Theor. Comput. Fluid Dyn. (2025) 39:36
https://doi.org/10.1007/s00162-025-00754-0

ORIGINAL ARTICLE

Kirill Goncharuk · Mukesh Kumar · Oz Oshri · Yuri Feldman

Implicit immersed boundary method integrated into the Vanka 'big box' smoother.

Received: 23 February 2025 / Accepted: 18 August 2025 / Published online: 8 September 2025

Abstract The current study introduces a novel fully coupled monolithic solver for the direct forcing immersed boundary method (IBM) in incompressible flows. The solver simultaneously integrates pressure, velocity, nonlinear convection terms, and Lagrangian forces into a unified framework, leveraging a modified big-box Vanka smoother extended with additional Lagrange multipliers arising from the IBM formulation. Central to the approach is the use of a Schur complement decomposition, which reduces the operator size by two-thirds while preserving both stability and accuracy. The solver's monolithic structure eliminates splitting errors and artificial pressure boundary conditions, common drawbacks of segregated methods. Additionally, the developed methodology enables high CFL numbers (up to 0.5), making it particularly effective for moving boundary simulations. Verification studies cover a broad set of benchmark problems, including both stationary and moving immersed bodies across a wide range of Reynolds numbers. These tests confirm that the solver achieves computational times comparable to existing semi-implicit methods while enhancing accuracy and stability. By addressing key challenges in high-fidelity incompressible flow simulations, the proposed method offers a robust and broadly applicable monolithic solver.

Keywords Immersed Boundary Method · Vanka Smoother · Fluid-Structure Interaction · Pressure-Velocity Coupling

Introduction

The immersed boundary method (IBM), introduced by Peskin [1], is a foundational computational tool for simulating fluid-structure interactions (FSI) in systems with complex geometrical interfaces. Its versatility and efficiency in handling boundary conditions on non-conforming meshes make it a widely used tool for applications encompassing biological fluid dynamics, particulate flows, and a wide spectrum of industrial applications including mixing processes, fluidized beds, membrane filtration systems, and wind turbine aerodynamics, to name but a few. For an in-depth review of IBM's formulations, implementations, and applications, refer to Verzicco et al. [2].

K. Goncharuk · M. Kumar · O. Oshri · Y. Feldman (✉)
Department of Mechanical Engineering, Ben-Gurion University of the Negev, David Ben Gurion Blvd, Beer-Sheva 8410501, Israel, Israel
E-mail: yurifeld@bgu.ac.il

K. Goncharuk
E-mail: kirillg@post.bgu.ac.il

M. Kumar
E-mail: mukesh@post.bgu.ac.il

O. Oshri
E-mail: oshrioz@bgu.ac.il

The immersed boundary method is not a self-contained solver; rather it operates in conjunction with an incompressible Navier–Stokes formulation, on which its accuracy and efficiency depend. In such formulations, pressure lacks an independent evolution equation and serves as a distributed Lagrange multiplier (DLM) to enforce the divergence-free constraint on the velocity field [3]. This gives rise to a saddle-point problem in which velocity and pressure are coupled through a differential constraint rather than through temporal dynamics. The resulting discrete system features a block structure with a zero diagonal in the pressure block, posing significant challenges for numerical solvers. To address this difficulty, one can consider decoupling strategies, such as the SIMPLE algorithm or projection-based methods. However, decoupling errors can lead to spurious pressure modes and mass conservation errors, especially at moderate-to-high Reynolds numbers.

Among IBM formulations, the direct forcing approach [4,5] has emerged as a computationally efficient method for enforcing no-slip boundary conditions. To implement this, Lagrangian forces are introduced to match the fluid velocity to that of the immersed surface, without assuming specific material properties. These Lagrangian forces serve as additional DLM, expanding the existing saddle-point structure. The resulting system simultaneously handles both the kinematic no-slip and the divergence-free constraints, which makes it considerably more challenging to solve.

Various strategies have been developed to solve the coupled saddle-point systems that arise in IBM formulations. Early methods used an explicit approach [6,7], where Lagrangian forces were computed from an intermediate velocity field that did not satisfy incompressibility. While simple, these methods required very small time steps to enforce the velocity constraint accurately. Semi-implicit formulations [8,9] improved stability by reformulating IBM into a projection-based saddle-point framework. However, they typically decoupled pressure and Lagrangian constraints from the velocity update. More recent methods based on the SIMPLE algorithm [10,11] improved coupling and efficiency but still involve operator splitting, which may reduce accuracy in highly unsteady or rapidly evolving flows.

Building on these developments, several recent studies have proposed fully monolithic approaches that couple all unknowns–velocity, pressure, and Lagrangian forces–within a single system [12–16]. Most of these methods, except for [15], solve the linear Stokes system using GMRES or direct solvers while treating the nonlinear convection terms explicitly. This strategy is adequate for low Reynolds number flows or smooth geometries but can compromise stability and accuracy in moderate to high Reynolds number regimes, particularly when dealing with complex or non-smooth geometries. Furthermore, the resulting linear systems are often large and ill-conditioned, requiring advanced and computationally demanding preconditioners to achieve convergence. For instance, [12,16] developed problem-specific preconditioners to manage these challenges. The method in [15] treats all nonlinear terms and constraints in a fully implicit manner, avoiding operator splitting. However, this comes with significant computational cost, making it practical primarily for applications such as slender-body dynamics where geometric stiffness dominates.

A key limitation of both explicit and semi-implicit IBM formulations is the decoupling of velocity from Lagrangian forces and pressure. This separation introduces consistency errors in the enforcement of kinematic and incompressibility constraints. This may result in fluid penetration, and spurious high-frequency oscillations of the Lagrangian forces. While acceptable in qualitative studies, such errors can undermine accuracy and stability in complex or unsteady simulations.

The purpose of the current study is twofold. First, we aim to develop an efficient solution strategy for the coupled saddle-point systems arising in the direct forcing IBM, by leveraging the proven effectiveness of the Vanka smoother for incompressible Navier–Stokes equations [17,18]. The Vanka smoother has demonstrated strong performance through its monolithic treatment of pressure–velocity coupling, thereby eliminating the need for artificial pressure boundary conditions. While Vanka-type smoothers have previously been explored in the context of implicit immersed boundary methods for the Stokes equations, for example as both solver and preconditioner in the continuous forcing formulation by [19], these approaches rely on force condensation and fully eliminate the Lagrangian variables from the system. In contrast, our contribution extends the Vanka framework to the direct forcing IBM by treating both Lagrangian forces and pressure as monolithic DLM. This fully coupled formulation enables direct enforcement of incompressibility and no-slip constraints at the solver level, without relying on elastic-body analogies or auxiliary constitutive laws.

Second, the monolithic solver developed in this study is well suited to serve as a computational framework for linear stability analysis of immersed boundary configurations. While time-stepping schemes are primarily used for transient simulations, they also provide a basis for computing steady states and analyzing their stability. In particular, they approximate the action of the Jacobian exponential, enabling matrix-free implementations of Newton–Krylov solvers and Arnoldi iterations [20]. Such methods are especially relevant for IBM, where linear stability studies remain limited to two-dimensional settings involving stationary [13,21,22], rotating

[23], or elastically mounted bodies [24–26]. This scarcity is largely due to the limitations of explicit and semi-implicit IBM formulations, which are generally incompatible with matrix-free iterative algorithms due to splitting errors and incomplete enforcement of velocity constraints. By directly coupling velocity, pressure, and Lagrangian forces within a unified solver, the present method overcomes these limitations and establishes a foundation for extending linear stability analysis to more complex immersed boundary problems in future studies.

Theoretical Background

Governing equations

The problem is governed by incompressible continuity and NS equations:

$$\begin{aligned} \nabla \cdot \mathbf{u} &= 0, \\ \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} &= -\nabla p + \frac{1}{Re}\nabla^2 \mathbf{u} + \mathbf{f}, \end{aligned} \tag{1}$$

where $\mathbf{u}$ is the velocity vector, p is the pressure field, $\mathbf{f}$ is the force density, and t corresponds to time. The equations were rendered dimensionless by utilizing L, U_0, L/U_0, ρU_0^2, and $\frac{\rho U_0^2}{L}$ to scale length, velocity, time, pressure, and force density, respectively, leading to a system governed by a single non-dimensional parameter, namely, the Reynolds number ($Re = \frac{LU_0}{\nu}$), where ν is the kinematic viscosity of the fluid.

According to the formalism of the IBM, the force density $\mathbf{f}$ entering the NS equations is regularized from the Lagrangian force density $\mathbf{F}$ determined around every point of the immersed body. This procedure, which constitutes the dynamic constraint, is implemented by introducing the regularization operator $R[\mathbf{F}]$. Closure of the overall system, as appears in Eq.(2), is achieved by introducing the kinematic constraint of no-slip, ensuring the same velocity values for the flow and for the surface of the immersed body,$\boldsymbol{U}_\Gamma$. This is accomplished by introducing an interpolation operator $I[\mathbf{u}]$ that interpolates the Eulerian velocities $\mathbf{u}$ on the surface of the immersed body. The two operators R and I are adjoint to each other and are implemented by using the discrete Dirac delta function. In the current study, we used the discrete Dirac delta functions developed in [27,28] for stationary immersed bodies, and in [29] for moving immersed bodies. The latter version of the discrete delta function aids in preventing spurious numerical oscillations of the solution at high CFL numbers.

$$\begin{aligned} \nabla \cdot \mathbf{u} &= 0, \\ \frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} &= -\nabla p + \frac{1}{Re}\nabla^2 \mathbf{u} + R[\mathbf{F}], \\ I[\mathbf{u}] &= \boldsymbol{U}_\Gamma. \end{aligned} \tag{2}$$

The key feature of the developed method is based on recognizing that both pressure and force density fields constitute DLMs, contributing to the enforcement of incompressibility and no-slip kinematic constraints, respectively. Consequently, we treat both fields as a single DLM vector. To avoid working with stiff operators implicitly coupling all the flow and the DLM fields, we utilize the Vanka smoother approach [17], originally developed for pressure-velocity coupling via the symmetrical coupled Gauss-Seidel (SCGS) operator. The novelty of the developed method lies in proposing a modified SCGS operator encompassing an extended DLM, consisting of pressure and Lagrangian force density fields, that are fully coupled.

