18. $Re = 400$, $\Delta x = 0.0047$, 45437 nodes: comparison with Ghia et al. [68].

Fig. 19. Domain (not to scale).

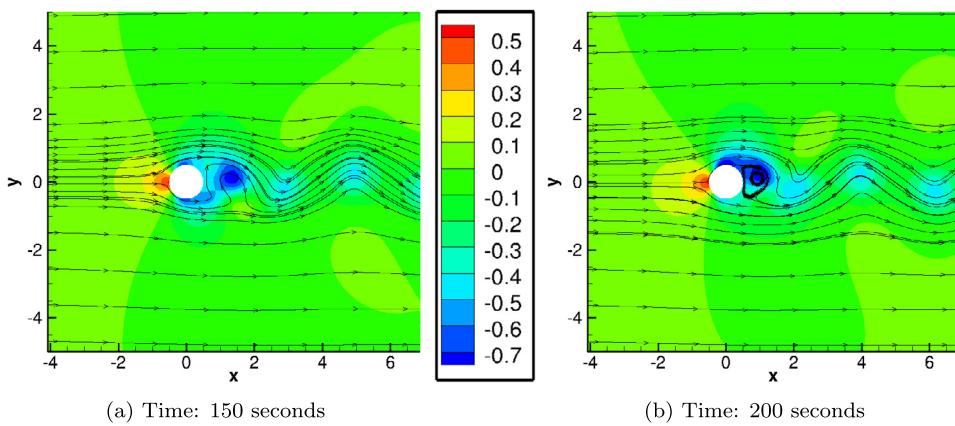


Fig. 20. Pressure contours with streamlines for Reynolds number: 200.

points are coarsened by a factor of 5 in the far stream region. This gives 105534 and 163798 nodes with an average Δx of 0.1377 and 0.1105 respectively. A polynomial degree of 5 is used for all simulations. The computations are started with a uniform flow field and time marching is performed using the second order Adams-Basforth method till stationary fields are obtained. Fig. 20 plots pressure contours with streamlines at two different time instants for a Reynolds number of 200.

Components of the stress tensor acting on the cylinder in terms of pressure, strain rates and dynamic viscosity are given by [65]:

$$\sigma_{xx} = -p + 2\mu \frac{\partial u}{\partial x} \quad \sigma_{yy} = -p + 2\mu \frac{\partial v}{\partial y} \quad \sigma_{xy} = \sigma_{yx} = \mu \left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \right) \quad (29)$$

Total force per unit length in the axial directions is computed by integrating appropriate stress components over the cylindrical surface:

$$\begin{aligned} F_x &= \frac{D}{2} \int_0^{2\pi} (\sigma_{xx} \cos \theta + \sigma_{yx} \sin \theta) d\theta \\ F_y &= \frac{D}{2} \int_0^{2\pi} (\sigma_{yy} \sin \theta + \sigma_{xy} \cos \theta) d\theta \end{aligned} \quad (30)$$

The above integrals are computed using the Simpson's rule [69] by first interpolating strain rates and pressure to 360 uniform points over the cylindrical surface. Drag (C_D) and lift (C_L) coefficients are defined as follows:

$$C_D = \frac{F_x}{\rho U_i^2 D / 2} \quad C_L = \frac{F_y}{\rho U_i^2 D / 2} \quad (31)$$

Fig. 21 plots temporal variation of the lift and drag coefficients for both the Reynolds numbers. We see a sinusoidal variation as expected [70]. The Strouhal number is defined as $St = fD/U_l$ where, f is the frequency of the lift coefficient. Table 3 compares these coefficients and Strouhal number for both the points resolutions with literature. Our computed estimates fall within the ranges reported in several previous studies.

7.3. Application to Euler equations

Although our primary focus in this work has been towards incompressible flows, we have also investigated its performance to compute hyperbolic flows governed by the Euler equations. For such equations, there is no physical viscosity and the Peclet numbers are infinity. Here, we have studied the evolution of a sharp shear layer in a double periodic domain. In order to implement a periodic boundary condition, the cloud of points is modified by coupling to the points on the other side of the domain. For instance, points near the right boundary are coupled with the points near the left boundary and vice-versa. Velocity field is initialized as follows [67]:

