Computers and Mathematics with Applications 214 (2026) 94–134

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

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

Very high-order accurate wall vorticity treatment on curved boundaries with polygonal meshes for incompressible vorticity equations

Ricardo Costa [a,b,*], Stéphane Clain [c], Gaspar J. Machado [d], João M. Nóbrega [a,b]

[a] Institute for Polymers and Composites, University of Minho, Azurém Campus, 4804-058, Guimarães, Portugal
[b] Department of Polymer Engineering, University of Minho, Azurém Campus, 4804-058, Guimarães, Portugal
[c] Centre of Mathematics, University of Coimbra, 3000-143, Coimbra, Portugal
[d] Centre of Mathematics, University of Minho, Azurém Campus, 4800-058, Guimarães, Portugal

ARTICLE INFO

Keywords:
Incompressible fluid flows
Streamfunction-vorticity formulation
Finite volume method
Very high-order convergence
Arbitrary curved boundaries
Immersed boundaries

ABSTRACT

Vorticity formulations of the incompressible Navier-Stokes equations circumvent the challenges associated with the pressure-velocity numerical coupling. Furthermore, vorticity dynamics in challenging fluid flow problems can be analysed more effectively within a vorticity-based numerical framework. However, deriving suitable vorticity boundary conditions for numerical simulations with high-order accuracy on curved domains remains a significant challenge for these formulations, particularly in three dimensions. Consequently, most existing work considers only rectangular domains with regular grids, limiting its application to geometries of practical interest. This study presents suitable wall vorticity boundary conditions for incompressible vorticity equations on arbitrary curved domains, introducing a simple, effective, and highly accurate numerical treatment based on a finite volume discretisation on unstructured meshes. Moreover, a novel technique is employed to discretise the boundary conditions on a linear piecewise approximation of the curved boundary, circumventing the challenges associated with curved meshes while preserving very high-order accuracy. The two-dimensional streamfunction-vorticity formulation is analysed both theoretically and numerically, but these advancements establish a foundation for other vorticity-based formulations and the three-dimensional case, in which the treatment of vorticity boundary conditions remains fundamentally unchanged. To demonstrate the accuracy and effectiveness of the proposed methodology, several benchmark incompressible fluid flow problems in non-trivial two-dimensional curved domains are presented, including the semi-elliptical lid-driven cavity problem, for which very high orders of convergence are achieved without any regularisation.

1. Introduction

The incompressible Navier-Stokes system in primitive variables (pressure and velocity) is the classical formulation for two- and three-dimensional flows of incompressible viscous fluids. However, this system poses significant numerical challenges due to its

* Corresponding author.
E-mail address: rcosta@dep.uminho.pt (R. Costa).

https://doi.org/10.1016/j.camwa.2026.04.004
Received 13 October 2025; Received in revised form 18 February 2026; Accepted 2 April 2026
0898-1221/© 2026 Published by Elsevier Ltd.

non-linear nature, pressure-velocity coupling through the continuity condition, and the absence of straightforward boundary conditions for the pressure. For many decades, extensive literature on numerical techniques and computer algorithms has been developed to address these issues regarding accuracy, stability, and efficiency. In addition to advanced numerical methods, alternative equivalent formulations of the incompressible Navier-Stokes equations (equivalent in the continuum, but generally not equivalent when discretised) have also received substantial attention.

Conventional non-primitive variables formulations of the incompressible Navier-Stokes equations are based on the vorticity transport equation, derived from taking the curl of the momentum balance equation in primitive variables, and, in the absence of source terms, is expressed as $\partial\omega/\partial t + \nabla\cdot(u\omega) - \nu\nabla^2\omega = 0$. The vorticity, ω, is a pseudo-vector field defined as the curl of the velocity vector that measures the local spinning motion of a continuum medium, and is useful for understanding fluid dynamics, particularly about turbulence, circulation, and the formation and behaviour of vortices. In two-dimensional flows, this formulation is particularly convenient, as the vorticity vector has a unique non-zero component (perpendicular to the plane of motion), hence reducing the vorticity transport equation to a scalar equation. Several formulations arise depending on the remaining physical quantities and associated equations chosen to ensure the continuity condition and close the system of partial differential equations.

Although numerical solvers for primitive variables formulations of the Navier-Stokes equations have achieved remarkable robustness over the past decades, the development of non-primitive variables formulations remains an active research area. Recent studies have focused on designing numerical frameworks based on non-primitive variables to address a wide range of applications, including natural convection problems [1,2], generalised Newtonian fluid flows [3], fluid transport in porous media [4], binary fluid flows [5], boundary-layer phenomena [6], shock-capturing problems [7], and three-dimensional flow configurations [8,9]. Particular attention has also been devoted to complex geometries [10,11], which are of central importance in many engineering applications.

1.1. Streamfunction-vorticity formulation

The streamfunction-vorticity formulation in two dimensions is the most classical form of the incompressible Navier-Stokes equations in non-primitive variables, whose development and application became more prominent following foundational papers such as those by Thom, 1933 [12], and Woods, 1954 [13]. The streamfunction, ψ, is a potential function whose derivatives are associated with the velocity components. Thus, a standard Poisson equation, in the form $\nabla^2\psi = -\omega$, ensures the continuity condition and closes the system of partial differential equations. The streamfunction-vorticity formulation has several advantages over the pressure-velocity formulation, such as:

- The pressure gradient, regarded as a source of numerical complexity, is implicitly handled and does not need to be computed separately, avoiding the difficulties associated with the pressure-velocity coupling, such as staggered meshes, stabilisation procedures, and pressure boundary conditions [14,15].
- For two-dimensional problems, the number of variables and equations is reduced, as the streamfunction-vorticity formulation uses two scalar quantities instead of velocity and pressure, thus reducing the computational burden.
- The velocity components are obtained from the streamfunction derivatives, guaranteeing that the continuity condition is always satisfied, both locally and globally, hence eliminating the need to enforce a solenoidal velocity field separately and ensuring that the mass balance remains conserved.
- Neither a coupled solution procedure nor a projection method for a segregated solution procedure is necessary, since the vorticity transport and streamfunction Poisson equations can be solved robustly using a simple iterative algorithm that updates the coupling terms at each iteration.
- Even in the presence of solution singularities, such as in the classical lid-driven cavity problem, high orders of convergence can still be achieved without the need for any regularisation techniques, since the streamfunction and vorticity preserve a higher degree of regularity than the primitive variables.
- The streamfunction-vorticity formulation and other vorticity-based formulations provide a more natural and direct framework for analysing vorticity dynamics, which is crucial for many engineering applications.

Despite the above advantages, the introduction of a potential function into the vorticity formulation also introduces a gauge freedom that must be fixed, since the potential functions ψ and $\psi + C$ represent the same velocity field for any constant C. A Dirichlet boundary condition for the streamfunction corresponds to the integral of the tangential velocity along the boundary, up to a constant that must be specified. In simply connected domains (comprising a single closed boundary), randomly fixing this constant suffices to resolve the streamfunction gauge freedom. However, in case of problems involving multiply connected domains, prescribing a Dirichlet boundary condition on each closed boundary leads to an unknown constant that is not independent of the others, since there is only one gauge freedom. A conventional approach to address this issue is to arbitrarily fix one of these constants and subsequently determine the others as part of the solution. Consequently, the system of linear equations is also augmented with an additional equation for each unknown constant, obtained from integrating the vorticity flux along the associated boundary.

1.2. Boundary conditions

Although the streamfunction introduces gauge freedom into the system, specifying the boundary conditions is straightforward and does not pose any difficulties. Indeed, two boundary conditions (Dirichlet and Neumann) can be derived directly from the normal and

tangential velocity components specified on the boundary, resulting in an over-constrained Poisson equation for the streamfunction. For vorticity, the conventional practice is to prescribe homogeneous Dirichlet boundary conditions at inlets (assuming that the fluid does not tend to rotate before entering the domain) and homogeneous Neumann boundary conditions at outlets (assuming that the domain is long enough for the flow to develop). The most important and challenging scenario is to prescribe a physically meaningful vorticity boundary condition on solid walls, as no reasonable physical principle seems to exist, which makes the vorticity transport equation under-constrained. This particular configuration of boundary conditions does not imply that the continuum problem is overdetermined for the streamfunction, but only that the vorticity must be compatible, in some sense, with the prescribed boundary conditions for the streamfunction. Nevertheless, the absence of boundary conditions for the vorticity transport equation on solid walls, combined with the over-specification of boundary conditions for the streamfunction Poisson equation, raises significant challenges for the numerical solution of the streamfunction-vorticity formulation.

The classical approach to circumvent the absence of vorticity boundary conditions on solid walls involves exploiting the relationship between vorticity and streamfunction at the boundary to derive approximate formulas for determining the wall vorticity. Artificial boundary conditions for vorticity are then imposed based on the implicit streamfunction (and, sometimes, also vorticity) values on the inner grid points in the vicinity, as well as the known boundary conditions for the streamfunction. This procedure has been widely investigated in the foundational works on the numerical solution of the streamfunction-vorticity formulation using the finite difference method [12,13,16–20]. Several formulae were proposed, and studies were carried out, usually based on a truncated Taylor series representing the discrete approximation of the streamfunction Poisson equation. These formulae incorporated varying numbers of terms, considering either inner grid values of the streamfunction or the streamfunction and vorticity together, with each formula providing a different level of accuracy and stability to the numerical procedure. These artificial vorticity boundary conditions are local in nature, requiring only knowledge of the values of the inner grid points nearby and having no direct connection to other boundary points through the prescribed formula. Similar techniques based on local vorticity boundary conditions were later proposed for the finite element method [21–25] and the control volume-based finite element method [26,27], where various approximate formulae for the wall vorticity were employed, based either on finite differences or the element function basis.

Most works on the numerical solution of the streamfunction-vorticity formulation adopted segregated approaches with relaxation solution procedures, as available computational capabilities at that time could not support a coupled solution procedure due to memory and calculation requirements. Consequently, the vorticity transport equation and the streamfunction Poisson equation were usually solved sequentially using iterative algorithms, with wall vorticity computed between iterations using the aforementioned formulae. Either the Dirichlet or Neumann boundary condition is therefore prescribed when solving the streamfunction Poisson equation, while the approximated wall vorticity is used to impose suitable Dirichlet boundary conditions on the vorticity transport equation. Some authors have proposed computing wall vorticity alongside the streamfunction Poisson equation to stabilise the calculation procedure. Indeed, solving the streamfunction and vorticity equations separately usually necessitates significant under-relaxation, particularly at high Reynolds numbers, thereby slowing down convergence. This issue stems from the coupling between the streamfunction and vorticity, as well as from the artificial boundary conditions used for the vorticity. In contrast, a directly coupled approach, which requires an implicit treatment of wall vorticity for boundary conditions on solid walls, can alleviate this issue, yielding a much more stable and robust procedure with faster convergence, although this was not the conventional practice at the time.

In the traditional coupled approach, the resulting algebraic equations from the discretisation of the vorticity transport and streamfunction Poisson equations are solved alongside the artificial vorticity boundary conditions. Several authors have, however, proposed coupled solution procedures that eliminate the need for such conditions. Examples include the finite difference method [28], which uses an implicit method to solve the boundary-layer equations, and the finite element method [29–32], which uses a mixed variational formulation for the Navier-Stokes equations in streamfunction-vorticity formulation. While the absence of artificial vorticity boundary conditions seems more natural in the finite element method, finite difference methods can also be implemented in a way that avoids the need for such conditions at solid walls. According to some authors, the key to avoiding this specification lies not in the use of finite element methods, but in solving the discrete system as a coupled set of equations rather than iterating between the vorticity transport and streamfunction Poisson equations [31]. Thus, in a finite difference method, the streamfunction Poisson equation can be discretised by imposing both Dirichlet and Neumann boundary conditions, yielding more equations than unknowns needed to determine the streamfunction. Conversely, the vorticity transport equation can be discretised without imposing any boundary conditions, resulting in fewer equations than unknowns to determine the vorticity. This results in an over-constrained system of linear equations for the streamfunction that cannot be solved, whereas the system of linear equations for the vorticity has no unique solution. However, the system of coupled linear equations can be solved to determine both the streamfunction and the vorticity simultaneously and uniquely, without the need to prescribe artificial vorticity boundary conditions.

1.3. High-order accuracy

Developments in the streamfunction-vorticity formulation have predominantly focused on the numerical treatment of vorticity boundary conditions, which are widely recognised as the principal challenge in this approach. The development of compact finite difference schemes [33–38] has also received considerable attention for achieving high-order of convergence without increasing the stencil size, which is typically required when implementing high-order accurate finite difference formulas. Although compact schemes are capable of delivering high-order accuracy, their application has been largely restricted to regular grids on simple rectangular domains. For general and complex geometries, more sophisticated techniques must be employed, such as immersed boundary methods [39–41] adapted to the high-order case [42–44], or formulations in general curvilinear coordinates [10,11].

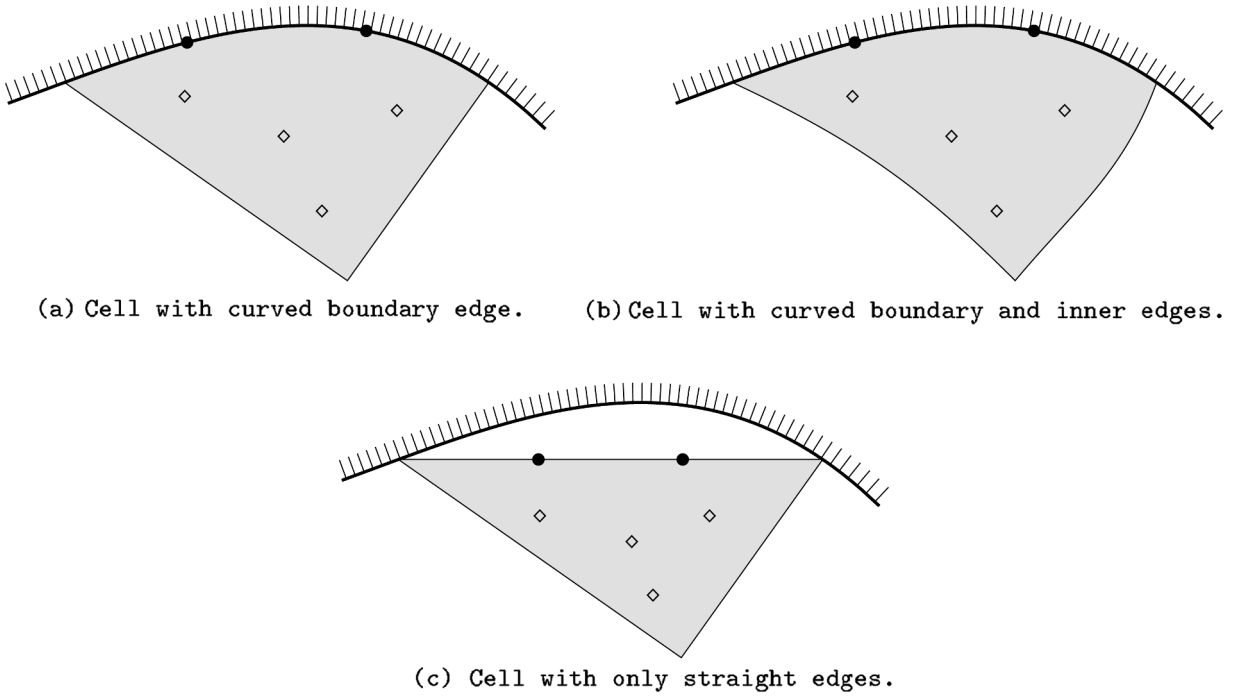


Fig. 1. Curved and polygonal mesh elements with illustrative quadrature points on the edges (solid circles) and in the cells (empty diamonds).

Nevertheless, research on the numerical solution of the streamfunction-vorticity formulation in complex geometries remains scarce, and no high-order accurate discretisation schemes on unstructured meshes for complex geometries have been reported.

1.4. Novelties of the article

Although some recent papers address the discretisation of the streamfunction-vorticity formulation using finite cell methods on unstructured meshes and immersed boundary methods on regular grids, the treatment of complex geometries with practical application remains underdeveloped. Apart from compact finite difference schemes, which are essentially limited to regular grids, no high-order accurate discretisation schemes have yet been developed for the streamfunction-vorticity formulation on general meshes and geometries. Such a limitation is primarily due to the wall vorticity treatment, since the vorticity transport and streamfunction Poisson equations can be discretised using any preferred method for parabolic partial differential equations, provided the suitable boundary conditions. Indeed, the key factor is to prescribe a simple and efficient numerical technique to determine the wall vorticity with high-order accuracy on arbitrary curved boundaries using unstructured meshes. This is the primary objective of the present work, which entails two main challenges, described below.

The approximation of wall vorticity for prescribing suitable vorticity boundary conditions is an extremely delicate issue, as an inaccurate evaluation can result in a significant loss of convergence order. In particular, on unstructured meshes, the adopted strategy cannot rely on classical Taylor expansion techniques and must consider the boundary curvature to achieve high-order convergence. Indeed, the inclusion of the boundary curvature in the determination of the wall vorticity is not mentioned in the literature, yet it is a fundamental factor for a consistent formulation.

Developing high-order accurate discretisation schemes for curved domains requires a specific boundary condition treatment to avoid second-order convergence limitations due to the geometrical mismatch between the physical and computational boundaries when using linear piecewise meshes. The classical approach to treating boundary conditions on arbitrary curved boundaries involves using curved mesh elements to eliminate this geometrical mismatch and recover optimal convergence orders.

Curved mesh approaches, such as the standard isoparametric elements method, utilise curved elements that fit the physical curved domain to a certain degree to prevent deterioration in accuracy. Sophisticated and computationally intensive algorithms are employed for this purpose, and, in addition to the boundary edges, curved inner edges may also be necessary to better accommodate the mesh elements (see Fig. 1(a) and(b)). Quadrature points on curved edges and cells may also be necessary, which is challenging, particularly in three-dimensional meshes, along with complex non-linear transformations for mapping the elements. Conversely, polygonal mesh approaches avoid these issues by using standard mesh generation algorithms and, consequently, standard quadrature rules (see Fig. 1(c)). However, accuracy deterioration can be avoided only if specific numerical techniques are employed to recover the optimal convergence order. The interested reader is referred to Costa et al., 2023 [45], for a comprehensive discussion and literature review.

The reconstruction for off-site data (ROD) method is a novel technique that was initially proposed for convection-diffusion equations [45–50], and recently extended to the incompressible Stokes and Navier-Stokes equations in primitive variables [51,52]. This

method handles arbitrary curved domains with a linear piecewise approximation of the physical boundary and employs polynomial reconstructions with specific linear constraints to convert the boundary conditions prescribed on the physical boundary into equivalent conditions on the computational boundary. Building on the ROD technique, this study proposes a high-order accurate wall vorticity treatment for a finite volume scheme on unstructured meshes to numerically solve the streamfunction-vorticity formulation, using linear piecewise meshes on arbitrary curved domains.

The proposed treatment is designed for seamless integration into the preferred space discretisation technique, including finite difference, finite volume, and discontinuous Galerkin methods. In this study, a high-order accurate finite volume scheme, combined with the ROD technique for the boundary treatment, is employed to discretise the vorticity transport and streamfunction Poisson equations, since the authors have previously developed this technique for solving the incompressible Navier-Stokes equations in primitive variables [51,52]. This approach allows for a direct comparison of the numerical properties and computational performance of both formulations using the same base discretisation scheme, as demonstrated in the semi-elliptical lid-driven cavity test case. Nevertheless, the novel concepts presented in this study are general enough to be applied to different numerical frameworks.

1.5. Organisation of the article

The remaining sections of the article are organised as follows. The problem statement for the two-dimensional incompressible Navier-Stokes equations and the mathematical derivation of the streamfunction-vorticity formulation are presented in Section 2, whereas the derivation of suitable boundary conditions for the non-primitive variables is discussed in Section 3. The numerical techniques related to the polynomial reconstruction method and the reconstruction for off-site data method are recalled in Section 4, and employed in Section 5 for the finite volume discretisation of the streamfunction-vorticity formulation. Section 6 is dedicated to the verification of the proposed method in terms of accuracy, convergence order, and computational efficiency, featuring several benchmark test cases with analytical or reference solutions. The article ends with the conclusions in Section 7.

2. Mathematical formulation

The steady-state flow of an incompressible Newtonian fluid is studied in a two-dimensional simply or multiply connected physical domain Ω with an arbitrary smooth curved physical boundary Γ. Using the Cartesian coordinate system $\boldsymbol{x} := (x, y)$, the associated outward unit normal vector is $\boldsymbol{n} := \boldsymbol{n}(\boldsymbol{x}) := (n_x, n_y)$ with $n_x := n_x(\boldsymbol{x})$ and $n_y := n_y(\boldsymbol{x})$, and the unit tangential vector is $\boldsymbol{t} := \boldsymbol{t}(\boldsymbol{x}) := (t_x, t_y)$ with $t_x := t_x(\boldsymbol{x})$ and $t_y := t_y(\boldsymbol{x})$, where $\boldsymbol{t}(\boldsymbol{x})$ and $\boldsymbol{n}(\boldsymbol{x})$ are positively oriented (see Fig. 2). For multiply connected domains (basically, any domain with holes), each closed path representing a physical boundary is denoted as Γ^k, such that $\Gamma = \cup_{k=0}^{K} \Gamma^k$, where K is the number of holes. For convenience, Γ^0 corresponds to the external physical boundary, such that for $K = 0$ (without holes), $\Gamma = \Gamma^0$ corresponds to a simply connected domain.

2.1. Primitive variables formulation

The primitive variables formulation of the steady-state two-dimensional incompressible Navier-Stokes equations for a Newtonian fluid with constant dynamic viscosity $\mu > 0$ and constant density $\rho > 0$ consists of the momentum and mass balance equations, given as

$$(\boldsymbol{u} \cdot \nabla)\boldsymbol{u} - \nu \nabla^2 \boldsymbol{u} + \frac{1}{\rho} \nabla p = \boldsymbol{f}, \qquad \text{in } \Omega, \tag{1}$$

$$\nabla \cdot \boldsymbol{u} = 0, \qquad \text{in } \Omega, \tag{2}$$

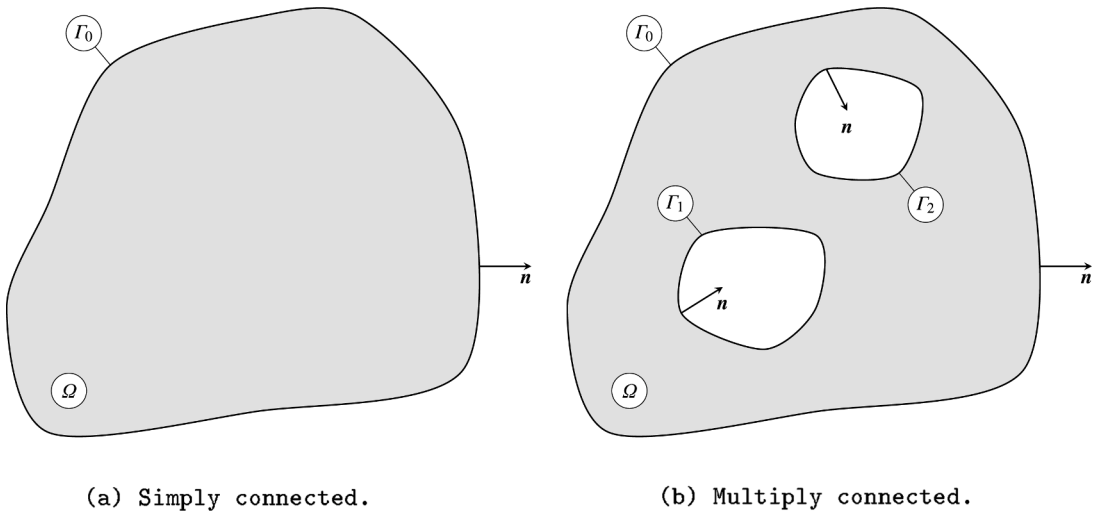
respectively, where $\nu = \mu/\rho$ is the constant kinematic viscosity, $\boldsymbol{u} := \boldsymbol{u}(\boldsymbol{x}) := (u_x, u_y)$ with $u_x := u_x(\boldsymbol{x})$ and $u_y := u_y(\boldsymbol{x})$ is the unknown velocity function, $p := p(\boldsymbol{x})$ is the unknown pressure function, while $\boldsymbol{f} := \boldsymbol{f}(\boldsymbol{x}) := (f_x, f_y)$ with $f_x := f_x(\boldsymbol{x})$ and $f_y := f_y(\boldsymbol{x})$ is a given body force function per unit area acting on the continuum, for instance, the gravitational force. The system of equations (1) and (2) is often referred to as the primitive steady-state formulation of the incompressible Navier-Stokes equations, which relates the pressure and velocity unknowns.

Remark 1. In the present work, the differential operator $\nabla \cdot \boldsymbol{A}$ for a matrix $\boldsymbol{A}$ follows the usual convention of a column-wise divergence operator, whereas $\nabla \boldsymbol{a}$ and $\nabla^2 \boldsymbol{a}$ for a vector $\boldsymbol{a}$ follow the standard conventions for component-wise gradient and Laplace operators, respectively.

2.2. Streamfunction-vorticity formulation

In three-dimensional fluid flows, vorticity is a physical pseudovector field that describes the local rotational motion of the fluid and is defined as the curl of the velocity function. For planar fluid flows, it follows that the vorticity vector always points perpendicularly to the plane of motion, such that it can be represented with a scalar function $\omega := \omega(\boldsymbol{x})$ in two-dimensional problems, given as

$$\omega = \frac{\partial u_y}{\partial x} - \frac{\partial u_x}{\partial y}. \tag{3}$$



(a) Simply connected. (b) Multiply connected.

Fig. 2. Curved physical domain.

The vorticity transport equation can be derived from taking the curl of the momentum balance equation in three dimensions (see Appendix A). For planar fluid flows, the components parallel to the plane of motion vanish and, therefore, in two-dimensional problems, the vorticity transport equation reduces to a scalar equation (the perpendicular component) for the scalar vorticity, given as

$$\nabla \cdot (\boldsymbol{u}\omega) - \nu \nabla^2 \omega = f, \qquad \text{in } \Omega, \tag{4}$$

where $f := f(\boldsymbol{x})$ is the perpendicular component of the curl of the body force function per unit volume, given as

$$f = \frac{\partial f_y}{\partial x} - \frac{\partial f_x}{\partial y}. \tag{5}$$

Remark 2. In two-dimensional fluid flows, the vorticity transport equation can also be obtained by subtracting the derivatives of the first and second components of Eq. (1) with respect to y and x, respectively, in that order.

In two-dimensional fluid flows, the incompressibility condition can be satisfied exactly if the velocity field is expressed in terms of a scalar potential $\psi := \psi(\boldsymbol{x})$, the streamfunction, whose derivatives correspond to the velocity components, given as

$$u_x = \frac{\partial \psi}{\partial y} \qquad \text{and} \qquad u_y = -\frac{\partial \psi}{\partial x}, \qquad \text{in } \Omega, \tag{6}$$

or, in compact form, given as $\boldsymbol{u}(\boldsymbol{x}) = \nabla^{\perp}\psi(\boldsymbol{x})$. From the above definition, it follows that the streamfunction is well-defined only up to an additive constant and always satisfies the continuity equation intrinsically since

$$\nabla \cdot \boldsymbol{u} = \frac{\partial u_x}{\partial x} + \frac{\partial u_y}{\partial y} = \frac{\partial}{\partial x}\left(\frac{\partial \psi}{\partial y}\right) + \frac{\partial}{\partial y}\left(-\frac{\partial \psi}{\partial x}\right) = 0. \tag{7}$$

The streamfunction-vorticity formulation currently has four variables ($\omega(\boldsymbol{x})$, $\psi(\boldsymbol{x})$, $u_x(\boldsymbol{x})$, and $u_y(\boldsymbol{x})$) but only three equations (the vorticity transport equation and the relationship between streamfunction and velocity components). To close the system for the incompressible Navier-Stokes equations, a fourth equation is obtained after substituting the definition of streamfunction (6) into the definition of vorticity (3), yielding

$$\nabla^2 \psi = -\omega, \qquad \text{in } \Omega. \tag{8}$$

The complete system for the streamfunction-vorticity formulation, apart from boundary conditions, is given as

$$\nabla \cdot (\boldsymbol{u}\omega) - \nu \nabla^2 \omega = f, \qquad \text{in } \Omega, \tag{9}$$

$$\boldsymbol{u} = \nabla^{\perp}\psi, \qquad \text{in } \Omega, \tag{10}$$

$$\nabla^2 \psi = -\omega, \qquad \text{in } \Omega, \tag{11}$$

which is non-linear due to the convective term in the vorticity transport equation. In that regard, the so-called Picard method (also known as fixed-point iteration) is employed to linearise the vorticity transport equation by considering an additional velocity function $\boldsymbol{v} := \boldsymbol{v}(\boldsymbol{x}) := (v_x, v_y)$ with $v_x := v_x(\boldsymbol{x})$ and $v_y := v_y(\boldsymbol{x})$, and removing the second equation from the system, yielding the linearised streamfunction-vorticity formulation, given as

$$\nabla \cdot (\boldsymbol{v}\omega) - \nu \nabla^2 \omega = f, \qquad \text{in } \Omega, \tag{12}$$

$$\nabla^2\psi + \omega = 0, \qquad \text{in } \Omega, \tag{13}$$

with the unknown terms arranged on the left-hand side.

An iterative procedure is applied such that, at each fixed-point iteration, function $\boldsymbol{v}(\boldsymbol{x})$ corresponds to the previously computed velocity function, $\boldsymbol{u}(\boldsymbol{x})$, obtained from the streamfunction. Therefore, the solution of the above system of equations converges to the solution of the steady-state streamfunction-vorticity formulation, provided that function $\boldsymbol{v}(\boldsymbol{x})$ converges to $\boldsymbol{u}(\boldsymbol{x})$. Notice that both functions $\boldsymbol{u}(\boldsymbol{x})$ and $\boldsymbol{v}(\boldsymbol{x})$ stand for a velocity field, where the latter is known and, therefore, becomes a physical coefficient in the linearised problem. Introducing function $\phi := \phi(\boldsymbol{x})$ corresponding to the streamfunction at the previous fixed-point iteration, then $\boldsymbol{v}(\boldsymbol{x}) = \nabla^{\perp}\phi(\boldsymbol{x})$.

Remark 3. Although the elimination of pressure is widely regarded as the most significant advantage of vorticity-based formulations, pressure distribution analysis is often still necessary for practical fluid flow applications. The conventional procedure involves taking the divergence of the momentum balance equation to produce a Poisson equation for the pressure, which can be simplified using the continuity equation in the case of incompressible fluid flows. Further simplifications take place if constant density and viscosity are also assumed. Once the velocity has been determined, the pressure Poisson equation can be solved using any discretisation scheme for elliptic equations to calculate the approximate pressure distribution. In the case of the streamfunction-vorticity formulation, the velocity components are obtained from the streamfunction derivatives, as discussed above. In discrete terms, explicitly deriving the approximate piecewise velocities in the right-hand side of the pressure Poisson equation generally results in some loss of accuracy and convergence order for the pressure. Therefore, obtaining an accurate streamfunction approximation is essential for determining the pressure distribution in industrial flows.

3. Boundary conditions

Deriving the appropriate boundary conditions for the streamfunction-vorticity formulation is a fundamental concern, especially in arbitrary curved domains. In this work, only Dirichlet boundary conditions are prescribed for the velocity, given as

$$\boldsymbol{u} = \boldsymbol{u}^{\mathrm{B}}, \qquad \text{on } \Gamma, \tag{14}$$

where $\boldsymbol{u}^{\mathrm{B}} := \boldsymbol{u}^{\mathrm{B}}(\boldsymbol{x}) := \left(u_x^{\mathrm{B}}, u_y^{\mathrm{B}}\right)$ with $u_x^{\mathrm{B}} := u_x^{\mathrm{B}}(\boldsymbol{x})$ and $u_y^{\mathrm{B}} := u_y^{\mathrm{B}}(\boldsymbol{x})$ is a given velocity function on the physical boundary.