We start by focusing on a typical segment of a staggered Eulerian grid with a fragment of the immersed body's surface superimposed on it, as shown in Fig.1. Note that the fragment of the surface of the immersed body is determined by a single point where the unknown Lagrangian force densities F_x and F_y are introduced. Next, we express all the values of the unknown fields at the current iteration $k+1$ in terms of the sum of the values taken from the previous iteration k and the corresponding correction fields:

$$\mathbf{u}^{k+1} = \mathbf{u}^k + \mathbf{u}'; \quad p^{k+1} = p^k + p'; \quad \mathbf{F}^{k+1} = \mathbf{F}^k + \mathbf{F}'. \tag{3}$$

Plugging the above expressions back into Eqs. (2) leads to an extended formulation of the Vanka smoother:

$$-\left(\frac{u'_e - u'_w}{\Delta x} + \frac{v'_n - v'_s}{\Delta y}\right) = \underbrace{\left(\frac{u_e - u_w}{\Delta x} + \frac{v_n - v_s}{\Delta y}\right)^k}_{\text{Continuity equation residual}}; \tag{4}$$

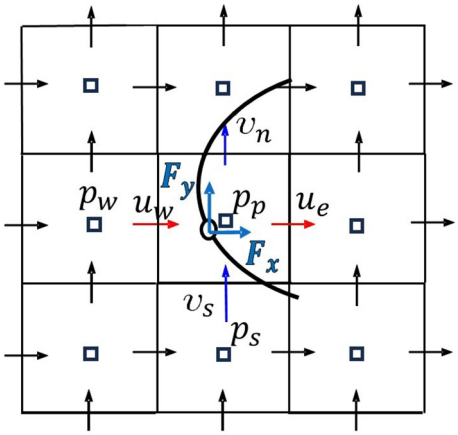


Fig. 1 A segment of a staggered Eulerian grid with a fragment of the immersed body's surface superimposed on it. The solid line represents the immersed body; symbols ∘ indicate Lagrangian points on the body with the corresponding body forces **F**. Symbols □ denote pressure field locations on the staggered grid, while → and ↑ represent horizontal and vertical velocity component locations, respectively

$$\left(\frac{3}{2}\Delta t + a_w \frac{1}{\mathrm{Re}}\right) u'_w + a_p p'_p - a_w p'_w + \sum_i R[F'_{x_i}]$$
$$= \underbrace{[-\partial_x p - \nabla\cdot(u\mathbf{u}) + \frac{1}{Re}\nabla^2 u - \frac{3}{2}\Delta t u_w]^k + (2u_w^n - \frac{1}{2}u_w^{n-1})\Delta t - \sum_i R[F^k_{x_i}]}_{\text{X-momentum equation residual}}; \tag{5}$$

$$\left(\frac{3}{2}\Delta t + a_s \frac{1}{\mathrm{Re}}\right) v'_s + a_p p'_p - a_s p'_s + \sum_i R[F'_{y_i}]$$
$$= \underbrace{[-\partial_y p - \nabla\cdot(v\mathbf{u}) + \frac{1}{Re}\nabla^2 v - \frac{3}{2}\Delta t v_s]^k + (2v_s^n - \frac{1}{2}v_s^{n-1})\Delta t - \sum_i R[F^k_{y_i}]}_{\text{Y-momentum equation residual}}; \tag{6}$$

$$\sum_i I[\mathbf{u}'_i] = \underbrace{\mathbf{U}^\Gamma - \sum_i I[\mathbf{u}^k_i]}_{\text{No-slip kinematic constraint residual}}, \tag{7}$$

where index i in the summation spans all the Lagrangian and Eulerian points for the regularization R and interpolation I operators, respectively. This provides an additional layer of coupling between the Lagrangian forces exerted by the body on the surrounding flow and the flow's velocity and pressure fields. Note that the above system of equations is not closed, as the momentum equations are formulated for u_w and v_s velocity components attached to the "west" and the "south" boundaries of a single finite volume (see Fig. 1), whereas the continuity equation includes the velocities attached to all the boundaries, i.e., "west", "east", "north" and "south". Closure is achieved by assembling the whole set of equations belonging to the entire computational domain in a 'big-box' manner, as proposed in [19,30]. Thereafter we exploit the special structure of the assembled operator to efficiently solve the system by employing the Schur complement approach and the direct open-source solver MUMPS [31,32], as detailed in the next section.

Numerical procedure

All the spatial terms in the governing equations were discretized using the standard finite volume method on a staggered grid [33], where the central difference was employed for approximating all the fluxes and shear stresses on the faces of the corresponding finite volumes. A second-order backward finite difference was utilized for the time derivative discretization.

To demonstrate the techniques that were further employed in the solution method that deals with the extended Vanka SCGS operator, we first use the original 'big-box' Vanka block operator as an example, without invoking the IBM formalism. We ensure that the currently built operator, given by $\begin{pmatrix} D & B^T \\ B & 0 \end{pmatrix}$, is symmetric by design. This symmetry is achieved by incorporating the discrete form of the continuity equation (matrix B) with a negative sign and by the diagonal structure of matrix D, with entries $a_{u,v}$ derived either from the conservation of linear momentum equations or from the boundary conditions. Overall, the system can be written as:

$$\begin{pmatrix} D & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u}' \\ \mathbf{p}' \end{pmatrix} = \begin{pmatrix} \mathbf{r_u} \\ \mathbf{r}_p \end{pmatrix}, \tag{8}$$

where the residuals $\mathbf{r_u}$ and $\mathbf{r}_p$ correspond to the right hand side of the momentum and continuity equations, respectively. The above matrix can be decomposed using the Schur complement approach as follows:

$$(BD^{-1}B^T)\mathbf{p}' = BD^{-1}\mathbf{r}_u - \mathbf{r}_p, \tag{9}$$

$$\mathbf{u}' = D^{-1}\mathbf{r_u} - D^{-1}B^T\mathbf{p}'. \tag{10}$$

To solve the pressure correction equation (Eq.(9)), the matrix $(BD^{-1}B^T)$ is first constructed and subsequently factorized using LU decomposition. Given that D is a diagonal matrix, its inverse D^{-1} can be directly computed as a diagonal matrix with reciprocal values along its main diagonal. The matrix $(BD^{-1}B^T)$ is then efficiently assembled through the utilization of standard MKL [34] utilities for sparse matrix-vector multiplications. This approach offers two principal advantages over the original formulation in Eq. (8): a significant reduction in system size (to one-third of the original) and the guarantee of symmetry and positive definiteness. Furthermore, the latter enables the MUMPS solver to leverage efficient decomposition into the LDL^T form for symmetric positive definite matrices. On average, the algorithm converges within $O(10)$ iterations per time step, with convergence behavior influenced by the CFL number, providing a mechanism for effective control.

For cases where $(BD^{-1}B^T)$ does not incorporate IBM forces, the Conjugate Gradient (CG) method can be applied efficiently without preconditioning, yielding solutions to machine precision faster than direct solvers. This efficiency is attributed to the relatively low condition number inherent to the system. However, the inclusion of IBM forces induces a substantial increase in the condition number, thereby rendering the CG method less effective in the absence of appropriate preconditioning. In such instances, the direct solver is preferred due to its robustness and reduced sensitivity to the condition number, ensuring both stability and accuracy in the solution process.