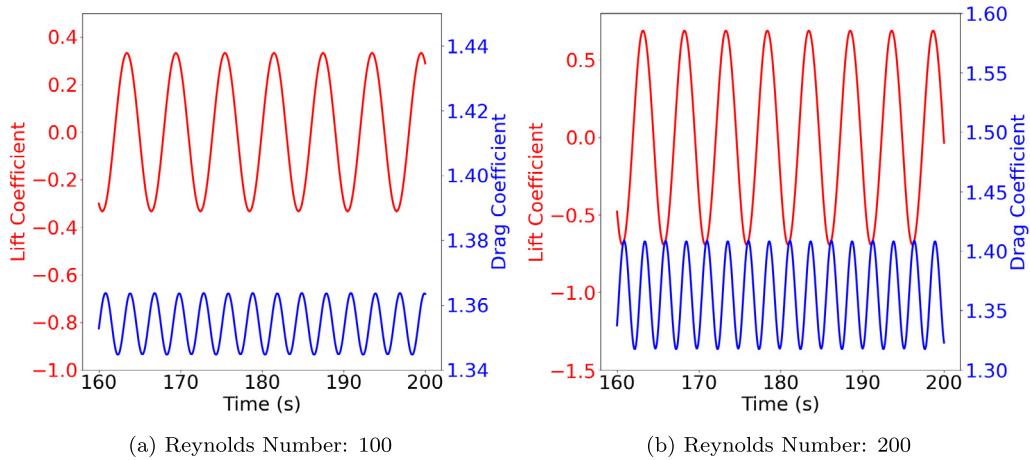


Fig. 21. Lift and drag coefficients (163798 nodes, average $\Delta x = 0.1105$, degree of appended polynomial: 5).

Table 3
Comparison of drag and lift coefficients with Strouhal number.

Reynolds number	Reference	Drag coefficient	Lift coefficient	Strouhal number
100	Braza et al. [71]	1.364 ± 0.015	± 0.25	0.16
	Liu et al. [72]	1.35 ± 0.012	± 0.339	0.165
	Ding et al. [70]	1.325 ± 0.008	± 0.28	0.164
	Present work: $\Delta x = 0.1377$	1.368 ± 0.009568	± 0.3359	0.1662
	Present work: $\Delta x = 0.1105$	1.354 ± 0.009361	± 0.3327	0.1663
200	Belov et al. [73]	1.19 ± 0.042	± 0.64	
	Braza et al. [71]	1.4 ± 0.05	± 0.75	0.193
	Liu et al. [72]	1.31 ± 0.049	± 0.69	0.192
	Ding et al. [70]	1.327 ± 0.045	± 0.60	0.196
	Present work: $\Delta x = 0.1377$	1.395 ± 0.04553	± 0.6976	0.1968
	Present work: $\Delta x = 0.1105$	1.364 ± 0.04451	± 0.6899	0.1972

$$u = \begin{cases} \tanh(30(y - 0.25)) & \text{if } y \leq 0.5 \\ \tanh(30(0.75 - y)) & \text{if } y > 0.5 \end{cases} \quad (32)$$

$$v = 0.05 \sin(2\pi x)$$

The Euler equations are solved by the fractional step method with the second order Adams-Basforth method for time integration till a time of 2 seconds. It is well known that the discretized differentiation matrices for Euler equations can contain eigenvalues with positive real parts [45,48,53,74,75]. This leads to spurious growth of the numerical solutions. Hence, following the previous works of Shankar and Fogelson [53] and Flyer et al. [48], we have added artificial hyper-viscosity terms $\kappa(\nabla^2)^\alpha u$ and $\kappa(\nabla^2)^\alpha v$ to the momentum equations in order to stabilize the time integration where, $\alpha \in \mathbb{N}$ and $\kappa = (-1)^{1-\alpha} 2^{-6} (\Delta x)^{2\alpha-1}$. We have currently experimented with $\alpha = 2$.

We have used 3 different point resolutions with 18711, 75206 and 301638 nodes which correspond to an average Δx of 0.0073, 0.0037 and 0.0018 respectively. Figs. 22 to 24 plot the contours of vorticity at four time instances for each of the point distributions. The initial condition has a steep gradient in the velocity field with the fluid in the middle section moving a direction opposite to the top and bottom regions. This evolves into multiple vortices with time eventually forming thin shear layers around these vortices. All three point distributions are able to capture the initial evolution until 0.8 seconds accurately. However, beyond this smaller point spacing is needed to resolve the thin layers with sharp gradients. Thus, the finer point distributions show sharper shear layers for the time instants of 1.2 and 1.8 seconds. These plots are in agreement with the results presented by Bell et al. [67].

Because of the absence of physical diffusion in the Euler equations, the kinetic energy defined as $\int(u^2 + v^2)dx dy$ [67] should remain constant in time. To compute this quantity, the velocity values are first interpolated using the PHS-RBF kernels with appended polynomials to a uniform Cartesian grid of points with spacing approximately equal to the average Δx corresponding to each point distribution. Simpson's rule [69] is further used to evaluate the integral. We see that the