3.1. Local coordinate system

It is convenient to define the problem variables in terms of a local coordinate system to deal with arbitrary smooth curved physical boundaries. For each point $\boldsymbol{p}$ on the physical boundary, a negatively oriented vector basis $\boldsymbol{E}^{\mathrm{L}}(\boldsymbol{p}) = \{\boldsymbol{t}_{\boldsymbol{p}}, \boldsymbol{n}_{\boldsymbol{p}}\}$ with $\boldsymbol{t}_{\boldsymbol{p}} = \boldsymbol{t}(\boldsymbol{p})$ and $\boldsymbol{n}_{\boldsymbol{p}} = \boldsymbol{n}(\boldsymbol{p})$ is introduced for a local Cartesian coordinate system defined as

$$\boldsymbol{x} \to \boldsymbol{\eta}_{\boldsymbol{p}} := \boldsymbol{\eta}_{\boldsymbol{p}}(\boldsymbol{x}) := \left(\eta_{\boldsymbol{p}}(\boldsymbol{x}), \xi_{\boldsymbol{p}}(\boldsymbol{x})\right) = \left((\boldsymbol{x}-\boldsymbol{p})\cdot\boldsymbol{t}_{\boldsymbol{p}}, (\boldsymbol{x}-\boldsymbol{p})\cdot\boldsymbol{n}_{\boldsymbol{p}}\right). \tag{15}$$

Note that this coordinate system does not involve curvilinear coordinates that follow the boundary. Moreover, the transformation from the global coordinate system to the local coordinate system (and vice versa) is a linear orthogonal transformation. As such, for general curved boundaries, the local coordinate system associated with a reference point $\boldsymbol{p}$ (local coordinate system origin) is aligned only with the boundary normal and tangential vectors at the reference point location. That is, for a point $\boldsymbol{q}$ in the vicinity of the reference point, the associated normal and tangential vectors, $\boldsymbol{n}(\boldsymbol{q})$ and $\boldsymbol{t}(\boldsymbol{q})$, are not generally aligned with the local coordinate system associated to $\boldsymbol{p}$ as it does not follow the boundary curvature.

From the definition of the local coordinate system, it follows that, for a fixed boundary point $\boldsymbol{p}$ and the associated local coordinate system, the problem variables and boundary conditions are defined as follows:

- The velocity field in the local coordinate system, $\boldsymbol{u}^{\mathrm{L}} := \boldsymbol{u}^{\mathrm{L}}(\boldsymbol{\eta}_{\boldsymbol{p}}) := (u_\eta, u_\xi)$, with $u_\eta := u_\eta(\boldsymbol{\eta}_{\boldsymbol{p}})$ and $u_\xi := u_\xi(\boldsymbol{\eta}_{\boldsymbol{p}})$, is defined as
$$u_\eta = \boldsymbol{u}\cdot\boldsymbol{n}_{\boldsymbol{p}} \qquad \text{and} \qquad u_\xi = \boldsymbol{u}\cdot\boldsymbol{t}_{\boldsymbol{p}}. \tag{16}$$
- The vorticity field in the local coordinate system, $\omega^{\mathrm{L}} := \omega^{\mathrm{L}}(\boldsymbol{\eta}_{\boldsymbol{p}})$, is defined as
$$\omega^{\mathrm{L}} = \frac{\partial u_\xi}{\partial \eta_{\boldsymbol{p}}} - \frac{\partial u_\eta}{\partial \xi_{\boldsymbol{p}}}. \tag{17}$$
- The streamfunction field in the local coordinate system, $\psi^{\mathrm{L}} := \psi^{\mathrm{L}}(\boldsymbol{\eta}_{\boldsymbol{p}})$, is defined as
$$u_\eta = \frac{\partial \psi^{\mathrm{L}}}{\partial \xi_{\boldsymbol{p}}} \qquad \text{and} \qquad u_\xi = -\frac{\partial \psi^{\mathrm{L}}}{\partial \eta_{\boldsymbol{p}}}. \tag{18}$$
- The boundary velocity in the local coordinate system, $\boldsymbol{u}^{\mathrm{L,B}} := \boldsymbol{u}^{\mathrm{L,B}}(\boldsymbol{\eta}_{\boldsymbol{p}}) := \left(u_\eta^{\mathrm{B}}, u_\xi^{\mathrm{B}}\right)$, with $u_\eta^{\mathrm{B}} := u_\eta^{\mathrm{B}}(\boldsymbol{\eta}_{\boldsymbol{p}})$ and $u_\xi^{\mathrm{B}} := u_\xi^{\mathrm{B}}(\boldsymbol{\eta}_{\boldsymbol{p}})$, is defined as
$$u_\eta^{\mathrm{B}} = \boldsymbol{u}^{\mathrm{B}}\cdot\boldsymbol{n}_{\boldsymbol{p}} \qquad \text{and} \qquad u_\xi^{\mathrm{B}} = \boldsymbol{u}^{\mathrm{B}}\cdot\boldsymbol{t}_{\boldsymbol{p}}. \tag{19}$$

For general curved boundaries, note that for any point $\boldsymbol{\eta}_{\boldsymbol{p}} \in \Gamma$ in the local coordinate system associated with a reference point $\boldsymbol{p}$, the components $u_\eta^{\mathrm{B}}(\boldsymbol{\eta}_{\boldsymbol{p}})$ and $u_\xi^{\mathrm{B}}(\boldsymbol{\eta}_{\boldsymbol{p}})$ correspond to the normal and tangential velocity only at the local coordinate system origin, $\boldsymbol{p}$. Finally, the vorticity definition in the local coordinate system should yield the same signal as Eq. (3), hence the orientation of the tangent vector on the boundary is not arbitrary and $\boldsymbol{t}(\boldsymbol{p}) = \left(-n_y(\boldsymbol{p}), n_x(\boldsymbol{p})\right)$ must be chosen.

3.2. Boundary streamfunction

From the streamfunction definition (18), it follows that the normal velocity on the physical boundary corresponds to its tangential derivative. Since the streamfunction is only defined up to an additive constant, assume an arbitrary point $\boldsymbol{x}_k$ on the physical boundary subset Γ^k, $k = 0, \dots, K$, which corresponds to an arbitrary streamfunction value C_k. Therefore, it follows that the boundary integration of $\boldsymbol{u}^{\mathrm{B}}(\boldsymbol{x}) \cdot \boldsymbol{n}(\boldsymbol{x})$ between $\boldsymbol{x}_k$ and another point $\boldsymbol{x}$ on the same physical boundary subset, gives the boundary streamfunction, $\psi^{\mathrm{B}} := \psi^{\mathrm{B}}(\boldsymbol{x})$, that is

$$\psi^{\mathrm{B}}(\boldsymbol{x}) = \int_{\widehat{\boldsymbol{x}_k \boldsymbol{x}}} \boldsymbol{u}^{\mathrm{B}} \cdot \boldsymbol{n} \, \mathrm{d}s + C_k, \qquad \text{on } \Gamma^k, \tag{20}$$

where $\widehat{\boldsymbol{x}_k \boldsymbol{x}}$ is the boundary path between $\boldsymbol{x}_k$ and $\boldsymbol{x}$ and each physical boundary subset is assumed to admit a parametrisation in the form $\boldsymbol{x} := (x(s), y(s))$. Note that, as a necessary condition for the streamfunction to exist, the total mass flux on each closed physical boundary must vanish [53], that is

$$\oint_{\Gamma^k} \boldsymbol{u}^{\mathrm{B}} \cdot \boldsymbol{n} \, \mathrm{d}s = 0, \qquad k = 0, \dots, K. \tag{21}$$

For multiply connected domains, the presence of $K + 1$ arbitrary constants C_k indicates that the problem has multiple solutions, as different choices of these constants generally lead to different velocity fields, all of which are valid solutions. However, since the streamfunction is only defined up to an additive constant, only one constant can be fixed arbitrarily, leaving the remaining K constants undetermined. For convenience, assume that C_0 is fixed arbitrarily and that C_k, $k = 1, \dots, K$, are to be determined. An effective practice is to determine these constants as part of the overall solution, meaning that the problem has K additional unknown scalar constants. Then, to close the system, an integral condition on the vorticity flux along the associated physical boundary subsets is prescribed (see Appendix B), given as

$$\oint_{\Gamma^k} (\boldsymbol{u}\,\omega - \nu \nabla \omega) \cdot \boldsymbol{n} \, \mathrm{d}s = \oint_{\Gamma^k} \boldsymbol{f} \cdot \boldsymbol{n} \, \mathrm{d}s, \qquad k = 1, \dots, K. \tag{22}$$

This approach provides K scalar constraints (compatibility conditions) on the vorticity flux, which fixes the K unknown constants for the boundary streamfunction.

From the streamfunction definition (18), it also follows that the tangential velocity on the physical boundary corresponds to the boundary streamfunction normal derivative, that is

$$\nabla \psi^{\mathrm{B}} \cdot \boldsymbol{n} = -\boldsymbol{u}^{\mathrm{B}} \cdot \boldsymbol{t}, \qquad \text{on } \Gamma. \tag{23}$$

Remark 4. In the special case of confined flows with the no-slip condition, $\boldsymbol{u}^{\mathrm{B}}(\boldsymbol{x}) \cdot \boldsymbol{t}(\boldsymbol{x}) = 0$, and impermeable walls, $\boldsymbol{u}^{\mathrm{B}}(\boldsymbol{x}) \cdot \boldsymbol{n}(\boldsymbol{x}) = 0$, on the physical boundary, the streamfunction boundary value and normal derivative are simply given as $\psi^{\mathrm{B}}(\boldsymbol{x}) = C_k$ and $\nabla \psi^{\mathrm{B}}(\boldsymbol{x}) \cdot \boldsymbol{n}(\boldsymbol{x}) = 0$, respectively.

3.3. Boundary vorticity

Obtaining the boundary vorticity function is more challenging since, in general, the given boundary velocity function is not sufficient to completely derive the associated boundary vorticity from the definition (17). In this regard, the conventional practice involves substituting the known derivative term in the vorticity definition from the given boundary velocity function and providing an implicit approximation for the unknown derivative term. Despite this hurdle, boundary vorticity is treated similarly for both simply and multiply connected physical domains.

3.3.1. Polygonal domains

Consider a rectangular physical domain with physical boundaries aligned with the global coordinate system axes. For instance, on the top horizontal physical boundary, $\partial u_y(\boldsymbol{x})/\partial x$ can be derived from the prescribed vertical boundary velocity function, given as

$$\frac{\partial u_y}{\partial x} = \frac{\partial u_y^{\mathrm{B}}}{\partial x}, \qquad \text{on } \Gamma. \tag{24}$$

Conversely, although the horizontal velocity on the same boundary is known since the function $u_x^{\mathrm{B}}(\boldsymbol{x})$ is given, its derivative in the y-direction cannot typically be determined and remains unknown. In this case, from the definition of vorticity (3), the boundary vorticity, $\omega^{\mathrm{B}} := \omega^{\mathrm{B}}(\boldsymbol{x})$, can be written as

$$\omega^{\mathrm{B}} = \frac{\partial u_y^{\mathrm{B}}}{\partial x} - \frac{\partial u_x}{\partial y}, \qquad \text{on } \Gamma, \tag{25}$$

where the unknown derivative term is to be determined implicitly using an appropriate numerical approximation. Most of the published studies employ regular grids only and, therefore, a simple finite difference formula is used to approximate $\partial u_x(\boldsymbol{x})/\partial y = \partial^2 \psi(\boldsymbol{x})/\partial y^2$ at the boundary using the unknown streamfunction values at the inner grid points together with the boundary streamfunction value. For unstructured meshes, more elaborate interpolation techniques are required.

For general polygonal domains, where the physical boundary sides are not necessarily aligned with the coordinate system axes, a similar approach is adopted, considering a local coordinate system for convenience. In this case, for a reference point $\boldsymbol{p}$ on the

physical boundary and associated local coordinate system, $\partial u_\eta(\boldsymbol{\eta_p})/\partial\xi_{\boldsymbol{p}}$ is determined from the prescribed normal velocity function as

$$\frac{\partial u_\eta}{\partial \xi_{\boldsymbol{p}}} = \frac{\partial u_\eta^{\mathrm{B}}}{\partial \xi_{\boldsymbol{p}}}, \qquad \text{at } \boldsymbol{p} \in \Gamma, \tag{26}$$

while $\partial u_\xi(\boldsymbol{\eta_p})/\partial\eta_{\boldsymbol{p}}$ remains undetermined at that stage. Then, from the definition of vorticity (17), the boundary vorticity can be written as

$$\omega^{\mathrm{B}} = \frac{\partial u_\xi}{\partial \eta_{\boldsymbol{p}}} - \frac{\partial u_\eta^{\mathrm{B}}}{\partial \xi_{\boldsymbol{p}}}, \qquad \text{at } \boldsymbol{p} \in \Gamma, \tag{27}$$

where the unknown derivative term is to be determined implicitly using an appropriate numerical approximation.

3.3.2. Curved domains

Consider a domain with arbitrary curved physical boundaries. In this scenario, the approach introduced for general polygonal domains becomes inappropriate since $\partial u_\eta(\boldsymbol{\eta_p})/\partial\xi_{\boldsymbol{p}}$ cannot be determined by simply deriving the prescribed normal velocity function in the $\xi_{\boldsymbol{p}}$-direction. Indeed, the local coordinate system axes are not a parametrisation of the boundary, and, apart from its origin, $\boldsymbol{p}$, they do not generally match with the boundary. As such, $\partial/\partial\eta_{\boldsymbol{p}}$ and $\partial/\partial\xi_{\boldsymbol{p}}$ do not correspond to the normal and tangential derivatives on the boundary, apart from $\partial/\partial\eta_{\boldsymbol{p}}$ at the reference point $\boldsymbol{p}$, as the local coordinate system axes do not follow the boundary curvature anywhere.

Going back to the global coordinates with $\boldsymbol{u}(\boldsymbol{x})$, $\boldsymbol{n}(\boldsymbol{x})$, and $\boldsymbol{t}(\boldsymbol{x})$ as functions of $\boldsymbol{x}$, the following identity can be derived

$$\frac{\partial(\boldsymbol{u}\cdot\boldsymbol{n})}{\partial \boldsymbol{t}} = \frac{\partial \boldsymbol{u}}{\partial \boldsymbol{t}}\cdot\boldsymbol{n} + \boldsymbol{u}\cdot\frac{\partial \boldsymbol{n}}{\partial \boldsymbol{t}}, \qquad \text{on } \Gamma. \tag{28}$$

In particular, $\partial\boldsymbol{n}(\boldsymbol{x})/\partial\boldsymbol{t}(\boldsymbol{x}) = \kappa(\boldsymbol{x})\boldsymbol{t}(\boldsymbol{x})$, where $\kappa := \kappa(\boldsymbol{x})$ is the normal curvature, positive if the centre of curvature lies on the same side as the physical domain. Moreover, $(\partial\boldsymbol{u}(\boldsymbol{x})/\partial\boldsymbol{t}(\boldsymbol{x}))\cdot\boldsymbol{n}(\boldsymbol{x}) = [(\nabla\boldsymbol{u}(\boldsymbol{x}))\boldsymbol{t}(\boldsymbol{x})]\cdot\boldsymbol{n}(\boldsymbol{x})$ and, therefore, the above identity can be rewritten as

$$\frac{\partial(\boldsymbol{u}\cdot\boldsymbol{n})}{\partial \boldsymbol{t}} = [(\nabla\boldsymbol{u})\boldsymbol{t}]\cdot\boldsymbol{n} + \kappa\boldsymbol{u}\cdot\boldsymbol{t}, \qquad \text{on } \Gamma. \tag{29}$$

Since the transformation from the global to the local coordinate system is a linear orthogonal transformation, the above identity can be rewritten in the local coordinate system associated with a reference point $\boldsymbol{p}$ on the physical boundary as

$$\frac{\partial(\boldsymbol{u}^{\mathrm{L}}\cdot\boldsymbol{n}^{\mathrm{L}})}{\partial \boldsymbol{t}^{\mathrm{L}}} = \left[(\nabla^{\mathrm{L}}\boldsymbol{u}^{\mathrm{L}})\boldsymbol{t}^{\mathrm{L}}\right]\cdot\boldsymbol{n}^{\mathrm{L}} + \kappa\boldsymbol{u}^{\mathrm{L}}\cdot\boldsymbol{t}^{\mathrm{L}}, \qquad \text{on } \Gamma, \tag{30}$$

where ∇^{L} is the gradient operator and $\boldsymbol{n}^{\mathrm{L}} := \boldsymbol{n}^{\mathrm{L}}(\boldsymbol{\eta_p})$ and $\boldsymbol{t}^{\mathrm{L}} := \boldsymbol{t}^{\mathrm{L}}(\boldsymbol{\eta_p})$ are the boundary normal and tangential vectors, respectively, with respect to the local coordinates. In particular, $\boldsymbol{n}^{\mathrm{L}}(\boldsymbol{0}) = (1,0)^{\mathrm{T}}$ and $\boldsymbol{t}^{\mathrm{L}}(\boldsymbol{0}) = (0,1)^{\mathrm{T}}$ at the local coordinate system origin, $\boldsymbol{p}$. Thus, by further simplifying the above identity after expanding the normal and tangential vectors in the local coordinate system at the reference point $\boldsymbol{p}$, yields

$$\frac{\partial(\boldsymbol{u}^{\mathrm{L}}\cdot\boldsymbol{n}^{\mathrm{L}})}{\partial \boldsymbol{t}^{\mathrm{L}}} = \frac{\partial u_\eta}{\partial \xi_{\boldsymbol{p}}} + \kappa u_\xi \quad \Rightarrow \quad \frac{\partial u_\eta}{\partial \xi_{\boldsymbol{p}}} = -\kappa u_\xi + \frac{\partial(\boldsymbol{u}^{\mathrm{L}}\cdot\boldsymbol{n}^{\mathrm{L}})}{\partial \boldsymbol{t}^{\mathrm{L}}}, \qquad \text{at } \boldsymbol{p} \in \Gamma. \tag{31}$$

Using the above identity, the boundary vorticity in the local coordinate system, $\omega^{\mathrm{L,B}} := \omega^{\mathrm{L,B}}(\boldsymbol{\eta_p})$, can be obtained as

$$\omega^{\mathrm{L,B}} = \frac{\partial u_\xi}{\partial \eta_{\boldsymbol{p}}} + \kappa u_\xi^{\mathrm{B}} - \frac{\partial(\boldsymbol{u}^{\mathrm{L,B}}\cdot\boldsymbol{n}^{\mathrm{L}})}{\partial \boldsymbol{t}^{\mathrm{L}}}, \qquad \text{at } \boldsymbol{p} \in \Gamma, \tag{32}$$

substituting $u_\xi(\boldsymbol{\eta_p})$ by $u_\xi^{\mathrm{B}}(\boldsymbol{\eta_p})$ and $\boldsymbol{u}^{\mathrm{L}}(\boldsymbol{\eta_p})$ by $\boldsymbol{u}^{\mathrm{L,B}}(\boldsymbol{\eta_p})$ since the boundary velocity function is given and its derivative with respect to $\boldsymbol{t}^{\mathrm{L}}(\boldsymbol{\eta_p})$ can be computed analytically. As for rectangular domains, the obtained boundary vorticity formula comprises one implicit derivative in the $\eta_{\boldsymbol{p}}$-direction to be determined. Moreover, since $u_\xi(\boldsymbol{\eta_p}) = -\partial\psi^{\mathrm{L}}(\boldsymbol{\eta_p})/\partial\eta_{\boldsymbol{p}}$, then the boundary vorticity can be rewritten as

$$\omega^{\mathrm{L,B}} = -\frac{\partial^2\psi^{\mathrm{L}}}{\partial\eta_{\boldsymbol{p}}^2} + \kappa u_\xi^{\mathrm{L,B}} - \frac{\partial(\boldsymbol{u}^{\mathrm{L,B}}\cdot\boldsymbol{n}^{\mathrm{L}})}{\partial \boldsymbol{t}^{\mathrm{L}}}, \qquad \text{at } \boldsymbol{p} \in \Gamma. \tag{33}$$

Remark 5. In the special case of confined flows with impermeable walls, $\boldsymbol{u}^{\mathrm{B}}(\boldsymbol{x})\cdot\boldsymbol{n}(\boldsymbol{x}) = 0$, on the physical boundary, the boundary vorticity simply writes $\omega^{\mathrm{L,B}}(\boldsymbol{x}) = -\partial^2\psi^{\mathrm{L}}(\boldsymbol{\eta_p})/\partial\eta_{\boldsymbol{p}}^2 + \kappa(\boldsymbol{\eta_p})u_\xi^{\mathrm{B}}(\boldsymbol{\eta_p})$.

3.4. Boundary conditions for the streamfunction-vorticity formulation

Finally, after deriving the necessary formulas for the boundary streamfunction and vorticity from the prescribed boundary velocity, the system of linear equations (12) and (13) is supplemented with appropriate boundary conditions as follows:

- For the streamfunction, a Cauchy boundary condition (Dirichlet and Neumann boundary conditions imposed simultaneously) is prescribed on the physical boundary, given as

$$\psi = \psi^{\mathrm{B}}, \qquad \text{on } \Gamma, \tag{34}$$

$$\nabla\psi\cdot\boldsymbol{n} = \nabla\psi^{\mathrm{B}}\cdot\boldsymbol{n}, \qquad \text{on } \Gamma, \tag{35}$$

  where function $\psi^{\mathrm{B}} := \psi^{\mathrm{B}}(\boldsymbol{x})$ and its normal derivative are given in Eqs. (20) and (23).

Table 1
Notation and geometric properties for the cells and edges.

Mesh elements	Notation	Properties	Definition	Choice
Cells	c_i $i \in \mathcal{I}$	∂c_i	Boundary	
		$\lvert c_i \rvert$	Area	
		$\boldsymbol{m}_i := (m_{i,x}, m_{i,y}, m_{i,z})$	Reference point (can be any point in c_i)	Mass centre
		$\boldsymbol{q}_{i,q} := (q_{i,q,x}, q_{i,q,y}, q_{i,q,z})$	Quadrature points, $q = 1, \dots, Q$	Gauss-Legendre
		$\mathcal{N}_i$	Indices of the adjacent cells and boundary subset	
Inner edges	e_{ij} $i, j \in \mathcal{I}$	$\lvert e_{ij} \rvert$	Length	
		$\boldsymbol{m}_{ij} := (m_{ij,x}, m_{ij,y})$	Reference point (can be any point on e_{ij})	Midpoint
		$\boldsymbol{q}_{ij,r} := (q_{ij,r,x}, q_{ij,r,y})$	Quadrature points, $r = 1, \dots, R$	Gauss-Legendre
		$\boldsymbol{s}_{ij} := (s_{ij,x}, s_{ij,y})$	Unit normal vector from cell c_i to cell c_j	
		$\boldsymbol{r}_{ij} := (r_{ij,x}, r_{ij,y})$	Unit tangential vector	
Boundary edges	$e_{i\mathrm{B}}$ $i \in \mathcal{I}^{\mathrm{B}}$	$\lvert e_{i\mathrm{B}} \rvert$	Length	
		$\boldsymbol{m}_{i\mathrm{B}} := (m_{i\mathrm{B},x}, m_{i\mathrm{B},y})$	Reference point (can be any point on $e_{i\mathrm{B}}$)	Midpoint
		$\boldsymbol{q}_{i\mathrm{B},r} := (q_{i\mathrm{B},r,x}, q_{i\mathrm{B},r,y})$	Quadrature points, $r = 1, \dots, R$	Gauss-Legendre
		$\boldsymbol{s}_{i\mathrm{B}} := (s_{i\mathrm{B},x}, s_{i\mathrm{B},y})$	Unit normal vector from cell c_i	
		$\boldsymbol{r}_{i\mathrm{B}} := (r_{i\mathrm{B},x}, r_{i\mathrm{B},y})$	Unit tangential vector	

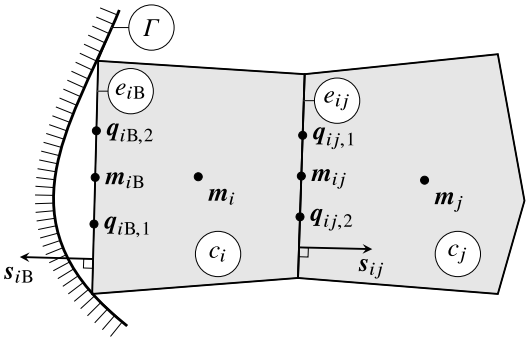


Fig. 3. Notation and geometric properties for the cells and edges.

- For the vorticity, the Dirichlet boundary condition is prescribed on the physical boundary, given as

$$\omega = \omega^{\mathrm{B}}, \qquad \text{on } \Gamma, \tag{36}$$

where the function $\omega^{\mathrm{B}} := \omega^{\mathrm{B}}(\boldsymbol{x})$ in global coordinates corresponds to the boundary vorticity in local coordinates, as given in Eq. (33).

For multiply connected domains, the system of linear equations is also supplemented with the compatibility conditions (22) to fix the unknown constants for the boundary streamfunction on inner physical boundary subsets.

4. Polynomial reconstruction

The polynomial reconstruction method is a powerful technique for computing local high-accurate approximations of the underlying solution and its derivatives on the mesh cell interfaces. In this regard, the reconstruction for off-site data (ROD) method [45–52] is a further development of the polynomial reconstruction method to enable treating general boundary conditions prescribed on arbitrary curved boundaries, while the problem is solved on polygonal meshes.

4.1. Polygonal meshes

The physical domain Ω is represented by a polygonal domain $\Omega_\Delta = \bigcup_{i\in\mathcal{I}} c_i$ consisting of N_{C} non-overlapping convex polygonal cells, with N_{E} edges, of which N_{B} are boundary edges. Cells are denoted as c_i, with $i \in \mathcal{I} = \{1, \dots, N_{\mathrm{C}}\}$, and an inner edge $e_{ij} = c_i \cap c_j$ is the common interface between two adjacent cells c_i and c_j, $j \neq i$. On the other hand, boundary edges $e_{i\mathrm{B}}$, $i \in \mathcal{I}^{\mathrm{B}}$, belong to the computational (surrogate) boundary $\Gamma_\Delta = \bigcup_{k=0}^{K} \Gamma_\Delta^k = \bigcup_{i\in\mathcal{I}} e_{i\mathrm{B}}$ with $\mathcal{I}^{\mathrm{B}} \subset \mathcal{I}$ being the index set of the boundary cells and Γ_Δ^k the linear piecewise approximation of the physical boundary subset Γ^k. For simplicity, each cell has at most one boundary edge.

The geometric properties for the cells are provided in Table 1, whereas Fig. 3 provides a schematic representation of the relevant data. Note that the inner edge e_{ij} is also denoted as e_{ji} and, therefore, reference and quadrature points are the same, that is, $\boldsymbol{m}_{ij} = \boldsymbol{m}_{ji}$ and $\boldsymbol{q}_{ij,r} = \boldsymbol{q}_{ji,r}$, whereas outward unit normal vectors satisfy $\boldsymbol{s}_{ij} = -\boldsymbol{s}_{ji}$.

Following the motivation presented in Section 1, the mesh consists solely of polygonal elements, therefore avoiding the difficulties of generating and employing curved meshes for curved domains. In this case, the computational domain, where the problem is

numerically solved, does not completely match the physical domain, where the governing equations are defined. More specifically, a geometrical mismatch exists between the computational boundary and the physical boundary, where the boundary conditions are prescribed. For uniform regular meshes of characteristic size h, the geometrical mismatch typically has an order of magnitude of $\mathcal{O}(h^2)$ between the physical and computational boundaries, potentially leading to a deterioration in accuracy without appropriate treatment.

4.2. Polynomial reconstruction method

Given a bounded scalar function $\phi := \phi(\boldsymbol{x})$, local polynomial approximations $\varphi := \varphi(\boldsymbol{x}) := \varphi(\boldsymbol{x}; \boldsymbol{a})$ of degree d are sought, written in compact form as

$$\varphi(\boldsymbol{x}; \boldsymbol{a}) = \boldsymbol{a} \cdot \boldsymbol{p}_d(\boldsymbol{x} - \boldsymbol{m}) = \sum_{\alpha=0}^{d} \sum_{\beta=0}^{d-\alpha} \left[a_{\alpha\beta} \left(x - m_x\right)^{\alpha} \left(y - m_y\right)^{\beta} \right], \tag{37}$$

where $\boldsymbol{p}_d(\boldsymbol{x})$ is a basis vector including all two-dimensional monomials up to degree d, $\boldsymbol{m} := \left(m_x, m_y\right)$ is a reference point, and vector $\boldsymbol{a} \in \mathbb{R}^n$ gathers the polynomial coefficients, with $n = (d+1)(d+2)/2$.

For each cell, ϕ_i, $i \in \mathcal{I}$, is a local piecewise mean-value approximation to function $\phi(\boldsymbol{x})$, defined as

$$\phi_i \approx \frac{1}{|c_i|} \int_{c_i} \phi \, \mathrm{d}\boldsymbol{x}. \tag{38}$$

Moreover, $\mathcal{S}$ denotes a stencil of the neighbouring cell indices in the vicinity of the polynomial reference, an edge or a cell. Each stencil must comprise a minimum of $s = n$ cells, but in practice, s is chosen higher than n to increase robustness and stability. The indices in $\mathcal{S}$ are selected from geometrically close cells, for which a neighbours-of-neighbours algorithm is implemented to ensure the contiguity of the stencil, that is, without gaps.

The linear least-squares method is then employed to minimise the squared errors between each approximate cell mean-value ϕ_i associated with the cells in the stencil and the polynomial function $\varphi(\boldsymbol{x}; \boldsymbol{a})$, introducing the weighted cost functional

$$F(\boldsymbol{a}; \boldsymbol{m}, \mathcal{S}, \boldsymbol{\omega}) = \sum_{\alpha \in \mathcal{S}} \omega_\alpha \left[\frac{1}{|c_\alpha|} \int_{c_\alpha} \varphi(\boldsymbol{x}; \boldsymbol{a}) \, \mathrm{d}\boldsymbol{x} - \phi_\alpha \right]^2, \tag{39}$$

where the integrals are evaluated numerically using quadrature rules of the same order as the polynomial degree plus one. In the cost functional, $\omega_\alpha = \omega\left(d_\alpha\right)$, $\alpha \in \mathcal{S}$, are positive weights determined from function $\omega\left(d_\alpha\right)$ given as

$$\omega\left(d_\alpha\right) = \frac{1}{\left(\sigma d_k\right)^{\delta} + 1}, \tag{40}$$

where $d_\alpha = \left\| \boldsymbol{m}_\alpha - \boldsymbol{m} \right\|$ is the distance from the reference point to the cell centre, and vector $\boldsymbol{\omega} \in \mathbb{R}^s$ gathers all the weights. Moreover, $\delta, \sigma \in \mathbb{R}$ are given parameters that control the sensitivity of the weights to the data distance, with optimal values obtained through numerical assessment.

4.2.1. Unconstrained polynomial reconstruction

Unconstrained polynomial reconstructions are computed for the cases where the numerical approximation requires only to minimise the cost functional $F(\boldsymbol{a}; \boldsymbol{m}, \mathcal{S}, \boldsymbol{\omega})$. In that case, the normal equations method [54] is employed in the present work to solve the linear least-squares problem and, for the sake of compactness, the reader is referred to Costa et al., 2019 [46], for a comprehensive and detailed description of the procedure. Unconstrained polynomial reconstructions are computed for each inner edge e_{ij} as the polynomial reference, assigning $\boldsymbol{m} := \boldsymbol{m}_{ij}$, $\mathcal{S} := \mathcal{S}_{ij}$, and $\boldsymbol{\omega} := \boldsymbol{\omega}_{ij}$. The optimal polynomial reconstruction is written as $\widetilde{\varphi}_{ij}(\boldsymbol{x}) := \varphi(\boldsymbol{x}; \widetilde{\boldsymbol{a}}_{ij}) = \widetilde{\boldsymbol{a}}_{ij} \cdot \boldsymbol{p}_d\left(\boldsymbol{x} - \boldsymbol{m}_{ij}\right)$, with $\widetilde{\boldsymbol{a}}_{ij} = \arg\min_{\boldsymbol{a}} F\left(\boldsymbol{a}; \boldsymbol{m}_{ij}, \mathcal{S}_{ij}, \boldsymbol{\omega}_{ij}\right)$.

4.2.2. Mean-value preservation polynomial reconstruction

In case the polynomial reference is a cell, preserving the conservation property is desired since it improves the robustness and stability of the numerical scheme while promoting the approximate solution to be physically meaningful. Thus, in addition to minimising the cost functional $F(\boldsymbol{a}; \boldsymbol{m}, \mathcal{S}, \boldsymbol{\omega})$, the linear least-squares procedure is subject to the linear constraint $G(\boldsymbol{a}) = 0$ that must be exactly fulfilled, and the procedure is referred to as constrained polynomial reconstruction. In that case, the so-called linearly constrained Lagrange multipliers method [55] is employed in the present work to solve the constrained linear least-squares problem, and the reader is referred to Costa et al., 2019 [46], for a comprehensive and detailed description of the procedure.