Embedding of the immersed boundary capability

In line with the idea of combining pressure and force density fields into a unified DLM field, we next determine the extended M and M^T operators as $M = \begin{pmatrix} B \\ I \end{pmatrix}$ and $M^T = \begin{pmatrix} B^T & R \end{pmatrix}$. The corresponding right hand side operators are extended accordingly as $\mathbf{m}' = \begin{pmatrix} \mathbf{p}' \\ \mathbf{f}' \end{pmatrix}$ and $\mathbf{r}_m = \begin{pmatrix} \mathbf{r}_p \\ \mathbf{r_f} \end{pmatrix}$, where $\mathbf{r_f}$ is a residual corresponding to the kinematic constraint of no-slip on the surface of the immersed body. By following the above conventions, the system of Eqs.(8) is reformulated to address the immersed boundary capability as follows:

$$\begin{pmatrix} D & M^T \\ M & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u}' \\ \mathbf{m}' \end{pmatrix} = \begin{pmatrix} \mathbf{r_u} \\ \mathbf{r_m} \end{pmatrix}. \tag{11}$$

Similarly to the configuration not invoking the immersed boundary capability, the above system is again decomposed by the Schur complement approach, resulting in:

$$(MD^{-1}M^T)\mathbf{m}' = MD^{-1}\mathbf{r_u} - \mathbf{r_m}. \tag{12}$$

$$\mathbf{u}' = D^{-1}(\mathbf{r_u} - B^T\mathbf{p}' - R\mathbf{f}'), \tag{13}$$

The operator $MD^{-1}M^T$ governing Eq. 12 is also symmetric and positive definite. Moreover, it can be built in a block matrix manner as:

$$(MD^{-1}M^T) = \begin{pmatrix} BD^{-1}B^T & BD^{-1}R \\ ID^{-1}B^T & ID^{-1}R \end{pmatrix}. \tag{14}$$

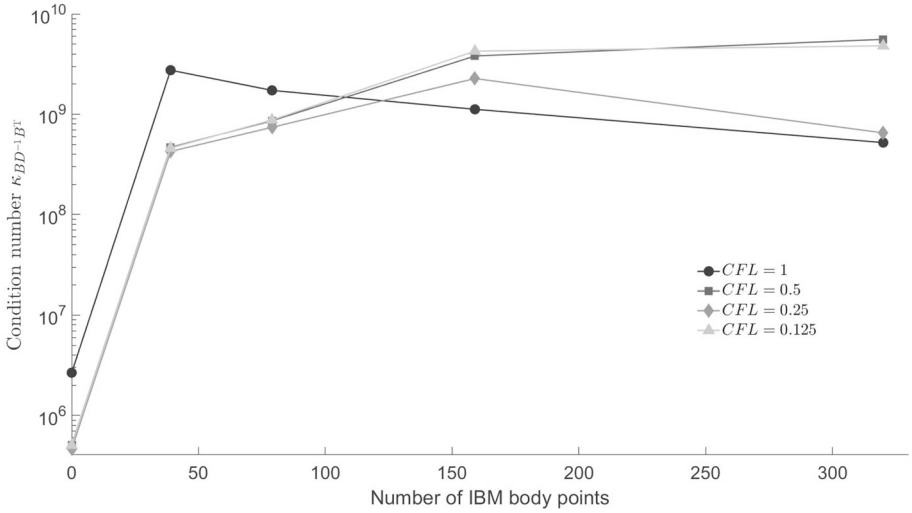


Fig. 2 Condition number $\kappa_{BD^{-1}B^{\mathrm{T}}}$ for lid-driven cavity problem simulated on 128^2 grid versus immersed-boundary point count for four CFL numbers. The points on the Y axis corresponds to the baseline case without an immersed boundary

In examining the above representation, it is clear that the biggest part is contributed by the original $(BD^{-1}B^T)$ block. As expected, the condition number of the extended formulation is significantly influenced by the operators I and R and can still be high. At the same time, the smoothing characteristics of the Vanka operator provide sustainable convergence, allowing for an accurate solution even for high values of the CFL number.

Condition number and convergence

We define the condition number of Schur complement as the 2-norm relation $\kappa_{BD^{-1}B^{\mathrm{T}}} = \|BD^{-1}B^{\mathrm{T}}\|_2 \, \|(BD^{-1}B^{\mathrm{T}})^{-1}\|_2$. For the Vanka 'big-box' solver, $\kappa_{BD^{-1}B^{\mathrm{T}}}$ increases by roughly three orders of magnitude once immersed-boundary (IB) forces are incorporated (Fig. 2).

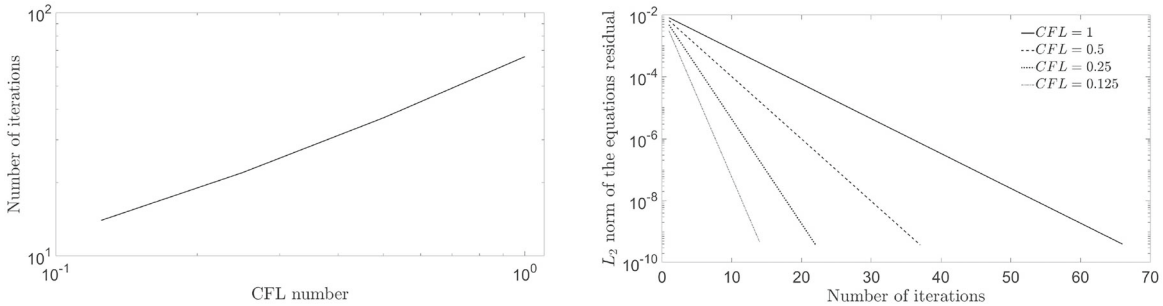
Increasing the amount of IB points from 39 to 320 was implemented by varying the diameter of the cylinder placed in the center of cavity from $d = 0.1$ to $d = 0.8$. Domain discretized by 128^2 grid and $\Delta s \approx \Delta x$ is kept constant. The change in number of Lagrangian markers does *not* exacerbate the ill-conditioning: all curves align near 10^9–10^{10}, irrespective of the CFL number. Without IBM, the Schur complement $BD^{-1}B^T$ is sensitive to CFL, yet this dependence vanishes once the velocity–force interaction is added. The condition number is further influenced by the problem physics and mesh resolution. More intricate bodies, such as a NACA airfoil, yield higher condition numbers than simpler configurations (e.g. a tandem cylinder system), and $\kappa_{BD^{-1}B^{\mathrm{T}}}$ growth as the grid is refined. Consequently, Krylov-subspace solvers become impracticable for this solver with IBM unless proper preconditioner is employed.

Even though the condition number plateaus, solver performance still remains governed by the CFL number. Figures 3a and 3b reveal an almost linear growth in iteration count with CFL. Thus, increasing the CFL slows convergence even when the underlying linear system is no further ill-conditioned.

Efficiency considerations

A significant advantage of the $MD^{-1}M^T$ operator structure, determined by Eq. (14), becomes evident when applying it to the simulation of moving boundary problems. In this case, the $(BD^{-1}B^T)$ block does not include the moving boundary characteristics and should, therefore, be computed only once at the beginning of the simulation. Consequently, only the remaining much smaller blocks, incorporating regularization or interpolation operators, need to be recalculated in each time step. This significantly boosts the overall computational efficiency of the developed method.

Another important advantage of the developed approach is its natural suitability for domain decomposition. While the current analysis uses the 'big-box' Vanka approach across the entire computational domain, the formulation can readily integrate domain decomposition techniques similar to those explored in [30]. This decomposition reduces operator stiffness within each patch of the decomposed domain. However, since domain



(a) Iterations required to reach $\|\mathbf{r}\|_2 < 10^{-9}$ as a function of CFL.

(b) L_2 residual norm versus iteration count for the four CFL numbers.

Fig. 3 Convergence behaviour of the Vanka big-box solver with an immersed boundary for lid-driven cavity problem simulated on 128^2 grid with a cylinder of $d = 0.5$ determined by 200 IB points. Smaller CFL numbers expedite convergence even though the condition number remains essentially unchanged (Fig. 2)

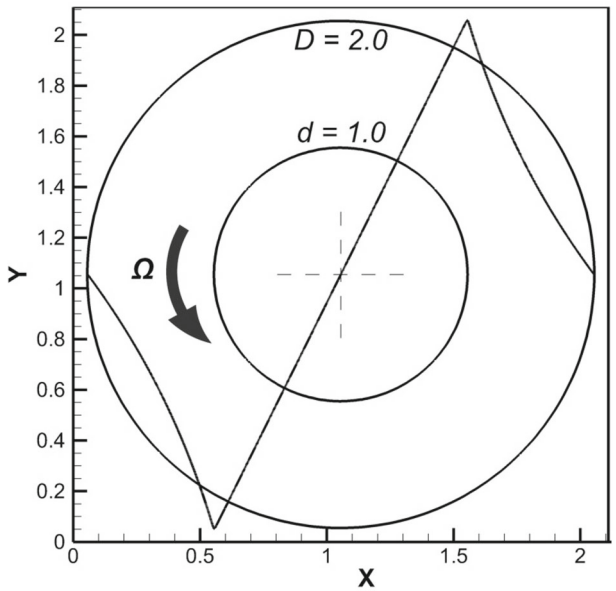


Fig. 4 Schematic of the computational setup and analytical solution for the u_θ velocity field. The setup consists of a rotating cylinder with diameter $d = 1.0$ placed inside a statiomary cylinder with diameter $D = 2.0$. Both cylinders placed within a periodic square domain of size $L = 2.11$. The inner cylinder rotates with an angular velocity $\Omega = 1 + \tanh((t - 0.2)/0.05)$, and the Reynolds number is set to $Re = 100$. The direction of rotation is indicated by the arrow. The black line corresponds to the analytical solution at steady state

decomposition typically increases the required iteration count, we opted for a single domain in these 2D configurations. Future research will explore domain decomposition for 3D flow simulations.

The convergence can be further accelerated using the multigrid method established in [17]. However, our numerical experiments showed minimal iteration reduction in 2D simulations. The current implementation achieves significant problem size reduction through the Schur complement approach with the 'big-box' Vanka solver without compromising solution convergence. Both domain decomposition and multigrid methods will be evaluated for future 3D configurations.