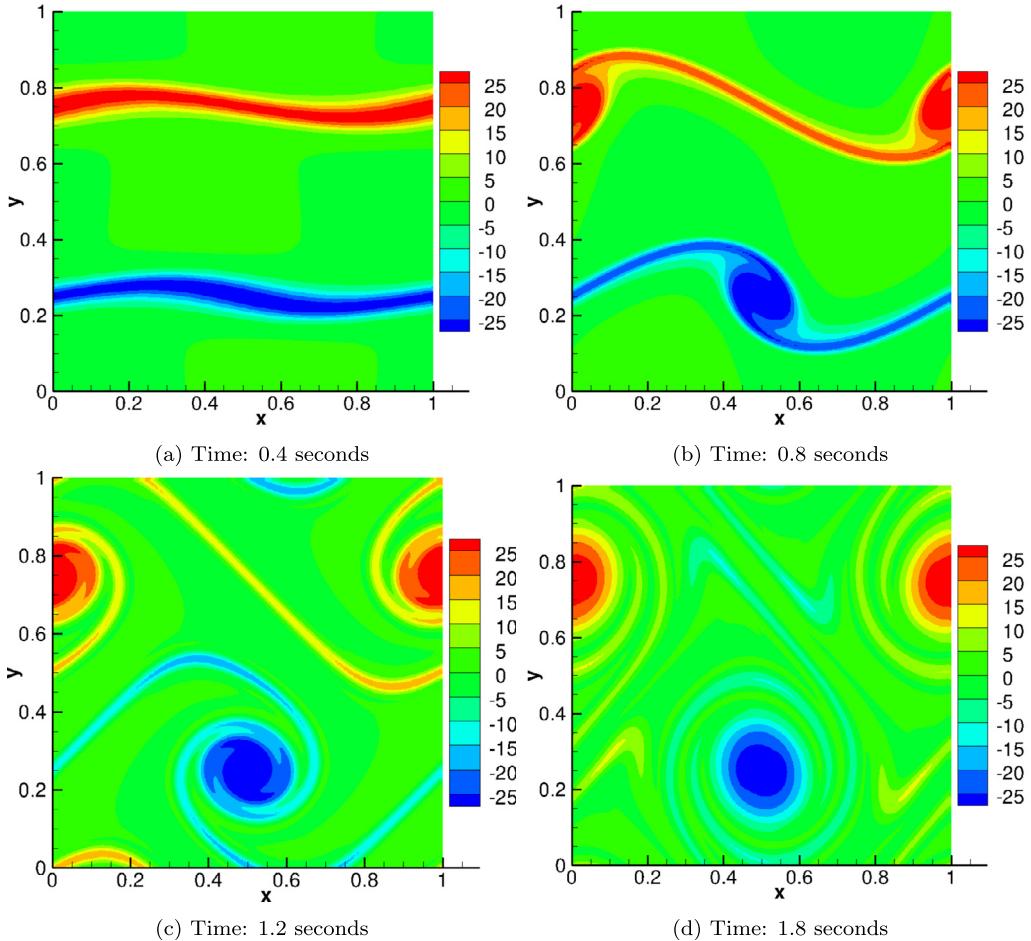


Fig. 22. Computed vorticity with 18711 points ($\Delta x = 0.0073$) and degree of appended polynomial of 6.

change in energy with time shown in Fig. 25 is small for all the cases. The maximum variations in kinetic energy for the three point distributions are 0.307, 0.075 and 0.019 percent respectively. Refinement improves accuracy as expected since adding more points helps in resolving the thin shear layers around the vortices. However, we observe that the rate of convergence is second-order instead of the expected fifth order. The two probable causes for this reduced accuracy could be the current form and magnitude of the hyper-viscosity term and insufficient number of points in the thin shear layers that develop in time. Although Shankar and Fogelson [53] have shown high order accuracy with the same hyper-viscosity term, their study considered only the linear scalar transport equation. The performance of such a term for nonlinear Euler equations is a topic of future research. In addition, high order polynomials of degree k require a cloud size given by $(k+1)(k+2)$ for two dimensional problems. For the thin shear layers, even with the finest grid, there are insufficient number of points to resolve the sharp gradients. In future, local mesh refinement techniques will be developed for such sharp fronts.

8. Conclusions

This paper presents a high order accurate meshless method for computing incompressible fluid flows in complex domains. Instead of using a finite volume or finite element method, it uses scattered points to discretize the partial differential equations. A given variable is interpolated between scattered points using the polyharmonic splines radial basis functions (PHS-RBF) with appended polynomials of a high degree. The basis functions are differentiated to calculate first and second derivatives. A fractional step algorithm for incompressible Navier-Stokes equations has been developed with time marching. The momentum equations are first solved explicitly without the pressure gradient terms, and a pressure Poisson equation is then solved to project the intermediate velocities to a divergence-free space. The computed pressure field is used to correct the velocities, resulting in a divergence-free field.

A modular computer software (MeMPhyS [56]) has been developed in C++ and applied to eight flow problems. The first two are flows with an exact analytical solution, which form verification cases for the software and the numerical method.