Constrained polynomial reconstructions are computed for each cell c_i as the polynomial reference, assigning $\boldsymbol{m} := \boldsymbol{m}_i$, $\mathcal{S} := \mathcal{S}_i$, and $\boldsymbol{\omega} := \boldsymbol{\omega}_i$. For the conservation of the cell mean-value, the linear constraint functional $G_i(\boldsymbol{a})$ from $\mathbb{R}^n$ to $\mathbb{R}$ is given as

$$G_i(\boldsymbol{a}) = \frac{1}{|c_i|} \int_{c_i} \varphi(\boldsymbol{x}; \boldsymbol{a}) \, \mathrm{d}\boldsymbol{x} - \phi_i. \tag{41}$$

The optimal polynomial reconstruction is written as $\widehat{\varphi}_i(\boldsymbol{x}) := \varphi(\boldsymbol{x}; \widehat{\boldsymbol{a}}_i) = \widehat{\boldsymbol{a}}_i \cdot \boldsymbol{p}_d\left(\boldsymbol{x} - \boldsymbol{m}_i\right)$, with $\widehat{\boldsymbol{a}}_i = \arg\min_{\boldsymbol{a}} F\left(\boldsymbol{a}; \boldsymbol{m}_i, \mathcal{S}_i, \boldsymbol{\omega}_i\right)$ subject to $G_i(\boldsymbol{a}) = 0$.

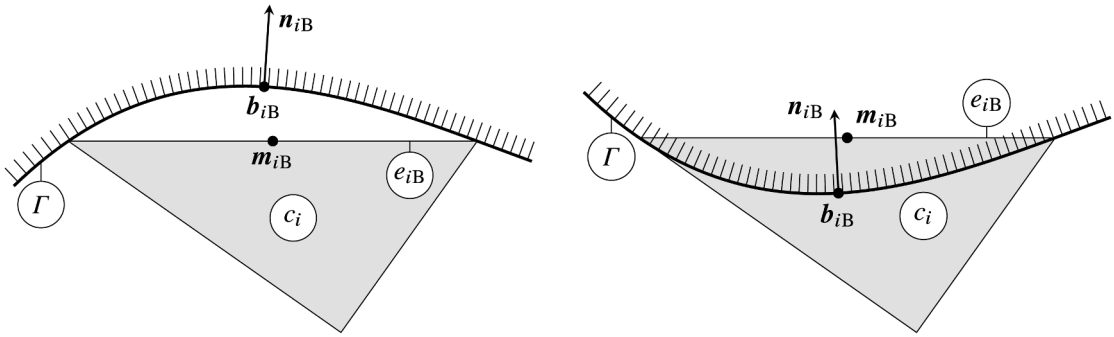


Fig. 4. Polygonal mesh with collocation point and associated outward unit normal vector on the curved physical boundary.

4.3. Reconstruction for off-site data method

Since arbitrary curved physical boundaries are discretised with linear piecewise edges, the associated boundary data is disconnected from the mesh structure. In this context, the ROD method has emerged as a sophisticated evolution of the polynomial reconstruction method, enabling the imposition of boundary data in the reconstruction procedure, while preserving high-order accuracy for arbitrary curved physical boundaries.

For each edge $e_{i\mathrm{B}}$ on the surrogate boundary, consider a collocation point $\boldsymbol{b}_{i\mathrm{B}} := (b_{i\mathrm{B},x}, b_{i\mathrm{B},y})$, which is selected to coincide with the physical boundary in the vicinity of the boundary edge, that is, $\boldsymbol{b}_{i\mathrm{B}} \in \Gamma$ (see Fig. 4). Since the physical boundary is described analytically, a simple projection of the edge midpoint onto the physical boundary can be employed to determine a suitable collocation point, and the associated outward unit normal vector $\boldsymbol{n}_{i\mathrm{B}} := (n_{i\mathrm{B},x}, n_{i\mathrm{B},y})$ is also determined and can be obtained from the normal vector function as $\boldsymbol{n}_{i\mathrm{B}} = \boldsymbol{n}(\boldsymbol{b}_{i\mathrm{B}})$. Then, the linear constraint functional $G_{i\mathrm{B}}(\boldsymbol{a})$ from $\mathbb{R}^n$ to $\mathbb{R}$ is defined at the collocation point as

$$G_{i\mathrm{B}}(\boldsymbol{a}) = \alpha\varphi(\boldsymbol{b}_{i\mathrm{B}};\boldsymbol{a}) + \beta\nabla\varphi(\boldsymbol{b}_{i\mathrm{B}};\boldsymbol{a})\cdot\boldsymbol{n}_{i\mathrm{B}} - g(\boldsymbol{b}_{i\mathrm{B}}), \tag{42}$$

where α and β are coefficients, and $g := g(\boldsymbol{x})$ is a scalar function that stands for the boundary condition value at the collocation point. This linear constraint represents a Robin boundary condition that imposes any prescribed scalar boundary condition by choosing the appropriate coefficients, α and β. For the Dirichlet boundary condition $\phi(\boldsymbol{x}) = g^{\mathrm{D}}(\boldsymbol{x})$, coefficients $\alpha = 1$ and $\beta = 0$ are chosen and the boundary condition value is $g(\boldsymbol{x}) = g^{\mathrm{D}}(\boldsymbol{b}_{i\mathrm{B}})$. For the Neumann boundary condition $\nabla\phi(\boldsymbol{x})\cdot\boldsymbol{n}(\boldsymbol{x}) = g^{\mathrm{N}}(\boldsymbol{x})$, coefficients $\alpha = 0$ and $\beta = 1$ are chosen and the boundary condition value is $g(\boldsymbol{x}) = g^{\mathrm{N}}(\boldsymbol{b}_{i\mathrm{B}})$. For the Robin boundary condition $\alpha^{\mathrm{R}}(\boldsymbol{x})\phi(\boldsymbol{x}) + \beta^{\mathrm{R}}(\boldsymbol{x})\nabla\phi(\boldsymbol{x})\cdot\boldsymbol{n}(\boldsymbol{x}) = g^{\mathrm{R}}(\boldsymbol{x})$, coefficients $\alpha = \alpha^{\mathrm{R}}(\boldsymbol{b}_{i\mathrm{B}})$ and $\beta = \beta^{\mathrm{R}}(\boldsymbol{b}_{i\mathrm{B}})$ are chosen and the boundary condition value is $g(\boldsymbol{x}) = \phi^{\mathrm{R}}(\boldsymbol{b}_{i\mathrm{B}})$. For the Cauchy boundary condition, two distinct conditions are imposed at each boundary point, and the scalar linear constraint $G_{i\mathrm{B}}(\boldsymbol{a})$ is substituted with the vectorial linear constraint $\boldsymbol{G}_{i\mathrm{B}}(\boldsymbol{a})$, given as

$$\boldsymbol{G}_{i\mathrm{B}}(\boldsymbol{a}) = \begin{bmatrix} \varphi(\boldsymbol{b}_{i\mathrm{B}};\boldsymbol{a}) \\ \nabla\varphi(\boldsymbol{b}_{i\mathrm{B}};\boldsymbol{a})\cdot\boldsymbol{n}_{i\mathrm{B}} \end{bmatrix} - \boldsymbol{g}(\boldsymbol{b}_{i\mathrm{B}}), \tag{43}$$

where $\boldsymbol{g} := \boldsymbol{g}(\boldsymbol{x})$ is a vector function that stands for the Dirichlet and Neumann boundary condition values at the collocation point.

Constrained polynomial reconstructions are computed for each boundary edge $e_{i\mathrm{B}}$ as the polynomial reference, assigning $\boldsymbol{m} := \boldsymbol{m}_{i\mathrm{B}}$, $\mathcal{S} := \mathcal{S}_{i\mathrm{B}}$, and $\boldsymbol{\omega} := \boldsymbol{\omega}_{i\mathrm{B}}$. The minimisation procedure in this case involves seeking the unique vector $\widehat{\boldsymbol{a}}_{i\mathrm{B}} \in \mathbb{R}^n$ that minimises the cost functional under the linear constraint $G_{i\mathrm{B}}(\boldsymbol{a}) = 0$ or $\boldsymbol{G}_{i\mathrm{B}}(\boldsymbol{a}) = \boldsymbol{0}$, that is, $\widehat{\boldsymbol{a}}_{i\mathrm{B}} = \arg\min_{\boldsymbol{a}} F(\boldsymbol{a};\boldsymbol{m}_{i\mathrm{B}},\mathcal{S}_{i\mathrm{B}},\boldsymbol{\omega}_{i\mathrm{B}})$ subject to $G_{i\mathrm{B}}(\boldsymbol{a}) = 0$ or $\boldsymbol{G}_{i\mathrm{B}}(\boldsymbol{a}) = \boldsymbol{0}$. The optimal polynomial reconstruction is written as $\widehat{\varphi}_{i\mathrm{B}}(\boldsymbol{x}) := \varphi(\boldsymbol{x};\widehat{\boldsymbol{a}}_{i\mathrm{B}}) = \widehat{\boldsymbol{a}}_{i\mathrm{B}}\cdot\boldsymbol{p}_d(\boldsymbol{x} - \boldsymbol{m}_{i\mathrm{B}})$. Similar to the conservation of the cell mean-value, the so-called linearly constrained Lagrange multipliers method [55] is employed to solve the constrained linear least-squares problem.

5. Numerical scheme

Polynomial reconstructions are employed for determining a highly accurate approximation to the boundary vorticity on the computational boundary Γ_Δ and for the discretisation of the vorticity transport and streamfunction Poisson equations in the computational domain Ω_Δ.

5.1. Generic finite volume scheme

Integrating the vorticity transport equation (12) over each cell c_i, $i \in \mathcal{I}$, and applying the divergence theorem yields

$$\int_{c_i}\left(\nabla\cdot(\boldsymbol{v}\omega) - \nu\nabla^2\omega\right)\mathrm{d}\boldsymbol{x} = \int_{c_i}\boldsymbol{f}\,\mathrm{d}\boldsymbol{x} \qquad \Rightarrow \qquad \int_{\partial c_i}(\boldsymbol{v}\omega - \nu\nabla\omega)\cdot\boldsymbol{s}_i\,\mathrm{d}\boldsymbol{x} = \int_{c_i}\boldsymbol{f}\,\mathrm{d}\boldsymbol{x}, \tag{44}$$

where ∂c_i stands for the cell boundary with $\boldsymbol{s}_i := \boldsymbol{s}_i(\boldsymbol{x}) := \big(s_{x,i}(\boldsymbol{x}), s_{y,i}(\boldsymbol{x})\big)$ the outward unit normal vector. An R-point Gauss-Legendre quadrature rule with weights ζ_r, $r = 1, \dots, R$, is used to approximate the integrals on the straight edges while a Q-point Gauss-Legendre quadrature rule with weights ξ_q, $q = 1, \dots, Q$, is used to approximate the integrals in the polygonal cells. Then, Eq. (44) is rewritten in the discrete residual form using the quadrature rules as

$$\sum_{j \in \mathcal{N}_i} \big|e_{ij}\big| \left[\sum_{r=1}^{R} \zeta_r \big(C_{ij,r} + D_{ij,r}\big) \right] - |c_i| f_i = \mathcal{O}\big(h_i^{\alpha}\big), \tag{45}$$

where $h_i = |c_i|^{1/2}$ is the characteristic cell size and α is the convergence order of the quadrature rules. Moreover, $C_{ij,r}$ and $D_{ij,r}$ represent the physical convective and diffusive fluxes of function $\omega(\boldsymbol{x})$, respectively, at the quadrature points of edge e_{ij}, and f_i denotes the approximate mean-value of the source term function $f(\boldsymbol{x})$ in cell c_i, given as

$$C_{ij,r} = \big(\boldsymbol{v}(\boldsymbol{q}_{ij,r}) \cdot \boldsymbol{s}_{ij}\big)\omega(\boldsymbol{q}_{ij,r}), \qquad D_{ij,r} = -\nu \nabla\omega(\boldsymbol{q}_{ij,r}) \cdot \boldsymbol{s}_{ij}, \qquad \text{and} \qquad f_i = \sum_{q=1}^{Q} \xi_q f\big(\boldsymbol{q}_{i,q}\big). \tag{46}$$

Similarly, integrating the streamfunction Poisson equation (13) over each cell c_i, $i \in \mathcal{I}$, and applying the divergence theorem yields

$$\int_{c_i} \big(\nabla^2\psi + \omega\big)\, \mathrm{d}\boldsymbol{x} = 0 \qquad \Rightarrow \qquad \int_{\partial c_i} \nabla\psi \cdot \boldsymbol{s}_i \, \mathrm{d}\boldsymbol{x} + \int_{c_i} \omega \, \mathrm{d}\boldsymbol{x} = 0, \tag{47}$$

which is rewritten in the discrete residual form using the quadrature rules as

$$\sum_{j \in \mathcal{N}_i} \big|e_{ij}\big| \left[\sum_{r=1}^{R} \zeta_r F_{ij,r} \right] - R_i = \mathcal{O}\big(h_i^{\alpha}\big), \tag{48}$$

where $F_{ij,r}$ represents the physical diffusive flux of function $\psi(\boldsymbol{x})$ at the quadrature points of edge e_{ij}, and R_i stands for the mean-value of function $\omega(\boldsymbol{x})$ in cell c_i, given as

$$F_{ij,r} = \nabla\psi\big(\boldsymbol{q}_{ij,r}\big) \cdot \boldsymbol{s}_{ij} \qquad \text{and} \qquad R_i = \int_{c_i} \omega \, \mathrm{d}\boldsymbol{x}. \tag{49}$$

5.2. Polynomial reconstructions

The discrete problem unknowns comprise cell mean-values ψ_i and ω_i, $i \in \mathcal{I}$, defined as

$$\psi_i \approx \frac{1}{|c_i|} \int_{c_i} \psi \, \mathrm{d}\boldsymbol{x} \qquad \text{and} \qquad \omega_i \approx \frac{1}{|c_i|} \int_{c_i} \omega \, \mathrm{d}\boldsymbol{x}, \tag{50}$$

for the piecewise approximations of functions $\psi(\boldsymbol{x})$ and $\omega(\boldsymbol{x})$, respectively. In addition, the unknown boundary point-values $\omega_{i\mathrm{B}}$, $i \in \mathcal{I}^{\mathrm{B}}$, represent an approximation of the boundary vorticity function $\omega^{\mathrm{B}}(\boldsymbol{x})$ at collocation points $\boldsymbol{b}_{i\mathrm{B}}$, that is $\omega_{i\mathrm{B}} \approx \omega^{\mathrm{B}}\big(\boldsymbol{b}_{i\mathrm{B}}\big)$.

To obtain highly accurate approximations of the physical fluxes for the finite volume scheme, polynomial reconstructions are computed based on the discrete problem cell mean-values and boundary point-values unknowns. Each polynomial reconstruction is specifically designed for its purpose (see the complete list in Table 2), as follows:

- For each inner edge e_{ij}, unconstrained polynomial reconstructions $\widetilde{\psi}_{ij}(\boldsymbol{x})$ for the streamfunction and $\widetilde{\omega}_{ij}(\boldsymbol{x})$ for the vorticity are computed using cell mean-values ψ_i and ω_i, respectively.
- For each cell c_i, constrained polynomial reconstructions $\widehat{\omega}_i(\boldsymbol{x})$ for the vorticity are computed using cell mean-values ω_i to preserve the conservation of the cell mean-value in the same cell.
- For each boundary edge $e_{i\mathrm{B}}$, constrained polynomial reconstructions $\widehat{\psi}_{i\mathrm{B}}(\boldsymbol{x})$ for the streamfunction are computed using cell mean-values ψ_i and associated boundary values and derivatives with functions $\psi^{\mathrm{B}}(\boldsymbol{x})$ and $\nabla\psi^{\mathrm{B}}(\boldsymbol{x}) \cdot \boldsymbol{n}(\boldsymbol{x})$ to impose the prescribed Cauchy boundary conditions (34) and (35). Following the ROD method, a collocation point $\boldsymbol{b}_{i\mathrm{B}}$ is chosen on the curved physical boundary with $\boldsymbol{n}_{i\mathrm{B}}$ as the associated outward unit normal vector. Then, the linear constraint (43) with the boundary data $\boldsymbol{g}\big(\boldsymbol{b}_{i\mathrm{B}}\big) = \big(\psi^{\mathrm{B}}\big(\boldsymbol{b}_{i\mathrm{B}}\big), \nabla\psi^{\mathrm{B}}\big(\boldsymbol{b}_{i\mathrm{B}}\big) \cdot \boldsymbol{n}_{i\mathrm{B}}\big)$ reads

$$\boldsymbol{G}_{i\mathrm{B}}(\boldsymbol{\eta}) = \begin{bmatrix} \varphi\big(\boldsymbol{b}_{i\mathrm{B}}\big) \\ \nabla\varphi\big(\boldsymbol{b}_{i\mathrm{B}}\big) \cdot \boldsymbol{n}_{i\mathrm{B}} \end{bmatrix} - \begin{bmatrix} \psi^{\mathrm{B}}\big(\boldsymbol{b}_{i\mathrm{B}}\big) \\ \nabla\psi^{\mathrm{B}}\big(\boldsymbol{b}_{i\mathrm{B}}\big) \cdot \boldsymbol{n}_{i\mathrm{B}} \end{bmatrix}. \tag{51}$$

- For each boundary edge $e_{i\mathrm{B}}$, polynomial reconstructions $\widehat{\omega}_{i\mathrm{B}}(\boldsymbol{x})$ for the vorticity are computed using cell mean-values ω_i and associated boundary values with point-values $\omega_{i\mathrm{B}}$ to impose the prescribed Dirichlet boundary condition (36). Following the ROD method, a collocation point $\boldsymbol{b}_{i\mathrm{B}}$ is chosen on the curved physical boundary. Then, the linear constraint (42) with coefficients $\alpha = 1$ and $\beta = 0$ and the boundary data $g\big(\boldsymbol{b}_{i\mathrm{B}}\big) = \omega_{i\mathrm{B}}$ reads

$$G_{i\mathrm{B}}(\boldsymbol{\eta}) = \varphi\big(\boldsymbol{b}_{i\mathrm{B}}\big) - \omega_{i\mathrm{B}}. \tag{52}$$

Remark 6. Note that the boundary data for the streamfunction polynomial reconstructions associated with the boundary edges is evaluated directly from the prescribed boundary velocity using Eqs. (20) and (23). In multiply connected domains, however, the evaluated streamfunction boundary value contains an unknown constant associated with each inner boundary subset, which is part of the discrete problem unknowns.

Table 2
Polynomial reconstructions based on the streamfunction and vorticity cell mean-values.

Underlying function	Polynomial reconstruction	Reference element	Condition type	Linear constraint
Streamfunction, $\phi(\boldsymbol{x})$	$\widetilde{\phi}_{ij}(\boldsymbol{x})$	Inner edge e_{ij}, $i,j \in \mathcal{I}$, $j \in \mathcal{N}_i$	None	—
	$\widetilde{\phi}_{i\mathrm{B}}(\boldsymbol{x})$	Boundary edge $e_{i\mathrm{B}}$, $i \in \mathcal{I}^{\mathrm{B}}$	None	—
Streamfunction, $\psi(\boldsymbol{x})$	$\widetilde{\psi}_{ij}(\boldsymbol{x})$	Inner edge e_{ij}, $i,j \in \mathcal{I}$, $j \in \mathcal{N}_i$	None	—
	$\widehat{\psi}_{i\mathrm{B}}(\boldsymbol{x})$	Boundary edge $e_{i\mathrm{B}}$, $i \in \mathcal{I}^{\mathrm{B}}$	Cauchy	$\boldsymbol{G}_{i\mathrm{B}}(\boldsymbol{\eta}) = \begin{bmatrix} \varphi(\boldsymbol{b}_{i\mathrm{B}}) \\ \nabla\varphi(\boldsymbol{b}_{i\mathrm{B}})\cdot\boldsymbol{n}_{i\mathrm{B}} \end{bmatrix} - \begin{bmatrix} \psi^{\mathrm{B}}(\boldsymbol{b}_{i\mathrm{B}}) \\ \nabla\psi^{\mathrm{B}}(\boldsymbol{b}_{i\mathrm{B}})\cdot\boldsymbol{n}_{i\mathrm{B}} \end{bmatrix}$
Vorticity, $\omega(\boldsymbol{x})$	$\widetilde{\omega}_{ij}(\boldsymbol{x})$	Inner edge e_{ij}, $i,j \in \mathcal{I}$, $j \in \mathcal{N}_i$	None	—
	$\widehat{\omega}_{i\mathrm{B}}(\boldsymbol{x})$	Boundary edge $e_{i\mathrm{B}}$, $i \in \mathcal{I}^{\mathrm{B}}$	Dirichlet	$G_{i\mathrm{B}}(\boldsymbol{\eta}) = \varphi(\boldsymbol{b}_{i\mathrm{B}}) - \omega_{i\mathrm{B}}$
	$\widehat{\omega}_i(\boldsymbol{x})$	Cell c_i, $i \in \mathcal{I}$	Mean-value	$G_i(\boldsymbol{\eta}) = \frac{1}{\lvert c_i\rvert}\int_{c_i} \varphi \,\mathrm{d}\boldsymbol{x} - \omega_i$

5.3. Physical fluxes approximation

For the vorticity transport equation, numerical approximations to the physical fluxes (46) are computed based on the previous polynomial reconstructions, as follows:

- For each inner edge e_{ij}, the physical convective and diffusive fluxes $C_{ij,r}$ and $D_{ij,r}$, respectively, are replaced with the numerical approximations $\mathcal{C}_{ij,r}$ and $\mathcal{D}_{ij,r}$, respectively, at quadrature points $\boldsymbol{q}_{ij,r}$, $r = 1,\dots,R$, given as

$$\mathcal{C}_{ij,r} = \left[\boldsymbol{v}_{ij,r}\cdot\boldsymbol{s}_{ij}\right]^{+}\widehat{\omega}_i\left(\boldsymbol{q}_{ij,r}\right) + \left[\boldsymbol{v}_{ij,r}\cdot\boldsymbol{s}_{ij}\right]^{-}\widehat{\omega}_j\left(\boldsymbol{q}_{ij,r}\right), \tag{53}$$

$$\mathcal{D}_{ij,r} = -\nu\nabla\widetilde{\omega}_{ij}\left(\boldsymbol{q}_{ij,r}\right)\cdot\boldsymbol{s}_{ij}, \tag{54}$$

  with the notation $[a]^{+} = \max(a,0)$, $[a]^{-} = \min(a,0)$, and $\boldsymbol{v}_{ij,r} := \left(v_{x,ij,r}, v_{y,ij,r}\right)$ representing an approximation of function $\boldsymbol{v}(\boldsymbol{x})$ at the edge quadrature points $\boldsymbol{q}_{ij,r}$, that is $\boldsymbol{v}_{ij,r} \approx \boldsymbol{v}\left(\boldsymbol{q}_{ii,r}\right)$.
- For each boundary edge $e_{i\mathrm{B}}$, the physical convective and diffusive fluxes $C_{i\mathrm{B},r}$ and $D_{i\mathrm{B},r}$, respectively, are replaced with the numerical approximations $\mathcal{C}_{i\mathrm{B},r}$ and $\mathcal{D}_{i\mathrm{B},r}$, respectively, at quadrature points $\boldsymbol{q}_{i\mathrm{B},r}$, $r = 1,\dots,R$, given as

$$\mathcal{C}_{i\mathrm{B},r} = \left[\boldsymbol{v}_{i\mathrm{B},r}\cdot\boldsymbol{s}_{i\mathrm{B}}\right]^{+}\widehat{\omega}_i\left(\boldsymbol{q}_{i\mathrm{B},r}\right) + \left[\boldsymbol{v}_{i\mathrm{B},r}\cdot\boldsymbol{s}_{i\mathrm{B}}\right]^{-}\widehat{\omega}_{i\mathrm{B}}\left(\boldsymbol{q}_{i\mathrm{B},r}\right), \tag{55}$$

$$\mathcal{D}_{i\mathrm{B},r} = -\nu\nabla\widehat{\omega}_{i\mathrm{B}}\left(\boldsymbol{q}_{i\mathrm{B},r}\right)\cdot\boldsymbol{s}_{i\mathrm{B}}, \tag{56}$$

  with $\boldsymbol{v}_{i\mathrm{B},r} := \left(v_{x,i\mathrm{B},r}, v_{y,i\mathrm{B},r}\right)$ representing an approximation of function $\boldsymbol{v}(\boldsymbol{x})$ at the boundary edge quadrature points $\boldsymbol{q}_{i\mathrm{B},r}$, that is $\boldsymbol{v}_{i\mathrm{B},r} \approx \boldsymbol{v}\left(\boldsymbol{q}_{i\mathrm{B},r}\right)$.

For the convective fluxes, an upwind strategy is employed to approximate the vorticity values based on the constrained polynomial reconstructions $\widehat{\omega}_i(\boldsymbol{x})$ and $\widehat{\omega}_j(\boldsymbol{x})$ that preserve the vorticity cell mean-values on each side of the inner edge e_{ij}. For the boundary edge $e_{i\mathrm{B}}$, the polynomial reconstruction $\widehat{\omega}_{i\mathrm{B}}(\boldsymbol{x})$, fulfilling the prescribed boundary condition, together with the polynomial reconstruction $\widehat{\omega}_i(\boldsymbol{x})$, preserving the vorticity cell mean-value preserving, are used. For the diffusive fluxes, either the unconstrained polynomial reconstruction $\widetilde{\omega}_{ij}(\boldsymbol{x})$, on the inner edge e_{ij}, or the constrained polynomial reconstruction $\widehat{\omega}_{i\mathrm{B}}(\boldsymbol{x})$, on the boundary edge $e_{i\mathrm{B}}$, are employed to evaluate an approximation of the vorticity gradient.

For the streamfunction equation, approximations to the physical fluxes (49) are computed based on the previous polynomial reconstructions, as follows:

- For each inner edge e_{ij}, the physical diffusive fluxes $F_{ij,r}$ are replaced with the numerical approximations $\mathcal{F}_{ij,r}$ at quadrature points $\boldsymbol{q}_{ij,r}$, $r = 1,\dots,R$, given as

$$\mathcal{F}_{ij,r} = \nabla\widetilde{\psi}_{ij}\left(\boldsymbol{q}_{ij,r}\right)\cdot\boldsymbol{s}_{ij}. \tag{57}$$

- For each boundary edge $e_{i\mathrm{B}}$, the physical diffusive fluxes $F_{i\mathrm{B},r}$ are replaced with the numerical approximations $\mathcal{F}_{i\mathrm{B},r}$ at quadrature points $\boldsymbol{q}_{i\mathrm{B},r}$, $r = 1,\dots,R$, given as

$$\mathcal{F}_{i\mathrm{B},r} = \nabla\widehat{\psi}_{i\mathrm{B}}\left(\boldsymbol{q}_{i\mathrm{B},r}\right)\cdot\boldsymbol{s}_{i\mathrm{B}}. \tag{58}$$

Similar to the diffusive fluxes in the vorticity transport equation, either the unconstrained polynomial reconstruction $\widetilde{\omega}_{ij}(\boldsymbol{x})$, on the inner edge e_{ij}, or the constrained polynomial reconstruction $\widehat{\omega}_{i\mathrm{B}}(\boldsymbol{x})$, on the boundary edge $e_{i\mathrm{B}}$, are employed to evaluate an approximation of the streamfunction gradient.

Providing an approximation to R_i in Eq. (49) is straightforward since it corresponds to the vorticity cell mean-value in cell c_i. Thus, for each cell c_i, R_i is approximated with $\mathcal{R}_i$, given as

$$\mathcal{R}_i = \omega_i. \tag{59}$$

Remark 7. Note that the prescribed boundary conditions for the vorticity and streamfunction are imposed through constrained polynomial reconstructions associated with the boundary edges. Therefore, there is no explicit reference to either the boundary condition or the physical boundary in the numerical scheme, which only addresses two situations: inner or boundary edges. Moreover, the numerical fluxes are determined solely at quadrature points belonging to the straight edges, without knowledge of the curved physical boundary at this stage.

5.4. Boundary vorticity approximation

The boundary vorticity function $\omega^{\mathrm{B}}(\boldsymbol{x})$ is unknown since, in general, the prescribed boundary velocity function alone is insufficient to compute all terms in Eq. (33). In that regard, polynomial reconstructions are used to approximate the unknown point-value $\omega_{i\mathrm{B}}$, evaluating the second derivative of the streamfunction numerically, while the remaining terms can be determined analytically. More specifically, for each boundary edge $e_{i\mathrm{B}}$, the constrained polynomial reconstruction $\widehat{\psi}_{i\mathrm{B}}(\boldsymbol{x})$ is employed approximate $\omega^{\mathrm{B}}(\boldsymbol{x})$ at the collocation point $\boldsymbol{b}_{i\mathrm{B}}$, given as

$$\omega_{i\mathrm{B}} = -\frac{\partial^2 \widehat{\psi}_{i\mathrm{B}}}{\partial \eta^2}\left(\boldsymbol{b}_{i\mathrm{B}}\right) + \kappa_t\left(\boldsymbol{b}_{i\mathrm{B}}\right) u_{\xi}^{\mathrm{B}}\left(\boldsymbol{b}_{i\mathrm{B}}\right) - \frac{\partial \boldsymbol{u}^{\mathrm{B}} \cdot \boldsymbol{n}}{\partial \boldsymbol{t}}\left(\boldsymbol{b}_{i\mathrm{B}}\right). \tag{60}$$

5.5. Velocity approximation

Polynomial reconstructions are employed to compute the approximate velocity point-values $\boldsymbol{v}_{ij,r}$ and $\boldsymbol{v}_{i\mathrm{B},r}$ in the approximate convective fluxes (53) and (55). Sincical $\boldsymbol{v}(\boldsymbol{x}) = \nabla^{\perp}\phi(\boldsymbol{x})$, where function $\phi(\boldsymbol{x})$ corresponds to the streamfunction from the previous fixed-point iteration, the unconstrained polynomial reconstructions $\widetilde{\phi}_{ij}(\boldsymbol{x})$ and $\widetilde{\phi}_{i\mathrm{B}}(\boldsymbol{x})$ for the inner edges e_{ij} and boundary edges $e_{i\mathrm{B}}$, respectively, are computed based on the cell mean-values ϕ_i, which correspond to the cell mean-values ψ_i from the previous fixed-point iteration. Then, the approximate velocity point-values are determined as follows:

- For each inner edge e_{ij}, velocity point-values $\boldsymbol{v}_{ij,r}$ at quadrature points $\boldsymbol{q}_{ij,r}$, $r = 1, \dots, R$, are given as $\boldsymbol{v}_{ij,r} = \nabla^{\perp}\widetilde{\phi}_{ij}\left(\boldsymbol{q}_{ij,r}\right)$, that is

$$v_{x,ij,r} = \frac{\partial \widetilde{\phi}_{ij}}{\partial y}\left(\boldsymbol{q}_{ij,r}\right) \qquad \text{and} \qquad v_{y,ij,r} = -\frac{\partial \widetilde{\phi}_{ij}}{\partial x}\left(\boldsymbol{q}_{ij,r}\right). \tag{61}$$

- For each boundary edge $e_{i\mathrm{B}}$, velocity point-values $\boldsymbol{v}_{i\mathrm{B},r}$ at quadrature points $\boldsymbol{q}_{i\mathrm{B},r}$, $r = 1, \dots, R$, are given as $\boldsymbol{v}_{i\mathrm{B},r} = \nabla^{\perp}\widetilde{\phi}_{i\mathrm{B}}\left(\boldsymbol{q}_{i\mathrm{B},r}\right)$, that is

$$v_{x,i\mathrm{B},r} = \frac{\partial \widetilde{\phi}_{i\mathrm{B}}}{\partial y}\left(\boldsymbol{q}_{i\mathrm{B},r}\right) \qquad \text{and} \qquad v_{y,i\mathrm{B},r} = -\frac{\partial \widetilde{\phi}_{i\mathrm{B}}}{\partial x}\left(\boldsymbol{q}_{i\mathrm{B},r}\right). \tag{62}$$

5.6. Compatibility conditions

Unknown constants C_k associated with physical boundary subsets Γ^k, $k = 1, \dots, K$, are determined as part of the overall solution. The compatibility conditions (22) for the boundary integral of the vorticity flux must be imposed to close the system, which can be cumbersome for arbitrary curved physical boundaries. Fortunately, these compatibility conditions can be transferred from the physical boundary Γ^k to the computational boundary Γ_{Δ}^k, thus avoiding integration on curved boundaries, provided that the boundary data is taken into account, yielding

$$\sum_{e_{i\mathrm{B}} \in \Gamma_{\Delta}^k} \left[\int_{e_{i\mathrm{B}}} (\boldsymbol{u}\,\omega - \nu \nabla \omega) \cdot \boldsymbol{s}_{i\mathrm{B}} \,\mathrm{d}s \right] = \sum_{e_{i\mathrm{B}} \in \Gamma_{\Delta}^k} \left[\int_{e_{i\mathrm{B}}} \boldsymbol{f} \cdot \boldsymbol{r}_{i\mathrm{B}} \,\mathrm{d}s \right]. \tag{63}$$

Then, the integrals on the boundary edges $e_{i\mathrm{B}}$ are approximated with constrained polynomial reconstructions $\widehat{\omega}_{i\mathrm{B}}(\boldsymbol{x})$ computed from the vorticity cell mean-values ω_i and the boundary vorticity point-values $\omega_{i\mathrm{B}}$. These are evaluated at an R-point Gauss-Legendre quadrature rule with weights ζ_r, $r = 1, \dots, R$, yielding

$$\sum_{e_{i\mathrm{B}} \in \Gamma_{\Delta}^k} |e_{i\mathrm{B}}| \left[\sum_{r=1}^{R} \zeta_r \left(\boldsymbol{v}_{i\mathrm{B},r}\, \widehat{\omega}_{i\mathrm{B}}\left(\boldsymbol{q}_{i\mathrm{B},r}\right) - \nu \nabla \widehat{\omega}_{i\mathrm{B}}\left(\boldsymbol{q}_{i\mathrm{B},r}\right) \right) \cdot \boldsymbol{s}_{i\mathrm{B}} \right] = \sum_{e_{i\mathrm{B}} \in \Gamma_{\Delta}^k} |e_{i\mathrm{B}}| \left[\sum_{r=1}^{R} \zeta_r \boldsymbol{f}\left(\boldsymbol{q}_{i\mathrm{B},r}\right) \cdot \boldsymbol{r}_{i\mathrm{B}} \right]. \tag{64}$$

Remark 8. Note that the computation of constrained polynomial reconstructions $\widehat{\omega}_{i\mathrm{B}}(\boldsymbol{x})$ takes into account the boundary vorticity point-values $\omega_{i\mathrm{B}}$, which are determined from Eq. (60). In turn, the constrained polynomial reconstructions $\widehat{\psi}_{i\mathrm{B}}$ are computed based on the associated boundary streamfunction values $\psi^{\mathrm{B}}\left(\boldsymbol{b}_{i\mathrm{B}}\right)$ and derivatives $\nabla\psi^{\mathrm{B}}\left(\boldsymbol{b}_{i\mathrm{B}}\right) \cdot \boldsymbol{n}_{i\mathrm{B}}$. Since the boundary streamfunction value $\psi^{\mathrm{B}}\left(\boldsymbol{b}_{i\mathrm{B}}\right)$ depends on the constant C_k on the associated physical boundary subset Γ^k, then the compatibility conditions (64) effectively close the system. This demonstrates the rationale for using these specific compatibility conditions to fix the constants C_k, provided that the boundary streamfunction $\psi^{\mathrm{B}}(\boldsymbol{x})$ is taken into account.

