Enhanced solution of 2D incompressible Navier–Stokes equations based on an immersed-boundary generalized harmonic polynomial cell method

Yu, Xueying; Fuhrman, David R.; Shao, Yanlin; Liao, Kangping; Duan, Wenyang; Zhang, Yunxing

Published in:
European Journal of Mechanics B - Fluids

Link to article, DOI:
10.1016/j.euromechflu.2021.04.009

Publication date:
2021

Document Version
Peer reviewed version

Citation (APA):
Enhanced solution of 2D incompressible Navier–Stokes equations based on an immersed-boundary generalized harmonic polynomial cell method

Xueying Yu, David R. Fuhrman, Yanlin Shao, Kangping Liao, Wenyang Duan, Yunxing Zhang

PII: S0997-7546(21)00056-X
DOI: https://doi.org/10.1016/j.euromechflu.2021.04.009
Reference: EJMFLU 103771

To appear in: *European Journal of Mechanics / B Fluids*

Received date: 20 August 2020
Revised date: 5 February 2021
Accepted date: 26 April 2021


This is a PDF file of an article that has undergone enhancements after acceptance, such as the addition of a cover page and metadata, and formatting for readability, but it is not yet the definitive version of record. This version will undergo additional copyediting, typesetting and review before it is published in its final form, but we are providing this version to give early visibility of the article. Please note that, during the production process, errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

© 2021 Elsevier Masson SAS. All rights reserved.
Enhanced solution of 2D incompressible Navier-Stokes equations based on an immersed-boundary generalized harmonic polynomial cell method

Xueying Yu1, David R. Fuhrman2, Yanlin Shao2,1,*, Kangping Liao1, Wenyang Duan1, Yunxing Zhang1

1 College of Shipbuilding Engineering, Harbin Engineering University, Harbin 150001, China
2 Department of Mechanical Engineering, Technical University of Denmark, DK-2800 Kgs. Lyngby, Denmark

Keywords: Pressure Poisson equation; Projection method; Immersed boundary method; Generalized Harmonic Polynomial Cell method; Incompressible Navier-Stokes equations

Abstract

Poisson equation lies in the heart of the numerical solutions of incompressible Navier-Stokes equations (NSEs) based on the popular projection methods, which decouple the pressure and velocities fields. This paper presents enhanced numerical solutions of the 2D incompressible NSEs inspired by a newly-developed Generalized Harmonic Polynomial Cell (GHPC) method for the Poisson equation, which has a fourth-order spatial accuracy. To achieve that, the original GHPC method is adapted to accommodate immersed boundaries, necessary when a staggered grid for velocities and pressure is applied. The finite difference method (FDM) is used in the spatial discretization of the diffusion and advection terms. Numerical analyses of lid-driven cavity flow, a Taylor-Green vortex, and flow around a smooth circular cylinder all show encouraging results, confirming the accuracy and efficiency of the new solver. Through comparison with an NSEs solver which applies second-order FDM for the Poisson equation, we demonstrate that the accuracy of the pressure solution can be greatly improved by applying the more accurate GHPC method for the Poisson equation, even though the accuracy of the velocities solutions are limited by the numerical schemes for advection-diffusion equations. The accuracy in the pressure solution can also be translated into CPU time saving to achieve a predefined accuracy. The present immersed-boundary GHPC Poisson equation solver can easily be ‘plugged’ into other existing NSEs solvers utilizing staggered grids.

1. Introduction

Viscous effects are found to be very important in many engineering applications, for example, in calculating the excitation and damping of marine structures, and other fluid-structure interaction problems at low Reynolds numbers. Compared with potential-flow models, NSEs with proper turbulence modeling, can give a better description of fluid-structure interaction, in particular when viscous effects are important. Even though the capacity of modern computers has been greatly increased, it is still considered a challenge to solve NSEs numerically for many applications, in large part due to the high computational costs. Thus, it is of great importance to improve the efficiency and accuracy of NSEs solvers [1,2].

The projection method is an effective numerical means to study fluid-structure interaction problems in incompressible fluid flows. The main advantage of the method is that the pressure and velocities fields are decoupled, and thus efficient methods can be applied for the resulting equations in different steps of the numerical procedure. It has been adopted in many existing NSEs solvers such as OpenFOAM [3,4] and STAR-CCM+ [5]. The first projection method was introduced by Chorin [6] and Témam [7] as a means to solve NSEs in two subsequent steps. The first step computes intermediate solutions for the velocities, ignoring the pressure gradient term, while the second corrects the intermediate velocity with pressure terms in the final solution at the new time step. The overall scheme has first-order temporal accuracy. Discussion of second-order schemes (in time) can be found in e.g. Timmermans et al. [8] and Guermond et al. [9]. Xiao [10] proposed

*Corresponding author: yshao@mek.dtu.dk (Yanlin Shao)
a three-step projection method to achieve a more general formulation that covers both compressible and incompressible flows. One obvious advantage of the three-step projection method is that it permits a large variety of numerical schemes when solving the sub-equations in different steps. A successful use can be found in [11] where a robust and accurate numerical approach for extremely nonlinear free-surface problems is established by Hu and Kashiwagi [11] with a CIP-based (Constrained Interpolation Profile) finite difference method.

For the spatial discretization of the NSEs, many traditional numerical schemes, such as the finite difference method (FDM), the finite element method (FEM), and the finite volume method (FVM), have been applied with successes in simulating various problems [12-14]. When the gradients of the flow solution are high, or small-scale phenomenon must be captured, very fine numerical meshes are necessary to ensure the accuracy of the solution. Although high-order schemes have also been applied in NS equation in recent years, such as the higher-order WENO (weighted essentially non-oscillatory) method [15-17] and CIP methods [11,18] for the discretization of convection terms, the most commonly used implementation yields second-order spatial accuracy, which might be due to the numerical instability and complexity of building high-order approximation schemes. Another major challenge of the projection method is how to obtain an accurate solution for the fluid pressure through the so-called pressure Poisson equation, which is essential for the overall accuracy and efficiency of the NSEs based on a projection method.

Recently, a highly efficient method with fourth-order accuracy, called the generalized harmonic polynomial cell (GHPC) method, was developed by Bardazzi et al. [19] to solve the 2D Poisson equation. In the GHPC method, the boundary value problem (BVP) is solved by decomposing the total solution into so-called homogeneous and particular solutions. The homogeneous solution satisfies the Laplace equation, which is discretized following the original 2D harmonic polynomial cell (HPC) methods [20-23]. The particular solution is constructed to the same order as the homogeneous solution by a bi-quadratic interpolation. In [19], the GHPC method has been validated by a series of test cases with analytical solutions, such as the well-known Taylor-Green vortex, the Lamb-Oseen vortex, and a circular wavefront, confirming very good performance with a fourth-order accuracy. The 2D GHPC method is a generalization of the 2D HPC, which was originally proposed by Shao and Faltinsen [20] as a general numerical method for the Laplace equation. In the 2D HPC method, the computational domain is discretized by overlapping cells containing four neighboring quadrilateral elements and nine grid points. Linear superposition of the harmonic polynomials which analytically satisfy the Laplace equation is then used to construct the local approximation of the velocity potential within each cell. An extension of the 2D HPC method to 3D was presented by Shao and Faltinsen [21], who have also applied the method in various 3D potential-flow problems, including liquid sloshing, water wave propagation, and wave-structure interactions.

The 3D HPC method is also applied in the liquid sloshing problem considering periodic and transient excitations [22]. The HPC methods have approximately fourth-order accuracy, while their computational costs are comparable to traditional second-order methods [20-23]. Later, Fredriksen et al. [24] present a domain decomposition strategy to investigate the piston-mode fluid resonance in moonpools, where the HPC method in the outer domain is coupled with an FVM-based NS equation solver in the inner domain.

Encouraged by the good performance of HPC method, an immersed boundary (IB) method for the 2D HPC method was proposed by Hansen et al. [25] to simulate a potential-flow problem governed by the Laplace equation, and it was validated by moving structures immersed in a Cartesian background grid. Their results show that the method is effective in accurately calculating forces on moving bodies, while at the same time overcoming problems associated with spurious oscillations. More recent developments [26, 27] in the 2D HPC method indicate that the method, in fact, has higher than fourth-order accuracy when the cells are close to squares. Increased errors were also seen with increasing distortion of the cells [26], where different grid strategies were proposed to accurately describe the boundary conditions on rigid or deformable surfaces in a potential-flow solver based on the concepts of the IB method and overlapping grids. This has motivated Hansen et al. [27], Zhu et al. [28], and Tong et al. [29] to introduce the IB strategies in the 2D HPC method in 2D fully-nonlinear potential flow analyses. The introduction of IBs makes it possible for the HPC method to deal with more complex structures.