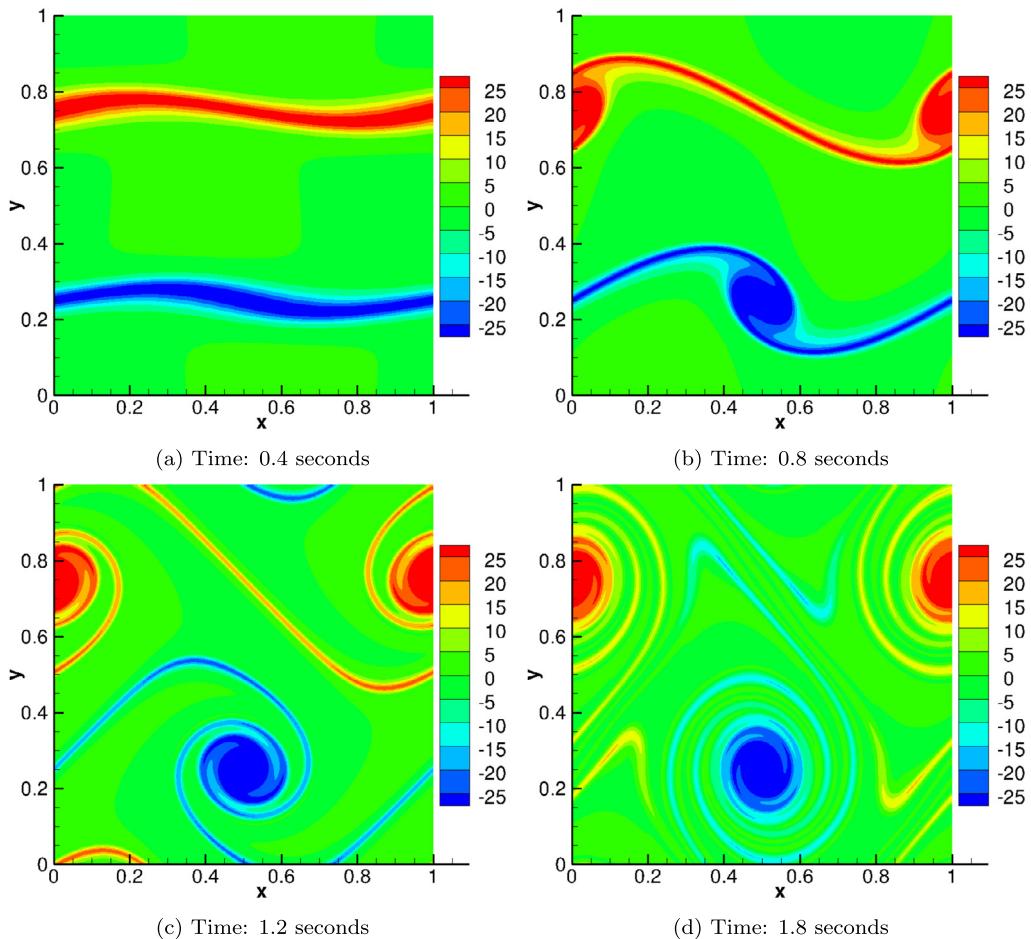


Fig. 23. Computed vorticity with 75206 points ($\Delta x = 0.0037$) and degree of appended polynomial of 6.

The next two cases demonstrate the ability of the method to represent complex domains. Systematic computations with varying point resolutions and increasing degree of appended polynomials have been done to investigate the discretization accuracy. Benchmark solutions of two problems have been generated using large numbers of scattered points and high degree of polynomials. Accuracy is evaluated by comparing the computed solutions with benchmark values. Further, a sample problem with a divergence free initial velocity field is simulated which develops to a null solution with time. For this case, the spatial accuracy is estimated using the Richardson extrapolation after a fixed amount of physical time at which the temporal variation is significant. It is shown that for a fixed number of points, the discretization error decreases rapidly as the polynomial order increases. Further, for a fixed degree of the polynomial, the discretization error decreases approximately as the average inter-point distance to power of the polynomial degree (between k and $k - 1$). The computed flow fields are presented, along with tabulations of the benchmark profiles at selected lines.

After showing the spatial convergence on four steady state and one transient problems, we have also applied the method to simulate the lid-driven cavity problem. Comparison of the velocities along center lines with a benchmark solution available in the literature shows a good agreement. We have further simulated vortex shedding over circular cylinder for two Reynolds numbers. The Strouhal number, lift and drag coefficients are shown to follow the expected trends in the literature. We have also analyzed the performance of this method for solution of Euler equations over a double periodic square domain. The explicit time integration is stabilized by adding an artificial hyper-viscosity term. The initial flow evolving into thin shear layers around vortices over time is resolved using multiple points distributions. The temporal variation of kinetic energy shows that this approach conserves energy within 0.307% error margin.