6. Verification benchmark

This section outlines the verification process for the numerical framework implemented using the highly accurate wall vorticity treatment and finite volume discretisation methods presented in previous sections. The verification process is based on simulations of specific benchmark test cases with known or manufactured analytical solutions, following the methodology described in the next section.

6.1. Verification methodology

The evaluation of the computed approximate solution is based on accuracy, convergence order, stability, and robustness. Each benchmark test case is solved numerically using progressively finer polygonal meshes for the associated physical domain, allowing the convergence order of the method to be determined under mesh refinement.

6.1.1. Errors and convergence orders

From the given analytical functions $\omega(\boldsymbol{x})$ and $\psi(\boldsymbol{x})$, the exact mean-values of the vorticity and streamfunction in the cells are evaluated as

$$\overline{\omega}_i = \frac{1}{|c_i|}\int_{c_i} \omega(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x} \qquad \text{and} \qquad \overline{\psi}_i = \frac{1}{|c_i|}\int_{c_i} \psi(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}, \qquad i \in \mathcal{I}, \tag{65}$$

respectively. Similarly, from the given analytical function $\boldsymbol{u}(\boldsymbol{x})$, the exact mean-values of the velocity on the edges are evaluated as

$$\overline{\boldsymbol{u}}_{ij} = \frac{1}{|e_{ij}|}\int_{e_{ij}} \boldsymbol{u}(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}, \qquad i \in \mathcal{I}, \qquad j \in \mathcal{N}_i, \tag{66}$$

where vector $\overline{\boldsymbol{u}}_{ij} := \left(\overline{u}_{x,ij}, \overline{u}_{y,ij}\right)$ stands for the exact mean-value of the velocity components in the x- and y-directions. Finally, from the given analytical function $\omega^{\mathrm{B}}(\boldsymbol{x})$, the exact point-values of boundary vorticity at the collocation points are evaluated as

$$\overline{\omega}_{i\mathrm{B}} = \omega\left(\boldsymbol{b}_{i\mathrm{B}}\right), \qquad i \in \mathcal{I}^{\mathrm{B}}. \tag{67}$$

The corresponding approximate variables are ω_i^*, ψ_i^*, $\boldsymbol{u}_{ij}^*$, and $\omega_{i\mathrm{B}}^{\mathrm{B},*}$ respectively, which are obtained from successively solving the system of linear equations until the fixed point convergence reaches a given tolerance.

With the exact and approximate values in hand, the associated errors E_1 and E_∞ in the L^1- and L^∞-norms, respectively, are determined as

$$E_1 = \frac{\displaystyle\sum_{i\in\mathcal{I}} \left|\xi_i^* - \overline{\xi}_i\right| |c_i|}{\displaystyle\sum_{i\in\mathcal{I}} |c_i|} \qquad \text{and} \qquad E_\infty = \max_{i\in\mathcal{I}} \left|\xi_i^* - \overline{\xi}_i\right|, \tag{68}$$

where ξ is taken as either the vorticity or streamfunction. For the velocity and boundary vorticity, the associated errors are computed similarly, adapting the above expressions for the edges and collocation points accordingly.

To determine the convergence orders, two meshes $\mathcal{M}_1$ and $\mathcal{M}_2$ are considered for the same physical domain having different characteristic sizes, with $N_{\mathrm{C},1}$ and $N_{\mathrm{C},2}$ cells, $N_{\mathrm{E},1}$ and $N_{\mathrm{E},2}$ edges, and $N_{\mathrm{B},1}$ and $N_{\mathrm{B},2}$ boundary edges, respectively. With the associated relative errors $E_{1,1}$ and $E_{1,2}$ in the L^1-norm and $E_{\infty,1}$ and $E_{\infty,2}$ in the L^∞-norm, respectively, the convergence orders O_1 and O_∞ for the relative errors in the L^1- and L^∞-norms, respectively, are determined as

$$O_1 = 2\frac{\left|\ln\left(\dfrac{E_{1,1}}{E_{1,2}}\right)\right|}{\left|\ln\left(\dfrac{N_{\mathrm{C},1}}{N_{\mathrm{C},2}}\right)\right|} \qquad \text{and} \qquad O_\infty = 2\frac{\left|\ln\left(\dfrac{E_{\infty,1}}{E_{\infty,2}}\right)\right|}{\left|\ln\left(\dfrac{N_{\mathrm{C},1}}{N_{\mathrm{C},2}}\right)\right|}, \tag{69}$$

for the vorticity and streamfunction. For the velocity and boundary vorticity, the associated convergence orders are computed similarly, adapting the above expressions for the edges and collocation points accordingly.

For compactness, the L^1-norm relative errors for the velocity presented thereafter correspond to the sum of the individual components errors, while the L^∞-norm relative errors correspond to the maximum of the individual components errors. The associated convergence orders are then determined accordingly.

6.1.2. Methods and boundary treatment

Seeking to investigate and improve the numerical accuracy and convergence order of the proposed techniques, various methods are explored regarding the boundary treatment. More specifically, for the location of the collocation points in Eqs. (42) and (43) and the associated boundary condition values, the following methods are considered:

- Naive method – collocation points correspond to the edge midpoints (computational boundary). For constant boundary conditions, the boundary condition values to be fulfilled correspond to the prescribed constant; otherwise, some extrapolation of the boundary condition function from the physical boundary to the computational boundary is necessary. For simplicity, for each edge, the projection of the edge midpoint onto the physical boundary is computed, and the associated value is obtained from the prescribed boundary condition function to be considered at the edge midpoint.
- Exact method – collocation points correspond to the edge midpoints (computational boundary). In contrast to the naive method, the exact or manufactured solution is used to obtain the boundary condition values to be fulfilled; hence, the computational boundary acts as a physical boundary with the appropriate values to be satisfied. In practice, such an approach cannot be applied since the analytical solution is not known, but it allows for a comparison of the ROD method with that of an ideal situation, that is, without a geometrical gap between the physical and computational boundaries.

- ROD method – collocation points are derived from the analytical boundary description (physical boundary), where either one or two collocation points for each polynomial reconstruction associated with the boundary edges are used. For one collocation point, a projection of the edge midpoint onto the physical boundary is computed, whereas for two collocation points, two points with barycentric coordinates $(1/3, 2/3)$ and $(2/3, 1/3)$ relative to the vertices of the edge are projected onto the physical boundary. The boundary condition values to be fulfilled are straightforwardly obtained from the prescribed boundary condition function.

Regarding the polynomial reconstructions for the vorticity and streamfunction, the following methods are considered:

- $\mathbb{P}_d$–$\mathbb{P}_d$ method – d-degree polynomials are employed for both the vorticity and streamfunction polynomial reconstructions.
- $\mathbb{P}_d$–$\mathbb{P}_{d+1}$ method – d-degree polynomials are employed for the vorticity reconstructions and $(d+1)$-degree polynomials are employed for the streamfunction polynomial reconstructions.
- $\mathbb{P}_d$–$\mathbb{P}_{d+2}$ method – d-degree polynomials are employed for the vorticity reconstructions and $(d+2)$-degree polynomials are employed for the streamfunction polynomial reconstructions.

Remark 9. Following previous works, for the ROD method to achieve optimal accuracy and convergence order, $(d+1)$-degree polynomials are employed for the polynomial reconstructions associated with the boundary edges imposing a Cauchy boundary condition, whenever d-degree polynomials are employed for those associated with the cells and inner edges.

6.2. Circular domain

This test case addresses an internal creeping flow (Stokes equations) in a circular domain with radius $r_{\mathrm{E}} > 0$ centred at the origin. The manufactured solutions for the vorticity and streamfunction in polar coordinates, $\omega := \omega(r,\theta)$ and $\psi := \psi(r,\theta)$, respectively, are given as

$$\omega = \frac{2u_{\mathrm{E}}}{r_{\mathrm{E}}}\left(r^2+1\right)\exp\left(r^2-r_{\mathrm{E}}^2\right) \qquad \text{and} \qquad \psi = \frac{u_{\mathrm{E}}}{2r_{\mathrm{E}}\exp\left(r_{\mathrm{E}}^2\right)}\left(\exp\left(r_{\mathrm{E}}^2\right)-\exp\left(r^2\right)\right), \tag{70}$$

where $u_{\mathrm{E}} \in \mathbb{R}$ is the linear velocity magnitude on the physical boundary. The velocity components in polar coordinates, $u_r := u_r(r,\theta)$ and $u_\theta := u_\theta(r,\theta)$, respectively, are obtained from the streamfunction as

$$u_r = 0 \qquad \text{and} \qquad u_\theta = \frac{u_{\mathrm{E}}}{r_{\mathrm{E}}} r \exp\left(r^2-r_{\mathrm{E}}^2\right). \tag{71}$$

The source term function in polar coordinates, $f := f(r,\theta)$, derives from the vorticity transport equation with the above-manufactured solution and is given as

$$f = -\frac{4\nu u_{\mathrm{E}}}{r_{\mathrm{E}}} r\left(r^2+2\right)\exp\left(r^2-r_{\mathrm{E}}^2\right). \tag{72}$$

Note that although the exact solution is given in polar coordinates, the problem is solved in Cartesian coordinates.

For the domain, the physical boundary has a radius of $r_{\mathrm{E}} = 1$ m (normal curvature of $\kappa_{\mathrm{E}} = 1$). For the fluid, a dynamic viscosity of $\mu = 1\ \mathrm{kg/(m\,s)}$, a density of $\rho = 1\ \mathrm{kg/m^3}$, and a velocity magnitude of $u_{\mathrm{E}} = 1\ \mathrm{m/s}$ are considered. Successively finer uniform Delaunay triangular meshes are used to discretise the physical domain, and the simulations are carried out for the $\mathbb{P}_d$–$\mathbb{P}_d$ method with $d = 1, 3, 5$, while the boundary is treated with the naive, exact or ROD methods. Fig. 5 illustrates the physical domain with a coarse mesh and the analytical solutions for the velocity, vorticity, and streamfunction fields.

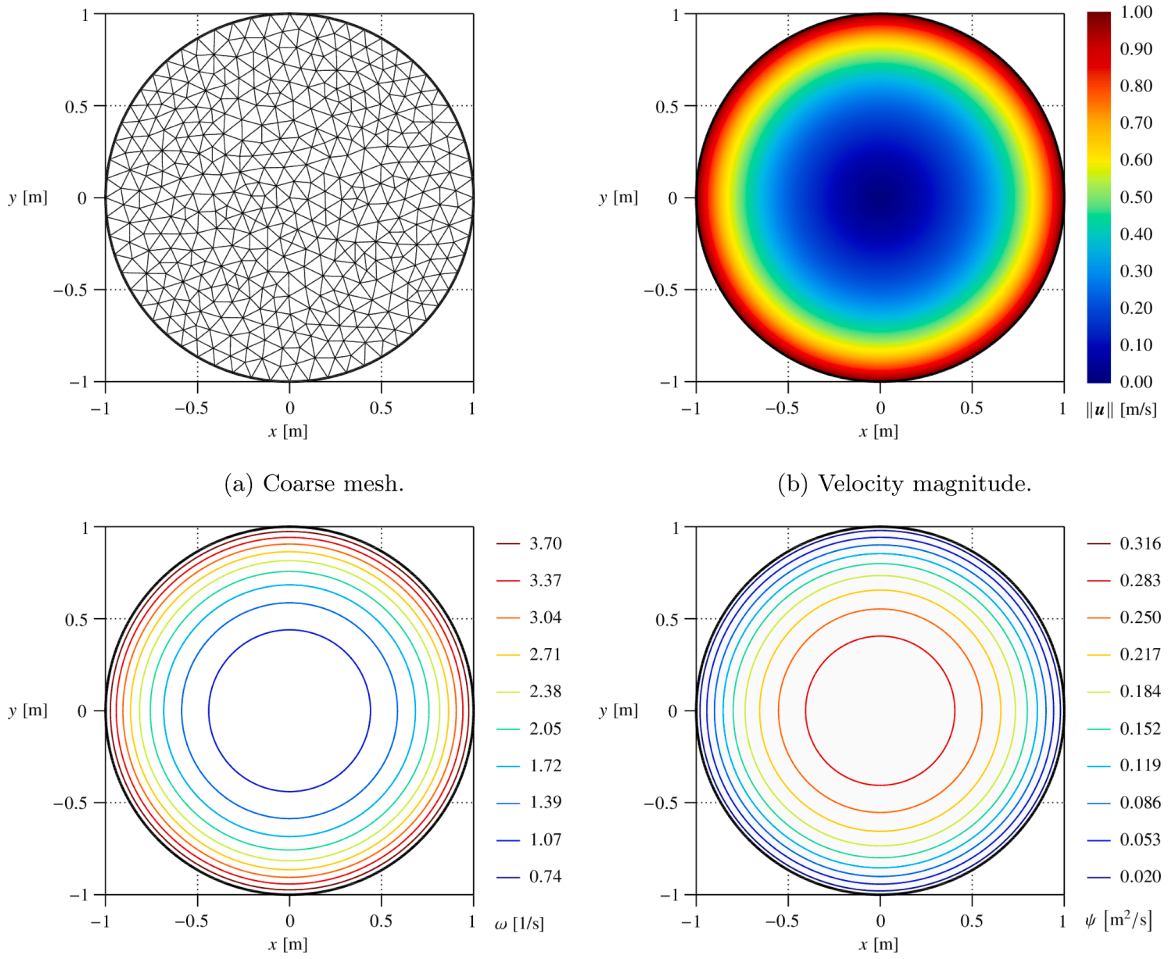
6.2.1. Boundary treatment assessment

To assess the different boundary treatment methods without the influence of the boundary vorticity approximation, the exact values derived from the corresponding analytical solution are imposed instead at this stage. In all the simulations, the $\mathbb{P}_d$–$\mathbb{P}_d$ method is employed, meaning that the same polynomial degree is considered for both the vorticity and streamfunction reconstructions.

The errors and convergence orders for the approximate vorticity and streamfunction are reported in Tables 3 and 4, respectively. In both cases, the naive method provides a second-order convergence limitation regardless of the employed polynomial degree for both norms, since the prescribed boundary conditions are fulfilled on the computational boundary rather than on the physical boundary. Conversely, the ROD method effectively achieves optimal convergence, obtaining the second-, fourth-, and sixth-orders convergence with $d = 1, 3, 5$, respectively, in both norms. Moreover, the ROD method provides comparable accuracy and convergence orders to the exact method, supporting its effectiveness relative to the ideal scenario.

The errors and convergence orders for the approximate velocity are reported in Table 5. Apart from the naive method, which fails to achieve the optimal convergence orders with $d = 3, 5$, the first-, third-, and fifth-orders convergence are obtained with $d = 1, 3, 5$, respectively, in the L^∞-norm, while slightly higher convergence orders are obtained in the L^1-norm. In this case, there is a reduction in the convergence orders compared to those obtained for the streamfunction, which is expected since the velocity is obtained from the streamfunction's first derivatives.

From the approximate streamfunction cell mean-values and associated boundary data, the approximate boundary vorticity is computed *a posteriori* using Eq. (60) (while the exact boundary vorticity was used for the previous simulations) and the associated errors and convergence orders are reported in Table 6. As observed, the ROD method achieves the second- and fourth-orders convergence with $d = 3, 5$, respectively, for both norms, whereas it fails to converge with $d = 1$ since the approximate boundary vorticity is obtained from the streamfunction second derivatives and, therefore, a reduction of two orders is expected compared to those obtained for the streamfunction.



(a) Coarse mesh.

(b) Velocity magnitude.

(c) Vorticity contours.

(d) Streamlines.

Fig. 5. Coarse mesh and analytical solution for the flow in a circular domain test case (for proper interpretation of the colour scale, the reader is referred to the electronic version of this article).

6.2.2. *Polynomial reconstructions assessment*

To assess the different combinations of polynomial degrees for the vorticity and streamfunction reconstructions, the exact boundary vorticity is substituted with the prescribed approximation (60) and the ROD method is employed for the boundary treatment in all simulations. The errors and convergence orders for the approximate vorticity and streamfunction are reported in Tables 7 and 8, respectively, while the results for the approximate boundary vorticity and velocity are reported in Tables 9 and 10, respectively. The discussion follows:

- The $\mathbb{P}_d$–$\mathbb{P}_d$ method does not converge with $d = 1$, either for the vorticity and streamfunction, whereas the obtained convergence orders are suboptimal with $d = 3, 5$. The convergence deterioration is more pronounced for the vorticity, where it achieves only the third- and fifth-orders convergence with $d = 3, 5$ in the L^1-norm, and the second- and fourth-orders convergence in the L^∞-norm, respectively. Such a deterioration in convergence order is expected since the boundary vorticity is approximated with second- and fourth-orders convergence with $d = 3, 5$, respectively, in both norms, while with $d = 1$ it does not converge at all. Consequently, there is substantial accuracy deterioration, especially in the vicinity of the boundary, which is reflected in the vorticity and, ultimately, also in the streamfunction. For the velocity, the third- and fifth-orders convergence are achieved with $d = 3, 5$, respectively, in both norms, while with $d = 1$ it does not converge, following the expected behaviour considering the obtained convergence orders for the streamfunction.
- Increasing the polynomial degree for the streamfunction reconstructions in the $\mathbb{P}_d$–$\mathbb{P}_{d+1}$ method improves the accuracy and increases the convergence orders for the boundary vorticity, with first-, third-, and fifth-orders convergence obtained with $d = 1, 3, 5$, respectively, in both norms. Consequently, accuracy and convergence orders for the vorticity are substantially improved, achieving second-, fourth-, and sixth-orders convergence in the L^1-norm and above first-, third-, and fifth-orders convergence in the L^∞-norm with $d = 1, 3, 5$, respectively. For the streamfunction, optimal convergence orders are achieved in both norms,

Table 3
Vorticity errors and convergence orders obtained in the flow in a circular domain test case with exact boundary vorticity.

	$d=1$				$d=3$				$d=5$			
N_C	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
Naive												
1,096	6.25E−03	—	2.12E−02	—	1.14E−02	—	1.16E−02	—	1.14E−02	—	1.14E−02	—
2,478	3.04E−03	1.76	1.33E−02	1.15	5.07E−03	1.99	5.12E−03	2.01	5.07E−03	1.98	5.07E−03	1.98
5,366	1.61E−03	1.64	7.04E−03	1.65	2.31E−03	2.03	2.33E−03	2.04	2.31E−03	2.03	2.31E−03	2.03
11,624	6.95E−04	2.18	3.53E−03	1.78	1.06E−03	2.01	1.07E−03	2.02	1.06E−03	2.01	1.06E−03	2.01
26,890	3.23E−04	1.83	1.49E−03	2.07	4.67E−04	1.96	4.68E−04	1.96	4.67E−04	1.96	4.67E−04	1.96
Exact												
1,096	1.76E−02	—	3.26E−02	—	7.87E−05	—	3.93E−04	—	1.52E−05	—	2.54E−05	—
2,478	8.11E−03	1.91	1.84E−02	1.41	1.54E−05	4.00	1.04E−04	3.25	1.89E−06	5.10	3.15E−06	5.12
5,366	3.92E−03	1.88	9.35E−03	1.75	3.20E−06	4.07	2.34E−05	3.88	2.19E−07	5.58	3.78E−07	5.48
11,624	1.76E−03	2.08	4.60E−03	1.84	6.47E−07	4.13	6.33E−06	3.38	2.52E−08	5.60	3.89E−08	5.89
26,890	7.90E−04	1.91	1.95E−03	2.04	1.18E−07	4.06	1.21E−06	3.93	2.04E−09	5.99	3.43E−09	5.79
ROD												
1,096	1.75E−02	—	3.25E−02	—	7.85E−05	—	3.93E−04	—	1.51E−05	—	2.53E−05	—
2,478	8.08E−03	1.90	1.83E−02	1.40	1.54E−05	3.99	1.04E−04	3.25	1.89E−06	5.10	3.14E−06	5.12
5,366	3.91E−03	1.88	9.33E−03	1.75	3.19E−06	4.07	2.33E−05	3.88	2.19E−07	5.58	3.78E−07	5.48
11,624	1.75E−03	2.08	4.59E−03	1.83	6.47E−07	4.13	6.32E−06	3.38	2.52E−08	5.60	3.88E−08	5.89
26,890	7.89E−04	1.91	1.95E−03	2.04	1.18E−07	4.06	1.21E−06	3.93	2.05E−09	5.99	3.43E−09	5.79

Table 4
Streamfunction errors and convergence orders obtained in the flow in a circular domain test case with exact boundary vorticity.

	$d=1$				$d=3$				$d=5$			
N_C	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
Naive												
1,096	3.29E−04	—	1.32E−03	—	7.87E−04	—	1.89E−03	—	7.86E−04	—	1.88E−03	—
2,478	1.60E−04	1.77	8.63E−04	1.05	3.51E−04	1.98	8.43E−04	1.98	3.51E−04	1.98	8.43E−04	1.97
5,366	8.59E−05	1.61	4.57E−04	1.64	1.60E−04	2.03	3.85E−04	2.03	1.60E−04	2.03	3.85E−04	2.03
11,624	3.68E−05	2.19	2.31E−04	1.77	7.38E−05	2.01	1.77E−04	2.01	7.38E−05	2.01	1.77E−04	2.01
26,890	1.70E−05	1.84	9.65E−05	2.08	3.24E−05	1.96	7.78E−05	1.96	3.24E−05	1.96	7.78E−05	1.96
Exact												
1,096	1.04E−03	—	2.24E−03	—	4.32E−06	—	2.07E−05	—	1.45E−06	—	3.50E−06	—
2,478	4.86E−04	1.86	1.21E−03	1.52	8.82E−07	3.89	5.32E−06	3.33	1.73E−07	5.21	4.17E−07	5.21
5,366	2.37E−04	1.86	6.22E−04	1.71	1.73E−07	4.22	1.22E−06	3.82	1.94E−08	5.66	4.65E−08	5.68
11,624	1.05E−04	2.10	3.13E−04	1.77	4.88E−08	3.27	3.22E−07	3.44	2.22E−09	5.61	5.24E−09	5.65
26,890	4.74E−05	1.89	1.34E−04	2.03	7.98E−09	4.32	6.16E−08	3.94	2.09E−10	5.64	4.94E−10	5.63
ROD												
1,096	1.03E−03	—	2.23E−03	—	4.29E−06	—	2.07E−05	—	1.46E−06	—	3.49E−06	—
2,478	4.84E−04	1.86	1.20E−03	1.51	8.87E−07	3.86	5.32E−06	3.33	1.73E−07	5.22	4.16E−07	5.21
5,366	2.36E−04	1.86	6.21E−04	1.71	1.72E−07	4.25	1.22E−06	3.82	1.94E−08	5.67	4.65E−08	5.68
11,624	1.05E−04	2.10	3.13E−04	1.77	4.89E−08	3.25	3.21E−07	3.44	2.22E−09	5.61	5.24E−09	5.64
26,890	4.74E−05	1.89	1.34E−04	2.03	8.00E−09	4.32	6.16E−08	3.94	2.10E−10	5.63	4.97E−10	5.62

resulting in improved accuracy and convergence orders for the velocity compared to the $\mathbb{P}_d$–$\mathbb{P}_d$ method, which achieves second-order convergence with $d=1$ and almost fourth- and sixth-orders convergence with $d=3,5$, respectively, in both norms.
- Further increasing the polynomial degree for the streamfunction reconstructions in the $\mathbb{P}_d$–$\mathbb{P}_{d+2}$ method slightly improves boundary vorticity accuracy in comparison with the $\mathbb{P}_d$–$\mathbb{P}_{d+1}$ method, in general. Thus, slightly more accurate vorticity and streamfunction cell mean-values are also computed, although such small improvements can often be limited in more practical domains due to the higher condition numbers of the resulting system of linear equations. There is, however, a substantial accuracy improvement in the velocity.

6.3. Annular domain

This test case addresses an internal creeping flow (Stokes equations) confined between two infinitely long cylinders rotating at specific constant angular velocities. The interior and exterior cylinders have radii $r_I > 0$ and $r_E > 0$, respectively, and rotate with linear velocity magnitudes u_I and u_E, respectively. The exterior cylinder is assumed to be centred at the origin, while the two cylinders are not

Table 5
Velocity errors and convergence orders obtained in the flow in a circular domain test case with exact boundary vorticity.

	$d=1$				$d=3$				$d=5$			
N_E	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
Naive												
1,680	1.65E−02	—	1.68E−01	—	3.79E−03	—	5.65E−03	—	3.83E−03	—	5.67E−03	—
3,771	9.08E−03	1.48	1.27E−01	0.68	1.69E−03	2.00	2.55E−03	1.97	1.70E−03	2.01	2.53E−03	1.99
8,129	5.12E−03	1.49	8.56E−02	1.04	7.72E−04	2.04	1.19E−03	1.98	7.74E−04	2.05	1.16E−03	2.04
17,554	3.00E−03	1.39	6.12E−02	0.87	3.55E−04	2.02	5.40E−04	2.05	3.56E−04	2.02	5.32E−04	2.02
40,513	1.75E−03	1.28	4.26E−02	0.87	1.56E−04	1.97	2.39E−04	1.95	1.56E−04	1.97	2.34E−04	1.97
Exact												
1,680	1.83E−02	—	1.73E−01	—	1.41E−04	—	1.84E−03	—	1.00E−05	—	1.19E−04	—
3,771	9.70E−03	1.57	1.30E−01	0.71	3.69E−05	3.31	6.99E−04	2.40	1.10E−06	5.47	1.57E−05	5.01
8,129	5.33E−03	1.56	8.67E−02	1.05	1.04E−05	3.29	3.38E−04	1.89	1.28E−07	5.59	2.45E−06	4.85
17,554	3.07E−03	1.43	6.17E−02	0.88	2.97E−06	3.26	8.16E−05	3.69	1.60E−08	5.42	4.03E−07	4.68
40,513	1.78E−03	1.31	4.28E−02	0.88	7.78E−07	3.21	2.38E−05	2.94	1.74E−09	5.30	6.02E−08	4.55
ROD												
1,680	1.83E−02	—	1.73E−01	—	1.41E−04	—	1.84E−03	—	1.00E−05	—	1.19E−04	—
3,771	9.70E−03	1.57	1.30E−01	0.71	3.69E−05	3.31	6.98E−04	2.40	1.10E−06	5.46	1.57E−05	5.01
8,129	5.32E−03	1.56	8.67E−02	1.05	1.04E−05	3.29	3.38E−04	1.89	1.28E−07	5.59	2.44E−06	4.85
17,554	3.07E−03	1.43	6.17E−02	0.88	2.97E−06	3.26	8.15E−05	3.69	1.60E−08	5.42	4.03E−07	4.68
40,513	1.78E−03	1.31	4.28E−02	0.88	7.78E−07	3.21	2.38E−05	2.94	1.74E−09	5.30	6.02E−08	4.55

Table 6
Boundary vorticity errors and convergence orders obtained in the flow in a circular domain test case with exact boundary vorticity.

	$d=1$				$d=3$				$d=5$			
N_B	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
Naive												
72	3.00E+00	—	3.00E+00	—	6.55E−02	—	7.23E−02	—	5.62E−03	—	6.43E−03	—
108	3.00E+00	—	3.00E+00	—	3.21E−02	1.76	3.94E−02	1.50	2.07E−03	2.47	2.30E−03	2.54
160	3.00E+00	—	3.00E+00	—	1.52E−02	1.91	1.91E−02	1.85	8.56E−04	2.24	9.23E−04	2.32
236	3.00E+00	—	3.00E+00	—	7.51E−03	1.81	9.67E−03	1.75	3.73E−04	2.14	3.90E−04	2.21
356	3.00E+00	—	3.00E+00	—	3.20E−03	2.07	4.39E−03	1.92	1.59E−04	2.07	1.64E−04	2.10
Exact												
72	2.99E+00	—	2.99E+00	—	6.17E−02	—	6.85E−02	—	1.82E−03	—	2.63E−03	—
108	3.00E+00	—	3.00E+00	—	3.04E−02	1.74	3.77E−02	1.47	3.74E−04	3.89	6.05E−04	3.63
160	3.00E+00	—	3.00E+00	—	1.44E−02	1.90	1.83E−02	1.84	8.54E−05	3.76	1.52E−04	3.52
236	3.00E+00	—	3.00E+00	—	7.15E−03	1.80	9.31E−03	1.73	1.83E−05	3.96	3.60E−05	3.71
356	3.00E+00	—	3.00E+00	—	3.05E−03	2.07	4.24E−03	1.92	3.41E−06	4.09	8.64E−06	3.47
ROD												
72	3.00E+00	—	3.00E+00	—	6.41E−02	—	7.09E−02	—	1.89E−03	—	2.72E−03	—
108	3.00E+00	—	3.00E+00	—	3.12E−02	1.78	3.85E−02	1.51	3.86E−04	3.92	6.18E−04	3.66
160	3.00E+00	—	3.00E+00	—	1.46E−02	1.92	1.85E−02	1.86	8.73E−05	3.78	1.54E−04	3.53
236	3.00E+00	—	3.00E+00	—	7.23E−03	1.81	9.40E−03	1.75	1.86E−05	3.98	3.64E−05	3.72
356	3.00E+00	—	3.00E+00	—	3.07E−03	2.08	4.26E−03	1.92	3.45E−06	4.10	8.70E−06	3.48

necessarily coaxial, resulting in a non-null eccentricity (the distance between the centres of the two cylinders) defined as $e = r_E - r_I$. Since the cylinder lengths are assumed to be infinite, the flow occurs only along the azimuthal axis, and, under these conditions, the flow has a known analytical solution for the vorticity and streamfunction [56], given as

$$\omega = 2B\frac{(-x+y+s)(x+y+s)}{c^2} + 2C\frac{(x+y-s)(x-y+s)}{d^2} - 4E - 8Fs\frac{\left(x^2-y^2+s^2\right)}{cd}, \text{ and} \tag{73}$$

$$\psi = A\log\left(\frac{a}{b}\right) + By\frac{s+y}{a} + Cy\frac{s-y}{b} + Dy + E\left(x^2+y^2+s^2\right) + Fy\log\left(\frac{a}{b}\right), \tag{74}$$

where the parameters a, b, c, and d are expressed as $a = x^2 + (s+y)^2$, $b = x^2 + (s-y)^2$, $c = x^2 + y^2 + s^2 + 2sy$, and $d = x^2 + y^2 + s^2 - 2sy$, and the coefficients A, B, C, D, E, F, G, and H are expressed as

$$A = -G\frac{d_I d_E - s^2}{2}, \qquad B = G\left(d_I + s\right)\left(d_E + s\right), \qquad C = G\left(d_I - s\right)\left(d_E - s\right), \tag{75}$$

Table 7
Vorticity errors and convergence orders obtained in the flow in a circular domain test case with approximate boundary vorticity.