While using IB strategy in the HPC method for Laplace equation has been seen as a success, it has not been attempted
for the GHPC method for the Poisson equation. Though it has yet to be proven, we expected that a method combining IB strategy with the GHPC, hereafter called the IB-GHPC method in this paper, can be used as a more powerful tool for general-purpose applications involving numerical solutions of the Poisson equation. One example is the numerical solution of incompressible NSEs based on the projection method.

The present paper will develop a numerical solver for incompressible NSEs based on the projection method and use the GHPC method to solve the resulting Poisson equation, with the hope of providing enhanced numerical solutions for fluid-structure interaction problems. As an example, the three-step projection method proposed by Xiao [10] will be implemented to efficiently solve the NSEs, where the involved spatial derivatives in the advection and diffusion terms will be discretized by a second-order central-difference scheme. To ensure that the pressure is properly represented in the numerical solution, a staggered grid is utilized, following Harlow and Welch [30]. One advantage of the staggered grid arrangement is a better coupling between pressure and velocities, with the added benefit of eliminating unphysical oscillations in the solutions. To model general fluid-structure interactions, a simple and efficient direct-forcing IB framework of Yang et al. [31-33] which is an extended use of an embedded-boundary formulation [34], is adopted in the present study, and the method will be further introduced in Section 4. The use of a staggered grid, and possibly other IB methods, requires the Poisson equation solver to be able to work with IB methods. Therefore, we will also present a new IB strategy for the original GHPC method [19].

As verification and validation of the new NSEs solver, a lid-driven cavity flow, a Taylor-Green vortex, and flow around a smooth circular cylinder will all be studied. To demonstrate that an enhanced solution of the pressure Poisson equation will lead to an improved overall pressure solution of the equations, the accuracy of the results will also be compared with another solver using a traditional second-order central-difference scheme for the Poisson equation. It should be noted that the stability of the new solver, which is dominated by diffusion and advection steps, will not be directly influenced by the use of the IB-GHPC method. Thus, the IB-GHPC method has the potential to be applied in other numerical solvers to achieve a large gain in accuracy, with only slight modifications required.

The remainder of this paper is organized as follows: In Section 2, the three-step projection method for the NSEs and spatial discretization for the velocities fields are introduced. In Section 3, the IB-GHPC method for the Poisson equation is introduced. In Section 4, the direct forcing IB method for general fluid-structure interaction problems solved by NSEs is briefly explained. In Section 5, the implemented IB-GHPC solver for the Poisson equation is verified for cases having analytical solutions. In Section 6, the numerical analyses of the lid-driven cavity flow, Taylor-Green vortex, and flow around a smooth circular cylinder are studied to investigate the accuracy and convergence of the new NSEs solver. Finally, conclusions are drawn in Section 7.

### 2. Governing equations and the three-step projection method

The NSEs for the unsteady, viscous, and incompressible flow can be written as follows:

\[
\begin{align*}
\frac{\partial u_i}{\partial t} + \frac{\partial (u_i u_j)}{\partial x_j} &= -\frac{1}{\rho} \frac{\partial p}{\partial x_i} + \frac{1}{\rho} \frac{\partial \tau_{ij}}{\partial x_j} + f_i, \quad (1) \\
\frac{\partial p}{\partial x_i} &= 0, \quad (2)
\end{align*}
\]

where $p$ and $\rho$ are the pressure and fluid density, respectively, $u_i (i = 1, 2)$ is the velocity component in the $i$-th direction (with subscripts $i=1$ and $i=2$ corresponding to the $x$ and $y$ directions, respectively), $f_i$ is the source term, $\tau_{ij}$ is the shear stress tensor having the form of $\tau_{ij} = \mu(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i})/2$, with $\mu$ being the dynamic viscosity. The Einstein summation convention is used in Eq. (1) and Eq. (2). In the above, $x_i (i = 1, 2)$ is the Cartesian coordinates and $t$ is time.

In what follows, the three-step fractional method [10], which belongs to the family of the projection methods, is employed as a basis to solve the NSEs. The three fractional steps in the method are: an advection step solving only the advection equations, a non-advection step (i) dealing with the diffusion and forcing terms, and another non-advection step (ii) to correct the fluid velocities thus to ensure a divergence-free flow. They can be expressed as follows

\[
\begin{align*}
\frac{\partial u_i}{\partial t} + \frac{\partial (u_i u_j)}{\partial x_j} &= -\frac{1}{\rho} \frac{\partial p}{\partial x_i} + \frac{1}{\rho} \frac{\partial \tau_{ij}}{\partial x_j} + f_i, \quad (1) \\
\frac{\partial p}{\partial x_i} &= 0, \quad (2)
\end{align*}
\]

...
Advection step:

\[
\frac{u_i^n - u_i^m}{\Delta t} + \frac{\partial (u_i^n u_j^n)}{\partial x_j} = 0.
\]

(3)

Non-advection step (i):

\[
\frac{u_i^n - u_i^{n-1}}{\Delta t} = \frac{1}{\rho} \frac{\partial \tau_{ij}}{\partial x_j} + f_i.
\]

(4)

Non-advection step (ii):

\[
\frac{u_i^{n+1} - u_i^n}{\Delta t} = -\frac{1}{\rho} \frac{\partial p^n}{\partial x_i},
\]

where,

\[
\frac{1}{\rho} \nabla^2 p^{n+1} = \frac{1}{\Delta t} \frac{\partial u_i^{n+1}}{\partial x_i}.
\]

Here the superscripts \( n \) and \( n + 1 \) indicate that the variable is evaluated at the \( n \)-th and \( (n+1) \)-th time step, respectively. The superscripts * and ** stand for the intermediate values as part of the prediction. The pressure Poisson equation, Eq. (5b), lies at the heart of the projection methods, accounting for most of the computational costs of the NSEs solver.

We have decided to use an implicit scheme for the diffusion terms in Eq. (4), while the original scheme in Xiao [10] is explicit for the diffusion terms. The reason for this choice is that, in what follows, we will study flow around a circular cylinder at very low Reynolds numbers, which would otherwise require impractically small time steps due to the stability requirements of the diffusion equation. There are other implicit schemes, e.g. the Alternating Direction Implicit (ADI) method [35] that can be used to solve the diffusion equation more efficiently, though these will not be pursued here. The pressure Poisson equation, Eq. (5b), will be solved immediately after Eq. (4) and Eq. (3) utilizing the GHPC method, after which the final flow velocities can be obtained through Eq. (5a). It has been shown in Xiao [10] that this three-step projection method leads to the first-order accuracy in the time integration of the governing equations.

Fig. 1 Example of a 2D staggered grid for a rectangular fluid domain. The blue thick line is the fluid boundary. ✑: grid point for the horizontal velocity of \( u_1 \), ✐: grid point for the vertical velocity of \( u_2 \), ✉: pressure point in the fluid, ✺: ghost point for pressure.

In the advection step, Eq. (3), the advection terms are discretized by the second-order explicit QUICK (Quadratic Upstream Interpolation for Convective Kinematics) scheme [36], meaning that two layers of ghost points outside of the fluid domain are required to facilitate the calculation of the spatial derivatives. See Fig. 1 for an illustration of the staggered grid for a rectangular fluid domain, which includes grid points for the horizontal velocity \( u_1 \) (red arrows), vertical velocity \( u_2 \) (pink arrows) and pressure \( p \) (blue crosses), respectively. The thick blue lines in the figure represents the fluid boundary. The QUICK scheme is essentially a bias-difference scheme, which utilizes more stencil points from the upstream than the downstream, providing conditional stability on a uniform grid as long as the CFL number is sufficiently small [1]. As shown in Fig. 1, pressure points are calculated in the center, and the velocities of \( u_1 \) and \( u_2 \) are placed in adjacent cell faces.
Consequently, not all grid points lie on the boundaries. For example, none of the pressure points are located on the fluid boundary in Fig. 1, while the vertical boundaries have no \( u_1 \) points, and the same for \( u_1 \) on the horizontal boundaries. To ensure the proper boundary conditions and application of the approximation schemes, extra ghost cells for velocities and pressure are introduced outside the boundary, as indicated in Fig. 1.