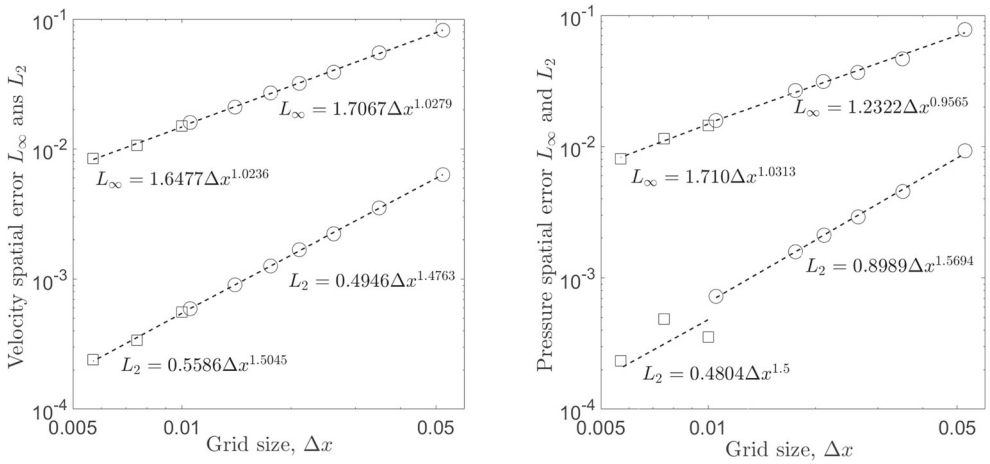


Fig. 5 Spatial error convergence for velocity and pressure in the rotating cylinder setup estimated in terms of L_∞ and L_2 norms of the errors. Squares and circles correspond to the currently obtained values and the corresponding data reported in [9], respectively. The dotted lines represent the corresponding least square power trends

Numerical results

Flow inside two concentric cylinders: spatial convergence and memory scaling

To investigate the efficiency of our method in terms of grid convergence and memory consumption, we conducted a comparison of the obtained results with those presented in [9] for the flow inside two concentric cylinders. This setup is most suitable as a test case for immersed boundary methods, as it has analytical solution to assess the spatial accuracy of numerical solvers.

The computational setup consists of two concentric cylinders: an inner rotating cylinder with prescribed angular velocity $\mathbf{\Omega}$ and a stationary outer cylinder (see Fig. 4). The Reynolds number was set to $Re = 100$, and the angular velocity of the inner cylinder was defined as $\mathbf{\Omega} = 1 + \tanh((t - 0.2)/0.05)$. The boundary conditions for the entire computational domain were set to be periodic, consistent with the configuration used in [9]. Spatial convergence was evaluated by comparing our results to an analytical solution for this setup in terms of L_1 and L_∞ norms of the difference between the numerical and the analytical solutions. For the sake of consistency in calculation of the pressure field, a Dirichlet point for pressure was placed precisely at (1.055, 1.055).

The difference norms plotted on a log-log scale (see Fig. 5) visualize the grid convergence rates, derived by fitting trendlines to the data points. The observed convergence rates matched the expected values reported in [9], verifying the accuracy of the currently developed method. For the L_∞ norm, the convergence rate was approximately 1.0 for both pressures and velocity fields, consistently with the reference results.

The present results can be meaningfully compared with those reported in [9] since periodic boundary conditions eliminate decoupling errors typical of semi-implicit IBM, enabling direct accuracy comparison with our fully coupled formulation. For non-periodic boundaries, our method should achieve superior accuracy by inherently avoiding these decoupling errors.

In addition to the spatial accuracy, we evaluate the memory consumption of the solver under varying grid resolutions. Despite the monolithic nature of the solver and the memory-intensive requirements typical of direct solvers using the MUMPS solver, we achieved reduced memory consumption due to the following factors:

- Only the BDB^T matrix requires LDL^T factorization, significantly reducing the matrix size.
- The BDB^T is symmetric positive definite matrix , enabling LDL^T decomposition without pivoting and requiring storage of only a single factorized L matrix and a main diagonal.

To optimize memory usage further, we examine the conjugate gradient (CG) method scaling. Using CG with the immersed boundary method requires symmetric preconditioning to enhance convergence, as previously discussed. The CG solver achieves approximately twofold reduction in memory consumption compared to MUMPS for the 2D case, while maintaining the same linear scaling. The results obtained for this benchmark confirm the first-order spatial accuracy of the L_∞ norm and demonstrate the linear scaling in memory con-

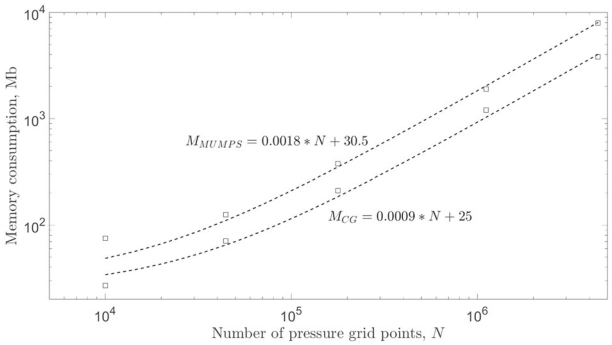


Fig. 6 Memory consumption as a function of grid resolution for the 2D simulation of flow between two concentric cylinders. Values are shown for both symmetric LDL^T factorization and conjugate gradient method solvers. The dotted lines represent the corresponding linear least square trends

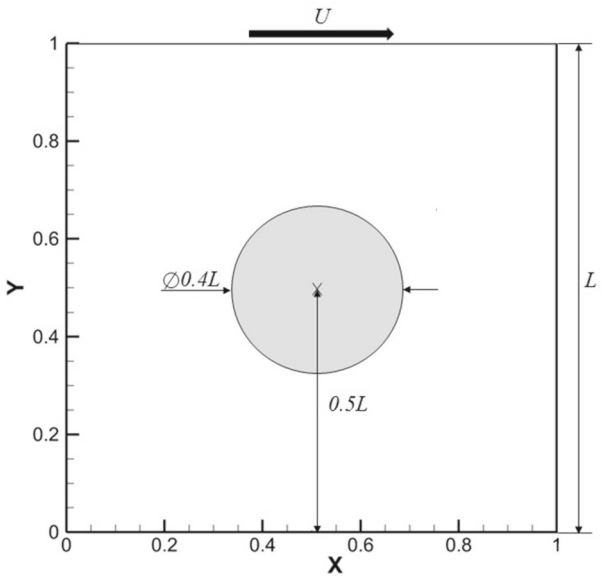


Fig. 7 Lid-driven cavity with a cylinder placed at the center: physical model

sumtion with the number of grid points in 2D simulations. Having verified the method's accuracy, we proceed to examine standard test cases.

Lid-driven cavity with a cylinder placed in a center

The configuration considered is the lid-driven cavity problem with a circular cylinder placed in the cavity center. The geometry consists of a square cavity of side L containing a cylinder of diameter $0.4L$ at its center. The top lid moves with constant velocity U in the x-direction, while all other boundaries remain stationary (see Figs. 6 and 7). No-slip boundary conditions were applied to all the cavity boundaries and to the cylinder surface. The steady-state criterion was established as $l_2 \leq 10^{-6}$, where l_2 is the second norm calculated for the relative deviation between two consecutive time steps for all the flow variables.

The results of the grid independence study performed for the above configuration are summarized in Table 1 in terms of the extreme values obtained for the x and y velocity components. The simulations were conducted for three different values of Re on three progressively refined grids, with grid resolution doubling each time. The relative deviations between the corresponding extreme values of all the velocity components obtained on the 256^2 and 512^2 grids did not exceed 0.5%, 1.6%, and 2.4% for $Re = 1000$, 5000, and 8000, respectively. Based on the above analysis, all further simulations of supercritical and subcritical flow regimes were conducted on a 512^2 grid.

Table 1 Results of a grid independence study obtained at three different Re values for the lid-driven cavity flow with a cylinder of diameter $D = 0.4$ placed at the cavity center

Grid	$Re = 1000$			$Re = 5000$			$Re = 8000$		
	$u_{x_{min}}$	$u_{y_{max}}$	$u_{y_{min}}$	$u_{x_{min}}$	$u_{y_{max}}$	$u_{y_{min}}$	$u_{x_{min}}$	$u_{y_{max}}$	$u_{y_{min}}$
128^2	−0.3371	0.3241	−0.4559	−0.4063	0.3924	−0.5148	−0.4027	0.3937	−0.5011
256^2	−0.3428	0.3291	−0.4634	−0.4289	0.4125	−0.5425	−0.4361	0.4242	−0.5438
512^2	−0.3445	0.3307	−0.4655	−0.4351	0.4181	−0.5509	−0.4462	0.4325	−0.5569

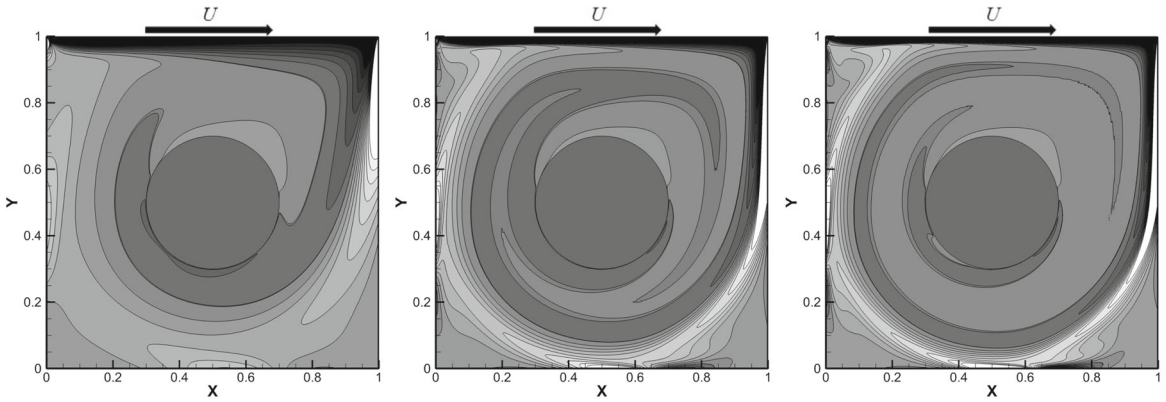


Fig. 8 Vorticity field inside the cavity with cylinder at $Re = 1000$ (left), $Re = 5000$ (center) and $Re = 8000$ (right). Contour levels were set from -12 to 12 in increments of 1.6 with an additional value of -2.35