	$d=1$				$d=3$				$d=5$			
N_C	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
$\mathbb{P}_d$–$\mathbb{P}_d$												
1,096	3.45E−01	—	1.73E+01	—	1.87E−04	—	2.96E−03	—	3.78E−06	—	3.95E−05	—
2,478	3.45E+01	—	1.00E+04	—	5.21E−05	3.14	2.04E−03	0.91	4.85E−07	5.04	8.33E−06	3.82
5,366	1.44E−01	—	1.23E+01	—	1.48E−05	3.26	7.73E−04	2.51	7.47E−08	4.84	1.86E−06	3.88
11,624	3.38E−01	—	1.54E+02	—	4.84E−06	2.89	3.43E−04	2.11	1.19E−08	4.74	5.81E−07	3.02
26,890	6.01E−02	—	2.87E+01	—	1.36E−06	3.03	1.99E−04	1.30	1.58E−09	4.83	9.73E−08	4.26
$\mathbb{P}_d$–$\mathbb{P}_{d+1}$												
1,096	4.46E−03	—	2.73E−02	—	9.09E−05	—	5.04E−04	—	3.75E−06	—	2.57E−05	—
2,478	2.17E−03	1.77	1.72E−02	1.14	1.62E−05	4.23	1.38E−04	3.17	3.75E−07	5.65	3.16E−06	5.13
5,366	9.77E−04	2.06	8.80E−03	1.73	3.08E−06	4.29	3.17E−05	3.82	3.49E−08	6.14	5.49E−07	4.53
11,624	4.62E−04	1.94	8.40E−03	0.12	6.30E−07	4.11	1.04E−05	2.87	3.66E−09	5.84	5.98E−08	5.73
26,890	1.96E−04	2.05	5.99E−03	0.81	1.17E−07	4.01	2.27E−06	3.64	3.08E−10	5.90	7.48E−09	4.96
$\mathbb{P}_d$–$\mathbb{P}_{d+2}$												
1,096	3.92E−03	—	1.35E−02	—	8.75E−05	—	3.75E−04	—	3.88E−06	—	1.23E−05	—
2,478	1.97E−03	1.69	9.24E−03	0.93	1.56E−05	4.22	9.83E−05	3.28	3.80E−07	5.69	1.72E−06	4.83
5,366	8.94E−04	2.04	4.60E−03	1.80	3.05E−06	4.23	2.10E−05	3.99	3.51E−08	6.17	2.21E−07	5.31
11,624	4.18E−04	1.97	2.75E−03	1.33	6.05E−07	4.19	4.76E−06	3.84	3.60E−09	5.89	2.86E−08	5.29
26,890	1.76E−04	2.06	1.26E−03	1.87	1.14E−07	3.97	1.15E−06	3.38	3.61E−10	5.48	1.65E−08	1.31

Table 8
Streamfunction errors and convergence orders obtained in the flow in a circular domain test case with approximate boundary vorticity.

	$d=1$				$d=3$				$d=5$			
N_C	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
$\mathbb{P}_d$–$\mathbb{P}_d$												
1,096	6.10E−03	—	9.07E−02	—	1.40E−05	—	2.69E−05	—	3.46E−07	—	7.57E−07	—
2,478	3.03E−01	—	7.31E+00	—	3.80E−06	3.21	6.82E−06	3.36	6.39E−08	4.14	1.07E−07	4.80
5,366	9.56E−04	—	1.90E−02	—	1.05E−06	3.32	1.65E−06	3.67	8.61E−09	5.19	1.43E−08	5.20
11,624	3.51E−03	—	7.91E−02	—	2.09E−07	4.19	3.76E−07	3.83	1.19E−09	5.12	1.71E−09	5.49
26,890	1.55E−04	—	4.22E−03	—	4.40E−08	3.72	7.55E−08	3.83	1.13E−10	5.61	1.71E−10	5.49
$\mathbb{P}_d$–$\mathbb{P}_{d+1}$												
1,096	2.70E−04	—	6.80E−04	—	1.02E−05	—	2.04E−05	—	3.33E−07	—	7.39E−07	—
2,478	1.41E−04	1.58	3.78E−04	1.44	2.14E−06	3.83	3.80E−06	4.12	3.45E−08	5.55	6.80E−08	5.85
5,366	6.07E−05	2.19	1.97E−04	1.69	4.52E−07	4.02	7.14E−07	4.33	3.93E−09	5.63	6.71E−09	5.99
11,624	2.88E−05	1.93	9.74E−05	1.82	9.62E−08	4.00	1.42E−07	4.18	3.77E−10	6.06	5.78E−10	6.35
26,890	1.15E−05	2.20	4.05E−05	2.10	1.74E−08	4.08	2.45E−08	4.19	2.56E−11	6.41	4.11E−11	6.30
$\mathbb{P}_d$–$\mathbb{P}_{d+2}$												
1,096	5.37E−04	—	8.86E−04	—	4.76E−06	—	7.72E−06	—	2.09E−07	—	3.77E−07	—
2,478	2.78E−04	1.62	4.52E−04	1.65	5.70E−07	5.20	9.13E−07	5.23	1.51E−08	6.44	2.63E−08	6.52
5,366	1.28E−04	2.01	2.10E−04	1.99	4.69E−08	6.47	1.01E−07	5.69	7.70E−10	7.71	1.60E−09	7.25
11,624	6.03E−05	1.94	9.89E−05	1.94	1.14E−08	3.65	3.93E−08	2.45	4.78E−11	7.19	1.27E−10	6.55
26,890	2.56E−05	2.05	4.30E−05	1.98	2.98E−09	3.21	1.02E−08	3.22	4.14E−12	5.83	1.14E−11	5.76

$$D = H\left(d_I \log\left(\frac{d_E+s}{d_E-s}\right) - d_E \log\left(\frac{d_I+s}{d_I-s}\right) - 2Hs\frac{r_E^2-r_I^2}{r_E^2+r_I^2}\right)\left(r_I u_I + r_E u_E\right) - \frac{r_I^2 r_E^2\left(r_I^{-1}u_I - r_E^{-1}u_E\right)}{\left(r_I^2+r_E^2\right)e}, \tag{76}$$

$$E = \frac{H}{2}\log\left(\frac{(d_I+s)(d_E-s)}{(d_I-s)(d_E+s)}\right)\left(r_I u_I + r_E u_E\right), \qquad F = He\left(r_I u_I + r_E u_E\right), \tag{77}$$

$$G = 2H\frac{d_E^2-d_I^2}{r_I^2+r_E^2}\left(r_I u_I + r_E u_E\right), \qquad \text{and} \qquad H = \left(\left(r_I^2+r_E^2\right)\log\left(\frac{(d_I+s)(d_E-s)}{(d_I-s)(d_E+s)}\right) - 4se\right)^{-1}, \tag{78}$$

and the distances s, d_I, and d_E are given as

$$s = \left(\frac{(r_E-r_I-e)(r_E-r_I+e)(r_E+r_I-e)(r_E+r_I+e)}{4e^2}\right)^{1/2}, \tag{79}$$

Table 9
Boundary vorticity errors and convergence orders obtained in the flow in a circular domain test case with approximate boundary vorticity.

	$d=1$				$d=3$				$d=5$			
N_B	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
$\mathbb{P}_d$–$\mathbb{P}_d$												
72	2.35E+00	—	2.38E+01	—	2.48E−03	—	4.86E−03	—	2.07E−05	—	6.08E−05	—
108	5.20E+02	—	1.70E+04	—	9.46E−04	2.38	2.99E−03	1.20	4.26E−06	3.90	1.25E−05	3.90
160	2.18E+00	—	1.72E+01	—	3.19E−04	2.77	1.07E−03	2.62	1.00E−06	3.68	2.79E−06	3.82
236	5.86E+00	—	2.01E+02	—	1.94E−04	1.27	5.63E−04	1.65	2.33E−07	3.75	8.65E−07	3.01
356	2.43E+00	—	4.08E+01	—	7.15E−05	2.43	2.82E−04	1.68	4.39E−08	4.06	1.54E−07	4.19
$\mathbb{P}_d$–$\mathbb{P}_{d+1}$												
72	1.73E−02	—	4.83E−02	—	1.15E−04	—	3.85E−04	—	1.54E−05	—	3.71E−05	—
108	9.37E−03	1.52	2.71E−02	1.43	3.82E−05	2.71	1.06E−04	3.19	1.98E−06	5.06	4.18E−06	5.38
160	5.19E−03	1.50	1.59E−02	1.36	8.51E−06	3.82	2.70E−05	3.47	2.35E−07	5.42	8.00E−07	4.20
236	3.88E−03	0.75	1.31E−02	0.49	4.37E−06	1.71	1.53E−05	1.46	3.19E−08	5.14	8.46E−08	5.78
356	2.46E−03	1.11	8.44E−03	1.07	1.07E−06	3.41	3.01E−06	3.95	3.24E−09	5.56	1.15E−08	4.87
$\mathbb{P}_d$–$\mathbb{P}_{d+2}$												
72	1.61E−02	—	2.81E−02	—	3.24E−05	—	1.06E−04	—	1.51E−05	—	1.88E−05	—
108	7.58E−03	1.86	1.92E−02	0.94	1.67E−05	1.64	4.52E−05	2.11	1.89E−06	5.13	2.54E−06	4.94
160	3.87E−03	1.71	1.08E−02	1.46	3.21E−06	4.19	1.16E−05	3.46	2.19E−07	5.48	2.98E−07	5.45
236	2.12E−03	1.55	6.25E−03	1.42	9.51E−07	3.13	4.19E−06	2.62	2.49E−08	5.59	4.22E−08	5.03
356	1.09E−03	1.62	3.25E−03	1.59	1.94E−07	3.87	5.96E−07	4.74	6.00E−09	3.46	2.70E−08	1.08

Table 10
Velocity errors and convergence orders obtained in the flow in a circular domain test case with approximate boundary vorticity.

	$d=1$				$d=3$				$d=5$			
N_E	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
$\mathbb{P}_d$–$\mathbb{P}_d$												
1,680	5.46E−02	—	8.85E−01	—	1.41E−04	—	1.99E−03	—	6.08E−06	—	1.08E−04	—
3,771	2.66E+00	—	1.90E+02	—	3.68E−05	3.32	6.48E−04	2.78	6.88E−07	5.39	1.41E−05	5.04
8,129	1.29E−02	—	3.19E−01	—	1.09E−05	3.16	3.35E−04	1.72	9.33E−08	5.20	2.30E−06	4.72
17,554	2.71E−02	—	1.69E+00	—	2.97E−06	3.39	8.06E−05	3.70	1.29E−08	5.14	3.84E−07	4.64
40,513	3.08E−03	—	1.58E−01	—	7.80E−07	3.20	2.28E−05	3.02	1.54E−09	5.09	5.79E−08	4.53
$\mathbb{P}_d$–$\mathbb{P}_{d+1}$												
1,680	2.83E−03	—	1.95E−02	—	2.75E−05	—	2.97E−04	—	1.24E−06	—	2.04E−05	—
3,771	1.25E−03	2.03	9.41E−03	1.80	6.32E−06	3.64	6.86E−05	3.62	1.27E−07	5.63	2.26E−06	5.44
8,129	5.59E−04	2.09	5.03E−03	1.63	1.62E−06	3.55	1.98E−05	3.23	1.47E−08	5.62	2.21E−07	6.05
17,554	2.53E−04	2.06	2.27E−03	2.07	3.84E−07	3.74	4.34E−06	3.94	1.71E−09	5.58	2.83E−08	5.34
40,513	1.06E−04	2.08	1.11E−03	1.70	8.16E−08	3.70	7.84E−07	4.09	1.83E−10	5.35	3.32E−09	5.12
$\mathbb{P}_d$–$\mathbb{P}_{d+2}$												
1,680	7.30E−04	—	3.49E−03	—	1.33E−05	—	1.26E−04	—	5.53E−07	—	3.93E−06	—
3,771	3.32E−04	1.95	1.31E−03	2.43	1.44E−06	5.49	1.69E−05	4.96	4.18E−08	6.39	4.91E−07	5.14
8,129	1.55E−04	1.99	8.41E−04	1.15	2.52E−07	4.54	2.80E−06	4.68	2.82E−09	7.02	3.14E−08	7.16
17,554	7.14E−05	2.01	4.32E−04	1.73	5.90E−08	3.77	4.25E−07	4.90	3.06E−10	5.77	4.12E−09	5.27
40,513	3.08E−05	2.01	1.76E−04	2.15	1.28E−08	3.65	7.21E−08	4.24	2.39E−11	6.10	1.86E−09	1.90

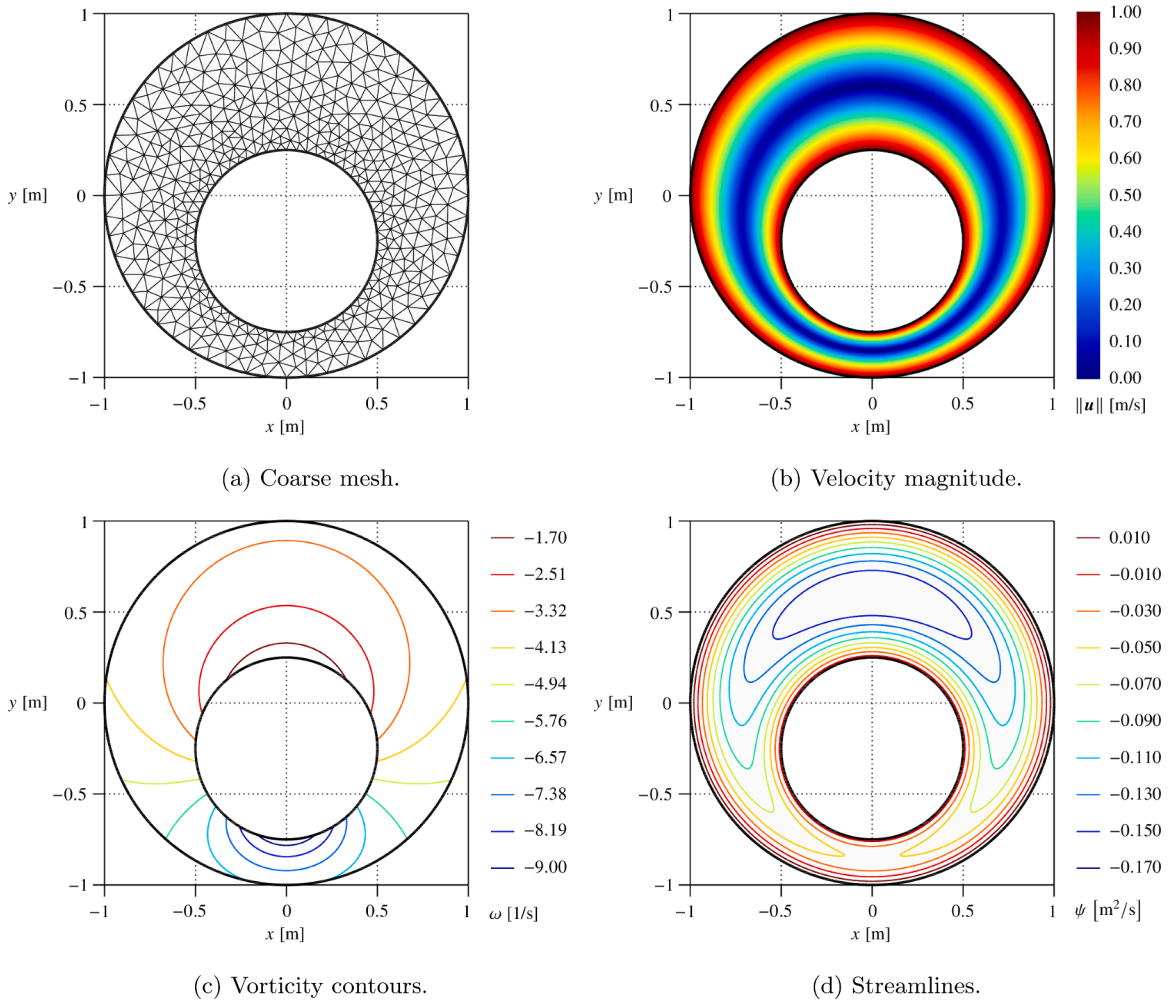
$$d_I = \frac{r_E^2 - r_I^2}{2e} - \frac{e}{2}, \qquad \text{and} \qquad d_E = \frac{r_E^2 - r_I^2}{2e} + \frac{e}{2}. \tag{80}$$

The velocity components are obtained from the streamfunction as

$$u_x = 4As\frac{x^2 - y^2 + s^2}{cd} + B\frac{sx^2 + sy^2 + 2x^2y + 2s^2y + s^3}{c^2} + C\frac{sx^2 + sy^2 - 2x^2y - 2s^2y + s^3}{d^2} + D + 2Ey + F\frac{2y((s-y)a + (s+y)b) + ab\log\left(\frac{a}{b}\right)}{ab}, \text{ and} \tag{81}$$

$$u_y = A\frac{8sxy}{cd} + B\frac{2xy(s+y)}{c^2} + C\frac{2xy(s-y)}{d^2} - 2Ex + F\frac{8sxy^2}{cd}. \tag{82}$$

For the domain, the interior physical boundary has a radius of $r_I = 1/2$ m (normal curvature of $\kappa_I = 2$), centre at $(0, -1/4)$ m, and rotates with $u_I = -1$ m/s (counter-clockwise). The exterior physical boundary has a radius of $r_E = 1$ m (normal curvature of $\kappa_E = 1$), centre at the origin, and rotates with $u_E = 1$ m/s (clockwise). For the fluid, a dynamic viscosity of $\mu = 1$ kg/(m s) and a density of



(a) Coarse mesh.

(b) Velocity magnitude.

(c) Vorticity contours.

(d) Streamlines.

Fig. 6. Coarse mesh and analytical solution for the flow in an annular domain test case (for proper interpretation of the colour scale, the reader is referred to the electronic version of this article).

$\rho = 1$ kg/m^3 are used. Successively finer uniform Delaunay triangular meshes are used to discretise the physical domain, and the simulations are carried out for the $\mathbb{P}_d$–$\mathbb{P}_d$, $\mathbb{P}_d$–$\mathbb{P}_{d+1}$, and $\mathbb{P}_d$–$\mathbb{P}_{d+2}$ methods with $d = 1, 3, 5$. Fig. 6 illustrates the physical domain with a coarse mesh and the analytical solutions for the velocity, vorticity, and streamfunction fields.

The errors and convergence orders for the approximate vorticity, streamfunction, boundary vorticity, and velocity are reported in Tables 11, 12, 13, and 14, respectively. The results exhibit the same behaviour as the previous test case, and, in general, the $\mathbb{P}_d$–$\mathbb{P}_d$ method does not achieve optimal convergence orders in both norms for the vorticity and streamfunction. Again, in accordance with the previous test case, the $\mathbb{P}_d$–$\mathbb{P}_{d+1}$ method substantially improves the boundary vorticity accuracy and convergence order, which turns out to be effective in recovering the optimal convergence orders for the vorticity and streamfunction, except for the L^∞-norm for the streamfunction. Moreover, the same convergence orders as for the streamfunction are achieved for the velocity, indicating that the accuracy deterioration in the streamfunction polynomial reconstruction derivatives is not significant. In turn, the $\mathbb{P}_d$–$\mathbb{P}_{d+2}$ method further improves the boundary vorticity accuracy, achieving fourth- and sixth-orders convergence with $d = 3, 5$, respectively, in both norms. On the contrary, the strategy is not effective with $d = 1$, which maintains a first-order convergence in both norms. Consequently, there is also no improvement in the convergence order for the vorticity, streamfunction, and velocity with $d = 1$ compared to the $\mathbb{P}_d$–$\mathbb{P}_{d+1}$ method. However, it is effective in recovering the optimal order for the vorticity in the L^∞-norm with $d = 3, 5$, while super-optimal convergence orders are obtained in the L^1-norm and for the streamfunction and velocity in both norms.

6.4. Rose-shaped domain

This test case addresses an internal flow (Navier-Stokes equations) confined between two rose-shaped boundaries, which correspond to a diffeomorphic transformation of an annular domain with interior and exterior physical boundaries with radii r_I and r_E,

Table 11
Vorticity errors and convergence orders obtained in the flow in an annular domain test case.

	$d=1$				$d=3$				$d=5$			
N_C	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
$\mathbb{P}_d$–$\mathbb{P}_d$												
1,178	4.65E−01	—	3.65E+01	—	5.15E−04	—	9.62E−03	—	3.38E−04	—	9.78E−03	—
2,374	7.04E−01	—	4.88E+01	—	1.74E−04	3.10	7.24E−03	0.81	5.75E−05	5.05	1.45E−03	5.44
5,236	1.43E+00	—	4.56E+02	—	5.22E−05	3.04	3.43E−03	1.89	6.40E−06	5.55	3.12E−04	3.89
10,378	1.15E+00	—	3.47E+02	—	1.53E−05	3.59	1.43E−03	2.56	8.74E−07	5.82	9.65E−05	3.43
22,072	2.71E−01	—	1.81E+02	—	5.39E−06	2.76	1.01E−03	0.92	1.26E−07	5.14	1.81E−05	4.44
$\mathbb{P}_d$–$\mathbb{P}_{d+1}$												
1,178	2.05E−03	—	3.62E−02	—	3.19E−04	—	5.32E−03	—	3.75E−04	—	1.40E−02	—
2,374	1.19E−03	1.55	2.55E−02	1.00	7.03E−05	4.31	1.71E−03	3.24	3.07E−05	7.14	1.11E−03	7.23
5,236	5.60E−04	1.90	1.76E−02	0.93	1.28E−05	4.31	8.25E−04	1.84	2.30E−06	6.56	1.03E−04	6.02
10,378	2.80E−04	2.03	1.19E−02	1.16	2.67E−06	4.58	2.88E−04	3.08	2.12E−07	6.96	2.46E−05	4.17
22,072	1.41E−04	1.81	9.11E−03	0.70	5.81E−07	4.04	8.75E−05	3.15	2.44E−08	5.73	2.94E−06	5.64
$\mathbb{P}_d$–$\mathbb{P}_{d+2}$												
1,178	1.84E−03	—	2.62E−02	—	4.64E−04	—	1.25E−02	—	2.97E−04	—	6.67E−03	—
2,374	9.94E−04	1.76	1.74E−02	1.16	4.52E−05	6.65	1.41E−03	6.22	2.68E−05	6.86	1.04E−03	5.32
5,236	5.11E−04	1.68	1.49E−02	0.39	5.06E−06	5.54	3.66E−04	3.42	1.70E−06	6.98	9.38E−05	6.07
10,378	2.46E−04	2.14	1.38E−02	0.22	7.92E−07	5.42	7.24E−05	4.73	1.14E−07	7.88	7.59E−06	7.35
22,072	1.24E−04	1.81	1.12E−02	0.55	1.34E−07	4.72	1.18E−05	4.82	7.49E−09	7.22	1.38E−06	4.51

Table 12
Streamfunction errors and convergence orders obtained in the flow in an annular domain test case.

	$d=1$				$d=3$				$d=5$			
N_C	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
$\mathbb{P}_d$–$\mathbb{P}_d$												
1,178	4.88E−03	—	7.60E−02	—	3.85E−05	—	1.42E−04	—	4.52E−06	—	2.59E−05	—
2,374	8.28E−03	—	8.87E−02	—	1.04E−05	3.74	3.26E−05	4.21	1.91E−06	2.45	6.52E−06	3.93
5,236	1.49E−02	—	4.02E−01	—	2.47E−06	3.63	7.29E−06	3.79	2.32E−07	5.33	6.28E−07	5.91
10,378	6.85E−03	—	2.80E−01	—	6.89E−07	3.73	1.85E−06	4.01	3.14E−08	5.85	8.29E−08	5.92
22,072	6.67E−04	—	3.58E−02	—	1.89E−07	3.43	4.88E−07	3.53	3.82E−09	5.59	1.04E−08	5.50
$\mathbb{P}_d$–$\mathbb{P}_{d+1}$												
1,178	7.95E−04	—	2.11E−03	—	1.84E−05	—	4.19E−05	—	1.53E−05	—	4.34E−05	—
2,374	3.81E−04	2.10	1.19E−03	1.63	2.66E−06	5.53	6.38E−06	5.37	5.65E−07	9.42	2.15E−06	8.57
5,236	1.75E−04	1.96	5.28E−04	2.07	4.17E−07	4.68	8.36E−07	5.14	5.08E−08	6.09	1.49E−07	6.75
10,378	8.99E−05	1.95	2.92E−04	1.73	8.77E−08	4.56	1.75E−07	4.57	4.71E−09	6.95	1.43E−08	6.86
22,072	4.13E−05	2.06	1.41E−04	1.93	1.94E−08	3.99	3.85E−08	4.02	4.34E−10	6.32	1.30E−09	6.35
$\mathbb{P}_d$–$\mathbb{P}_{d+2}$												
1,178	8.46E−04	—	2.22E−03	—	7.30E−06	—	3.39E−05	—	5.34E−06	—	3.06E−05	—
2,374	4.14E−04	2.04	1.30E−03	1.53	1.32E−06	4.89	4.36E−06	5.85	5.11E−07	6.70	1.71E−06	8.23
5,236	1.92E−04	1.95	5.50E−04	2.17	1.46E−07	5.55	4.29E−07	5.86	3.34E−08	6.90	1.23E−07	6.65
10,378	9.69E−05	1.99	3.04E−04	1.74	1.89E−08	5.99	5.33E−08	6.09	2.53E−09	7.55	6.62E−09	8.54
22,072	4.56E−05	2.00	1.48E−04	1.91	2.66E−09	5.19	7.71E−09	5.13	1.31E−10	7.85	4.67E−10	7.03

respectively. The resulting interior and exterior rose-shaped boundaries are parametrised in polar coordinates as

$$\Gamma_I : (r,\theta) = \big(R_I(\theta),\theta\big) \qquad \text{and} \qquad \Gamma_E : (r,\theta) = \big(R_E(\theta),\theta\big), \tag{83}$$

respectively, where functions $R_I := R_I(\theta)$ and $R_E := R_E(\theta)$ correspond to a periodic radius perturbation of frequency $\alpha_I, \alpha_E \in \mathbb{N}^+$ and magnitude $\beta_I, \beta_E \in \mathbb{R}$, given as

$$R_I = r_I - \beta_I + \beta_I \cos\big(\alpha_I\theta\big) \qquad \text{and} \qquad R_E = r_E - \beta_E + \beta_E \cos\big(\alpha_E\theta\big), \tag{84}$$

respectively. The normal curvatures of the interior and exterior physical boundaries, $\kappa_I := \kappa_I(\theta)$ and $\kappa_E := \kappa_E(\theta)$, respectively, can be determined analytically (see Appendix C) as

$$\kappa_I = -\frac{R_I^2 + 2\big(R_I'\big)^2 - R_I R_I''}{\Big(R_I^2 + \big(R_I'\big)^2\Big)^{3/2}} \qquad \text{and} \qquad \kappa_E = \frac{R_E^2 + 2\big(R_E'\big)^2 - R_E R_E''}{\Big(R_E^2 + \big(R_E'\big)^2\Big)^{3/2}}, \tag{85}$$

Table 13
Boundary vorticity errors and convergence orders obtained in the flow in an annular domain test case.

	$d=1$				$d=3$				$d=5$			
N_B	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
$\mathbb{P}_d$–$\mathbb{P}_d$												
128	2.44E+00	—	4.95E+01	—	2.83E−03	—	1.38E−02	—	1.65E−03	—	1.34E−02	—
184	3.92E+00	—	7.34E+01	—	1.38E−03	1.97	1.03E−02	0.81	3.24E−04	4.49	2.03E−03	5.20
272	9.36E+00	—	6.03E+02	—	6.98E−04	1.75	5.19E−03	1.75	4.94E−05	4.81	4.76E−04	3.71
384	1.20E+01	—	4.75E+02	—	3.23E−04	2.24	2.10E−03	2.62	1.30E−05	3.87	1.51E−04	3.33
560	5.41E+00	—	2.34E+02	—	1.56E−04	1.92	1.52E−03	0.86	2.35E−06	4.53	2.66E−05	4.61
$\mathbb{P}_d$–$\mathbb{P}_{d+1}$												
128	7.26E−03	—	5.58E−02	—	1.27E−03	—	8.19E−03	—	1.67E−03	—	1.96E−02	—
184	5.72E−03	0.65	4.06E−02	0.88	3.93E−04	3.23	2.48E−03	3.29	1.72E−04	6.26	1.44E−03	7.20
272	3.33E−03	1.38	2.30E−02	1.45	1.18E−04	3.09	1.18E−03	1.90	2.00E−05	5.51	1.80E−04	5.31
384	2.89E−03	0.41	1.80E−02	0.71	4.09E−05	3.06	4.23E−04	2.98	2.99E−06	5.51	3.68E−05	4.61
560	1.36E−03	1.99	1.26E−02	0.95	1.22E−05	3.20	1.33E−04	3.06	4.70E−07	4.91	4.25E−06	5.72
$\mathbb{P}_d$–$\mathbb{P}_{d+2}$												
128	1.07E−02	—	5.46E−02	—	2.01E−03	—	1.99E−02	—	1.46E−03	—	1.10E−02	—
184	6.17E−03	1.53	5.01E−02	0.24	2.98E−04	5.25	2.12E−03	6.17	1.39E−04	6.48	1.55E−03	5.40
272	4.59E−03	0.76	2.75E−02	1.53	4.57E−05	4.80	5.06E−04	3.67	1.14E−05	6.40	1.66E−04	5.72
384	2.98E−03	1.25	2.43E−02	0.35	1.09E−05	4.16	1.21E−04	4.15	1.12E−06	6.73	1.19E−05	7.64
560	2.09E−03	0.95	1.93E−02	0.61	1.87E−06	4.68	1.74E−05	5.14	1.07E−07	6.21	2.05E−06	4.67

Table 14
Velocity errors and convergence orders obtained in the flow in an annular domain test case.

	$d=1$				$d=3$				$d=5$			
N_E	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
$\mathbb{P}_d$–$\mathbb{P}_d$												
1,831	7.14E−02	—	1.11E+00	—	5.00E−04	—	1.13E−02	—	2.58E−04	—	1.45E−02	—
3,653	7.16E−02	—	9.00E−01	—	1.19E−04	4.15	4.99E−03	2.36	3.62E−05	5.69	2.71E−03	4.86
7,990	1.10E−01	—	5.02E+00	—	2.88E−05	3.63	1.32E−03	3.40	3.83E−06	5.74	3.86E−04	4.98
15,759	7.09E−02	—	3.89E+00	—	8.45E−06	3.61	4.49E−04	3.18	5.56E−07	5.68	6.28E−05	5.34
33,388	1.23E−02	—	1.16E+00	—	2.46E−06	3.28	1.75E−04	2.51	6.84E−08	5.58	1.24E−05	4.32
$\mathbb{P}_d$–$\mathbb{P}_{d+1}$												
1,831	2.41E−03	—	5.57E−02	—	2.73E−04	—	8.48E−03	—	2.47E−04	—	1.60E−02	—
3,653	1.09E−03	2.30	3.36E−02	1.46	4.81E−05	5.03	2.60E−03	3.42	1.68E−05	7.78	1.41E−03	7.03
7,990	4.55E−04	2.23	1.33E−02	2.36	7.49E−06	4.75	4.80E−04	4.32	1.19E−06	6.76	1.47E−04	5.77
15,759	1.86E−04	2.63	5.01E−03	2.88	1.49E−06	4.76	1.43E−04	3.56	1.21E−07	6.73	1.71E−05	6.35
33,388	9.98E−05	1.66	3.84E−03	0.71	2.82E−07	4.43	3.87E−05	3.49	1.03E−08	6.57	3.42E−06	4.28
$\mathbb{P}_d$–$\mathbb{P}_{d+2}$												
1,831	2.16E−03	—	3.35E−02	—	3.25E−04	—	1.71E−02	—	2.21E−04	—	1.70E−02	—
3,653	9.48E−04	2.39	9.45E−03	3.67	3.71E−05	6.28	2.96E−03	5.07	1.06E−05	8.79	1.00E−03	8.21
7,990	4.04E−04	2.18	6.72E−03	0.87	3.91E−06	5.75	4.09E−04	5.06	5.17E−07	7.72	8.73E−05	6.23
15,759	1.94E−04	2.16	2.69E−03	2.70	5.74E−07	5.65	7.09E−05	5.16	3.80E−08	7.69	8.05E−06	7.02
33,388	8.87E−05	2.08	1.43E−03	1.69	7.11E−08	5.57	1.37E−05	4.38	2.21E−09	7.58	6.36E−07	6.76

where $R'_I := R'_I(\theta)$ and $R'_E := R'_E(\theta)$ are the first derivatives and $R''_I := R''_I(\theta)$ and $R''_E := R''_E(\theta)$ are the second derivatives of $R_I(\theta)$ and $R_E(\theta)$ with respect to θ, that is

$$R'_I = \frac{\partial R_I}{\partial \theta} = -\alpha_I \beta_I \sin\left(\alpha_I \theta\right), \qquad R''_I = \frac{\partial R_I^2}{\partial \theta^2} = -\alpha_I^2 \beta_I \cos\left(\alpha_I \theta\right), \tag{86}$$

$$R'_E = \frac{\partial R_E}{\partial \theta} = -\alpha_E \beta_E \sin\left(\alpha_E \theta\right), \quad \text{and} \qquad R''_E = \frac{\partial R_E^2}{\partial \theta^2} = -\alpha_I^2 \beta_E \cos\left(\alpha_E \theta\right). \tag{87}$$

Notice that such non-trivial parametrisation poses significant difficulties in curved mesh approaches to determining quadrature rules and implementing non-linear transformations.