The present discretization procedure and solution algorithm have potential to solve practical flow problems in complex geometries. We plan to extend this algorithm to several other flows, including to Direct and Large Eddy Simulations of turbulence in complex domains. Implementation on high-performance computers including GPUs is also being planned.

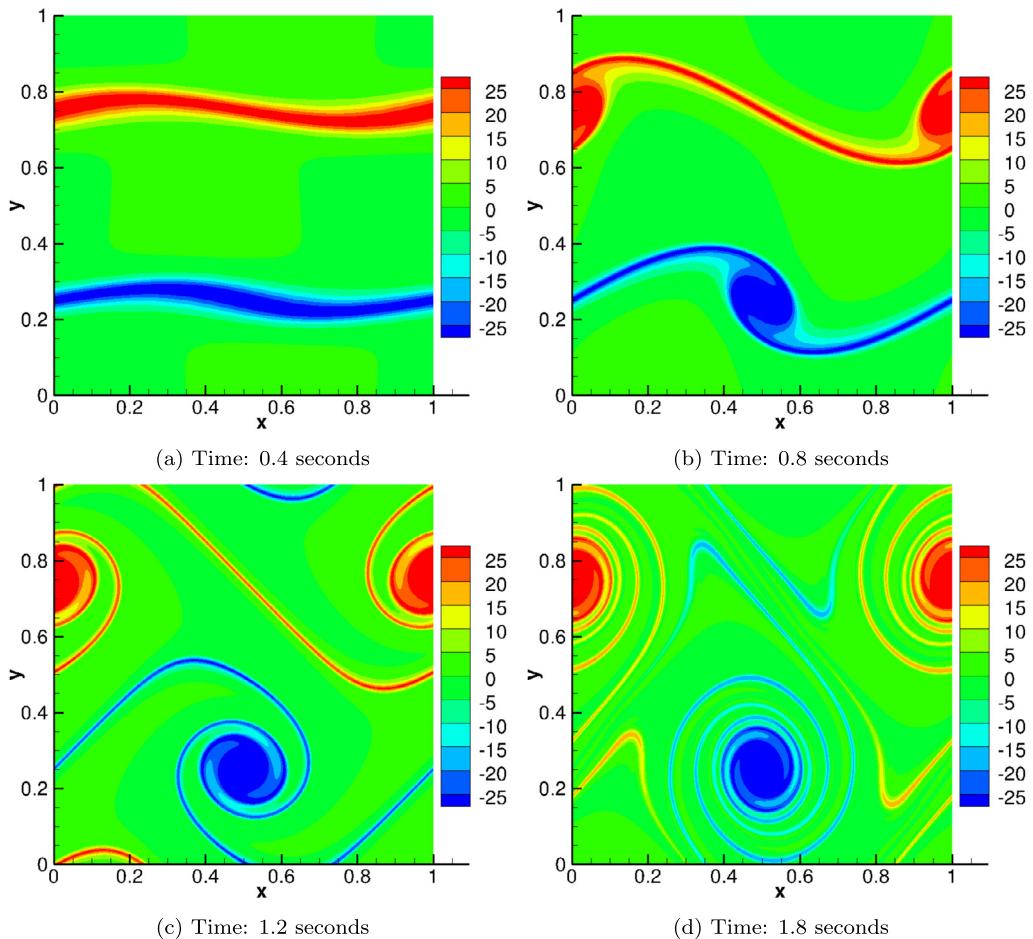


Fig. 24. Computed vorticity with 301638 points ($\Delta x = 0.0018$) and degree of appended polynomial of 6.

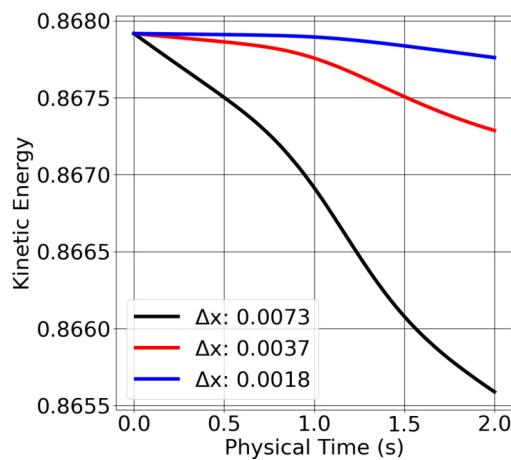


Fig. 25. Temporal variation of kinetic energy.

CRediT authorship contribution statement

Shantanu Shahane: Conceptualization, Methodology, Software development, Analysis, Writing: original draft, review and editing. **Anand Radhakrishnan:** Simulations, Software development, Plotting. **Surya Pratap Vanka:** Conceptualization, Investigation, Project administration, Supervision, Writing: review and editing.