In this study, the other two non-advection steps, described by Eqs. (4) and (5), are discretized by the second-order central difference method as an example. In principle, other higher-order methods can also be applied for the same purpose. This is not pursued here, as our purpose is to demonstrate that the use of IB-GHPC alone is able to enhance the pressure solution. The Poisson equation, Eq. (5b), will be solved by the IB GHPC method that will be introduced in the next section.

3. IB-GHPC method for Poisson equation

Following the three-step projection method described in Section 2, \( u^{**} \) is obtained through the advection step and non-advection step (i), after which \( \frac{\partial u^{**}}{\partial x_i} \) can be approximated by a central difference scheme. Therefore, the forcing term

\[
\sigma(x) = \frac{1}{\Delta t} \frac{\partial u^{**}}{\partial x_i}
\]

on the right-hand side of the Poisson equation (5b) is known at all pressure grid points. Since the staggered grid leads to one extra layer of pressure points outside the fluid domain, the IB strategy in GHPC for the Poisson equation will be introduced in this section.

The GHPC is a generalization of the HPC method for Laplace equations. The homogeneous part of the solution of the Poisson equation is taken care of by the HPC method (more details about the HPC method are presented in Appendix A), while the particular solution is constructed locally through high-order Lagrange polynomials. For completeness, we will only give a brief introduction of the mathematical formulation of this method in Section 3.1 and discuss how the GHPC could be applied together with an IB strategy. Interested readers are referred to Bardazzi et al. [19] for further details of the GHPC method on a boundary-fitted grid and Shao and Faltinsen [20, 21] for the HPC method.

![Diagram](image)

a) The global index in the domain  
b) Local index in a cell

Fig. 2 local index of the grid nodes in a cell.

3.1 Mathematical formulation of GHPC

In a fluid domain \( \Omega \), for an arbitrary function \( \psi \), the boundary value problem (BVP) governed by the Poisson equation, Dirichlet or Neumann conditions can be respectively described as [19]:

\[
\Delta \psi = \sigma(x) \quad \text{in} \quad \Omega
\]

\[
\psi = \psi_D \quad \text{on} \quad \Gamma_D
\]

\[
\frac{\partial \psi}{\partial n} = \psi_N \quad \text{on} \quad \Gamma_N
\]
\[
\begin{align*}
\nabla^2 \psi(x_1,x_2) &= \sigma(x_1,x_2) \\
\psi(x_1,x_2) &= g_p(x_1,x_2) \\
\frac{\partial \psi(x_1,x_2)}{\partial n} &= g_N(x_1,x_2)
\end{align*}
\]

Here, \( \sigma(x_1,x_2) \) is a known function. The solution \( \psi(x_1,x_2) \) is divided into two parts as below

\[
\psi(x_1,x_2) = \bar{\psi}(x_1,x_2) + \tilde{\psi}(x_1,x_2),
\]

where \( \bar{\psi}(x_1,x_2) \) is a homogenous solution satisfying the Laplace equation, and \( \tilde{\psi}(x_1,x_2) \) is a particular solution satisfying

\[
\Delta \tilde{\psi}(x_1,x_2) = \sigma(x_1,x_2)
\]

In the GHPC method, the domain is divided into quadrilateral elements with a finite number of computational points. Overlapping cells, each of which includes nine grid points and four neighbor elements, are used for the local approximation of the solution. As shown in Fig. 2, each point is assigned a global index and a local index. Then, the variables \( \bar{\psi}, \tilde{\psi}, \sigma \) within a cell can be approximated by a linear superposition of the corresponding values on the cell nodes as:

\[
\bar{\psi}(x_1,x_2) = \sum_{\lambda=1}^{8} a_\lambda q_\lambda(x_1,x_2) \tag{9a}
\]

\[
\sigma(x_1,x_2) = \sum_{\lambda=1}^{8} c_\xi h_\xi(x_1,x_2) \tag{9b}
\]

\[
\tilde{\psi}(x_1,x_2) = \sum_{\lambda=1}^{8} c_\xi \tilde{h}_\xi(x_1,x_2) \tag{9c}
\]

Eq. (9a) constructs the homogenous solution \( \bar{\psi}(x_1,x_2) \) within each cell by a weighted sum of harmonic polynomials \( q_\lambda (\lambda = 1, \ldots, 8) \), which are given in Table B.1 in Appendix B. More explanations of Eq. (9) can be found in Appendix B. Eq. (9b) approximates the forcing term \( \sigma(x_1,x_2) \) by a bi-quadratic interpolation, with the coefficients \( c_\xi (\xi = 1, \ldots, 9) \) determined by

\[
c_\xi = [h^{-1}]_{\xi k} \sigma_k, \quad \text{with } \xi, k = 1, \ldots, 9. \tag{10}
\]

Here \( [h^{-1}]_{\xi k} \) is the inverse of the matrix \( h_{\xi k} \), whose entries are the values of the polynomial \( h_\xi \) at the \( k \)-th local cell point. To this end, the local solution within each cell is expressed as

\[
\psi(x_1,x_2) = \sum_{\lambda=1}^{8} a_\lambda q_\lambda(x_1,x_2) + \sum_{\lambda=1}^{8} c_\xi \tilde{h}_\xi(x_1,x_2) \tag{11}
\]

Here \( a_\lambda (\lambda = 1, \ldots, 8) \) are the unknown coefficients, which can be determined through boundary values on the eight nodes of the cell.

\[
\psi_m = q_{m\lambda} a_\lambda + g_{m\xi} c_\xi \Rightarrow a_\lambda = [q^{-1}]_{\lambda m} (\psi_m - g_{m\xi} c_\xi)
\]

\( \lambda, m = 1, \ldots, 8; \quad \xi, k = 1, \ldots, 9. \)

Here \( \psi_m \) is the value of \( \psi(x_1,x_2) \) at the \( m \)-th node, \( q_{m\lambda} \) and \( g_{m\xi} \) are the matrices whose elements are \( q_\lambda \) and \( g_\xi \) evaluated at the \( m \)-th point of the cell, respectively, and \( [q^{-1}]_{\lambda m} \) is the inverse matrix of \( q_{m\lambda} \). Therefore, the solution in Eq. (11) within each cell is expressed as

\[
\psi(x_1,x_2) = \sum_{\lambda=1}^{8} a_\lambda q_\lambda(x_1,x_2) + \sum_{\lambda=1}^{8} c_\xi \tilde{h}_\xi(x_1,x_2)
\]

\[
= q_{m\lambda}(x_1,x_2)[q^{-1}]_{\lambda m}(\psi_m - g_{m\xi} [h^{-1}]_{\xi k} \sigma_k) + g_{m\xi}(x_1,x_2) [h^{-1}]_{\xi k} \sigma_k \tag{12}
\]

\( \lambda, m = 1, \ldots, 8; \quad \xi, k = 1, \ldots, 9 \)

It can be seen from Eq. (12) that the solution \( \psi(x_1,x_2) \) at any point within the cell can be obtained by the linear combination of \( \bar{\psi} \) and \( \sigma \) values on the cell nodes. Application of Eq. (12) on all fluid points ensures the continuity of the...
solution across the fluid domain which, together with the boundary conditions, leads to a global sparse-matrix equation. Even though it has not been done in the literature, the 2D GHPC method can be straightforwardly extended to three dimensions by following a similar procedure described in this section. The 3D HPC method by Shao and Faltinsen [21] for the Laplace equation can be readily used to take care of the homogeneous solution of a 3D Poisson equation, while a particular solution can be obtained by using tri-quadratic interpolation based on 27 polynomials. In the 2D GHPC method, one works with $3 \times 3$ stencils, while a typical stencil in three dimensions will be $3 \times 3 \times 3$. Since the 3D HPC method has been shown in [21] to be 3rd-4th order accurate, similar accuracy is expected for the resulting GHPC method for the Poisson equation. As future work, it is of great interest to extend the GHPC method to 3D and apply it in the numerical solution of 3D NSEs.

3.2 IB strategy in GHPC

In our IB strategy for the GHPC method, ghost points are introduced outside of the fluid domain to allow for a smooth and artificial extension of the fluid solution across the fluid boundary. Fig. 3a shows an example of a box-shape fluid domain, whose boundary is denoted by the thick blue lines. To construct a local approximation of the fluid solution at each grid point by the GHPC method, a local harmonic polynomial cell must be formed. All fluid points (blue crosses) in the domain can be considered as the center of a cell, which is defined by the eight points surrounding it. For each ghost point (red filled squares) which cannot be the center of a cell, an associated cell must be assigned. In Fig. 3a, an arrow is drawn from each ghost point to its associated cell center. The boundary conditions will be enforced at individual boundary nodes denoted by the green filled circles. An example of an associated cell close to a corner of the rectangular domain is shown in Fig. 3b.