The manufactured solutions for the vorticity and streamfunction in polar coordinates, $\omega := \omega(r,\theta)$ and $\psi := \psi(r,\theta)$, respectively, read as

$$\omega = \frac{u}{ar^2}\left(R''_I\left(r - R_E\right) + R''_E\left(r - R_I\right) - 2R'_I R'_E - r\left(4r - R_I - R_E\right)\right), \text{ and} \tag{88}$$

Table 15
Vorticity errors and convergence orders obtained in the flow in a rose-shaped domain test case.

	$d=1$				$d=3$				$d=5$			
N_C	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
1 collocation point												
2,216	8.84E−02	—	8.31E−01	—	1.17E−02	—	1.68E−01	—	9.38E−03	—	1.14E−01	—
5,784	3.12E−02	2.17	3.69E−01	1.69	1.65E−03	4.08	4.07E−02	2.95	4.39E−04	6.39	1.41E−02	4.37
15,522	1.19E−02	1.96	1.10E−01	2.45	2.35E−04	3.95	6.61E−03	3.68	3.28E−05	5.25	1.28E−03	4.85
41,996	4.32E−03	2.04	4.54E−02	1.78	3.35E−05	3.91	1.27E−03	3.31	1.14E−06	6.75	1.06E−04	5.02
113,256	1.61E−03	1.98	2.73E−02	1.03	4.81E−06	3.91	2.27E−04	3.48	6.07E−08	5.91	6.29E−06	5.69
2 collocation points												
2,216	8.32E−02	—	5.70E−01	—	1.78E−02	—	2.87E−01	—	1.36E−02	—	2.19E−01	—
5,784	3.10E−02	2.06	1.88E−01	2.31	2.02E−03	4.54	4.70E−02	3.78	5.68E−04	6.62	2.02E−02	4.98
15,522	1.19E−02	1.94	1.03E−01	1.23	2.36E−04	4.34	6.79E−03	3.92	3.60E−05	5.59	1.64E−03	5.08
41,996	4.32E−03	2.03	4.95E−02	1.46	3.37E−05	3.91	1.32E−03	3.29	1.14E−06	6.93	1.02E−04	5.59
113,256	1.61E−03	1.98	2.87E−02	1.10	4.82E−06	3.92	2.64E−04	3.24	6.06E−08	5.92	5.90E−06	5.74

Table 16
Streamfunction errors and convergence orders obtained in the flow in a rose-shaped domain test case.

	$d=1$				$d=3$				$d=5$			
N_C	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
1 collocation point												
2,216	9.98E−04	—	2.07E−03	—	1.72E−04	—	3.12E−04	—	8.24E−05	—	2.06E−04	—
5,784	4.87E−04	1.50	9.85E−04	1.55	1.12E−05	5.71	3.03E−05	4.86	4.35E−06	6.14	8.22E−06	6.72
15,522	2.07E−04	1.73	3.76E−04	1.95	9.54E−07	4.98	3.31E−06	4.49	1.06E−07	7.53	3.35E−07	6.48
41,996	5.98E−05	2.50	1.20E−04	2.30	1.23E−07	4.12	4.55E−07	3.99	5.01E−09	6.13	1.62E−08	6.09
113,256	2.32E−05	1.90	4.57E−05	1.94	2.28E−08	3.40	5.43E−08	4.28	1.36E−10	7.26	4.76E−10	7.11
2 collocation points												
2,216	1.18E−03	—	2.26E−03	—	2.55E−04	—	4.26E−04	—	8.61E−05	—	2.06E−04	—
5,784	4.56E−04	1.99	9.45E−04	1.81	1.58E−05	5.79	3.83E−05	5.02	4.97E−06	5.95	9.23E−06	6.48
15,522	2.03E−04	1.64	3.71E−04	1.89	8.66E−07	5.88	3.12E−06	5.08	1.20E−07	7.54	3.63E−07	6.56
41,996	5.88E−05	2.49	1.18E−04	2.30	1.11E−07	4.13	4.24E−07	4.01	4.74E−09	6.50	1.58E−08	6.29
113,256	2.30E−05	1.89	4.54E−05	1.93	2.63E−08	2.90	6.14E−08	3.90	1.27E−10	7.30	5.20E−10	6.89

Table 17
Boundary vorticity errors and convergence orders obtained in the flow in a rose-shaped domain test case.

	$d=1$				$d=3$				$d=5$			
N_B	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
1 collocation point												
200	2.04E−01	—	9.22E−01	—	4.03E−02	—	2.51E−01	—	5.25E−02	—	1.98E−01	—
330	8.00E−02	1.86	5.12E−01	1.17	6.50E−03	3.64	5.61E−02	2.99	3.57E−03	5.37	1.78E−02	4.81
546	3.09E−02	1.89	1.89E−01	1.98	1.47E−03	2.95	8.07E−03	3.85	3.19E−04	4.80	2.03E−03	4.32
900	1.37E−02	1.62	8.36E−02	1.63	2.82E−04	3.31	1.50E−03	3.37	1.78E−05	5.77	1.44E−04	5.30
1,484	6.58E−03	1.47	4.69E−02	1.15	4.84E−05	3.52	3.13E−04	3.13	1.27E−06	5.28	1.00E−05	5.32
2 collocation points												
400	1.97E−01	—	1.03E+00	—	9.60E−02	—	4.96E−01	—	1.13E−01	—	5.42E−01	—
660	7.57E−02	1.91	4.26E−01	1.75	1.24E−02	4.08	7.57E−02	3.76	6.77E−03	5.62	3.75E−02	5.33
1,092	3.12E−02	1.76	2.14E−01	1.37	1.56E−03	4.13	9.01E−03	4.23	5.95E−04	4.83	2.68E−03	5.24
1,800	1.37E−02	1.65	1.06E−01	1.40	2.91E−04	3.35	1.77E−03	3.26	2.57E−05	6.29	2.07E−04	5.13
2,968	6.57E−03	1.47	5.43E−02	1.34	5.02E−05	3.52	3.75E−04	3.10	1.52E−06	5.65	1.12E−05	5.83

$$\psi = \frac{u}{a}\big(r - R_{\mathrm{I}}\big)\big(r - R_{\mathrm{E}}\big), \tag{89}$$

where $u \in \mathbb{R}$ is a reference linear velocity magnitude on the physical boundary and $a \in \mathbb{R}$ is a normalisation factor. The velocity components in polar coordinates, $u_r := u_r(r,\theta)$ and $u_\theta := u_\theta(r,\theta)$, are obtained from the streamfunction as

$$u_r = \frac{u}{ar}\big(R_{\mathrm{I}}'\big(R_{\mathrm{E}} - r\big) + R_{\mathrm{E}}'\big(R_{\mathrm{I}} - r\big)\big) \qquad \text{and} \qquad u_\theta = \frac{u}{a}\big(R_{\mathrm{I}} + R_{\mathrm{E}} - 2r\big). \tag{90}$$

Table 18
Velocity errors and convergence orders obtained in the flow in a rose-shaped domain test case.

	$d=1$				$d=3$				$d=5$			
N_E	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞	E_1	O_1	E_∞	O_∞
1 collocation point												
3,424	1.13E−02	—	1.13E−01	—	2.16E−03	—	4.61E−02	—	2.00E−03	—	6.55E−02	—
8,841	3.28E−03	2.61	3.49E−02	2.48	2.72E−04	4.37	7.57E−03	3.81	9.62E−05	6.39	4.21E−03	5.79
23,556	1.24E−03	1.99	1.76E−02	1.40	3.57E−05	4.15	1.84E−03	2.89	6.60E−06	5.47	5.33E−04	4.22
63,444	4.25E−04	2.16	5.12E−03	2.49	4.67E−06	4.11	2.26E−04	4.23	1.89E−07	7.18	2.03E−05	6.60
170,626	1.56E−04	2.03	2.07E−03	1.83	1.01E−06	3.09	3.84E−05	3.59	8.27E−09	6.32	1.43E−06	5.36
2 collocation points												
3,424	9.16E−03	—	7.89E−02	—	3.42E−03	—	6.18E−02	—	2.07E−03	—	6.39E−02	—
8,841	3.26E−03	2.18	3.50E−02	1.71	3.56E−04	4.77	1.05E−02	3.73	9.40E−05	6.52	4.26E−03	5.71
23,556	1.24E−03	1.98	1.76E−02	1.40	3.57E−05	4.69	1.84E−03	3.57	6.57E−06	5.43	5.35E−04	4.24
63,444	4.25E−04	2.16	5.13E−03	2.49	4.66E−06	4.11	2.26E−04	4.23	1.88E−07	7.17	2.03E−05	6.61
170,626	1.56E−04	2.03	2.07E−03	1.84	1.01E−06	3.09	3.83E−05	3.59	8.23E−09	6.32	1.43E−06	5.36

Table 19
Errors obtained with the naive method in the regularised semi-elliptical lid-driven cavity test case with $Re = 1$.

d	N_C	E_x [%]	E_y [%]	E_ω [%]	E_ψ [%]
$(\psi,\omega)/\mathbb{P}_d$–$\mathbb{P}_{d+1}$					
1	1,044	1.20E+02	5.56E−01	1.08E+00	1.65E−01
	4,070	8.39E+01	1.65E−01	3.25E−01	6.39E−02
	16,398	1.31E+01	6.32E−02	1.00E−01	2.47E−02
	65,786	3.01E+00	1.32E−02	2.41E−02	7.13E−03
3	1,044	1.95E+00	2.18E−02	2.18E−02	2.18E−02
	4,070	5.85E−02	5.70E−03	5.67E−03	5.60E−03
	16,398	4.04E−03	1.51E−03	1.49E−03	1.41E−03
	65,786	8.40E−04	3.78E−04	3.73E−04	3.52E−04
5	1,044	2.47E−02	2.32E−02	2.30E−02	2.14E−02
	4,070	1.17E−02	6.05E−03	5.98E−03	5.63E−03
	16,398	2.47E−03	1.51E−03	1.49E−03	1.41E−03
	65,786	6.24E−04	3.78E−04	3.74E−04	3.52E−04
$(\psi,\omega)/\mathbb{P}_d$–$\mathbb{P}_{d+2}$					
1	1,044	2.33E+02	2.88E−01	3.34E−01	3.30E−01
	4,070	1.27E+02	6.59E−02	6.40E−02	1.02E−01
	16,398	4.46E+01	4.69E−02	3.35E−02	2.44E−02
	65,786	3.17E+01	1.78E−02	1.57E−02	6.13E−03
3	1,044	2.37E+00	2.31E−02	2.29E−02	2.14E−02
	4,070	1.04E−02	5.58E−03	5.52E−03	5.60E−03
	16,398	1.81E−03	1.51E−03	1.49E−03	1.41E−03
	65,786	5.25E−04	3.78E−04	3.73E−04	3.52E−04
5	1,044	3.27E−02	2.31E−02	2.29E−02	2.15E−02
	4,070	1.20E−02	6.05E−03	5.98E−03	5.63E−03
	16,398	2.48E−03	1.51E−03	1.49E−03	1.41E−03
	65,786	6.24E−04	3.78E−04	3.74E−04	3.52E−04

The source term function in polar coordinates, $f := f(r,\theta)$, derives from the vorticity transport equation with the above-manufactured solution, the expression of which is omitted for compactness. Notice that although the exact solution is given in polar coordinates, the problem is solved in Cartesian coordinates.

For the domain, the interior physical boundary has parameters $r_I = 1/2$ m, $\alpha_I = 8$, and $\beta_I = r_I/10$ m, and centre at the origin. The exterior physical boundary has parameters $r_E = 1$ m, $\alpha_E = 8$, and $\beta_E = r_E/10$ m, and centre at the origin. The fluid has a reference

Table 20
Errors obtained with the ROD method in the regularised semi-elliptical lid-driven cavity test case with $Re = 1$.

d	N_C	E_x [%]	E_y [%]	E_ω [%]	E_ψ [%]	N_{FP}	N_{GMRES}	T_{EXE} [s]	M_{USE} [Gb]
$(\psi,\omega)/\mathbb{P}_d$–$\mathbb{P}_{d+1}$									
1	1,044	1.20E+02	5.77E−01	1.10E+00	1.43E−01	7	240	0.07	0.02
	4,070	8.40E+01	1.71E−01	3.31E−01	5.83E−02	7	490	0.31	0.06
	16,398	1.31E+01	6.47E−02	1.02E−01	2.33E−02	8	1,110	3.29	0.22
	65,786	3.01E+00	1.35E−02	2.45E−02	6.78E−03	9	2,850	38.40	0.85
3	1,044	2.00E+00	1.34E−03	1.03E−03	3.09E−04	7	380	0.19	0.03
	4,070	6.86E−02	3.49E−04	3.11E−04	3.28E−05	8	860	0.94	0.10
	16,398	1.25E−02	1.00E−05	9.04E−06	2.03E−06	9	1,630	8.04	0.42
	65,786	2.15E−04	4.98E−07	2.52E−07	1.46E−07	9	6,400	115.35	1.35
5	1,044	6.28E−02	2.98E−05	1.22E−04	7.26E−05	8	590	0.54	0.05
	4,070	1.92E−03	7.79E−07	1.87E−06	1.59E−06	8	1,050	2.97	0.20
	16,398	2.50E−05	1.40E−08	2.91E−08	1.46E−08	9	3,040	19.28	0.60
	65,786	—	—	—	—	9	9,860	240.24	2.27
$(\psi,\omega)/\mathbb{P}_d$–$\mathbb{P}_{d+2}$									
1	1,044	3.46E+01	4.26E−01	4.01E−01	4.95E−02	7	330	0.10	0.02
	4,070	7.44E+00	2.10E−01	1.95E−01	4.65E−03	7	630	0.51	0.07
	16,398	5.38E+00	6.63E−02	6.13E−02	8.73E−04	8	1,610	4.87	0.24
	65,786	4.80E+00	1.30E−02	1.30E−02	1.63E−04	9	4,410	64.86	0.94
3	1,044	1.46E+00	6.13E−03	6.89E−03	3.69E−04	7	490	0.28	0.04
	4,070	1.50E−02	4.74E−04	4.87E−04	3.08E−05	8	1,110	1.36	0.11
	16,398	6.84E−04	2.18E−06	1.43E−06	1.12E−06	9	2,520	11.46	0.40
	65,786	9.91E−05	4.18E−07	3.18E−07	3.75E−08	9	7,960	155.00	1.51
5	1,044	7.07E−02	1.61E−05	3.65E−05	5.44E−06	8	1,120	0.87	0.05
	4,070	1.97E−03	3.36E−07	1.13E−07	2.96E−07	8	1,550	3.31	0.18
	16,398	1.72E−05	1.62E−08	1.78E−08	1.34E−09	9	3,700	25.63	0.68
	65,786	—	—	—	—	9	27,790	677.04	2.59
$(\boldsymbol{u},p)$									
1	1,044	1.05E+02	1.22E+00	—	—	7	940	0.20	0.04
	4,070	1.92E+01	3.70E−01	—	—	7	2,080	1.94	0.13
	16,398	8.49E+00	7.36E−02	—	—	8	5,780	60.11	0.65
	65,786	6.83E−01	2.20E−02	—	—	8	21,440	2,121.02	3.59
3	1,044	6.40E−01	2.08E−03	—	—	7	810	0.32	0.05
	4,070	2.58E−02	1.58E−04	—	—	7	2,360	3.21	0.18
	16,398	1.83E−03	9.74E−06	—	—	7	6,870	86.42	0.85
	65,786	1.20E−04	6.50E−07	—	—	8	23,030	2,434.13	4.28
5	1,044	8.27E−03	7.96E−05	—	—	7	920	0.70	0.07
	4,070	4.01E−04	1.25E−06	—	—	7	2,550	5.96	0.27
	16,398	1.51E−05	4.11E−08	—	—	8	8,020	119.03	1.08
	65,786	—	—	—	—	8	24,710	2,877.39	5.53

linear velocity magnitude of $u = 1$ m/s, which corresponds to a counter-clockwise and clockwise rotation on the interior and exterior physical boundaries, respectively, a dynamic viscosity of $\mu = 1$ kg/(m s), and a density of $\rho = 1$ kg/m^3. The normalisation parameter is chosen as $a = 0.6021$. Successively finer uniform Delaunay triangular meshes are used to discretise the physical domain, and the simulations are carried out for the $\mathbb{P}_d$–$\mathbb{P}_{d+1}$ method with $d = 1, 3, 5$. Either one or two collocation points per boundary edge are considered to fulfil the prescribed boundary conditions on the vorticity and streamfunction polynomial reconstructions. For more than one collocation point, the ROD method follows the same least-squares procedure, considering more linear restrictions on the constraint functional. Fig. 7 illustrates the physical domain with a coarse mesh and the analytical solutions for the velocity, vorticity, and streamfunction fields.

The errors and convergence orders for the approximate vorticity, streamfunction, boundary vorticity, and velocity are reported in Tables 15, 16, 17, and 18, respectively. Regardless of the number of collocation points, the optimal second-, fourth-, and sixth-orders convergence are achieved with $d = 1, 3, 5$, respectively, in both norms for the streamfunction and in the L^1-norm for the vorticity. Moreover, more than the first-, third-, and fifth-orders convergence are obtained with $d = 1, 3, 5$, respectively, in both norms for the boundary vorticity, whereas the second-, fourth-, and sixth-orders convergence are obtained with $d = 1, 3, 5$, respectively, in both norms for the velocity. The results are in accordance with the previous test cases for the $\mathbb{P}_d$–$\mathbb{P}_{d+1}$ method, demonstrating the capability of the ROD method to preserve its effectiveness in recovering the optimal convergence orders in non-trivial curved

Table 21
Errors obtained with the ROD method in the regularised semi-elliptical lid-driven cavity test case with $Re = 100$.

d	N_C	E_x [%]	E_y [%]	E_ω [%]	E_ψ [%]	N_{FP}	N_{GMRES}	T_{EXE} [s]	M_{USE} [Gb]
$(\psi,\omega)/\mathbb{P}_d$–$\mathbb{P}_{d+1}$									
1	1,044	4.02E+00	9.88E−01	1.73E+00	2.34E−01	21	960	0.15	0.02
	4,070	1.17E+00	2.63E−01	5.07E−01	1.19E−01	22	1,880	0.83	0.06
	16,398	4.44E−01	6.11E−02	1.25E−01	3.73E−02	23	3,700	9.69	0.22
	65,786	6.25E−02	1.54E−02	3.18E−02	9.99E−03	23	8,360	108.67	0.85
3	1,044	3.56E−01	7.65E−03	1.01E−01	3.37E−02	22	1,450	0.42	0.04
	4,070	1.78E−02	1.19E−03	2.66E−03	1.04E−03	22	2,890	2.05	0.10
	16,398	1.40E−03	1.02E−04	1.38E−04	6.34E−05	23	6,330	22.19	0.36
	65,786	7.07E−05	9.00E−06	7.12E−06	3.02E−06	25	15,500	266.50	1.35
5	1,044	1.72E−02	2.44E−03	8.74E−03	2.80E−03	22	2,040	0.85	0.05
	4,070	6.80E−04	3.89E−05	1.45E−04	7.81E−05	22	4,400	4.67	0.16
	16,398	9.80E−06	9.69E−07	1.20E−06	1.13E−06	24	10,420	51.22	0.60
	65,786	—	—	—	—	24	19,940	456.98	2.27
$(\psi,\omega)/\mathbb{P}_d$–$\mathbb{P}_{d+2}$									
1	1,044	2.64E+00	8.31E−01	1.14E+00	3.98E−02	23	1,190	0.24	0.02
	4,070	7.93E−02	2.20E−01	2.66E−01	1.29E−02	22	2,620	1.31	0.06
	16,398	1.91E−02	2.89E−02	4.43E−02	5.06E−03	23	5,210	13.84	0.24
	65,786	1.39E−02	8.82E−03	1.17E−02	9.14E−04	24	12,440	175.91	0.94
3	1,044	2.31E−01	2.77E−03	2.31E−02	1.54E−02	22	1,680	0.48	0.04
	4,070	8.75E−03	4.69E−04	1.88E−03	7.62E−04	23	3,520	2.86	0.11
	16,398	8.15E−04	2.89E−05	8.46E−05	1.90E−05	23	8,260	31.14	0.40
	65,786	2.15E−05	3.24E−06	1.43E−06	1.77E−06	24	19,141	356.02	1.51
5	1,044	2.60E−02	1.38E−03	8.33E−03	3.83E−03	22	2,350	1.36	0.07
	4,070	4.87E−04	4.10E−05	1.19E−04	5.04E−05	23	5,210	6.01	0.18
	16,398	3.22E−06	8.00E−07	1.07E−06	4.17E−07	24	11,540	61.76	0.68
	65,786	—	—	—	—	24	24,100	591.23	2.59
$(\boldsymbol{u}, p)$									
1	1,044	5.68E+00	9.85E−01	—	—	12	1,560	0.30	0.04
	4,070	9.57E−01	2.48E−01	—	—	12	3,180	2.95	0.13
	16,398	1.84E−01	6.65E−02	—	—	13	10,020	108.43	0.66
	65,786	1.60E−02	1.77E−02	—	—	13	28,990	2,840.35	3.59
3	1,044	7.08E−02	2.39E−03	—	—	11	1,430	0.31	0.05
	4,070	7.07E−03	5.44E−05	—	—	12	3,340	4.45	0.18
	16,398	3.40E−04	6.19E−06	—	—	12	10,420	130.03	0.85
	65,786	2.74E−05	1.18E−06	—	—	13	34,340	3,557.23	4.28
5	1,044	7.66E−03	3.70E−04	—	—	11	1,520	0.88	0.07
	4,070	2.98E−04	5.64E−06	—	—	12	3,991	8.37	0.27
	16,398	4.44E−06	1.61E−07	—	—	13	11,710	176.37	1.15
	65,786	—	—	—	—	13	35,060	4,006.00	5.53

physical domains. Furthermore, comparable accuracy and convergence orders are obtained with one and two collocation points per boundary edge, providing strong evidence of the simplicity and efficiency of the proposed method. Indeed, these results distinctly emphasise the versatility of the ROD method in handling arbitrary curved physical boundaries prescribed with general boundary conditions.

6.5. Semi-elliptical lid-driven cavity

To finalise the numerical verification of the proposed method, the classical lid-driven cavity benchmark test case is addressed by replacing the rectangular cavity with a semi-ellipse centred at the origin, given implicitly as

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1, \tag{91}$$

where $a, b > 0$ are the semi-axis lengths, or parametrically in polar coordinates as $(r, \theta) = (R(\theta), \theta)$ with $R(\theta)$ given as

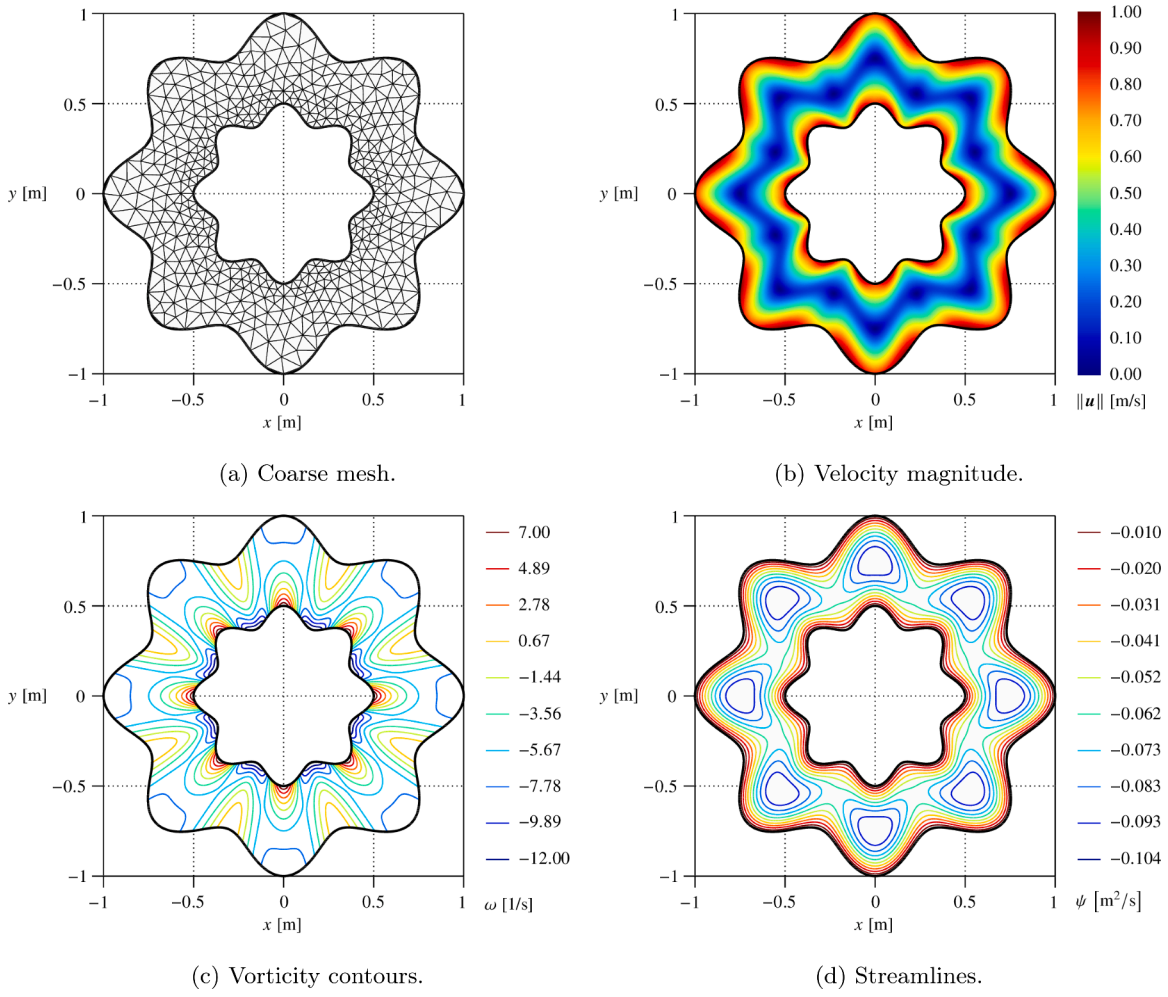
$$R(\theta) = \frac{ab}{\sqrt{(a\sin(\theta))^2 + (b\cos(\theta))^2}}, \tag{92}$$

Table 22
Errors obtained with the ROD method in the conventional semi-elliptical lid-driven cavity test case with $Re = 1$.

d	N_C	E_x [%]	E_y [%]	E_ω [%]	E_ψ [%]	N_{FP}	N_{GMRES}	T_{EXE} [s]	M_{USE} [Gb]
$(\psi,\omega)/\mathbb{P}_d$–$\mathbb{P}_{d+1}$									
1	1,044	8.72E+02	8.63E−01	1.26E+00	1.55E−01	7	230	0.07	0.02
	4,070	3.16E+02	2.05E−01	3.12E−01	5.80E−02	7	480	0.31	0.06
	16,398	2.57E+01	6.03E−02	8.78E−02	1.50E−02	8	1,070	3.17	0.22
	65,786	7.87E+00	1.94E−02	2.64E−02	3.73E−03	9	2,580	35.21	0.85
3	1,044	3.91E+00	6.78E−04	6.42E−04	7.13E−05	7	400	0.22	0.03
	4,070	6.37E−02	8.41E−05	8.79E−05	9.83E−06	8	810	0.90	0.10
	16,398	1.93E−03	3.22E−06	2.39E−06	4.43E−07	8	1,820	7.57	0.36
	65,786	1.53E−04	9.29E−08	1.44E−07	4.51E−09	9	5,870	106.31	1.35
5	1,044	1.13E−01	8.88E−05	2.31E−04	5.46E−05	7	440	0.69	0.06
	4,070	2.85E−02	9.75E−07	3.52E−06	1.28E−06	8	990	2.94	0.20
	16,398	1.69E−03	1.50E−07	6.64E−07	1.54E−07	9	2,280	19.84	0.72
	65,786	—	—	—	—	9	7,320	183.97	2.27
$(\boldsymbol{u}, p)$									
1	1,044	2.02E+02	1.16E+00	—	—	7	840	0.19	0.04
	4,070	9.18E+01	3.77E−01	—	—	7	2,090	2.09	0.13
	16,398	3.41E+01	7.74E−02	—	—	7	5,980	65.33	0.66
	65,786	6.85E+00	1.44E−02	—	—	8	18,160	1,763.87	3.59
3	1,044	2.90E+00	2.23E+00	—	—	6	730	0.30	0.05
	4,070	3.87E−01	1.12E+00	—	—	7	2,310	3.20	0.18
	16,398	1.75E−01	5.75E−01	—	—	7	6,910	88.17	0.85
	65,786	3.45E−02	2.68E−01	—	—	8	20,420	2,181.51	4.41
5	1,044	2.36E+01	1.96E+00	—	—	7	2,030	0.82	0.06
	4,070	6.09E+00	1.22E+00	—	—	7	2,430	6.38	0.30
	16,398	2.42E+00	6.38E−01	—	—	7	7,970	126.64	1.15
	65,786	4.73E−01	3.13E−01	—	—	8	25,130	2,987.98	5.90

Table 23
Errors obtained with the ROD method in the conventional semi-elliptical lid-driven cavity test case with $Re = 100$.

d	N_C	E_x [%]	E_y [%]	E_ω [%]	E_ψ [%]	N_{FP}	N_{GMRES}	T_{EXE} [s]	M_{USE} [Gb]
$(\psi,\omega)/\mathbb{P}_d$–$\mathbb{P}_{d+1}$									
1	1,044	5.62E+00	8.74E−01	1.45E+00	7.09E−02	24	1,130	0.18	0.02
	4,070	1.27E+00	2.20E−01	3.35E−01	5.83E−02	24	2,030	0.88	0.06
	16,398	6.22E−01	4.15E−02	7.93E−02	1.49E−02	24	3,560	9.55	0.22
	65,786	8.94E−02	1.32E−02	2.04E−02	4.39E−03	26	8,770	114.51	0.85
3	1,044	6.79E−02	1.10E−04	1.57E−03	1.22E−03	24	1,570	0.38	0.03
	4,070	1.35E−02	2.44E−05	1.17E−04	4.54E−05	24	2,720	1.98	0.11
	16,398	5.03E−04	4.12E−06	3.62E−06	5.81E−06	25	6,730	22.93	0.36
	65,786	5.45E−05	5.64E−07	2.57E−08	7.92E−07	26	15,980	275.02	1.35
5	1,044	8.02E−03	4.76E−05	1.48E−03	4.96E−04	23	1,971	0.81	0.05
	4,070	2.90E−03	5.25E−06	6.60E−05	2.44E−05	26	3,030	4.38	0.19
	16,398	8.81E−05	3.38E−07	1.13E−07	8.82E−07	26	10,101	49.43	0.60
	65,786	—	—	—	—	26	22,020	503.91	2.26
$(\boldsymbol{u}, p)$									
1	1,044	2.51E+00	1.13E+00	—	—	12	1,650	0.32	0.04
	4,070	1.23E+00	3.39E−01	—	—	12	3,980	3.79	0.14
	16,398	9.49E−01	1.43E−01	—	—	14	11,940	126.39	0.68
	65,786	3.45E−01	7.27E−02	—	—	14	31,701	3,029.64	3.58
3	1,044	1.82E+00	2.20E+00	—	—	12	1,670	0.51	0.05
	4,070	9.10E−01	1.11E+00	—	—	13	4,190	5.41	0.18
	16,398	4.69E−01	5.69E−01	—	—	14	12,700	156.03	0.85
	65,786	2.18E−01	2.70E−01	—	—	14	36,320	3,779.81	4.28
5	1,044	1.74E+00	2.31E+00	—	—	13	1,620	0.91	0.07
	4,070	9.37E−01	1.25E+00	—	—	13	4,570	9.29	0.27
	16,398	5.31E−01	6.32E−01	—	—	14	13,700	207.78	1.15
	65,786	2.62E−01	3.17E−01	—	—	14	40,610	4,633.66	5.53



(a) Coarse mesh.

(b) Velocity magnitude.

(c) Vorticity contours.

(d) Streamlines.