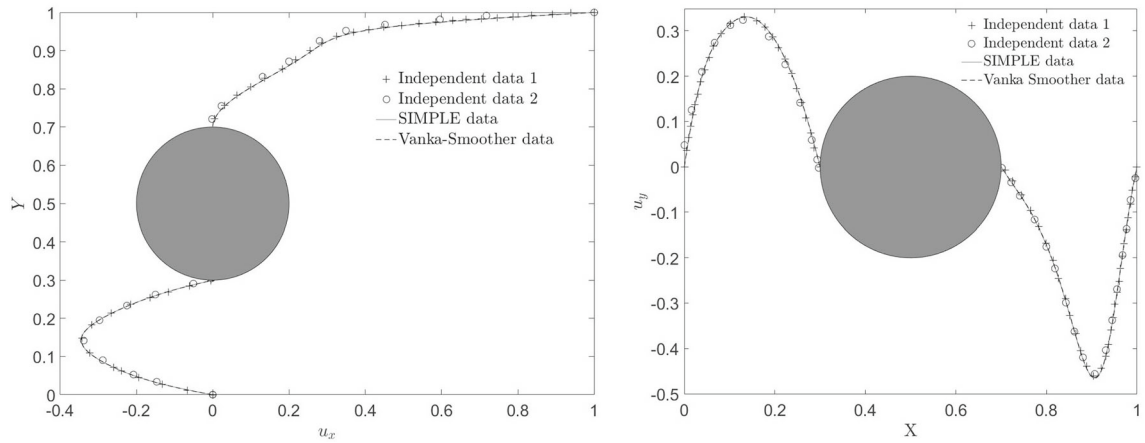


Fig. 9 Velocity components measured along the cavity center lines, $Re = 1000$: (a) v_x velocity component measured along the vertical center line; (b) v_y velocity component measured along the horizontal centre line. The solid line corresponds to the data obtained by SIMPLE algorithm[10], the dashed line represents currently obtained data, ∘ symbols correspond to the data reported in [35], and + symbols correspond to the data reported in [36]

To verify the obtained data, we compared vorticity fields calculated at $Re = 1000$, 5000, and 8000 with the corresponding results available in the literature. Our results presented in Fig. 8 are in good qualitative agreement with the corresponding data reported in [36] as well as with our previous study [10]. Additionally, we conducted a quantitative comparison of the u_x and u_y velocity components along the cavity's vertical and horizontal center lines at $Re = 1000$. The comparison was performed between our results and the values reported in [35,36]. Furthermore, we compared our results with those from our previous study that used the SIMPLE-based algorithm [10]. The comparison shown in Fig. 9 shows excellent agreement between our results and the data reported in all the previous studies.

We conclude this section by presenting results obtained for supercritical flow regimes at $Re = 8600$, 8700, 8800, and 8900. Table 2 summarises the amplitude values for the first three harmonics and the oscillation

Table 2 Oscillation frequencies f and amplitudes A at the control point $(0.9, 0.1)$ for slightly super-critical lid-driven-cavity flow. Present results are compared with the [10], for data that was available. A dash (—) indicates that no value was given in the earlier work

Re	Harmonic 1 ($n = 1$)				Harmonic 2 ($n = 2$)			Harmonic 3 ($n = 3$)	
	f	f Ref. [10]	A	ARef. [10]	f	f Ref. [10]	A	f	A
8900	0.500	–	$1.14 * 10^{-3}$	–	1.000	–	$7.84 * 10^{-5}$	1.500	$3.39 * 10^{-6}$
8800	0.500	0.489	$1.00 * 10^{-3}$	$6.96 * 10^{-4}$	1.000	0.978	$5.93 * 10^{-5}$	1.500	$2.89 * 10^{-6}$
8700	0.500	0.489	$8.40 * 10^{-4}$	$5.11 * 10^{-4}$	1.000	0.978	$3.78 * 10^{-5}$	1.500	$1.89 * 10^{-6}$
8600	0.500	0.491	$6.61 * 10^{-4}$	$6.23 * 10^{-5}$	1.000	0.982	$1.54 * 10^{-5}$	1.500	$1.24 * 10^{-6}$

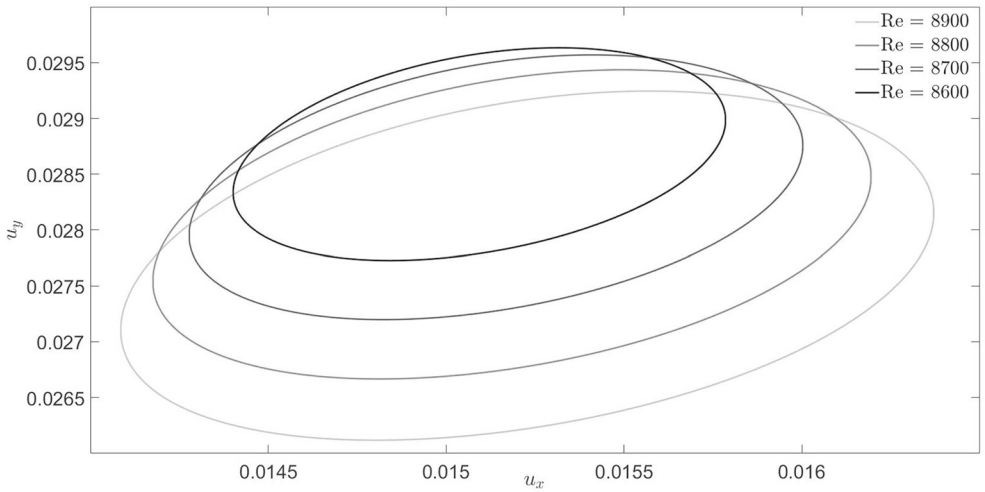


Fig. 10 Velocity trajectories measured at (0.9,0.1) in the supercritical mode for four different Re numbers: 8600, 8700, 8800 and 8900

frequencies f values obtained for different Re numbers. Remarkably, the frequency values of the main harmonic and its multipliers, typically showing up for non-linear dynamic systems, coincide with each other for the entire range of the Re values and are in agreement (not exceeding 2% frequency discrepancy) with the corresponding data previously reported by us in [10].

The accuracy of the obtained slightly supercritical results was further confirmed by examining the corresponding velocity trajectories at the point (0.9,0.1) (see Fig. 10). These trajectories establish clearly recognizable limit cycles, indicating that periodic stationary flow was reached for the entire range of Re values.

Three-dimensional lid-driven cavity with an embedded sphere

We next examine confined shear flow in a cubic lid–driven cavity of side L containing a sphere of diameter $D = 0.25L$ placed in the center of the cavity (Fig. 11). The top wall moves with a uniform velocity U in the x-direction, while the remaining walls and the sphere remain stationary. All quantities are scaled using the characteristic chamber length L and lid velocity U, yielding a characteristic time scale of L/U.

Immersed-boundary discretisation.

The spherical surface is represented by uniformly distributed Lagrangian markers generated with the non-iterative algorithm of Leopardi [37]. The marker spacing is chosen to be close to the local Eulerian grid spacing around the immersed surface to maximize interpolation accuracy.

Grid-independence study.

To establish a suitable mesh, simulations were carried out on N^3 Cartesian grids with $N = \{100, 150, 200\}$ at $Re = 1,\ 100,\ 400$. Figure 12 illustrates contours of the streamwise velocity u_x in the cross mid-plane obtained on the finest grid; the distributions remain symmetric with respect to the vertical centre-plane for all Reynolds numbers, and the boundary layer narrowing with Re matches the trends reported by [38].

Table 3 summarises the the grid independance study. The focus is on the extreme values of the centre-line velocities. For each Reynolds number we extract $u_x^{\max}(z),\ u_x^{\min}(z)$ along the vertical centre-line, and

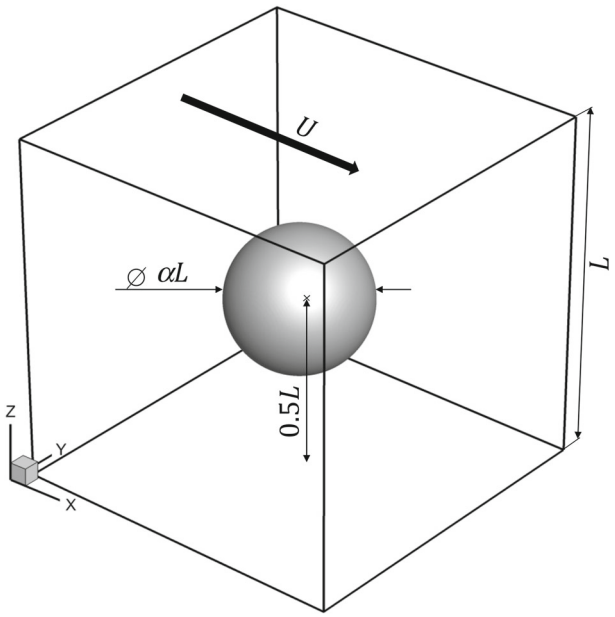


Fig. 11 Cubic lid-driven cavity with a sphere placed in the center of cavity: physical model