Declaration of competing interest

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

References

- [1] J. Monaghan, Smoothed particle hydrodynamics and its diverse applications, *Annu. Rev. Fluid Mech.* 44 (2012) 323–346.
- [2] T. Ye, D. Pan, C. Huang, M. Liu, Smoothed particle hydrodynamics (SPH) for complex fluid flows: recent developments in methodology and applications, *Phys. Fluids* 31 (2019) 011301.
- [3] A. Zhang, P. Sun, F. Ming, A. Colagrossi, Smoothed particle hydrodynamics and its applications in fluid-structure interactions, *J. Hydron.* 29 (2017) 187–216.
- [4] N. Perrone, R. Kao, A general finite difference method for arbitrary meshes, *Comput. Struct.* 5 (1975) 45–57.
- [5] T. Liszka, J. Orkisz, The finite difference method at arbitrary irregular grids and its application in applied mechanics, *Comput. Struct.* 11 (1980) 83–95.
- [6] L. Gavete, F. Ureña, J. Benito, A. García, M. Ureña, E. Salete, Solving second order non-linear elliptic partial differential equations using generalized finite difference method, *J. Comput. Appl. Math.* 318 (2017) 378–387.
- [7] W. Liu, S. Jun, Y. Zhang, Reproducing kernel particle methods, *Int. J. Numer. Methods Fluids* 20 (1995) 1081–1106.
- [8] T. Huang, A Stabilized Reproducing Kernel Formulation for Shock Modeling in Fluids and Fluid-Structure Interactive Systems, Ph.D. thesis, UC, San Diego, 2020.
- [9] V.G. Patel, N.V. Rachchh, Meshless method—review on recent developments, *Mater. Today, Proc.* 26 (2020) 1598–1603.
- [10] L. Wang, Z. Qian, A meshfree stabilized collocation method (SCM) based on reproducing kernel approximation, *Comput. Methods Appl. Mech. Eng.* 371 (2020) 113303.
- [11] T. Belytschko, Y. Lu, L. Gu, Element-free Galerkin methods, *Int. J. Numer. Methods Eng.* 37 (1994) 229–256.
- [12] M. Abbaszadeh, M. Dehghan, A. Khodadadian, C. Heitzinger, Analysis and application of the interpolating element free Galerkin (IEFG) method to simulate the prevention of groundwater contamination with application in fluid flow, *J. Comput. Appl. Math.* 368 (2020) 112453.
- [13] L. Zhang, J. Ouyang, X. Zhang, On a two-level element-free Galerkin method for incompressible fluid flow, *Appl. Numer. Math.* 59 (2009) 1894–1904.
- [14] T. Liszka, C. Duarte, W. Tworzydlo, hp-Meshless cloud method, *Comput. Methods Appl. Mech. Eng.* 139 (1996) 263–288.
- [15] C. Duarte, The hp cloud method, Ph.D. thesis, University of Texas at Austin, USA, 1996.
- [16] C. Duarte, J. Oden, An hp adaptive method using clouds, *Comput. Methods Appl. Mech. Eng.* 139 (1996) 237–262.
- [17] Y. Chen, J. Lee, A. Eskandarian, An overview on meshless methods and their applications, in: *Meshless Methods in Solid Mechanics*, 2006, pp. 55–67.
- [18] J. Melenk, I. Babuška, The partition of unity finite element method: basic theory and applications, in: *Research Report/Seminar für Angewandte Mathematik*, vol. 1996, Eidgenössische Technische Hochschule, Seminar für Angewandte Mathematik, 1996.
- [19] I. Babuška, J. Melenk, The partition of unity method, *Int. J. Numer. Methods Eng.* 40 (1997) 727–758.
- [20] B. Boroomand, M. Najjar, E. Oñate, The generalized finite point method, *Comput. Mech.* 44 (2009) 173–190.
- [21] E. Oñate, S. Idelsohn, O. Zienkiewicz, R. Taylor, A finite point method in computational mechanics. applications to convective transport and fluid flow, *Int. J. Numer. Methods Eng.* 39 (1996) 3839–3866.
- [22] E. Oñate, C. Sacco, S. Idelsohn, A finite point method for incompressible flow problems, *Comput. Vis. Sci.* 3 (2000) 67–75.
- [23] R. Hardy, Multiquadric equations of topography and other irregular surfaces, *J. Geophys. Res.* 76 (1971) 1905–1915.