Fig. 7. Coarse mesh and analytical solution for the flow in a rose-shaped domain test case (for proper interpretation of the colour scale, the reader is referred to the electronic version of this article).

which is more practical for projecting the edge midpoints onto the physical boundary after finding the corresponding angular coordinate. The outward unit normal and tangential vectors on the semi-elliptical boundary are given as

$$\boldsymbol{n} = \frac{\left(\frac{2x}{a^2}, \frac{2y}{b^2}\right)^{\mathrm{T}}}{\sqrt{\left(\frac{2x}{a^2}\right)^2 + \left(\frac{2y}{b^2}\right)^2}} \quad \text{and} \quad \boldsymbol{t} = \frac{\left(-\frac{2y}{b^2}, \frac{2x}{a^2}\right)^{\mathrm{T}}}{\sqrt{\left(\frac{2x}{a^2}\right)^2 + \left(\frac{2y}{b^2}\right)^2}}, \tag{93}$$

respectively. The straight top and semi-elliptical bottom physical boundaries are Γ^{T} and Γ^{B}, respectively.

As in the classical lid-driven cavity test case in a rectangular domain, the top physical boundary is prescribed with a horizontal velocity, u, whereas the no-slip condition is imposed on the semi-elliptical physical boundary. Thus, both the boundary streamfunction and its normal derivative are homogeneous on the whole physical boundary, apart from the top physical boundary, where its normal derivative corresponds to the imposed horizontal velocity. For the vorticity boundary condition, it is noted that the normal curvature of the semi-ellipse does not need to be determined since the no-slip boundary condition is imposed on the semi-elliptical physical boundary.

As is well-known, when prescribing a constant horizontal velocity on the lid, pressure and velocity singularities arise at each corner due to the incompatible boundary velocity prescribed on both sides, negatively affecting the accuracy and convergence order of any approximate solution [57]. This phenomenon naturally arises in non-primitive variables, as both the vorticity and streamfunction are derived from the velocity. Thus, a regularised semi-elliptical lid-driven test case is firstly addressed, replacing the constant horizontal velocity with a polynomial function, given as

$$\boldsymbol{u}(x) = u\left((x-a)^4(x+a)^4/a^8, 0\right)^{\mathrm{T}}, \qquad \text{on } \Gamma^{\mathrm{T}}. \tag{94}$$

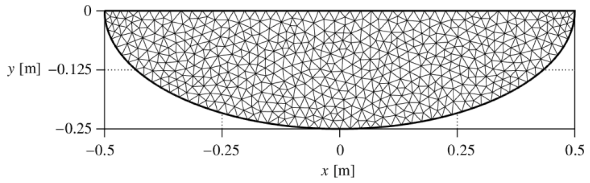
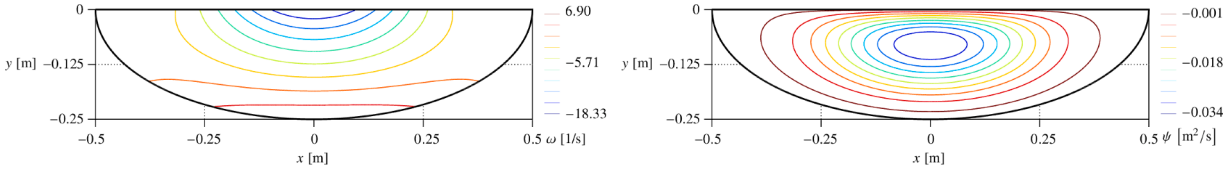


Fig. 8. Coarse mesh for the semi-elliptical lid-driven cavity test case.



(a) Vorticity contours. (b) Streamlines.

Fig. 9. Reference solutions for the regularised semi-elliptical lid-driven cavity test case with $Re = 1$ (For proper interpretation of the colour scale, the reader is referred to the electronic version of this article).

In both the regularised and conventional lid-driven cavity test cases, a primary vortex arises, and its centre coordinates, along with the corresponding vorticity and streamfunction values, are used for assessing the accuracy and efficiency of the proposed method. A reference solution is computed on a fine mesh to evaluate the errors E_x, E_y, E_ω, and E_ψ for the x-coordinate, y-coordinate, vorticity, and streamfunction, respectively, at the primary vortex centre. A gradient descent method, supplied with polynomial reconstructions for the streamfunction, is employed to determine the primary vortex centre up to an appropriate tolerance. Vorticity and streamfunction values at the vortex centre are also determined based on polynomial reconstructions.

Besides the numerical verification, a computational assessment is performed with the number of fixed point and GMRES iterations, N_{FP} and N_{GMRES}, respectively, to reach the corresponding stopping criteria reported. Moreover, the execution time (total time of the simulation) and the memory usage, T_{EXE} and M_{USE}, respectively, are also reported. The GMRES method is supplemented with a preconditioning technique following previous works [14,15] and restarts every 200 iterations. Relative residual tolerances of $\varepsilon_{\text{FP}} = 10^{-8}$ and $\varepsilon_{\text{GMRES}} = 10^{-10}$ are imposed for the fixed point and GMRES algorithms, respectively. For simplicity, the same residual tolerances are considered, regardless of the mesh characteristic size or the nominal convergence order of the numerical scheme. It is also important to note that incomplete convergence of the fixed point and GMRES algorithms, that is, admitting approximate solutions with a residual exceeding machine precision, undermines the strict conservativity property inherent in finite volume methods. However, upon inspection, the specified relative residual tolerances ensure that the calculated approximate solutions sufficiently minimise deviations from the conservation property, rendering any detrimental impact on the measured primary vortex centre, along with the corresponding vorticity and streamfunction values, negligible.

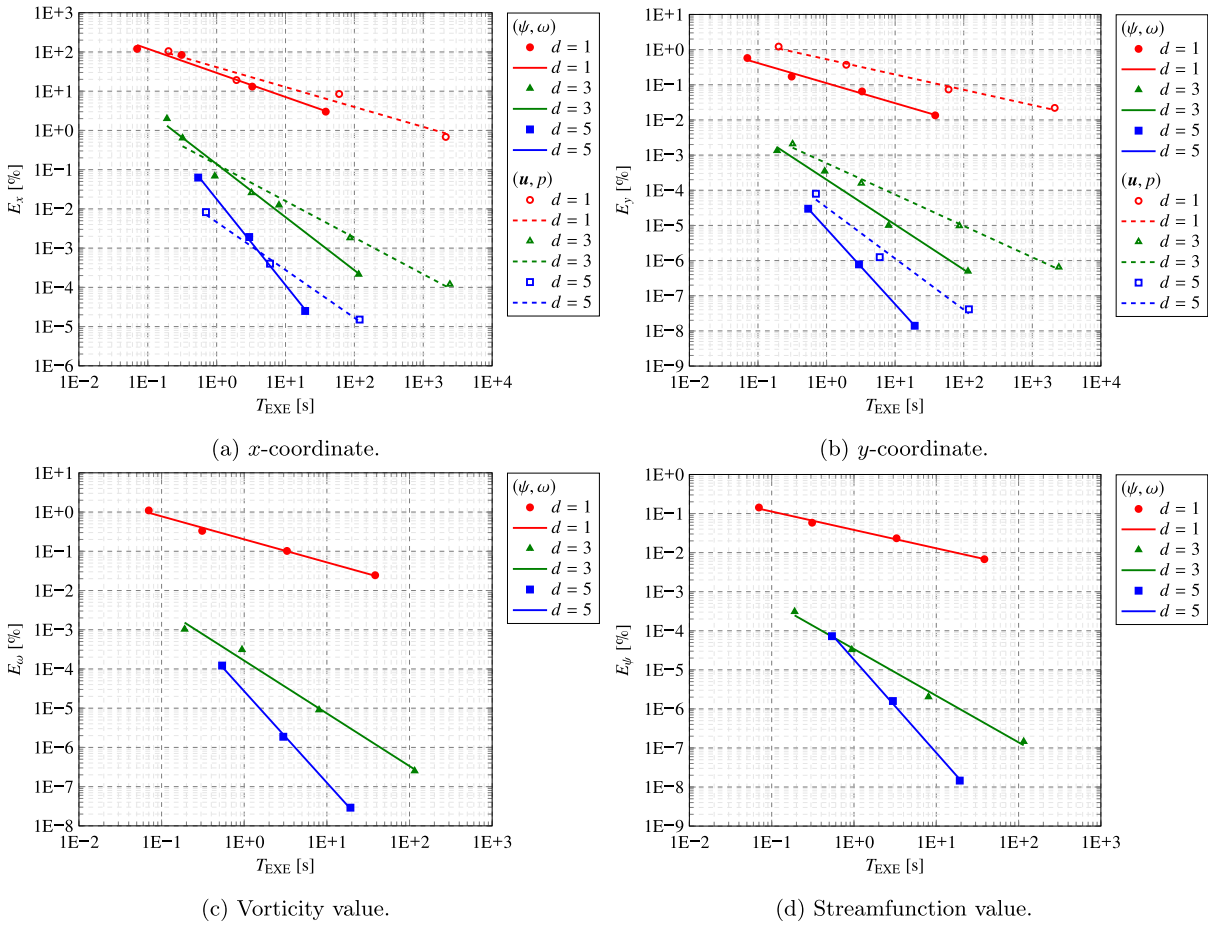
To assess the premises drawn in the introduction section that motivate the formulation of the incompressible Navier-Stokes equations based on vorticity and streamfunction, thereafter referred to as (ψ, ω) formulation, a comparison with the primitive variables formulation, thereafter referred to as $(\boldsymbol{u}, p)$ formulation, is carried out. A very high-order accurate finite volume scheme on staggered meshes [51] discretises the momentum and mass balance equations, and the ROD method is employed to preserve accuracy on the semi-elliptical physical boundary. Then, the same simulations are run to compute velocity and pressure, and a gradient descent method is employed to determine the primary vortex centre, where polynomial reconstructions for the velocity provide the search direction (perpendicular to the velocity vectors). Only the errors for the primary vortex x- and y-coordinates are evaluated in this comparison.

For the domain, the semi-axes have lengths of $a = 1/2$ m and $b = 1/4$ m. The fluid has a reference linear velocity magnitude of $u = 1$ m/s, a dynamic viscosity of either $\mu = 1$ kg/(m s) or $\mu = 0.01$ kg/(m s), and a density of $\rho = 1$ kg/m^3, corresponding to Reynolds numbers of $Re = 1$ and $Re = 100$, respectively. Successively finer uniform Delaunay triangular meshes are used to discretise the physical domain, and the simulations are carried out for the $\mathbb{P}_d$–$\mathbb{P}_{d+1}$ and $\mathbb{P}_d$–$\mathbb{P}_{d+2}$ methods with $d = 1, 3, 5$. A zero initial guess solution is set in all simulations. Fig. 8 illustrates the physical domain with a coarse mesh.

6.5.1. *Regularised semi-elliptical lid-driven cavity with* $Re = 1$

Fig. 9 illustrates the associated reference solutions for the vorticity and streamfunction computed on the finest mesh with $d = 5$. The errors for the primary vortex centre coordinates, vorticity, and streamfunction are provided in Table 19 when the naive method is employed for imposing the prescribed boundary conditions. Although there is a substantial accuracy improvement from $d = 1$ to $d = 3$, the error convergence rates under mesh refinement do not improve further from $d = 3$ to $d = 5$. Moreover, the coarser mesh with $d = 3$ achieves comparable accuracy to the finest mesh with $d = 1$. Additionally, both the $\mathbb{P}_d$–$\mathbb{P}_{d+1}$ and $\mathbb{P}_d$–$\mathbb{P}_{d+2}$ methods provide similar accuracy regardless of the polynomial degree.

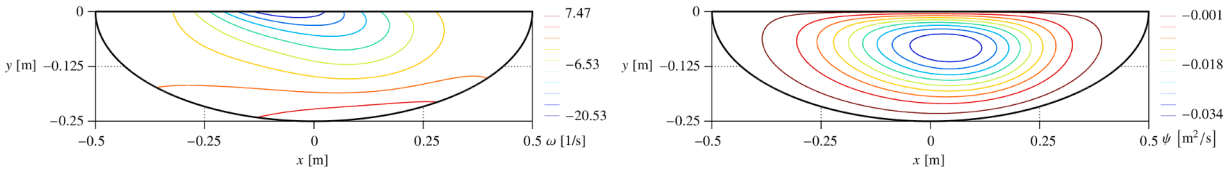
Table 20 reports the errors for the same parameters when the ROD method is employed instead. As observed, the error convergence rates under mesh refinement significantly improve with $d = 3, 5$ compared to the Naive method, whereas comparable accuracy and



(a) x-coordinate. (b) y-coordinate.

(c) Vorticity value. (d) Streamfunction value.

Fig. 10. Errors evolution as a function of execution time in the regularised semi-elliptical lid-driven cavity test case with $Re = 1$ (for proper interpretation of the colour scale, the reader is referred to the electronic version of this article).

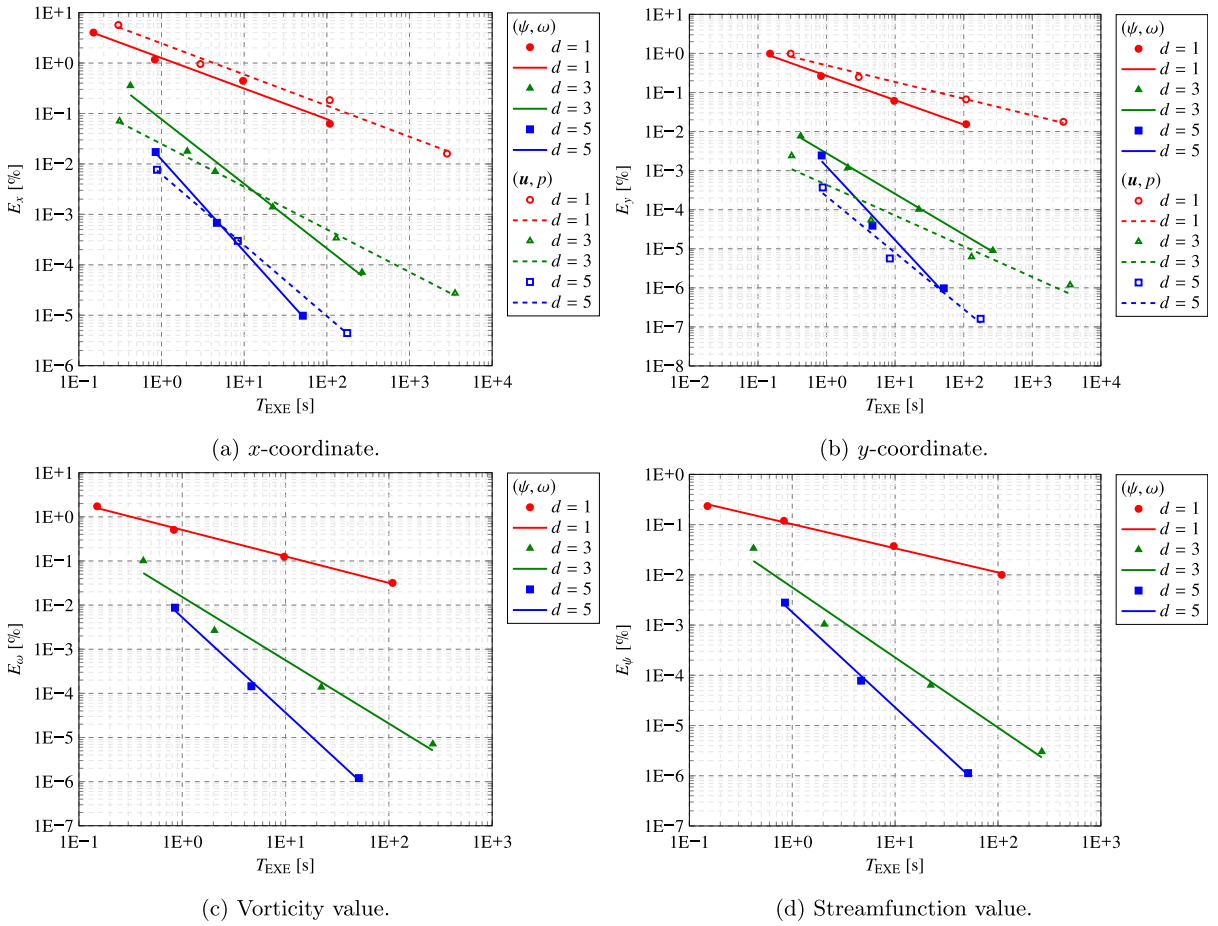


(a) Vorticity contours. (b) Streamlines.

Fig. 11. Reference solutions for the regularised semi-elliptical lid-driven cavity test case with $Re = 100$ (for proper interpretation of the colour scale, the reader is referred to the electronic version of this article).

convergence are obtained with $d = 1$. In this case, the coarser mesh with $d = 3$ achieves even smaller errors than the finest mesh with $d = 1$, whereas $d = 5$ further improves accuracy. In general, the $\mathbb{P}_d$–$\mathbb{P}_{d+2}$ method provides better accuracy than the $\mathbb{P}_d$–$\mathbb{P}_{d+1}$ method, although improvement ratios are inconsistent and vary significantly under mesh refinement and for each polynomial degree. For the $(\boldsymbol{u}, p)$ formulation, comparable accuracy to the (ψ, ω) formulation is achieved for the primary vortex centre coordinates, with improvement ratios below one order of magnitude relative to both methods.

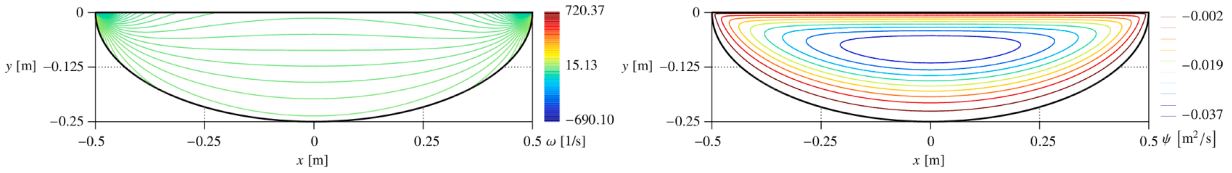
In terms of computational efficiency, both the $\mathbb{P}_d$–$\mathbb{P}_{d+1}$ and $\mathbb{P}_d$–$\mathbb{P}_{d+2}$ methods in the (ψ, ω) and $(\boldsymbol{u}, p)$ formulations require roughly the same number of fixed point iterations, which remains constant under mesh refinement and for higher polynomial degrees. However, the number of GMRES iterations increases with mesh refinement and higher polynomial degrees due to larger systems of linear equations and higher coefficients matrix condition numbers, respectively. For the same mesh, the $\mathbb{P}_d$–$\mathbb{P}_{d+2}$ method requires more GMRES iterations than the $\mathbb{P}_d$–$\mathbb{P}_{d+1}$ method since higher polynomial degrees are employed for streamfunction reconstructions in the former. Consequently, the $\mathbb{P}_d$–$\mathbb{P}_{d+2}$ method also requires more execution time and leaves a slightly higher memory footprint, hence it is less efficient than the $\mathbb{P}_d$–$\mathbb{P}_{d+1}$ method, since both provide comparable accuracy. For the $(\boldsymbol{u}, p)$ formulation, considerably



(a) x-coordinate. (b) y-coordinate.

(c) Vorticity value. (d) Streamfunction value.

Fig. 12. Errors evolution as a function of execution time in the regularised semi-elliptical lid-driven cavity test case with $Re = 100$ (for proper interpretation of the colour scale, the reader is referred to the electronic version of this article).

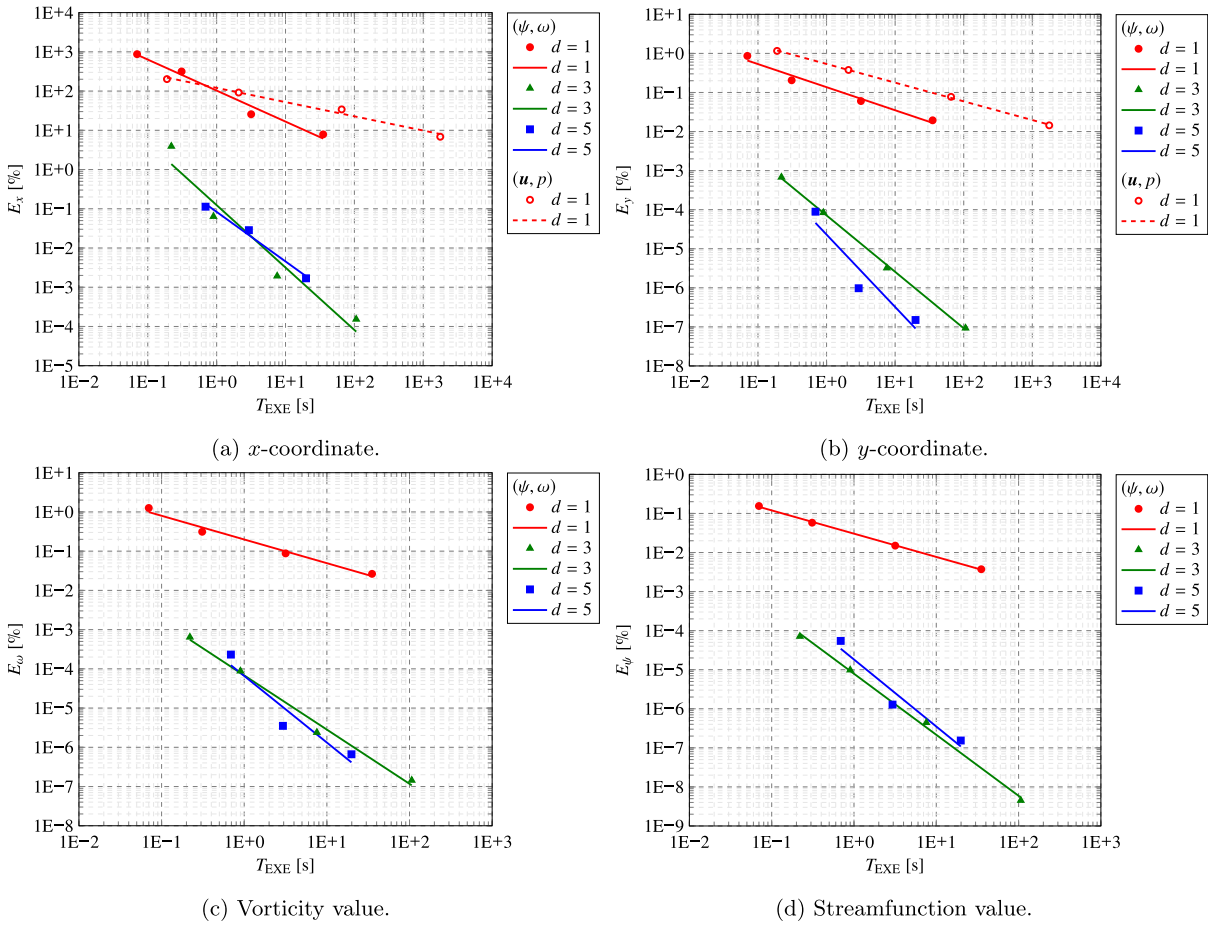


(a) Vorticity contours. (b) Streamlines.

Fig. 13. Reference solutions for the conventional semi-elliptical lid-driven cavity test case with $Re = 1$ (for proper interpretation of the colour scale, the reader is referred to the electronic version of this article).

more GMRES iterations are necessary than for the (ψ, ω) formulation with the same mesh, and the difference between polynomial degrees is less significant. Consequently, the $(\boldsymbol{u}, p)$ formulation requires considerably more execution time, hence it is less efficient than the (ψ, ω) formulation since both formulations provide comparable accuracy. It is also worth noting that the $(\boldsymbol{u}, p)$ formulation uses up to around four, three, and two times more memory than the (ψ, ω) formulation with $d = 1, 3, 5$, respectively, with the same mesh.

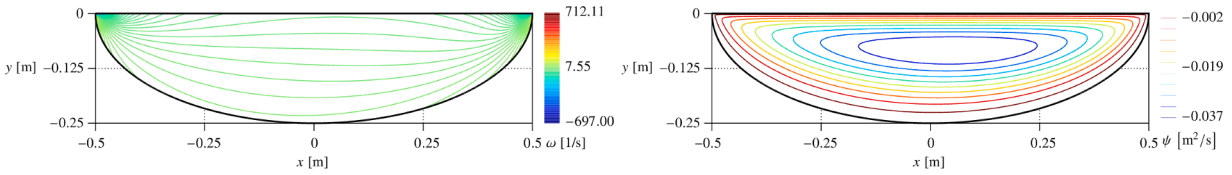
To provide better insight into the efficiency comparison, the error evolution as a function of execution time for the (ψ, ω) formulation with the $\mathbb{P}_d$–$\mathbb{P}_{d+1}$ method and the $(\boldsymbol{u}, p)$ formulation is illustrated in Fig. 10. Consider a target accuracy of $E_x = 1$ % and $E_y = 0.01$ % for the x- and y-coordinates of the primary vortex centre. In that case, for the x-coordinate, the $(\boldsymbol{u}, p)$ formulation with $d = 1$ requires $T_{EXE} \approx 1,000$ s in comparison to $T_{EXE} \approx 100$ s for the (ψ, ω) formulation with the same polynomial degree. Similarly, for the y-coordinate, the $(\boldsymbol{u}, p)$ formulation with $d = 1$ requires $T_{EXE} \approx 1,500$ s, whereas the (ψ, ω) formulation with the same polynomial degree requires $T_{EXE} \approx 50$ s. Thus, the (ψ, ω) formulation offers improvement ratios of one order of magnitude in the execution time. Moreover, higher polynomial degrees further enhance this efficiency, as the same target accuracy is achieved in less than one second



(a) x-coordinate. (b) y-coordinate.

(c) Vorticity value. (d) Streamfunction value.

Fig. 14. Errors evolution as a function of execution time in the conventional semi-elliptical lid-driven cavity test case with $Re = 1$ (for proper interpretation of the colour scale, the reader is referred to the electronic version of this article).



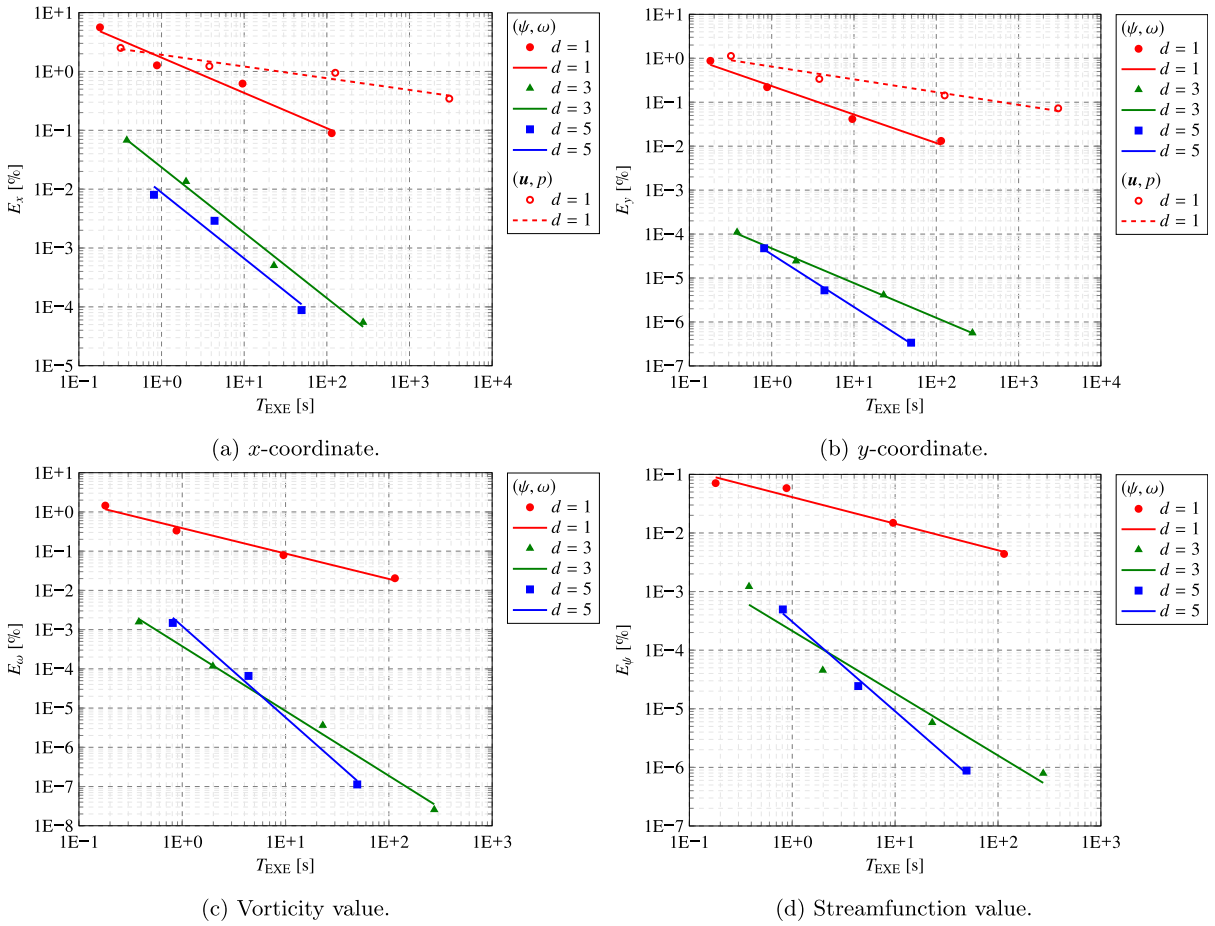
(a) Vorticity contours. (b) Streamlines.

Fig. 15. Reference solutions for the conventional semi-elliptical lid-driven cavity test case with $Re = 100$ (for proper interpretation of the colour scale, the reader is referred to the electronic version of this article).

with $d = 3$ in both formulations (comparable accuracy is obtained on coarser meshes), representing improvement ratios of three to four orders of magnitude in execution time compared to the $(\boldsymbol{u}, p)$ formulation with $d = 1$.

6.5.2. *Regularised semi-elliptical lid-driven cavity with* $Re = 100$

Fig. 11 illustrates the reference solutions for the vorticity and streamfunction computed on the finest mesh with $d = 5$. The errors for the primary vortex centre coordinates, vorticity, and streamfunction are provided in Table 21 when the ROD method is used to impose the prescribed boundary conditions. In terms of accuracy and convergence, the $\mathbb{P}_d$–$\mathbb{P}_{d+1}$ and $\mathbb{P}_d$–$\mathbb{P}_{d+2}$ methods in the (ψ, ω) formulation follow the same behaviour as for the previous test case with $Re = 1$, with the latter generally providing better accuracy. Conversely, while the $(\boldsymbol{u}, p)$ formulation in the previous test case provided accuracy improvement ratios always below an order of magnitude compared to the (ψ, ω) formulation, it appears that increasing the Reynolds numbers favours the former formulation. Indeed, improvement ratios up to around 20 and 10 are observed for the $\mathbb{P}_d$–$\mathbb{P}_{d+1}$ and $\mathbb{P}_d$–$\mathbb{P}_{d+2}$ methods, respectively, especially for high polynomial degrees. In terms of computational efficiency, the $\mathbb{P}_d$–$\mathbb{P}_{d+2}$ method requires more GMRES iterations than the



(a) x-coordinate. (b) y-coordinate.

(c) Vorticity value. (d) Streamfunction value.

Fig. 16. Errors evolution as a function of execution time in the conventional semi-elliptical lid-driven cavity test case with $Re = 100$ (for proper interpretation of the colour scale, the reader is referred to the electronic version of this article).

$\mathbb{P}_d$–$\mathbb{P}_{d+1}$ method with the same mesh, but the same number of fixed point iterations, following the same behaviour as previously with $Re = 1$. The main difference arises in the comparison with the $(\boldsymbol{u}, p)$ formulation, which requires fewer fixed point iterations than the (ψ, ω) formulation, favouring its efficiency.

To provide better insight into the efficiency comparison, the evolution of the error as a function of execution time for the (ψ, ω) formulation with the $\mathbb{P}_d$–$\mathbb{P}_{d+1}$ method and the $(\boldsymbol{u}, p)$ formulation is illustrated in Fig. 12. As observed, although the errors for the primary vortex centre coordinates decrease faster as a function of execution time in the (ψ, ω) formulation, the crossover point only occurs at significantly lower error levels with $d = 3, 5$, especially for the y-coordinate. Indeed, both the improved accuracy and the faster fixed point convergence of the $(\boldsymbol{u}, p)$ formulation compared to the (ψ, ω) formulation contribute to better computational performance with high polynomial degrees. In conclusion, the (ψ, ω) formulation can still enhance computational efficiency over the $(\boldsymbol{u}, p)$ formulation, but only when a very high accuracy for the primary vortex centre is required or with $d = 1$.