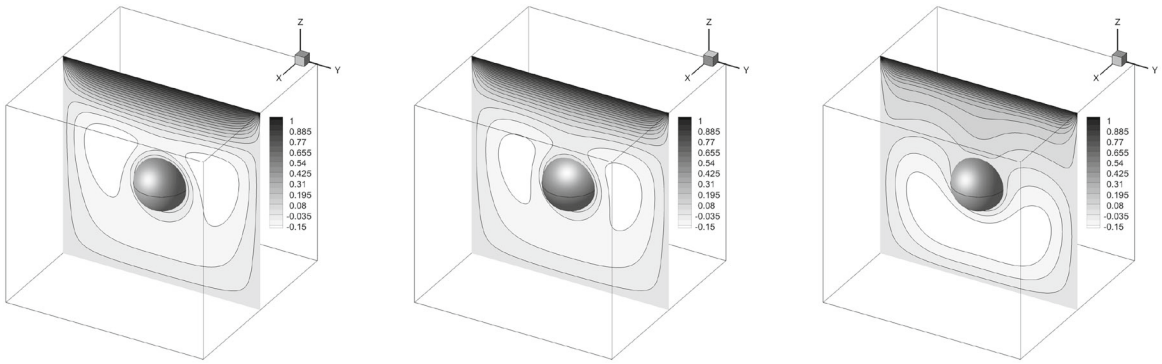


Fig. 12 3D lid driven cavity contours of the streamwise velocity u_x in the mid-plane for obtained for $Re = 1$ (left), $Re = 100$ (center) and $Re = 400$ (right)

$u_z^{\max}(x)$, $u_z^{\min}(x)$ along the horizontal centre-line, together with the coordinates at which the extrema occur. Table 3 lists these four quantities obtained on the grids 100^3, 150^3, and 200^3. Convergence is deemed satisfactory when the change between the two finest meshes falls below 1%.

Verification study.

Following the performed grid independence study, velocity profiles computed on the 200^3 mesh were compared with the data from [38] and with the numerical data from our previous research [10]. Table 4 reports the comparison at twelve test locations, while we define the relative error $\varepsilon_{\text{rel}} = |\phi_{\text{present}} - \phi_{\text{ref}}|/|\phi_{\text{ref}}|$. For the majority of points the error stays below 5%; the largest deviations (up to 10%) occur where the reference velocity is close to zero, consistent with earlier studies.

In addition to the velocity profiles, we report the drag (C_D), lift (C_L), and spanwise (C_S) force coefficients for the three Reynolds numbers considered. In our formulation, the immersed-boundary method computes the regularized force density $\mathbf{f}_k$ at each of the N_b Lagrangian markers. This force density is distributed within a narrow shell of one Eulerian cell width surrounding each Lagrangian point, as described by the integration procedure in Eqs. (16b) of the Appendix. The total hydrodynamic load on the sphere is then approximated by

Table 3 Grid-independence study: extreme centre-line velocities and the coordinates at which they occur

	100^3		150^3		200^3	
	z or x	u_x or u_z	z or x	u_x or u_z	z or x	u_x or u_z
$Re = 1$						
$u_x^{\max}$	1.0	1.0	1.0	1.0	1.0	1.0
$u_x^{\min}$	0.235	−0.1168	0.230	−0.1202	0.232	−0.1159
$u_z^{\max}$	0.195	0.1755	0.197	0.1751	0.197	0.1743
$u_z^{\min}$	0.805	−0.1759	0.803	−0.1793	0.803	−0.1748
$Re = 100$						
$u_x^{\max}$	1.0	1.0	1.0	1.0	1.0	1.0
$u_x^{\min}$	0.235	−0.1234	0.23	−0.1248	0.233	−0.1227
$u_z^{\max}$	0.195	0.1576	0.197	0.1588	0.198	0.1562
$u_z^{\min}$	0.815	−0.215	0.817	−0.219	0.818	−0.216
$Re = 400$						
$u_x^{\max}$	1.0	1.0	1.0	1.0	1.0	1.0
$u_x^{\min}$	0.225	−0.1915	0.23	−0.1958	0.228	−0.1904
$u_z^{\max}$	0.165	0.1880	0.17	0.1915	0.168	0.1866
$u_z^{\min}$	0.855	−0.322	0.857	−0.328	0.857	−0.320

Table 4 Centre-line streamwise velocity $u_x(z)$ at twelve locations for three Reynolds numbers. Present results are compared with[10,38]

	$Re = 1$			$Re = 100$			$Re = 400$		
z	[38]	[10]	Present	[38]	[10]	Present	[38]	[10]	Present
0.95	0.713	0.678	0.685	0.657	0.628	0.634	0.525	0.464	0.464
0.90	0.423	0.402	0.414	0.352	0.340	0.347	0.246	0.227	0.223
0.85	0.1932	0.1708	0.1956	0.1423	0.1467	0.1556	0.1665	0.1574	0.1491
0.80	0.0214	0.00838	0.0340	0.0257	0.0241	0.0356	0.1360	0.1254	0.1175
0.75	−0.0828	−0.0866	−0.0701	−0.0349	−0.0441	−0.0362	0.1079	0.0998	0.0926
0.70	−0.1317	−0.1292	−0.1134	−0.0726	−0.0760	−0.0679	0.0802	0.0704	0.0655
0.30	−0.1053	−0.1028	−0.0981	−0.1125	−0.1120	−0.1046	−0.1842	−0.1779	−0.1557
0.25	−0.1182	−0.1199	−0.1147	−0.1276	−0.1308	−0.1217	−0.218	−0.217	−0.1876
0.20	−0.1168	−0.1177	−0.1135	−0.1235	−0.1282	−0.1196	−0.214	−0.218	−0.1862
0.15	−0.1027	−0.1040	−0.1006	−0.1076	−0.1130	−0.1055	−0.1805	−0.1926	−0.1623
0.10	−0.0791	−0.0807	−0.0780	−0.0820	−0.0874	−0.0818	−0.1345	−0.1488	−0.1234
0.05	−0.0461	−0.0470	−0.0453	−0.0465	−0.0510	−0.0476	−0.0739	−0.086	−0.0713

summing the contributions over all markers:

$$\mathbf{F} \approx \sum_{k=1}^{N_b} \mathbf{f}_k \, \Delta V_k, \qquad \Delta V_k \approx (\Delta s)^3,$$

where Δs is the uniform marker spacing along the surface.

Using the sphere diameter $D = 0.25L$ scaled by the reference length L, the drag, lift and spanwise coefficients become

$$C_D = \frac{F_x}{\frac{1}{2}\rho U_\infty^2 D}, \quad C_L = \frac{F_y}{\frac{1}{2}\rho U_\infty^2 D}, \quad C_S = \frac{F_z}{\frac{1}{2}\rho U_\infty^2 D}.$$

With the present non-dimensionalisation $C_D = 8\,F_x$, $C_L = 8\,F_y$, $C_S = 8\,F_z$. The latter coefficient is 0 because of the symmetry of the flow thus effectively establishing an achieved numerical 0 for the current force calculations.

Flow around a stationary cylinder

The incident flow around a stationary circular cylinder is considered. The computational domain contains a cylinder of diameter $D = 1$ placed at the center of a square domain of dimensions $(60D, 60D)$. These dimensions of the computational domain (see Fig. 13) were chosen to ensure that the boundary conditions

Table 5 Drag (C_D), lift (C_L), and spanwise (C_S) coefficients for the embedded sphere at three Reynolds numbers

Coefficient	$Re = 1$	$Re = 100$	$Re = 400$
C_D	-7.11	$-6.48 \cdot 10^{-2}$	$-7.31 \cdot 10^{-3}$
C_L	$2.90 \cdot 10^{-2}$	$6.53 \cdot 10^{-2}$	$1.200 \cdot 10^{-2}$
C_S	$5.00 \cdot 10^{-6}$	$4.88 \cdot 10^{-6}$	$-2.01 \cdot 10^{-5}$

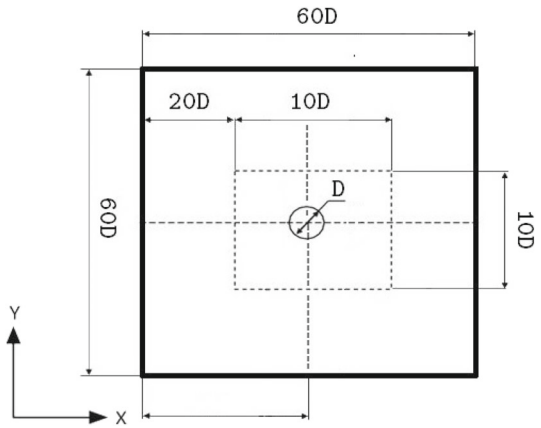


Fig. 13 Schematic representation of the geometrical model and characteristics of the computational domain for a single cylinder within the incident flow

did not influence the flow near the cylinder. To address the challenges posed by spatial discretization in a large domain, a non-uniform grid was used in both the x and y directions. The cylinder was discretized with an arc length step of $\Delta s \approx 0.015$, aligning with the smallest grid steps near the cylinder, uniformly set at $\Delta x_{min} = 0.015$ within a range of $\pm 5D$. Then, the size of the grid was gradually increased to a maximum value of $\Delta x_{min} = 0.3$ near the boundaries, employing a smooth sinusoidal transition to preserve the accuracy of the results. A uniform value of the velocity in the x direction $\mathbf{u} = (1, 0)$ was applied along the left, top, and bottom boundaries, while a convective boundary condition $\partial_t \mathbf{u} + \partial_x \mathbf{u} = 0$ was applied at the outlet (right boundary) to facilitate the undisturbed exit of the vorticity. A uniform value of $\mathbf{u} = (1, 0)$ was set as the initial condition throughout the computational domain.