It is straightforward to extend the above IB strategy to consider complex arbitrary boundaries in the fluid. Similar strategies have been successfully applied by Hanssen et al. [25] and Ma et al. [25, 26] in the HPC method for Laplace equation as part of their solution for potential-flow problems. For the GHPC method for the Poisson equation, IB strategies can be similarly applied. Fig. 4 depicts a part of a complex arbitrary boundary immersed in the Cartesian background grid. As seen in Fig. 3, ghost points, fluid points, and boundary points are marked by red filled squares, blue crosses, and green filled circles, respectively. Each ghost point is associated with a cell, whose center is inside the fluid domain. In contrast to the case of a rectangular fluid boundary in Fig. 3, where the intersection points between the grid lines and the fluid boundary are selected as the boundary points, the boundary points can be determined by searching for the point on the boundary.
nearest to each ghost point. When the position of the boundary points and the associated harmonic polynomial cells are defined for each ghost points, the boundary conditions can be enforced exactly on the boundary points. Generally, the type of boundary conditions can be categorized as Dirichlet, Neumann, and Robin boundary conditions. Dirichlet conditions are satisfied by applying Eq. (12) at the boundary nodes. Concerning Neumann boundary conditions, the normal derivative can be obtained by taking the normal derivatives on both sides of Eq. (12) as:

$$\frac{\partial \psi(x_1, x_2)}{\partial n} = \frac{\partial q(x_1, x_2)}{\partial n} \left[ q^{-1} \mu_m \psi_m - g_{m\xi} [h^{-1}] \xi_k \sigma_k \right] + \frac{\partial g(x_1, x_2)}{\partial n} \left[ h^{-1} \xi_k \sigma_k \right]$$

(13)

where \(\vec{n}\) is the normal vector on the boundary nodes. In the global linear system, the sparse matrix is composed of (at most) nine non-zeros in each row. In this study, a sparse matrix solver called parallel direct sparse solver (PARDISO) [37] is applied to solve the final sparse-matrix equation system.

Fig. 4 IB strategy for complex arbitrary boundaries in GHPC. The blue thick line is the boundary: \(\times\) fluid point in the domain, \(\bullet\) intersection point at the boundary, \(\blacksquare\) ghost point outside the boundary.

4. The direct forcing IB method for general fluid-structure interaction problems

The IB method is a popular strategy to deal with general fluid-structure interaction problems involving complex geometries. To demonstrate the accuracy of the new solver, we will also study the flow around a smooth circular cylinder in a later section as a validation case involving fluid-structure interaction. For simplicity, an easy-to-implement IB method proposed by Yang et al. [31] is adopted as an example, which is sufficient in the present study, since the focus here is on the enhanced numerical solution by using IB-GHPC as the Poisson equation solver. This IB method has been extended later by Liu et al. [38] to suppress spurious pressure oscillations on moving structures. Note that the presented numerical solution procedure for the Poisson equation with the IB strategy in Section 3.2 can also be used in any other IB methods for the NSEs. In the following, we will only briefly describe the IB method. More details can be found in Yang et al. [31-33].
As shown in Fig. 5a, the first step is to distinguish the three types of points, namely the solid points (dark filled squares), fluid points (blue filled squares), and forcing points (red filled triangles), which are located inside the fluid, outside the fluid, and in the first layer near the boundary, respectively. Another type of point on the boundary, called intersection or boundary points (green filled cycles), is used to get the boundary conditions. Then, velocities are assigned to all the points including the fluid and solid points. If a point is in the fluid, its velocity is obtained from the solution of the NSEs. The (horizontal and vertical) velocities on a solid point and the intersection points are determined by the rigid-body motions of the structure. The velocities on a forcing point are reconstructed by the velocities on the nearby solid and fluid points, to ensure that non-slip boundary conditions are satisfied on the boundary points. Fig. 5b shows a linear triangle-based reconstruction scheme for forcing points, which involves an intersection point and two fluid points. Any variables in the solution within the triangle (see Fig. 5) can be linearly approximated as:

\[
\varphi = B_1 + B_2 x_1 + B_3 x_2. \tag{14}
\]

Thus, the coefficient of \(B_1\), \(B_2\), and \(B_3\) can be obtained from the following system:

\[
\begin{bmatrix}
B_1 \\
B_2 \\
B_3
\end{bmatrix} =
\begin{bmatrix}
1 & (x_2)_a \\
1 & (x_2)_b \\
1 & (x_2)_c
\end{bmatrix}^{-1}
\begin{bmatrix}
\varphi_a \\
\varphi_b \\
\varphi_c
\end{bmatrix}. \tag{18}
\]

To be able to use the direct forcing IB method, the three-step projection method described in Section 2 will be adapted as the following three steps:

**Step A**: Perform the advection step using Eq. (3) and get the intermediate values of \(u_i^*\).

**Step B**: For the non-advection step (i) described in Eq.(4), we further split this equation into the following two equations.
\[ \frac{u_i^* - u_i^h}{\Delta t} = v \nabla^2 u_i^* \quad (19) \]
\[ \frac{u_i^h - u_i^s}{\Delta t} = f_i^n \quad (20) \]

Here \( u_i^h \) stand for the intermediate velocity, and it should be mentioned that \( u_i^h = u_i^* \) for the fluid points. Thus, we obtain the actual forcing term as
\[ f_i^n = \frac{u_i^* - u_i^h}{\Delta t} - v \nabla^2 u_i^h + \frac{\partial (u_i u_j)}{\partial x_j} \quad (21) \]

An alternative description can be found in Eq. (20). It is particularly convenient to use Eq. (20) for the evaluation of the momentum forcing term. Therefore, the force exerted on the body by the fluid can be expressed as,
\[ F_i = -\int \rho_f f_i dS = -\int \rho_f \frac{u_i^* - u_i^h}{\Delta t} dS, \quad (22) \]

Here we need to calculate all the solid and forcing points, where \( dS = dx dx_2 \). Note that it does not make sense to integrate on the fluid domain since \( f_i^n \) equals to 0 for the fluid points.

**Step C:** For the non-advection step (ii) described in Eq. (5b), after invoking \( u_i^* \) into Eq. (5b), the solution can be updated with Eq. (5a).

It should be mentioned that no boundary treatment for the body is considered in the calculation because we solve the Poisson equation in the entire domain, based on the IB method above. We will also discuss the boundary treatment on a square domain in Section 5. Validation of the Poisson equation solver, which could contribute to the boundary treatment for other approaches on a complex body.

**5. Validation of the Poisson equation solver**

**5.1 Setup of the verification case**

The performance of the IB-GHPC method for the Poisson equation introduced in Section 3.2 is investigated in the section. The following pressure solution of a 2D Taylor-Green vortex at a given time instant is used as an analytical function for the verification of the Poisson equation solver
\[ \frac{p}{\rho u^2} (x_1, x_2, t) = \frac{L}{4} (\cos (2k x_1) + \cos (2k x_2)) e^{(-\alpha t)} \quad (23) \]

Here \( k = \frac{2\pi}{L} \) is the wavenumber, in which \( L \) is the length of the (doubly-periodic) domain, \( t^* = k^2 v t = \frac{4\pi^2 (Ut_0)^2}{L} \frac{1}{Re} \) is dimensionless time and \( Re = \frac{U_0 L}{v} \) is the Reynolds number. Note that, in contrast to many presentations throughout the literature, the Taylor-Green vortex solution above (see also the forthcoming Section 6.2) is posed in a dimensionally-consistent manner, with all characteristic physical variables properly included. We have also introduced the arbitrary coefficient \( \alpha \) for convenience, which only acts as a factor in the definition of the Reynolds number, as clearly indicated above. This enables the formulation above (combined with that in Section 6.2) to be easily reconciled with other presentations of this problem throughout the literature, where various definitions of the Reynolds number (i.e. various values for \( \alpha \)) are seemingly in use. We here set \( \alpha = 1 \) and consider \( U_0 = 1 \) m/s, \( v = 1 \) m$^2$/s, and \( k = 1 \) m$^{-1}$, yielding \( Re = 2\pi \). To simulate the same case as Bardazzi et al. [19], we set \( \gamma = 1 \) and consider the solution at the dimensionless time \( t^* = 10^{-4} \) in the analysis which follows.