6.5.3. *Conventional semi-elliptical lid-driven cavity with* $Re = 1$

Fig. 13 illustrates the reference solutions for the vorticity and streamfunction computed on the finest mesh with $d = 5$. The errors for the primary vortex centre coordinates, vorticity, and streamfunction are provided in Table 22 when the ROD method is employed to impose the prescribed boundary conditions. In terms of accuracy and convergence, the $\mathbb{P}_d$–$\mathbb{P}_{d+1}$ method in the (ψ, ω) formulation follows essentially the same behaviour as for the regularised test case, except $d = 5$ that converges at almost the same rate as $d = 3$ and, therefore, offers little to no accuracy improvement. This behaviour is linked to the singularity points at the cavity corners, producing non-physical oscillations in the approximate solution, which exacerbate as the polynomial degree increases. Nevertheless, polynomial reconstructions with $d = 3$ do not appear to be affected, hence the same convergence behaviour as for the regularised test case is observed. For the $\mathbb{P}_d$–$\mathbb{P}_{d+2}$ method, omitted for compactness, the same consequence is observed with $d = 5$, whereas with $d = 1, 3$ the same observations as for the regularised test case apply. Conversely, the singularity issue has a more pronounced impact in the $(\boldsymbol{u}, p)$ formulation, with $d = 3, 5$ exhibiting the same convergence rates as $d = 1$ for the error in the primary vortex centre x-coordinate, and even lower convergence rates and higher errors for the primary vortex centre y-coordinate. As such, the numerical

accuracy benefit of using higher polynomial degrees is diminished, making the $(\boldsymbol{u}, p)$ formulation more computationally efficient with $d = 1$ than with $d = 3, 5$. Indeed, it appears that the (ψ, ω) formulation is more robust and proficient in withstanding the non-physical oscillations produced at the cavity corners than the $(\boldsymbol{u}, p)$ formulation.

To provide better insight into the efficiency comparison, the evolution of the error as a function of execution time for the (ψ, ω) formulation with the $\mathbb{P}_d$–$\mathbb{P}_{d+1}$ method and the $(\boldsymbol{u}, p)$ formulation is shown in Fig. 14. For the $(\boldsymbol{u}, p)$ formulation, only $d = 1$ is reported. As observed, both $d = 3, 5$ offer the same computational efficiency for the (ψ, ω) formulation, and are significantly superior to the $(\boldsymbol{u}, p)$ formulation with $d = 1$.

6.5.4. Conventional semi-elliptical lid-driven cavity with $Re = 100$

Fig. 15 illustrates the associated reference solutions for the vorticity and streamfunction computed on the finest mesh with $d = 5$. The errors for the primary vortex centre coordinates, vorticity, and streamfunction are provided in Table 23 when the ROD method is used to impose the prescribed boundary conditions. Fig. 16 provides a better insight into the efficiency comparison, showing errors evolution as a function of execution time for the (ψ, ω) formulation with the $\mathbb{P}_d$–$\mathbb{P}_{d+1}$ method and the $(\boldsymbol{u}, p)$ formulation with $d = 1$. This final test case supports the conclusions drawn from the previous test case with $Re = 1$, indicating that the $(\boldsymbol{u}, p)$ formulation is only effective with $d = 1$, whereas the (ψ, ω) formulation continues to provide the expected accuracy and convergence rates improvement with $d = 3$. Hence, it also becomes more computationally efficient, either with $d = 1$ and $d = 3$, when compared to the $(\boldsymbol{u}, p)$ formulation.

7. Conclusions

In the present work, a high-order accurate wall vorticity treatment is proposed to solve the steady-state incompressible Navier-Stokes equations in vorticity-based formulations in domains with arbitrary smooth curved boundaries. Firstly, the appropriate boundary conditions for the vorticity and streamfunction are derived from the prescribed boundary velocity. Multiply connected domains are also discussed since prescribing the appropriate streamfunction boundary conditions requires a more careful derivation. For the numerical scheme, the physical fluxes are approximated from polynomial reconstructions computed in the least-squares sense based on the approximate piecewise cell mean-values. Polygonal meshes are generated with conventional meshing algorithms to discretise the curved domain, leading to a geometrical mismatch between the curved physical boundaries and the associated piecewise linear representation. In this regard, the ROD method is employed to overcome the accuracy deterioration and achieve very high orders of convergence for the approximate solution. The method requires only a set of points on the physical boundary and avoids the difficulties of generating curved meshes and discretising on curved elements. The resulting system of linear equations is solved using a coupled approach, and a fixed point algorithm iteratively computes an approximate solution for the non-linear Navier-Stokes problem.

The accuracy, convergence order, robustness, and stability of the proposed method were assessed with a comprehensive numerical benchmark of test cases with known or manufactured solutions, with the main conclusions summarised below:

- The proposed method achieves the second-, fourth-, and sixth-orders convergence for the approximate vorticity and streamfunction with $d = 1, 3, 5$, respectively, except for the vorticity in the L^∞-norm, which generally achieves the first-, third-, and fifth-orders convergence for the same degrees.
- Employing a higher polynomial degree for the streamfunction reconstructions that impose the prescribed boundary conditions can further improve the accuracy and the convergence order of the method.
- The proposed method recovers the optimal convergence orders in domains with arbitrary smooth curved boundaries, while employing only piecewise linear approximations of the physical boundary.

The numerical and computational performance of the method in a more practical scenario without a known solution is also assessed with the semi-elliptical lid-driven cavity benchmark and compared to the same simulations performed with a primitive variables formulation, with the main conclusions summarised below:

- In the regularised configuration, the streamfunction-vorticity formulation provides comparable accuracy to the velocity-pressure formulation for low Reynolds numbers, but the latter is generally an order of magnitude more accurate with higher polynomial degrees for higher Reynolds numbers.
- In the conventional configuration, the velocity-pressure formulation fails to recover the optimal convergence rates for higher polynomial degrees, whereas the streamfunction-vorticity formulation withstands the non-physical oscillations produced at the cavity corners for $d = 1, 3$ without significant accuracy deterioration.
- The performance of the fixed point algorithm depends solely on the Reynolds number and, therefore, obtaining a very high-order of convergence requires roughly the same number of fixed point iterations as the low-order of convergence methods.
- The streamfunction-vorticity formulation typically requires more fixed point iterations than the velocity-pressure formulation for high Reynolds numbers, although it requires fewer GMRES iterations in total.

In conclusion, the streamfunction-vorticity formulation demonstrates a significant potential to solve the incompressible Navier-Stokes equations more efficiently than the velocity-pressure formulation for fluid flow problems at low Reynolds numbers and, particularly, problems with singularity points in the solution at any Reynolds number. This potential is further enhanced by the proposed method to achieve very high orders of convergence, even in domains with arbitrary curved boundaries, providing the same results as the low-order of convergence, with improvement ratios of several orders of magnitude in execution time.

Further developments of the proposed method include extensions to three-dimensional fluid flow problems. Deriving a streamfunction-vorticity formulation for three-dimensional incompressible Navier-Stokes equations requires both vorticity and streamfunction to have three-dimensional analogues. While the vorticity vector naturally arises from applying the curl operator to the momentum balance equation in three dimensions, the three-dimensional streamfunction, referred to as the vector potential, has caused considerable confusion regarding its physical meaning and numerical solution. Unlike its two-dimensional counterpart, determining suitable boundary conditions for the vector potential from those prescribed for the velocity is far from trivial. Additionally, there is a gauge freedom associated with the three-dimensional streamfunction-vorticity formulation that must be resolved by imposing a divergence-free condition on the vector potential. This issue can be alleviated by considering a scalar potential, together with the vector potential; however, a more convenient form of the three-dimensional incompressible Navier-Stokes equations in non-primitive variables is the vorticity-velocity formulation. This latter formulation is simpler and more practical than the vorticity-vector potential formulation concerning boundary conditions. Furthermore, integral conditions to ensure a single-valued vector potential are unnecessary when solving multiply connected problems, since there is no potential in the formulation. Vorticity-velocity formulations are also employed in two dimensions and consist of a scalar vorticity and two velocity components.

Regardless of the vorticity-based formulation applied for the three-dimensional case, the wall vorticity issue remains fundamentally unchanged. Therefore, the techniques and concepts presented in this work for the wall vorticity treatment, concerning its mathematical derivation on curved boundaries and numerical solution on polygonal meshes, can be readily extended, remaining agnostic to the spatial discretisation scheme employed.

Data availability

The data that supports the findings of this study is openly available in the CFDBench repository at https://github.com/ricardodpcosta/CFDBench.

Declaration of competing interest

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgements

R. Costa and J.M. Nóbrega acknowledge the financial support of FEDER – Fundo Europeu de Desenvolvimento Regional, through COMPETE 2020 – Programa Operational Fatores de Competitividade, and FCT – Foundation for Science and Technology, I.P., projects no. UIDB/05256/2020 (https://doi.org/10.54499/UIDB/05256/2020) and UIDP/05256/2020 (https://doi.org/10.54499/UIDP/05256/2020).

R. Costa also acknowledges the financial support of FCT – Foundation for Science and Technology, I.P., project no. 2023.09481.CEECIND/CP2841/CT0008 (https://doi.org/10.54499/2023.09481.CEECIND/CP2841/CT0008).

S. Clain acknowledges the financial support of FEDER – Fundo Europeu de Desenvolvimento Regional, through COMPETE 2020 – Programa Operational Fatores de Competitividade, and FCT – Foundation for Science and Technology, I.P., project no. UIDB/00324/2020 (https://doi.org/10.54499/UIDB/00324/2020).

G.J. Machado acknowledges the financial support of FCT – Foundation for Science and Technology, I.P., project no. UID/00013/2025 (https://doi.org/10.54499/UID/00013/2025).

S. Clain, G.J. Machado, and J.M. Nóbrega acknowledge the financial support of FCT – Foundation for Science and Technology, I.P., within the scope of the project HiPMoS4C with reference 2023.16854.ICDT2023.16854.ICDT (https://doi.org/10.54499/2023.16854.ICDT).

The authors also acknowledge the infrastructure support of Search-ON2 – Revitalization of HPC infrastructure of UMinho, project no. NORTE-07-0162-FEDER-000086, co-funded by the North Portugal Regional Operational Programme (ON.2 – O Novo Norte), under the National Strategic Reference Framework (NSRF), through the European Regional Development Fund (ERDF).

The authors thank the reviewers for their insightful comments, which have contributed to enhancing the quality of this article.

Appendix A. Derivation of the vorticity transport equation

Using the Cartesian coordinate system $\boldsymbol{x} := (x, y, z)$, the steady-state three-dimensional Navier-Stokes equations for a Newtonian fluid with constant dynamic viscosity $\mu > 0$ and constant density $\rho > 0$ consists of the momentum and mass balance equations, given as

$$(\boldsymbol{u} \cdot \nabla)\boldsymbol{u} - \nu \nabla^2 \boldsymbol{u} + \frac{1}{\rho} \nabla p = \boldsymbol{f}, \tag{A.1}$$

$$\nabla \cdot \boldsymbol{u} = 0, \tag{A.2}$$

respectively, where $\nu = \mu/\rho$ is the constant kinematic viscosity, $\boldsymbol{u} := \boldsymbol{u}(\boldsymbol{x}) := \left(u_x, u_y, u_z\right)$ with $u_x := u_x(\boldsymbol{x})$, $u_y := u_y(\boldsymbol{x})$, and $u_z := u_z(\boldsymbol{x})$ is the unknown velocity function, $p := p(\boldsymbol{x})$ is the unknown pressure function, while $\boldsymbol{f} := \boldsymbol{f}(\boldsymbol{x}) := \left(f_x, f_y, f_z\right)$ with $f_x := f_x(\boldsymbol{x})$,$f_y := f_y(\boldsymbol{x})$, and $f_z := f_z(\boldsymbol{x})$ is a given body force function per unit volume acting on the continuum, for instance, the gravitational force.

Taking the curl of the momentum balance equation (A.1), where each term can be rewritten as

$$\nabla \times ((\boldsymbol{u} \cdot \nabla)\boldsymbol{u}) = (\boldsymbol{\omega} \cdot \nabla)\boldsymbol{u} - (\boldsymbol{u} \cdot \nabla)\boldsymbol{\omega}, \qquad \nabla \times \left(\nu \nabla^2 \boldsymbol{u}\right) = \nu \nabla^2 (\nabla \times \boldsymbol{u}) = \nu \nabla^2 \boldsymbol{\omega}, \qquad \text{and} \qquad \nabla \times \nabla p = \boldsymbol{0}, \tag{A.3}$$

using the definition of the vorticity vector in three-dimensions, $\boldsymbol{\omega}(\boldsymbol{x}) = \nabla \times \boldsymbol{u}(\boldsymbol{x})$, yields the vorticity vector transport equation, given as

$$(\boldsymbol{u} \cdot \nabla)\boldsymbol{\omega} - (\boldsymbol{\omega} \cdot \nabla)\boldsymbol{u} - \nu \nabla^2 \boldsymbol{\omega} = \nabla \times \boldsymbol{f}, \tag{A.4}$$

where $(\boldsymbol{\omega}(\boldsymbol{x}) \cdot \nabla)\boldsymbol{u}(\boldsymbol{x})$ represents the vortex stretching, an important phenomenon in three-dimensional fluid flows, which vanishes for planar fluid flows since the vorticity vector is perpendicular to the plane of motion. Thus, in two-dimensional fluid flows using the Cartesian coordinate system $\boldsymbol{x} := (x, y)$, the vorticity vector transport equation reduces to the scalar equation

$$(\boldsymbol{u} \cdot \nabla)\omega - \nu \nabla^2 \omega \;=\; f, \tag{A.5}$$

for the scalar vorticity defined as $\omega(\boldsymbol{x}) = \partial u_y(\boldsymbol{x})/\partial x - \partial u_x(\boldsymbol{x})/\partial y$, where $f := f(\boldsymbol{x}) = \partial f_y(\boldsymbol{x})/\partial x - \partial f_x(\boldsymbol{x})/\partial y$, which can be rewritten in conservative form, using the identity $(\boldsymbol{u}(\boldsymbol{x}) \cdot \nabla)\omega(\boldsymbol{x}) = \nabla \cdot (\boldsymbol{u}(\boldsymbol{x})\omega(\boldsymbol{x})) - \omega(\boldsymbol{x})(\nabla \cdot \boldsymbol{u}(\boldsymbol{x}))$ and the incompressibility constraint $\nabla \cdot \boldsymbol{u}(\boldsymbol{x}) = 0$, as

$$\nabla \cdot (\boldsymbol{u}\omega) - \nu \nabla^2 \omega \;=\; f. \tag{A.6}$$

Appendix B. Derivation of the compatibility conditions

Following the notations and definitions for the local coordinate system introduced in Section 3.1, the equation for the tangential balance momentum along each physical boundary subset in terms of normal and tangential velocities is given as

$$\left(u_\eta \frac{\partial}{\partial \eta} + u_\xi \frac{\partial}{\partial \xi}\right) u_\xi - \nu \left(\frac{\partial^2}{\partial \eta^2} + \frac{\partial^2}{\partial \xi^2}\right) u_\xi + \frac{1}{\rho}\frac{\partial p}{\partial \xi} = f_\xi, \qquad \text{on } \Gamma^k, \quad k = 1, \dots, K, \tag{B.1}$$

where $f_\xi := f_\xi(\boldsymbol{\eta}) = \boldsymbol{f}(\boldsymbol{x}) \cdot \boldsymbol{t}(\boldsymbol{x})$ is the tangential body force function per unit area acting on the continuum. Using the identity $\partial u_\xi(\boldsymbol{x})/\partial \eta(\boldsymbol{x}) = \omega(\boldsymbol{x}) + \partial u_\eta(\boldsymbol{x})/\partial \xi$ from the vorticity definition in the local coordinate system, the convective term in Eq. (B.1) can be rewritten as

$$\left(u_\eta \frac{\partial}{\partial \eta} + u_\xi \frac{\partial}{\partial \xi}\right) u_\xi = u_\eta \omega + u_\eta \frac{\partial u_\eta}{\partial \xi} + u_\xi \frac{\partial u_\xi}{\partial \xi} = u_\eta \omega + \frac{\partial}{\partial \xi}\left(\frac{1}{2} u_\eta^2\right) + \frac{\partial}{\partial \xi}\left(\frac{1}{2} u_\xi^2\right). \tag{B.2}$$

Similarly, using the identity $\partial u_\xi(\boldsymbol{x})/\partial \xi = -\partial u_\eta(\boldsymbol{x})/\partial \eta$ from the mass conservation equation for an incompressible fluid in local coordinates, that is $\partial u_\eta(\boldsymbol{x})/\partial \eta + \partial u_\xi(\boldsymbol{x})/\partial \xi = 0$, the viscous term in Eq. (B.1) can be rewritten as

$$-\nu \left(\frac{\partial^2}{\partial \eta^2} + \frac{\partial^2}{\partial \xi^2}\right) u_\xi = -\nu \left(\frac{\partial^2 u_\xi}{\partial \eta^2} + \frac{\partial}{\partial \xi}\left(-\frac{\partial u_\eta}{\partial \eta}\right)\right) = -\nu \frac{\partial}{\partial \eta}\left(\frac{\partial u_\xi}{\partial \eta} - \frac{\partial u_\eta}{\partial \xi}\right) = -\nu \frac{\partial \omega}{\partial \eta}. \tag{B.3}$$

Combining the convective and viscous terms, the equation for the tangential balance momentum along each physical boundary subset can be rewritten as

$$u_\eta \omega + \frac{\partial}{\partial \xi}\left(\frac{1}{2} u_\eta^2\right) + \frac{\partial}{\partial \xi}\left(\frac{1}{2} u_\xi^2\right) - \nu \frac{\partial \omega}{\partial \eta} + \frac{1}{\rho}\frac{\partial p}{\partial \xi} = f_\xi, \qquad \text{on } \Gamma^k, \quad k = 1, \dots, K, \tag{B.4}$$

and transformed back into global coordinates as

$$(\boldsymbol{u} \cdot \boldsymbol{n})\omega + \nabla\left(\frac{1}{2}(\boldsymbol{u} \cdot \boldsymbol{n})^2\right) \cdot \boldsymbol{t} + \nabla\left(\frac{1}{2}(\boldsymbol{u} \cdot \boldsymbol{t})^2\right) \cdot \boldsymbol{t} - \nu \nabla \omega \cdot \boldsymbol{n} + \frac{1}{\rho}\nabla p \cdot \boldsymbol{n} = \boldsymbol{f} \cdot \boldsymbol{t}, \qquad \text{on } \Gamma^k, \quad k = 1, \dots, K. \tag{B.5}$$

Then, integrating the previous equation along the associated physical boundary subset and using the gradient theorem for the line integral of the tangential derivative along a closed path, yields

$$\oint_{\Gamma^k} (\boldsymbol{u}\omega - \nu \nabla \omega) \cdot \boldsymbol{n} \, \mathrm{d}s = \oint_{\Gamma^k} \boldsymbol{f} \cdot \boldsymbol{t} \, \mathrm{d}s, \quad k = 1, \dots, K. \tag{B.6}$$

Appendix C. Calculation of line curvature

Consider a twice-differentiable plane curve represented parametrically in Cartesian coordinates as $\gamma := \gamma(t) = (X, Y)$ with $X := X(t)$ and $Y := Y(t)$. Assuming that the derivative $\mathrm{d}\gamma(t)/\mathrm{d}t$ is defined, differentiable, and nowhere equal to the zero vector on the domain, then the normal curvature $\kappa := \kappa(t)$ as a function of the curve parameter can be determined as

$$\kappa = \frac{\left|X'Y'' - X''Y'\right|}{\left(\left(X'\right)^2 + \left(Y'\right)^2\right)^{3/2}}, \tag{C.1}$$

where $X' := X'(t)$ and $Y' := Y'(t)$ are the first derivatives and $X'' := X''(t)$ and $Y'' := Y''(t)$ are the second derivatives of $\gamma(t)$ with respect to the parameter t, that is

$$X' = \frac{\mathrm{d}X}{\mathrm{d}t}, \qquad Y' = \frac{\mathrm{d}Y}{\mathrm{d}t}, \qquad X'' = \frac{\mathrm{d}^2X}{\mathrm{d}t^2}, \qquad \text{and} \qquad Y'' = \frac{\mathrm{d}^2Y}{\mathrm{d}t^2}. \tag{C.2}$$

Consider a twice-differentiable plane curve represented parametrically in polar coordinates as $\gamma := \gamma(\theta) = (R, \theta)$ with $R := R(\theta)$, that is, the polar angle is the curve parameter and the radius of the curve is a function of the polar angle. This particular curve can be easily parametrised in Cartesian coordinates as $\gamma(\theta) = (X, Y)$ with $X := X(\theta) = R(\theta)\cos(\theta)$ and $Y := Y(\theta) = R(\theta)\sin(\theta)$. Thus, substituting X and Y in the normal curvature formula for a general parametrisation and making the necessary simplifications, the normal curvature $\kappa := \kappa(\theta)$ as a function of the polar angle is given as

$$\kappa = \frac{\left|R^2 + 2\left(R'\right)^2 - RR''\right|}{\left(R^2 + \left(R'\right)^2\right)^{3/2}}, \tag{C.3}$$

where $R' := R'(\theta)$ and $R'' := R''(\theta)$ are the first and second derivatives of $R(\theta)$ with respect to the polar angle, that is

$$R' = \frac{\mathrm{d}R}{\mathrm{d}\theta} \qquad \text{and} \qquad R'' = \frac{\mathrm{d}^2R}{\mathrm{d}\theta^2}. \tag{C.4}$$

References

[1] N.B. Balam, A. Gupta, A fourth-order accurate finite difference method to evaluate the true transient behaviour of natural convection flow in enclosures, Int. J. Numer. Methods Heat Fluid Flow 30 (3) (2020) 1233–1290.
[2] S. Wu, Y. Yan, Z. Chen, Z. Tian, A streamfunction-velocity formulation-based higher-order upwind compact difference method for solving unsteady natural convection, Phys. Fluids 37 (6) (2025) 063604.
[3] G. Kefayati, Implementing vorticity-velocity formulation in a finite difference lattice Boltzmann method for two-dimensional incompressible generalized Newtonian fluids, Phys. Fluids 36 (1) (2024) 013128.
[4] G. Kefayati, Integration of vorticity-velocity formulation in a lattice Boltzmann method for porous media, Phys. Fluids 36 (4) (2024) 043607.
[5] Y. Zhu, Y. Yang, F. Ren, General vorticity-streamfunction formulation for incompressible binary flow with arbitrary density ratio, Int. J. Numer. Methods Fluids 96 (4) (2024) 561–573.
[6] P. Yu, B. Wang, H. Ouyang, A high-order upwind compact difference scheme for solving the streamfunction-velocity formulation of the unsteady incompressible Navier-Stokes equations, Phys. Fluids 36 (6) (2024) 063601.
[7] M. Latini, O. Schilling, D.I. Meiron, Analysis of single-mode Richtmyer-Meshkov instability using high-order incompressible vorticity-streamfunction and shock-capturing simulations, Phys. Fluids 36 (2) (2024) 024115.
[8] B. Peng, X. Guo, Y. Zu, Z. Tian, A physics-preserving pure streamfunction formulation and high-order compact solver with high-resolution for three-dimensional steady incompressible flows, Phys. Fluids 35 (4) (2023) 043104.
[9] G. Kefayati, Three-dimensional vorticity-velocity formulation in a lattice Boltzmann method, Phys. Fluids 36 (9) (2024) 093121.
[10] J. Chen, P. Yu, Z.F. Tian, H. Ouyang, A high-order compact scheme for solving the 2D steady incompressible Navier-Stokes equations in general curvilinear coordinates, Int. J. Numer. Methods Fluids 92 (5) (2020) 456–477.
[11] B. Wang, P. Yu, X. Tong, H. Ouyang, A high-order pure streamfunction method in general curvilinear coordinates for unsteady incompressible viscous flow with complex geometry, Int. J. Numer. Methods Fluids 96 (12) (2024) 1960–1989.
[12] A. Thom, The flow past circular cylinders at low speeds, Proc. R. Soc. Lond. Ser. A, Contain. Pap. Math. Phys. Char. 141 (845) (1933) 651–669.
[13] L.C. Woods, A note on the numerical solution of fourth order differential equations, Aeronaut. Q 5 (4) (1954) 176–184.
[14] R. Costa, S. Clain, G.J. Machado, A sixth-order finite volume scheme for the steady-state incompressible Stokes equations on staggered unstructured meshes, J. Comput. Phys 349 (2017) 501–527.
[15] R. Costa, S. Clain, G.J. Machado, R. Loubère, A very high-order accurate staggered finite volume scheme for the stationary incompressible Navier-Stokes and Euler equations on unstructured meshes, J. Sci. Comput. 71 (2017) 1375–1411.
[16] J.E. Fromm, A method for computing nonsteady, incompressible, viscous fluid flows. No LA-2910. Los Alamos Scientific Lab, N. Mex, 1963.
[17] J.E. Fromm, F.H. Harlow, Numerical solution of the problem of vortex street development, Phys. Fluids 6 (7) (1963) 975–982.
[18] C.E. Pearson, A computational method for viscous flow problems, J. Fluid Mech. 21 (4) (1965) 611–622.
[19] S.A. Orszag, M. Israeli, Numerical simulation of viscous incompressible flows, Annu. Rev. Fluid Mech. 6 (1) (1974) 281–318.
[20] M.M. Gupta, R.P. Manohar, Boundary approximations and accuracy in viscous flow computations, J. Comput. Phys 31 (2) (1979) 265–288.
[21] R.T.S. Cheng, Numerical solution of the Navier-Stokes equations by the finite element method, Phys. Fluids 15 (12) (1972) 2098–2105.
[22] A.J. Baker, Finite element solution algorithm for viscous incompressible fluid dynamics, Int. J. Numer. Meth. Eng. 6 (1) (1973) 89–101.
[23] C. Taylor, P. Hood, A numerical solution of the Navier-Stokes equations using the finite element technique, Comput. Fluids 1 (1) (1973) 73–100.
[24] T.E. Tezduyar, R. Glowinski, J. Liou, Petrov-Galerkin methods on multiply connected domains for the vorticity-stream function formulation of the incompressible Navier-Stokes equations, Int. J. Numer. Meth. Fluids 8 (10) (1988) 1269–1290.
[25] G. Comini, M. Manzan, C. Nonino, Finite element solution of the streamfunction-vorticity equations for incompressible two-dimensional flows, Int. J. Numer. Meth. Fluids 19 (6) (1994) 513–525.
[26] C.F. Kettleborough, S.R. Husain, C. Prakash, Solution of fluid flow problems with the vorticity-streamfunction formulation and the control-volume-based finite-element method, Numer. Heat Transf. B 16 (1) (1989) 31–58.
[27] D. Elkaim, M. Reggio, R. Camarero, Numerical solution of reactive laminar flow by a control-volume based finite-element method and the vorticity-streamfunction formulation, Numer. Heat Transf. B 20 (2) (1991) 223–240.
[28] R.T. Davis, Numerical solution of the Navier-Stokes equations for symmetric laminar incompressible flow past a parabola, J. Fluid Mech. 51 (3) (1972) 417–433.
[29] A. Campion-Renson, M.J. Crochet, On the stream function-vorticity finite element solutions of Navier-Stokes equations, Int. J. Numer. Meth. Eng. 12 (12) (1978) 1809–1818.
[30] G. Dhatt, B.K. Fomo, C. Bourque, A $\psi - \omega$ finite element formulations for the Navier-Stokes equations, Int. J. Numer. Meth. Eng. 17 (2) (1981) 199–212.
[31] M.D. Gunzburger, J.S. Peterson, On the finite element approximation of the stream function-vorticity equations, NASA STI/Recon Technical Report N 84 (1984) 32758.
[32] M.F. Peeters, W.G. Habashi, E.G. Dueck, Finite element stream function-vorticity solutions of the incompressible Navier-Stokes equations, Int. J. Numer. Meth. Fluids 7 (1) (1987) 17–27.
[33] S.C.R. Dennis, J.D. Hudson, Compact h4 finite-difference approximations to operators of Navier-Stokes type, J. Comput. Phys 85 (2) (1989) 390–416.
[34] M.M. Gupta, High accuracy solutions of incompressible Navier-Stokes equations, J. Comput. Phys 93 (2) (1991) 343–359.
[35] M. Li, T. Tang, B. Fornberg, A compact fourth-order finite difference scheme for the steady incompressible Navier-Stokes equations, Int. J. Numer. Meth. Fluids 20 (10) (1995) 1137–1151.

[36] W.F. Spotz, G.F. Carey, High-order compact scheme for the steady stream-function vorticity equations, Int. J. Numer. Meth. Eng. 38 (20) (1995) 3497–3512.
[37] E. Weinan, J.G. Liu, Essentially compact schemes for unsteady viscous incompressible flows, J. Comput. Phys 126 (1) (1996) 122–138.
[38] M. Li, T. Tang, A compact fourth-order finite difference scheme for unsteady viscous incompressible flows, J. Sci. Comput. 16 (2001) 29–45.
[39] M. Ben-Artzi, I. Chorev, J.P. Croisille, D. Fishelov, A compact difference scheme for the biharmonic equation in planar irregular domains, SIAM J. Numer. Anal. 47 (4) (2009) 3087–3108.
[40] D. Calhoun, A Ccartesian grid method for solving the two-dimensional streamfunction-vorticity equations in irregular regions, J. Comput. Phys 176 (2) (2002) 231–275.
[41] Z. Li, C. Wang, A fast finite difference method for solving Navier-Stokes equations on irregular domains, Commun. Math. Sci. 1 (2003) 180–196.
[42] D. Fishelov, A new fourth-order compact scheme for the Navier-Stokes equations in irregular domains, Comput. Math. Appl. 74 (1) (2017) 6–25.
[43] M. Ben-Artzi, J.P. Croisille, D. Fishelov, A cartesian compact scheme for the Navier-Stokes equations in streamfunction formulation in irregular domains, J. Sci. Comput. 81 (3) (2019) 1386–1408.
[44] M.N. Linnick, H.F. Fasel, A high-order immersed interface method for simulating unsteady incompressible flows on irregular domains, J. Comput. Phys 204 (1) (2005) 157–192.
[45] R. Costa, S. Clain, R. Loubère, G.J. Machado, Very high-order accurate finite volume scheme on curved boundaries for the two-dimensional steady-state convection-diffusion equation with Dirichlet condition, Appl. Math. Model. 54 (2018) 752–767.
[46] R. Costa, J.M. Nóbrega, S. Clain, G.J. Machado, R. Loubère, Very high-order accurate finite volume scheme for the convection-diffusion equation with general boundary conditions on arbitrary curved boundaries, Int. J. Numer. Meth. Eng. 117 (2) (2019) 188–220.
[47] R. Costa, J.M. Nóbrega, S. Clain, G.J. Machado, Very high-order accurate polygonal mesh finite volume scheme for conjugate heat transfer problems with curved interfaces and imperfect contacts, Comput. Meth. Appl. Mech. Eng. 357 (2019) 112560.
[48] R. Costa, J.M. Nóbrega, S. Clain, G.J. Machado, Efficient very high-order accurate polyhedral mesh finite volume scheme for 3D conjugate heat transfer problems in curved domains, J. Comput. Phys 445 (2021) 110604.
[49] R. Costa, J.M. Nóbrega, S. Clain, G.J. Machado, High-order accurate conjugate heat transfer solutions with a finite volume method in anisotropic meshes with application in polymer processing, Int. J. Numer. Meth. Eng. 123 (4) (2022) 1146–1185.
[50] M. Santos, A. Araújo, S. Barbeiro, S. Clain, R. Costa, G.J. Machado, Very high-order accurate discontinuous Galerkin method for curved boundaries with polygonal meshes, J. Sci. Comput. 100 (2024) 66.
[51] R. Costa, S. Clain, G.J. Machado, J.M. Nóbrega, Very high-order accurate finite volume scheme for the steady-state incompressible Navier-Stokes equations with polygonal meshes on arbitrary curved boundaries, Comput. Meth. Appl. Mech. Eng. 396 (2022) 115064.
[52] R. Costa, S. Clain, G.J. Machado, J.M. Nóbrega, H. Beirão da Veiga, F. Crispo, Imposing general slip conditions on curved boundaries for 3D incompressible flows with a very high-order accurate finite volume scheme on polygonal meshes, Comput. Meth. Appl. Mech. Eng. 415 (2023) 116274.
[53] V. Girault, P.A. Raviart, Finite element approximation of the Navier-Stokes equations, 749 of Berlin, Springer, 1979.
[54] D.E. Wells, E.J. Krakiwsky, The method of least-squares, Technical Report, Department of Surveying Engineering, University of New Brunswick, Fredericton, 1971.
[55] D.P. Bertsekas, Constrained optimization and Lagrange multiplier methods, Academic Press, 1982.
[56] G.H. Wannier, A contribution to the hydrodynamics of lubrication, Q. Appl. Math 8 (1) (1950) 1–32.
[57] H.C. Kuhlmann, F. Romanò, The lid-driven cavity, in: A. Gelfgat (Ed.), Computational Modelling of Bifurcations and Instabilities in Fluid Dynamics, 2019, pp. 233–309.