- [24] E. Kansa, Multiquadratics—a scattered data approximation scheme with applications to computational fluid-dynamics—I surface approximations and partial derivative estimates, *Comput. Math. Appl.* 19 (1990) 127–145.
- [25] E. Kansa, Multiquadratics—a scattered data approximation scheme with applications to computational fluid-dynamics—II solutions to parabolic, hyperbolic and elliptic partial differential equations, *Comput. Math. Appl.* 19 (1990) 147–161.
- [26] E. Kansa, Y. Hon, Circumventing the ill-conditioning problem with multiquadric radial basis functions: applications to elliptic partial differential equations, *Comput. Math. Appl.* 39 (2000) 123–137.
- [27] C. Shu, H. Ding, K. Yeo, Local radial basis function-based differential quadrature method and its application to solve two-dimensional incompressible Navier–Stokes equations, *Comput. Methods Appl. Mech. Eng.* 192 (2003) 941–954.
- [28] H. Ding, C. Shu, K. Yeo, D. Xu, Numerical computation of three-dimensional incompressible viscous flows in the primitive variable form by local multiquadric differential quadrature method, *Comput. Methods Appl. Mech. Eng.* 195 (2006) 516–533.
- [29] E. Larsson, B. Fornberg, A numerical study of some radial basis function based solution methods for elliptic PDEs, *Comput. Math. Appl.* 46 (2003) 891–902.
- [30] G. Wright, B. Fornberg, Scattered node compact finite difference-type formulas generated from radial basis functions, *J. Comput. Phys.* 212 (2006) 99–123.
- [31] Y. Sanyasiraju, G. Chandhini, Local radial basis function based gridfree scheme for unsteady incompressible viscous flows, *J. Comput. Phys.* 227 (2008) 8922–8948.
- [32] Y. Sanyasiraju, G. Chandhini, A note on two upwind strategies for RBF-based grid-free schemes to solve steady convection–diffusion equations, *Int. J. Numer. Methods Fluids* 61 (2009) 1053–1062.
- [33] G. Chandhini, Y. Sanyasiraju, Local RBF-FD solutions for steady convection–diffusion problems, *Int. J. Numer. Methods Eng.* 72 (2007) 352–378.
- [34] A. Vidal, A. Kassab, E. Divo, A direct velocity–pressure coupling meshless algorithm for incompressible fluid flow simulations, *Eng. Anal. Bound. Elem.* 72 (2016) 1–10.
- [35] R. Zamolo, E. Nobile, Solution of incompressible fluid flow problems with heat transfer by means of an efficient RBF-FD meshless approach, *Numer. Heat Transf., Part B, Fundam.* (2019) 1–24.
- [36] K. Kosec, B. Šarler, Solution of thermo-fluid problems by collocation with local pressure correction, *Int. J. Numer. Methods Heat Fluid Flow* 18 (2008) 868–882.
- [37] G. Kosec, Local meshless method for multi-phase termo-fluid problems, Ph.D. thesis, Univerza v Novi Gorici, Fakulteta za podiplomski študij, 2011.
- [38] L. Wang, J.-S. Chen, H.-Y. Hu, Subdomain radial basis collocation method for fracture mechanics, *Int. J. Numer. Methods Eng.* 83 (2010) 851–876.
- [39] B. Fornberg, G. Wright, Stable computation of multiquadric interpolants for all values of the shape parameter, *Comput. Math. Appl.* 48 (2004) 853–867.
- [40] B. Fornberg, G. Wright, E. Larsson, Some observations regarding interpolants in the limit of flat radial basis functions, *Comput. Math. Appl.* 47 (2004) 37–55.
- [41] E. Larsson, B. Fornberg, Theoretical and computational aspects of multivariate interpolation with increasingly flat radial basis functions, *Comput. Math. Appl.* 49 (2005) 103–130.
- [42] B. Fornberg, E. Larsson, N. Flyer, Stable computations with Gaussian radial basis functions, *SIAM J. Sci. Comput.* 33 (2011) 869–892.
- [43] G. Fasshauer, M. McCourt, Stable evaluation of Gaussian radial basis function interpolants, *SIAM J. Sci. Comput.* 34 (2012) A737–A762.
- [44] B. Fornberg, E. Lehto, C. Powell, Stable calculation of Gaussian-based RBF-FD stencils, *Comput. Math. Appl.* 65 (2013) 627–637.