Here, we consider a rectangular fluid domain \( \Omega = L \times L = [0, 2\pi] \times [0, 2\pi] \) m$^2$. Both Dirichlet and Neumann boundary value problems (BVPs) are studied to test the accuracy of the IB-GHPC Poisson solver. The \( L_2 \) error of the pressure in the fluid domain will be calculated as:
\[ E_{L2} = \sqrt{\frac{\sum_{i=1}^{N} (p_i^{num} - p_i^{an})^2}{\sum_{i=1}^{N} (p_i^{an})^2}} \quad (24) \]
where \( N \) is the total number of pressure points on the grid, \( p_i^{num} \) and \( p_i^{an} \) are the numerical and analytical solutions at the \( i \)-th point, respectively.
5.2 Convergence study and the effect of IB treatment

In contrast to a boundary-fitted grid, the IB method enforces the corresponding (Dirichlet, Neumann or Robin) boundary conditions on the boundaries immersed in a Cartesian background grid. The unknowns are the pressure on the fluid points and the ghost points. The effect of the boundary position in the IB-GHPC method described in Section 3.2 is also discussed in this section. For the Taylor-Green vortex problem over a square domain, an example of the Cartesian background grid and immersed boundaries are shown in Fig. 6. There is one layer of ghost points (marked by red filled squares) outside of the fluid domain. We consider a uniform grid with \( dh = dx_1 = dx_2 \) and define the distance between the fluid boundary and the layer of ghost points as \( d \). Both Dirichlet- and Neumann-type BVPs are solved for different mesh resolutions and distance \( d \). The same case has also been studied by Bardazzi et al. [19], who used a GHPC method with a boundary-fitted grid. Four different \( d \) values, namely \( d = 0, \frac{dh}{4}, \frac{dh}{2} \) and \( dh \) will be considered in our analysis to investigate how the position of the immersed boundaries will influence the accuracy of the solution. When \( d = 0 \), the IB method becomes the same approach as using a boundary-fitted grid.

Fig. 7 shows the \( L_2 \) errors of the present numerical results for fluid points as a function of \( dh/L \) for different \( d \) values. \( L \) is the length and width of the domain. The numerical results digitalized from [19] based on a boundary-fitted GHPC method and that based on a traditional second-order FDM are included for comparison. The two dashed straight lines denoted by ‘2nd order’ and ‘4th order’ represent the second- and fourth-order convergency rate, respectively. The results for Dirichlet- and Neumann-type BVPs are presented in Fig.7a and Fig.7b, respectively. Our results with \( d = 0 \) are seen to be very close to those of Bardazzi et al. [19]. Furthermore, a fourth-order accuracy is confirmed for the IB-GHPC method. As can be understood from Fig. 7a, the position of the IB has a negligible effect on the \( L_2 \) errors for the Dirichlet BVP. However, it is also clear from Fig. 7b that the \( L_2 \) errors for the Neumann BVP vary for different \( d \) values. In general, the errors are the smallest for \( d = dh \), corresponding to a case where the fluid boundary coincides with the first outer layer of the fluid points (blue crosses). This observation can be understood through Eq. (13), which was applied to satisfy the Neumann boundary conditions on the boundary points (green filled circles). When the boundary points are closer to the cell center, the normal derivatives calculated by Eq. (13) are more accurate. One can make an analogy to a finite difference approximation, where a central difference scheme is more accurate than a forward or backward difference scheme using the same number of stencil nodes.

As expected, the fourth-order GHPC methods show much faster convergence compared with the second-order FDM. Taking the Neumann BVP as an example, which is relevant when solving the NSEs with solid walls based on a projection method, the FDM and IB-GHPC require \( \frac{dh}{L} \approx 0.001 \) and 0.04 respectively to achieve the same accuracy of \( 10^{-5} \). A fair comparison of the efficiency of different methods would be to compare the required CPU time to achieve the same level of accuracy. In Table 1, the CPU time to achieve two different accuracy levels (\(-1.5E-4\) and \(-3.0E-5\)) is presented for the GHPC and FDM methods. \( d = dh \) has been used in the GHPC method. Much less CPU time is required for the GHPC method to reach the same accuracy as the second-order FDM. In addition to the total CPU time, the CPU time used in setting up the problem (grid generation, stencil preparation, setting up matrices, etc.) and in solving the sparse-matrix equations is also presented, respectively. In the case of time-domain simulation using a fixed Cartesian background grid, the stencil coefficients only need to be updated close to the moving boundaries as part of the IB method, and thus solving the sparse-matrix equation will be the most time-consuming task.
Fig. 6 IB and background grid used in the IB GHPC method for the Taylor-Green vortex problem over a rectangular domain. $d$ is the distance between the fluid boundary and the layer of ghost points. The blue thick line is the boundary. ➔: fluid point in the domain, ■: ghost point outside the boundary, ◦: intersection point.

a) Dirichlet

b) Neumann

Fig. 7 $L_2$ errors of the numerical solution of the pressure field for the Dirichlet- and Neumann-type boundary value problems. $d$ is the distance between the fluid boundary and the layer of ghost points, $d_h = dx_1 = dx_2$.

Table 1. Comparison of CPU time to achieve the same target accuracy.

<table>
<thead>
<tr>
<th>Target accuracy</th>
<th>Required grid resolution</th>
<th>Total time [s]</th>
<th>Setting-up time [s]</th>
<th>Sparse-matrix time [s]</th>
</tr>
</thead>
<tbody>
<tr>
<td>GHPC $\sim 1.5E-4$</td>
<td>28×28</td>
<td>0.3</td>
<td>0.2</td>
<td>0.1</td>
</tr>
<tr>
<td>FDM</td>
<td>278×278</td>
<td>3.5</td>
<td>0.8</td>
<td>2.7</td>
</tr>
<tr>
<td>GHPC $\sim 3.0E-5$</td>
<td>41×41</td>
<td>0.4</td>
<td>0.1</td>
<td>0.3</td>
</tr>
<tr>
<td>FDM</td>
<td>598×598</td>
<td>13.7</td>
<td>3.8</td>
<td>9.9</td>
</tr>
</tbody>
</table>
6. Studies on accuracy of the new NSEs solver

6.1. Lid-driven cavity flow

In this simulation, the lid-driven cavity flow where Reynolds number ($Re = \frac{UL}{\nu}$) equals to 1000 is chosen as the first validation case. The square-shape fluid cavity with the length and height $L$ equal to 1 m is studied. The upper plane moves at a constant velocity $U = 1$ m/s. The fluid density is $10^3$ kg/m$^3$. The non-slip boundary conditions for the velocities on four faces are shown in Fig. 8, where $\frac{\partial p}{\partial n} = 0$ is used for pressure [39-41]. The problem is solved as an initial boundary value problem in the time domain. Five different grid resolutions, namely 16×16, 32×32, 64×64, 128×128, and 256×256, have been used to investigate the convergence of the numerical solution. The minimum horizontal velocity $u_{1,\text{min}}$ along the vertical centerline is plotted in Fig. 9 as a function of time to identify the required simulation duration for the flow to become steady. It can be seen from Fig. 9 that 100 s is sufficient for this purpose. Thus, later steady results presented in this section will be based on the solution at $t = 100$ s. The contour plots of the steady horizontal and vertical velocities based on the 128×128 grid are shown in Fig. 10a and Fig. 10b, respectively.

![Diagram of boundary conditions](image1)

**Fig. 8** Definition of the boundary conditions of the lid-driven cavity problem.

![Time series of minimum horizontal velocity](image2)

**Fig. 9** Time series of the minimum horizontal velocity $u_{1,\text{min}}$ along the vertical centerline. Five different grid resolutions are considered.
a) The contour of $u_1$

b) The contour of $u_2$

Fig. 10 Contour plots of the steady horizontal velocity $u_1$ and vertical velocity $u_2$ based on 128×128 grid.

To reveal the convergence of the solution, the steady velocities of $u_1$ and $u_2$ along the vertical and horizontal centerlines are presented in Fig. 11a and Fig. 11b for five different grids, respectively. It is evident that the results based on 128×128 and 256×256 grids are almost identical, indicating that it is adequate to use a 128×128 grid for such a simulation at $Re = 1000$.

The resulting streamlines based on a 128×128 grid are presented in Fig. 12. Three vortices are shown in Fig. 12 and the positions of vorticities are presented in Table 1. Benchmark results from the previous works of Ghia et al. [39] and Vanka [40] are also provided for comparison in Table 2, which serve to validate the present results. It can be concluded that all the results are in good agreement with the benchmark results, indicating that the present NS solver has been properly implemented with convergence.
Fig. 12 Streamlines of the steady numerical results based on a 128 × 128 grid resolution at Re = 1000.