The results obtained for the steady-state flow at $Re = 20$ and $Re = 40$ were verified against published drag and lift coefficients, as well as against the wake geometry illustrated in Fig. 14. The hydrodynamic forces on the cylinder are evaluated directly from the immersed-boundary forcing field $\mathbf{f}_k$ defined on the Lagrangian markers so that the non-dimensional coefficients read

$$C_d = \frac{F_x}{\frac{1}{2}\rho U_\infty^2 D}, \qquad C_l = \frac{F_y}{\frac{1}{2}\rho U_\infty^2 D},$$

with the cylinder diameter D as the reference length. Under the current scaling these reduce to $C_d = 2F_x$ and $C_l = 2F_y$.

All the compared results are summarised in Tables 5and 6. The present values agree closely with both numerical and experimental benchmarks [9,39–42]. Figure 15 shows the spanwise vorticity fields for $Re = 20$ and $Re = 40$; the computed structures reproduce the patterns reported in literature.

Following the verification of the sub-critical flow regimes, we extended our simulations to the value of $Re = 200$ to investigate the periodic vortex shedding phenomenon. The simulations started from a uniform initial flow, and the onset of vortex shedding was naturally triggered by the smallest numerical perturbations in the calculations. The numerical results replicate the periodic shedding of vortices, forming the typical von Kármán vortex street. The vorticity contours and shedding patterns are illustrated in Fig. 16. The resulting lift and drag coefficients are compared to the values reported in previous studies. In addition, we compare the currently obtained Strouhal number, which is defined as $St = \frac{f_s d}{u_\infty}$, where f_s denotes the shedding frequency with values of the literature[9,42–46]. As shown in Table 7, good agreement between our and the independent results was obtained for all the supercritical flow characteristics.

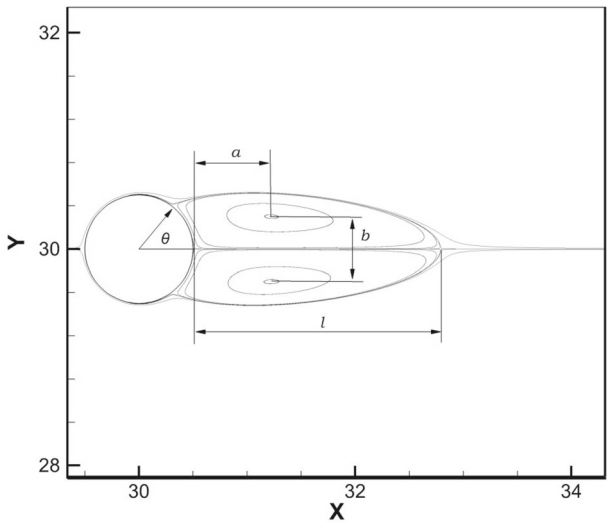


Fig. 14 Geometric characteristics of the flow wake developing behind the cylinder: θ – flow separation angle; l – length of the recirculating bubble; a – distance between the cylinder's trailing edge and the center of the recirculating bubble; b – distance between the opposite recirculating bubbles

Table 6 Comparison of the currently obtained results with the corresponding data available in the literature for the steady state flow at $Re = 20$ and $Re = 40$

Re	Study	l/d	a/d	b/d	θ	C_d
20	Coutanceau & Bouard [39]	0.93	0.33	0.46	45.0°	–
	Tritton [40]	–	–	–	–	2.09
	Dennis & Chang [41]	0.94	–	–	43.7°	2.05
	Linnick & Fasel [42]	0.93	0.36	0.43	43.5°	2.06
	Taira & Colonius (Case A)[9]	0.97	0.39	0.43	44.1°	2.07
	Taira & Colonius (Case B)[9]	0.94	0.37	0.43	43.3°	2.06
	Current study	0.943	0.366	0.431	42.6°	2.05
40	Coutanceau & Bouard [39]	2.13	0.76	0.59	53.8°	–
	Tritton [40]	–	–	–	–	1.59
	Dennis & Chang [41]	2.35	–	–	53.8°	1.52
	Linnick & Fasel [42]	2.28	0.72	0.60	53.6°	1.54
	Taira & Colonius (Case A)[9]	2.33	0.75	0.60	54.1°	1.55
	Taira & Colonius (Case B)[9]	2.30	0.73	0.60	53.7°	1.54
	Current study	2.29	0.724	0.599	53.8°	1.534

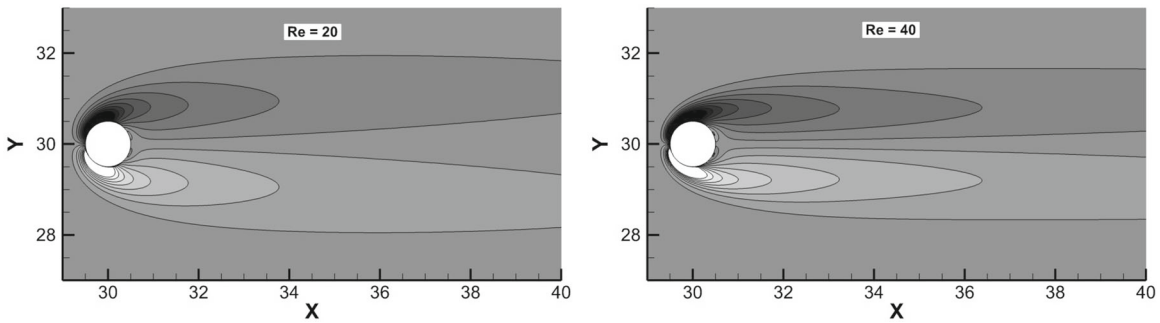


Fig. 15 Vorticity distribution obtained for $Re = 20$ (left) and $Re = 40$ (right). Contour levels are set from -3 to 3 in increments of 0.4

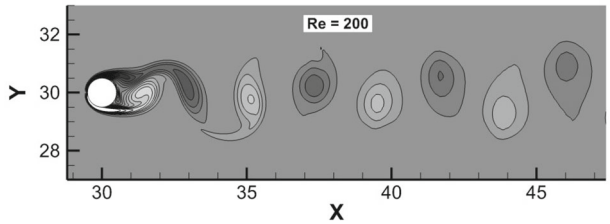


Fig. 16 Snapshot of the vorticity field around the cylinder obtained for $Re = 200$. Contour levels are set from -3 to 3 in increments of 0.4

Table 7 Comparison of the Strouhal number and the drag and lift coefficients for the incident flow over a cylinder from experimental and numerical studies at $Re = 200$

Study	St	C_d	C_l
Belov et al. [43]	0.193	1.19 ± 0.042	± 0.64
Liu et al. [44]	0.192	1.31 ± 0.049	± 0.69
Lai and Peskin [45]	0.190	–	–
Roshko [46]	0.19	–	–
Linnick and Fasel [42]	0.197	1.34 ± 0.044	± 0.69
Taira & Colonius (Case B)[9]	0.196	1.35 ± 0.048	± 0.68
Taira & Colonius (Case C)[9]	0.195	1.34 ± 0.047	± 0.68
Taira & Colonius (Case D)[9]	0.197	1.36 ± 0.043	± 0.69
Current study	0.1960	1.340 ± 0.0445	± 0.682

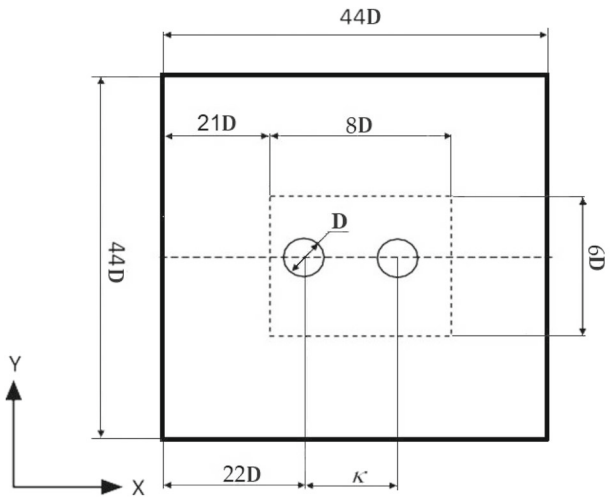


Fig. 17 Schematic representation of the geometrical model and discretization of the computational domain for the flow around a tandem of horizontally aligned cylinders

Flow around a tandem of two horizontally aligned cylinders

In this section, the verification of the currently developed computational methodology is extended by examining the fluid dynamics around a tandem of two horizontally aligned circular cylinders at $Re = 200$. This configuration serves as a benchmark to demonstrate the capabilities of capturing the effects of multiple stationary bodies on flow behavior. The computational domain, schematically shown in Fig. 17, is discretized by a 1075×1075 non-uniform mesh. A central region surrounding the cylinders, sized $8d \times 6d$, is refined with a 800×600 uniform mesh, where the grid steps are $\Delta x = \Delta y = 0.01$. Beyond this detailed area, the mesh size gradually expands to $\Delta x \approx \Delta y \approx 0.20$ at the domain boundaries. Different vortex shedding dynamics were assessed by setting the cylinders at varying center-to-center distances, $\kappa = L_x/d = [1.5, 2.3, 5]$. The boundary and initial conditions were chosen to be exactly the same as for the single cylinder configuration.