- [45] G.A. Barnett, A robust RBF-FD formulation based on polyharmonic splines and polynomials, Ph.D. thesis, University of Colorado, Boulder, 2015.
- [46] V. Bayona, N. Flyer, B. Fornberg, G. Barnett, On the role of polynomials in RBF-FD approximations: II. Numerical solution of elliptic PDEs, *J. Comput. Phys.* 332 (2017) 257–273.
- [47] V. Bayona, N. Flyer, B. Fornberg, On the role of polynomials in rbf-fd approximations: III. Behavior near domain boundaries, *J. Comput. Phys.* 380 (2019) 378–399.
- [48] N. Flyer, G. Barnett, L. Wicker, Enhancing finite differences with radial basis functions: experiments on the Navier–Stokes equations, *J. Comput. Phys.* 316 (2016) 39–62.
- [49] N. Flyer, B. Fornberg, V. Bayona, G. Barnett, On the role of polynomials in RBF-FD approximations: I. Interpolation and accuracy, *J. Comput. Phys.* 321 (2016) 21–38.
- [50] L. Santos, N. Manzanares-Filho, G. Menon, E. Abreu, Comparing rbf-fd approximations based on stabilized gaussians and on polyharmonic splines with polynomials, *Int. J. Numer. Methods Eng.* 115 (2018) 462–500.
- [51] V. Bayona, Comparison of moving least squares and rbf+ poly for interpolation and derivative approximation, *J. Sci. Comput.* 81 (2019) 486–512.
- [52] V. Shankar, The overlapped radial basis function-finite difference (rbf-fd) method: a generalization of rbf-fd, *J. Comput. Phys.* 342 (2017) 211–228.
- [53] V. Shankar, A. Fogelson, Hyperviscosity-based stabilization for radial basis function-finite difference (rbf-fd) discretizations of advection-diffusion equations, *J. Comput. Phys.* 372 (2018) 616–639.
- [54] M. Jančić, J. Slak, G. Kosec, Analysis of high order dimension independent rbf-fd solution of Poisson's equation, arXiv preprint, arXiv:1909.01126, 2019.
- [55] D. Gundersen, N. Flyer, B. Fornberg, Transport schemes in spherical geometries using spline-based rbf-fd with polynomials, *J. Comput. Phys.* (2020) 109256.
- [56] S. Shahane, S.P. Vanka, MeMPhyS: meshless multi-physics software, <https://github.com/shahaneshantanu/memphys>, 2021.
- [57] F. Harlow, J. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, *Phys. Fluids* 8 (1965) 2182–2189.
- [58] Poisson equation with pure Neumann boundary conditions, <https://fenicsproject.org/docs/dolfin/1.4.0/python/demo/documentation/neumann-poisson/python/documentation.html>, 2019.
- [59] Poisson's equation – medusa: coordinate free meshless method implementation, http://e6.ijs.si/medusa/wiki/index.php?title=Poisson%27s_equation&oldid=2804, 2019.
- [60] A. George, J. Liu, E. Ng, *Computer Solution of Sparse Linear Systems*, Oak Ridge National Laboratory, 1994.
- [61] E. Cuthill, J. McKee, Reducing the bandwidth of sparse symmetric matrices, in: Proceedings of the 1969 24th National Conference, 1969, pp. 157–172.
- [62] L. Kovasznay, Laminar flow behind a two-dimensional grid, in: Mathematical Proceedings of the Cambridge Philosophical Society, vol. 44, Cambridge University Press, 1948, pp. 58–62.
- [63] C. Geuzaine, J. Remacle, Gmsh: a 3-d finite element mesh generator with built-in pre-and post-processing facilities, *Int. J. Numer. Methods Eng.* 79 (2009) 1309–1331.
- [64] NEK5000: examples, <http://doc.nektar.info/userguide/4.3.4/user-guides45.html>, 2020.
- [65] F. White, *Fluid Mechanics*, McGraw-Hill, 2011.
- [66] A. Tiwari, S. Vanka, A ghost fluid lattice Boltzmann method for complex geometries, *Int. J. Numer. Methods Fluids* 69 (2012) 481–498.
- [67] J.B. Bell, P. Colella, H.M. Glaz, A second-order projection method for the incompressible Navier-Stokes equations, *J. Comput. Phys.* 85 (1989) 257–283.
- [68] U. Ghia, K.N. Ghia, C. Shin, High-re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method, *J. Comput. Phys.* 48 (1982) 387–411.
- [69] E. Süli, D.F. Mayers, *An Introduction to Numerical Analysis*, Cambridge University Press, 2003.
- [70] H. Ding, C. Shu, K. Yeo, D. Xu, Simulation of incompressible viscous flows past a circular cylinder by hybrid fd scheme and meshless least square-based finite difference method, *Comput. Methods Appl. Mech. Eng.* 193 (2004) 727–744.
- [71] M. Braza, P. Chassaing, H.H. Minh, Numerical study and physical analysis of the pressure and velocity fields in the near wake of a circular cylinder, *J. Fluid Mech.* 165 (1986) 79–130.
- [72] C. Liu, X. Zheng, C. Sung, Preconditioned multigrid methods for unsteady incompressible flows, *J. Comput. Phys.* 139 (1998) 35–57.
- [73] A. Belov, L. Martinelli, A. Jameson, A new implicit algorithm with multigrid for unsteady incompressible flow calculations, in: 33rd Aerospace Sciences Meeting and Exhibit, 1995, p. 49.
- [74] N. Flyer, E. Lehto, S. Blaise, G.B. Wright, A. St-Cyr, A guide to rbf-generated finite differences for nonlinear transport: shallow water simulations on a sphere, *J. Comput. Phys.* 231 (2012) 4078–4095.
- [75] B. Fornberg, E. Lehto, Stabilization of rbf-generated finite difference methods for convective pdes, *J. Comput. Phys.* 230 (2011) 2270–2285.