### Table 2. Comparison of the positions of the main vortex left and right vortices with the benchmark results.

<table>
<thead>
<tr>
<th>Comparison</th>
<th>Main vortex</th>
<th>Left vortex</th>
<th>Right vortex</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>x/L</td>
<td>y/L</td>
<td>x/L</td>
</tr>
<tr>
<td>Present results</td>
<td>0.5344</td>
<td>0.5625</td>
<td>0.0828</td>
</tr>
<tr>
<td>Ghia et al. [39]</td>
<td>0.5313</td>
<td>0.5625</td>
<td>0.0859</td>
</tr>
<tr>
<td>Vanka [40]</td>
<td>0.5313</td>
<td>0.5664</td>
<td>0.0859</td>
</tr>
</tbody>
</table>

#### 6.2. Taylor-Green vortex

To assess the accuracy of the present simulation method, the evolution of a periodic decaying Taylor-Green vortex is studied in this section [42]. The unsteady velocities and pressure of the flow can be described by the following solutions which satisfy the NSEs [43]

\[
\begin{align*}
\frac{u_1}{U_0}(x_1, x_2, t) &= -\cos(kx_1)\sin(kx_2)e^{-2t^*}, \\
\frac{u_2}{U_0}(x_1, x_2, t) &= \sin(kx_1)\cos(kx_2)e^{-2t^*},
\end{align*}
\]

The pressure follows the function in Eq. (23), now with \( \gamma = -1 \). The dimensionless time \( t^* \) and Reynolds number \( Re \) are as defined in Section 5.1. The numerical simulation is performed on a 2 m × 2 m square domain. Utilizing the parameters \( k=\pi \text{ m}^{-1}, U_0=1 \text{ m/s}, \nu=0.1 \text{ m}^2/\text{s} \) yields a Reynolds number \( Re=10 \), where the factor \( \alpha=1/2 \) (see Section 5.1) has been invoked to bring our definition in line with that used by Kim et al. [43]. Periodic boundary conditions are imposed on the four boundaries of the domain. At each time instance, the error norms of velocities and pressure in the fluid domain can be calculated as

\[
E_2 = \left[ \frac{1}{N} \sum_{k=1}^{N} (\zeta_{k}^{\text{num}} - \zeta_{k}^{\text{an}})^2 \right]^{1/2}
\]

\[
E_{\text{max}} = |\zeta_{k}^{\text{num}} - \zeta_{k}^{\text{an}}|_{\text{max}}
\]
Here $\zeta_\lambda^{num}$ and $\zeta_\lambda^{an}$ are the numerical and analytical solutions of velocity or pressure at the $i$-th point in the fluid.

![Pressure fields at the time $t = 0.3s$.](image1)

**Fig. 13** Pressure fields at the time at $t = 0.3s$.

![Vorticity at the time $t = 0.3s$.](image2)

**Fig. 14** Vorticity at the time at $t = 0.3s$, $Vorticity = (\frac{\partial u_2}{\partial x_1} - \frac{\partial u_1}{\partial x_2})(L/U_0)$.

The contour plots of the pressure and vorticity at time $t=0.3s$ are shown in Fig. 13 and Fig. 13, respectively. To study the accuracy, four different uniform grids, namely $16\times16$, $32\times32$, $64\times64$, and $128\times128$, are considered to calculate the two errors norms $E_2$ and $E_{max}$ defined in Eq. (27) and Eq. (28), respectively. In Fig. 15, the solutions are compared with a second-order finite difference solver, where the GHPC method is replaced by another Poisson equation solver based on classical second-order FDM. In the figure, $U$-FDM and $P$-FDM represent the results for the horizontal velocity $u_1$ and the pressure $p$ obtained by the second-order FDM solver. The results of Kim et al. [43] based on a second-order FVM solver are also included in the comparisons. A dashed straight line representing a second-order convergence rate is included in the plots as well for comparison. The time step $\Delta t$ has been chosen to be sufficiently small such that the truncation error in the time integration has negligible effects in the convergence study for these grid sizes. In general, all the methods demonstrate second-order accuracy for both the velocities and pressure, which is expected as all the spatial derivatives in the advection and diffusion terms are discretized by second-order schemes. It seems that the convergence rate of the pressure solution is limited by the accuracy of the forcing term on the right-hand side of the Poisson equation Eq. (5b), which is evaluated by a second-order scheme. However, a very interesting observation is that much lower errors are achieved in the present results using IB-GHPC as the Poisson equation solver, indicating that much better accuracy for the
pressure has been achieved. Compared with second-order FDM and FVM, both error norms $E_2$ and $E_{\text{max}}$ for the pressure are consistently reduced. This can be explained by the high accuracy of the IB-GHPC method. It is worth mentioning that the pressure on structures is typically the most important parameter when evaluating their flow-induced hydrodynamic loads, which implies that with the embedded GHPC Poisson equation solver one can apply coarser meshes to achieve the same accuracy as the traditional second-order solvers.

Fig. 15 Spatial convergence of the pressure and velocity field at $t = 0.3\text{s}$. U-FDM, P-FDM are the results for horizontal velocity $u_1$ and the pressure $p$ obtained by FDM, respectively. U-FVM is the results for velocity $u_1$ taken from Kim et al. [43].

6.3 Flow over a stationary free circular cylinder

Flow around a circular cylinder is a classical test case within hydrodynamics, and it is considered as a canonical numerical test, since a great amount of theoretical and experimental studies at different Reynolds numbers are available in the literature. In Section 6.3.1, an accuracy study is conducted for Stokes flow around a free circular cylinder, where analytical solution exists for very small Reynold numbers, to test the performance of the present solver when dealing with a solid body in a fluid. Results for some larger Reynold numbers are presented in Section 6.3.2.

6.3.1 Study of accuracy

The accuracy of the presented solver is analyzed by considering the steady uniform Stokes flow past an isolated free cylinder. A uniform Stokes flow ($Re = UD/v = 0.001$) passing a cylinder of diameter $D=2\text{ m}$ with undisturbed velocity $U = 1\text{ m/s}$, is studied on a domain with size $L \times H = 3D \times 3D$. The solution near the cylinder in the polar coordinate $(r, \theta)$ is given in [44][45], and a general dimensionless form centered on the cylinder could be described as,

$$\chi = \frac{-DU \sin \theta (r_0 \log(r_0) - \frac{r_0}{2} + \frac{1}{2r_0})}{2v \log(Re/2)}$$

$$u_1 = \frac{\partial \chi}{\partial x_2}, \quad u_2 = -\frac{\partial \chi}{\partial x_1}$$

where $\chi$ is the stream function and $r_0 = 2r/Re$ with $r = O(1)$. The contour plots for the horizontal and vertical velocities close to the cylinder are presented in Fig. 16.

The calculations are carried out with four levels of uniform grids: $30 \times 30, 90 \times 90, 150 \times 150$, and $450 \times 450$. Here, the grid number is not selected by doubling the specific grids as usual which is limited by the staggered grid. Since the
analytical solutions are not directly available from [44] and [45], we have used the finest grid (450×450) to solve for a reference solution for pressure. To do that, the analytical solutions for the velocities in Eq. (29) and Eq. (30) are used as the initial condition and Dirichlet boundary conditions for the velocities.

To minimize the errors due to temporal discretization, a very small CFL number ($CFL = 10^{-4}$) has been used in all analyses. The error calculated with Eq. (24) is presented in Fig. 17 for different uniform meshes denoted by different $D/dh$ values. The solutions based on a traditional second-order FDM are also included for comparisons. We can find that the velocity errors decrease with $\Delta x^2$, confirming that both solvers provide second-order accuracy for the velocity solutions. It is also clear that a more accurate solution for the pressure is achieved in the present results, compared with the second-order FDM solver. The present pressure solution has an accuracy close to second order, in particular for finer numerical meshes. However, it is also noticed that the increase of accuracy after using the GHPC method is not as significant as that in the Taylor-Green vortex problem studied in Section 5.2, which might be due to the use of the 1st order immersed-boundary method. It is expected that the accuracy of the present solver can be further improved by applying a higher-order immersed-boundary method, and higher-order discretization for the convection and diffusion terms.

![Fig. 16 Referenced solution for a Stokes flow past a fixed cylinder.](image)

![Fig. 17 The accuracy of the present solver. $U_1$-FDM, $U_2$-FDM and $P$-FDM are the results for horizontal velocity $u_1$, the vertical $u_2$ and pressure $p$ obtained by FDM, respectively.](image)

### 6.3.2 Validations

In this section, we will further validate the present solver by studying cases over an even greater range of Reynolds
numbers. The results will be compared with other references results in the literature. A sketch of the computational domain and boundaries is shown in Fig. 18. Uniform velocity \( U \) is imposed at the left inlet boundary. Non-slip boundary conditions are applied on the wall and \( \frac{\partial u_1}{\partial x_1} = 0, \frac{\partial u_2}{\partial x_2} = 0 \) are used for outlet boundaries. The cylinder diameter \( D = 1 \) m, \( L = 40D \), and \( H = 20D, S = 10D \). \( U \) is fixed as 1 m/s in our later analyses, while \( v \) varies to achieve different Reynolds numbers. The cylinder is treated by the direct forcing IB method in a uniform Cartesian grid, following the procedure described in Section 4.

A uniform grid \( dx_1 = dx_2 = dh \) is used in this study. To decide a proper grid resolution and time step for the flow over a circular cylinder, we have considered the following normalized physical parameters:

1. \( D_n = \frac{D}{dh} \) is used to describe the grid number along the diameter of the cylinder;

2. \( C_n = \frac{\Delta t U}{dx} \) is the Courant-Friedrichs-Lewy (CFL) number, which should be less than 1 as required by the advection equation. When it is solved by an explicit time-integration scheme, the CFL requirement due to the diffusion equation is not needed, as the diffusion terms are treated by an implicit scheme;

3. \( T_n = \frac{\Delta t U}{D} \) is proportional to the time it takes for the flow to travel from one end to the other end of the cylinder;

The grid is fixed at 1000×500 with 25 grid points along the diameter of the cylinder. The time step \( \Delta t \) is mainly determined by the advection CFL number \( C_n \), as an implicit scheme for diffusion term has been applied. \( \Delta t \) is set in the range of 0.001–0.01 s in this study for different Reynolds numbers to ensure \( C_n < 1 \). Fig. 19 shows the simulated vorticity fields at four Reynolds numbers (Re=0.1, 1, 10, 100) from the fine grid simulation. The dimensionless form of vorticity is defined as \( \text{Vorticity} = (\partial u_2/\partial x_1 - \partial u_1/\partial x_2)(D/U) \). The plots in the left column present the flow in the entire computational domain, while those in the right column are zoomed-in plots close to the cylinder. The flow solutions at low Reynolds numbers appear to be steady, which are in close agreement with those reported in the literature [2]. For Reynolds number equal to 0.1 and 1, creeping flow without separation is predicted, as shown in Fig. 19a and Fig. 19b. When the Reynolds number increases to 10, a fixed pair of symmetric vortices are observed in the wake of the cylinder, as shown in Fig. 19c. As shown in Fig. 19d, for Re=100, a change of flow structure appears which gives birth to vortex shedding with a laminar vortex street behind the cylinder.

Different wake structures imply the drag and lift force acting on the cylinder. The force on the cylinder does change periodically in time oscillating around mean, although the incoming flow is completely symmetric. The drag and lift coefficients \( (C_D, C_L) \) can be easily described as
Here $F_x$ and $F_x'$ are the drag and lift forces in the horizontal and vertical directions, respectively. Similarly, the mean drag coefficient ($C_D$) and fluctuations ($C_D'$, $C_L'$) are defined as

$$C_D = \frac{2F_x}{\rho_D U^2}$$  \hspace{1cm} (31)$$
$$C_D' = \frac{2F_x'}{\rho_D U^2}$$  \hspace{1cm} (32)$$
$$C_L' = \frac{2F_x'}{\rho_D U^2}$$  \hspace{1cm} (33)$$

$F_x'$ is the oscillating part of the drag force defined as $F_{x1}' = F_{x1} - \bar{F}_{x1}$. $F_{x2}'$ is the oscillating lift force $F_{x2}' = F_{x2} - \bar{F}_{x2}$, where $\bar{F}_{x2}$ stands for mean lift force which is 0 for the stationary cylinder  \cite{2}\cite{46}.

Fig. 20 shows the comparisons for the average drag coefficients at different Reynolds numbers ($Re = 0.1, 1, 10, 100, 200$). The reference data include the results based on Oseen-Lamb laminar theory \cite{46,48}, experimental results by Wieselsberger \cite{44}, and the rest come from \cite{2}. Good agreement between the present numerical solution and the reference results is observed in Fig. 20. The evolution of the drag and lift coefficients as a function of time for a period of 200 s ($T = 200$ s) at $Re = 100, 200$ are plotted in Fig. 21. Meanwhile, drag and lift coefficients, as well as Strouhal number ($St = f_v D/|U|$) which is a normalized vortex-shedding frequency ($f_v$), are also summarized in Table 3, where some reference data are also presented for comparison. The simulation results from Uhlmann \cite{50}, Mimeau et al. \cite{51}, Xu & Wang \cite{52}, and Ji et al. \cite{53} are based on IB methods, while Braza et al. \cite{54}, Liu et al. \cite{55}, and Taira & Colonius \cite{56} have applied body-fitted grids. It can be seen from Table 2 that our results agree very well with the above-mentioned numerical results.

The Strouhal number is a function of the Reynolds number. The results at $Re = 50, 100, 150, 200$ are compared with physical experimental data obtained from Williamson \cite{49} and Roshko \cite{57}, as shown in Fig. 22. All our above results are in excellent agreement with the reference data as expected, which further demonstrates the accuracy of our approach.
Fig. 19 Vorticity contours at four Reynolds numbers, \( \text{Vorticity} = (\partial u_2 / \partial x_1 - \partial u_1 / \partial x_2)(D/U) \). a) \( Re = 0.1 \); b) \( Re = 1 \); c) \( Re = 10 \); d) \( Re = 100 \).

d) \( Re = 100 \)

Fig. 20 Comparison of time-averaged drag coefficients as a function of Reynolds number.
Fig. 21 Evolution of the instantaneous drag and lift coefficients as a function of time. Here \( T = 200 \text{s} \). a) \( Re = 100 \); b) \( Re = 200 \).

Table 3 Comparison of the drag and lift coefficients for the flow around a stationary cylinder at \( Re = 100 \) and 200.

<table>
<thead>
<tr>
<th>Re</th>
<th>Reference data</th>
<th>( C_D )</th>
<th>( C_D' )</th>
<th>( C_L' )</th>
<th>( St )</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>Present results</td>
<td>1.50 ±0.024</td>
<td>±0.39</td>
<td>±0.167</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Williamson (1989) [49]</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.164</td>
</tr>
<tr>
<td></td>
<td>Uhlmann (2005) [50]</td>
<td>1.453 ±0.011</td>
<td>±0.339</td>
<td>±0.169</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ji et al. (2012) [53]</td>
<td>1.376 ±0.010</td>
<td>±0.339</td>
<td>±0.169</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Liu et al. (1998) [55]</td>
<td>1.350 ±0.019</td>
<td>±0.293</td>
<td>±0.16</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mimeau et al. (2015) [51]</td>
<td>1.40 ±0.010</td>
<td>±0.32</td>
<td>±0.165</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Xu &amp; Wang (2006) [52]</td>
<td>1.423 ±0.013</td>
<td>±0.34</td>
<td>±0.171</td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>Present results</td>
<td>1.48 ±0.05</td>
<td>±0.77</td>
<td>±0.200</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Williamson (1989) [49]</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.197</td>
</tr>
<tr>
<td></td>
<td>Taira &amp; Colonius (2007) [56]</td>
<td>1.35 ±0.048</td>
<td>±0.68</td>
<td>±0.196</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Ji et al. (2012) [53]</td>
<td>1.354 ±0.044</td>
<td>±0.682</td>
<td>±0.2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Braza et al. (1986) [54]</td>
<td>1.386 ±0.040</td>
<td>±0.766</td>
<td>±0.2</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Liu et al. (1998) [55]</td>
<td>1.31 ±0.049</td>
<td>±0.69</td>
<td>±0.192</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Mimeau et al. (2015) [51]</td>
<td>1.44 ±0.05</td>
<td>±0.75</td>
<td>±0.200</td>
<td></td>
</tr>
</tbody>
</table>

Fig. 22 Variation of Strouhal number as a function of Reynolds number for flow passing a stationary smooth circular cylinder.
7. Conclusions

This paper presents enhanced numerical solutions of the 2D incompressible NSEs by adopting a newly developed generalized harmonic polynomial cell (GHPC) method for the Poisson equation, which has a fourth-order accuracy. The introduction of the IB strategy in the GHPC method is shown to further improve the accuracy of the GHPC method, while making it possible to study problems with complex fluid boundaries. This enables us to use the new IB-GHPC Poisson equation solver in the numerical solution of incompressible NSEs based on a standard staggered grid for flow velocities and pressure. For simplicity, but sufficient for demonstration purposes, a new solver for the incompressible NSEs has been developed by following a three-step projection method of Xiao [10] and a simple and efficient direct forcing IB framework of Yang et al.[31-33]. The solver has been verified and validated through the study of lid-driven cavity flow, Taylor-Green vortex, and flow around a smooth circular cylinder. It is also confirmed in the numerical results that the pressure solution can be greatly improved by only replacing the second-order pressure Poisson solver with the IB-GHPC solver, without changing other parts of the code. This feature is attractive from a practical point of view, as the IB-GHPC Poisson equation solver can easily be ‘plugged’ into other existing NS equation solvers based on staggered grids.

Acknowledgments

This work was supported by the Chinese Scholarship Council (CSC) [grant number is 201906680072]. Additionally, great thanks to the Department of Mechanical Engineering at the Technical University of Denmark for hosting the first author as a visiting PhD student.

Appendix A. The summary of the 2D HPC method

In the 2D HPC method, the fluid domain is divided into quadrilateral elements, and four neighboring quadrilateral elements with nine grid points are defined as a cell as illustrated in Fig. 2. The harmonic polynomials used in the HPC method satisfy the Laplace equation everywhere in the space, and they are given by the real and imaginary parts of the complex polynomial:

\[ z^n = (x_1 + i x_2)^n = \chi_n(x_1, x_2) + i \Gamma_n(x_1, x_2) \]  

(A1)

Here \( n \) is an integer and \( i^2 = -1 \). \( \chi_n \) and \( \Gamma_n \) are the components in \( x \) and \( y \) directions. The corresponding polynomials for \( n \leq 4 \) are shown in Table A1.

<table>
<thead>
<tr>
<th>( n )</th>
<th>( z^n )</th>
<th>( \chi_n(x_1, x_2) )</th>
<th>( \Gamma_n(x_1, x_2) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>( x_1 + i x_2 )</td>
<td>( x_1 )</td>
<td>( x_2 )</td>
</tr>
<tr>
<td>2</td>
<td>( (x_1^2 - x_2^2) + 2i x_1 x_2 )</td>
<td>( x_1^2 - x_2^2 )</td>
<td>( 2x_1 x_2 )</td>
</tr>
<tr>
<td>3</td>
<td>( (x_1^3 - 3 x_1 x_2^2) + i(3 x_1^2 x_2 - x_2^3) )</td>
<td>( x_1^3 - 3 x_1 x_2^2 )</td>
<td>( 3 x_1^2 x_2 - x_2^3 )</td>
</tr>
<tr>
<td>4</td>
<td>( (x_1^4 - 6 x_1^2 x_2^2 + x_2^4) + i(4 x_1^3 x_2 - 4 x_1 x_2^3) )</td>
<td>( x_1^4 - 6 x_1^2 x_2^2 + x_2^4 )</td>
<td>( 4 x_1^3 x_2 - 4 x_1 x_2^3 )</td>
</tr>
</tbody>
</table>

Table A1. The complex polynomials and harmonic polynomials for \( n \leq 4 \) in 2D.

The velocity in the cell is approximated by the eight surrounding points. The solution within each cell can be approximated by the first eight harmonic polynomials in Table A1:

\[ \phi(x_1, x_2) = \sum_{\ell=1}^{8} b_\ell q_\ell(x_1, x_2) \]  

(A2)
\[ \phi = b_1 + b_2 x_1 + b_3 x_2 + b_4 (x_1^2 - x_2^2) + b_5 x_1 x_2 + b_6 (x_1^3 - 3x_1 x_2^2) + b_7 (3x_1^2 x_2 - x_2^3) + b_8 (x_1^4 - 6x_1^2 x_2^2 + x_2^4) \] (A3)

where \( \phi \) is the velocity potential in a cell, \( q_\xi (\xi = 1, \ldots, 8) \) are the first eight harmonic polynomials listed in Table A1, and \( b_\xi (\xi = 1, \ldots, 8) \) are unknown coefficients that can be achieved by the linear system associated with the boundary conditions:

\[ b_\lambda = \sum_{\xi=1}^{8} w_{\lambda \xi} \phi_\xi \quad \lambda = 1, \ldots, 8 \] (A4)

Here \( w_{\lambda \xi} (\lambda, \xi = 1, \ldots, 8) \) are the elements of the inverse of the matrix \([D]\), whose elements are \( d_{\lambda \xi} = q_\xi (x_\lambda, x_\xi) \), where \( \lambda \) and \( \xi \) are the row and column index in the matrix, respectively [20-23].

Finally, the velocity potential in a cell can be expressed by a linear combination of values on the eight boundary nodes,

\[ \phi (x_\mu, x_\nu) = \sum_{\lambda=1}^{8} \sum_{\xi=1}^{8} w_{\lambda \xi} q_\xi (x_\mu, x_\nu) \phi_\lambda \] (A5)

**Appendix B. Computation of Eq. (9) in the 2D GHPC method**

In Eq. (9), we get three terms in similar forms as Eq. (A2) and \( \tilde{u}(x) \) is approximated by a series of harmonic polynomials defined in Table A1. The \( \sigma(x_1, x_2) \) term is approximated by a bi-quadratic interpolation, which has the same order as \( \tilde{u}(x) \). A particular solution \( \tilde{\psi}(x) \) can be constructed by integrating \( \Delta \tilde{\psi}(x_1, x_2) = \sigma(x_1, x_2) \) in Eq. (8). The polynomials \( q_\lambda (x_1, x_2), h_\xi (x_1, x_2), \) and \( g_\xi (x_1, x_2) \) are listed in Table B1, where \( \lambda = 1, \ldots, 8; \ \xi = 1, \ldots, 9 \) [19].

<table>
<thead>
<tr>
<th>( n )</th>
<th>( q_\lambda (x_1, x_2) )</th>
<th>( h_\xi (x_1, x_2) )</th>
<th>( g_\xi (x_1, x_2) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>( x_1^2 + x_2^2 )/4</td>
</tr>
<tr>
<td>2</td>
<td>( x_1 )</td>
<td>( x_1 )</td>
<td>( x_1 x_2^2 )/2</td>
</tr>
<tr>
<td>3</td>
<td>( x_2 )</td>
<td>( x_2 )</td>
<td>( x_1^2 x_2 )/2</td>
</tr>
<tr>
<td>4</td>
<td>( x_1^2 - x_2^2 )</td>
<td>( x_1^2 )</td>
<td>( x_1^4 )/12</td>
</tr>
<tr>
<td>5</td>
<td>( x_1 x_2 )</td>
<td>( x_1 x_2 )</td>
<td>( x_1^2 x_2 + x_1 x_2^2 )/12</td>
</tr>
<tr>
<td>6</td>
<td>( x_1^2 - 3x_1 x_2^2 )</td>
<td>( x_1^2 )</td>
<td>( x_1^4 )/12</td>
</tr>
<tr>
<td>7</td>
<td>( 3x_1^2 x_2 - x_2^3 )</td>
<td>( x_1^2 x_2 )</td>
<td>( x_1^2 x_2^2 )/12</td>
</tr>
<tr>
<td>8</td>
<td>( x_1^4 - 6x_1^2 x_2^2 + x_2^4 )</td>
<td>( x_1 x_2^2 )</td>
<td>( x_1 x_2^2 )/12</td>
</tr>
<tr>
<td>9</td>
<td>( x_1^2 x_2^2 )</td>
<td>( x_1^4 (-x_1^2 + 15x_2^2) + x_1^2 (-x_1^2 + 15x_2^2) )/360</td>
<td></td>
</tr>
</tbody>
</table>
References


620 https://doi.org/10.1016/j.jfluidstructs.2007.08.002.
[40] S.P. Vanka, Block-implicit multigrid solution of Navier-Stokes equations in primitive variables, Journal of


Research highlight

- A new 2D Navier-Stokes equations solver based on generalized harmonic polynomial cell (GHPC) method
- Immersed-boundary strategy introduced in the original GHPC method
- Demonstration of the enhanced pressure field solution by using the new solver
- The present Immersed-boundary GHPC can be ‘plugged’ in other existing Navier-Stokes equations solvers based on stagger grids
Declaration of interests

☒ 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.

☐ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests:

[Blank space for declaration]