Each separation distance, κ, induces a distinct vortex shedding scenario:

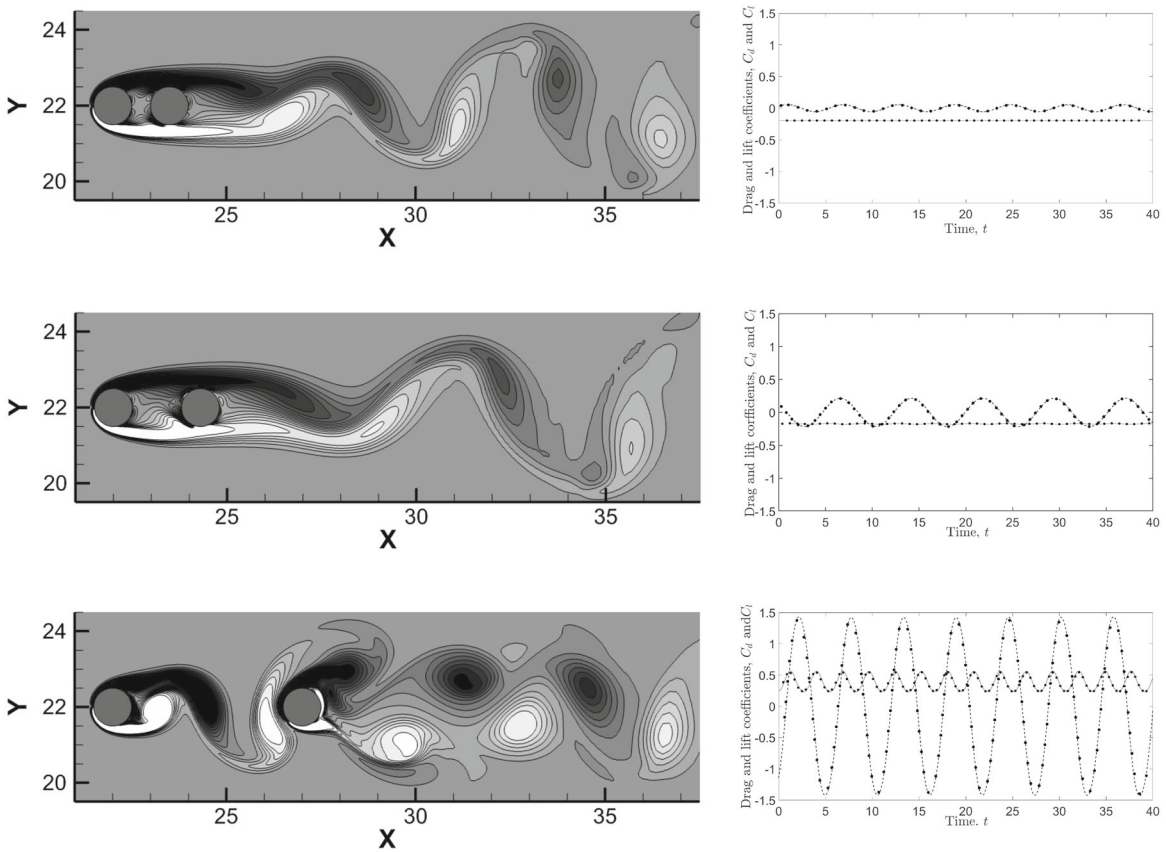


Fig. 18 The vorticity field contours on the left illustrate vortex shedding regimes around two cylinders in a horizontal tandem alignment for the values of $\kappa = 1.5, 2.3, 5.0$, in that specific order. Contour levels range from -2.2 to 2.2 in increments of 0.3. The graphs on the right depict the time evolution of the drag (gray solid line) and lift (black dashed line) coefficients, as calculated for the downstream cylinder. Black points are the reference data from [47]

Table 8 Comparison of the oscillating frequencies and average drag coefficients for different configurations at $Re = 200$

κ	St_L(Present/Ref[48]/Ref[47])	St_D(Present/Ref[48]/Ref[47])	$\overline{C_D}$(Present/Ref[48]/Ref[47])
1.5	0.333/0.338/0.332	0.167/0.169/0.166	−0.193/−0.196/−0.194
2.3	0.267/0.263/0.264	0.133/0.132/0.132	−0.174/−0.173/−0.172
5	0.357/0.360/0.355	0.178/0.181/0.178	0.387/0.423/0.401

- **SG (Symmetric in the Gap) for** $\kappa = 1.5$: Exhibits symmetric vortex shedding in the gap between the cylinders.
- **AG (Alternating in the Gap) for** $\kappa = 2.3$: Features alternating vortices with significant oscillations in lift and drag.
- **WG (Wake in the Gap) for** $\kappa = 5$: Mimics the behavior around a single cylinder with stable and well-defined vortex shedding downstream.

The vorticity contours and the vortex shedding patterns obtained for the three scenarios are illustrated in Fig.18. The resulting lift and drag coefficients calculated for the downstream cylinder were compared to the values reported in previous studies [47,48]. Furthermore, we compared with the same studies the Strouhal number values, defined as $St_L = \frac{f_{CL}d}{u_\infty}$ and $St_D = \frac{f_{CD}d}{u_\infty}$, where f_{CL} and f_{CD} denote the shedding frequencies of lift and drag oscillations, respectively. The time-averaged drag coefficient of the downstream cylinder, $\overline{C_D}$, was also compared. A summary of all the results is presented in Table 8.

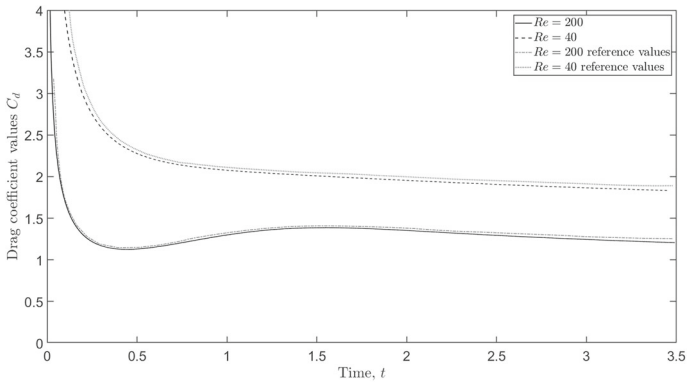


Fig. 19 Time evolution of the cylinder drag coefficient calculated for $Re = 200$ and $Re = 40$. Reference values are obtained by digital photogrammetry from [9]

Spontaneous start

We next verify the developed method for the simulation of moving boundary configurations. Specifically, a spontaneous start of a circular cylinder was simulated. Initially at rest, a cylinder of diameter $D = 1$ was impulsively set to move to the left with a constant velocity $u_0 = -1$. The simulations were performed at $Re = 40$ and 200. A square computational domain with dimensions $[0, 30D] \times [0, 30D]$ was chosen. The cylinder was placed at the center of the domain to diminish the influence of no-slip boundary conditions for all the velocity components.

We employed a non-uniform grid, maintaining a finer resolution near the cylinder ($\Delta x_{\min} = 1/256$) to capture the flow dynamics accurately. A constant time step $\Delta t = 5 \times 10^{-4}$ was used, setting the upper bound value of 0.128 for the CFL number to ensure numerical stability throughout the simulation. The simulations spanned a range of $0 \le t \le 3.5$. In this specific case, an integrated version of the discrete Dirac delta function from [29] was utilized to avoid fully spurious oscillations (see Appendix A for details on the interpolation and regularization operators).

The cylinder started to move spontaneously within a quiescent fluid, generating complex flow patterns, including wake formation with consequent vortex shedding behind it. Instantaneous vorticity fields at times $t = 1$, 2.5, and 3.5 are shown in Fig. 20, demonstrating the time evolution of the flow. The obtained contours agree well with the data reported in [9].

In Fig. 19, we plotted the time evolution of the drag coefficients for $Re = 40$ and 200. Initially, the force implied for a spontaneous start was, as expected, singular [49], and the solution scheme successfully captured this behavior. The resulting drag coefficients were slightly lower than those provided by [9], which appears to be a consequence of the higher grid resolution and lower CFL numbers employed in the current study. Concluding this benchmark, the results demonstrate the capability of the computational model to accurately replicate the spontaneous start scenario, verifying the adequate accuracy of the developed method in simulating flow developing around a moving boundary.

Flow around an airfoil at supercritical angle of attack

The flow around a NACA0012 airfoil at $Re = 1000$ at the supercritical angle of attack of 34 degrees was considered. The current set-up was similar to that used for the simulation of a tandem of two cylinders in incident flow: the computational domain is discretized by a 1075×1075 non-uniform grid. A central region surrounding the airfoil, of dimensions $6L \times 8L$, is refined with an 600×600 uniform mesh, with grid steps equal to $\Delta x = \Delta y = 0.01$. Beyond this refined domain, the mesh size gradually expands, reaching $\Delta x \approx \Delta y \approx 0.20$ at the domain boundaries. All the length dimensions are scaled by a chord length $L = 1$, as illustrated in Figure 21. Additionally, the NACA profile is carefully built to preserve an almost constant arc-length spacing close to that of the minimum grid size.

For the 34-degree angle of attack, we compared our results with both experimental and numerical [50–53] data in terms of streamline snapshots captured at different times. The currently obtained results shown in Fig.

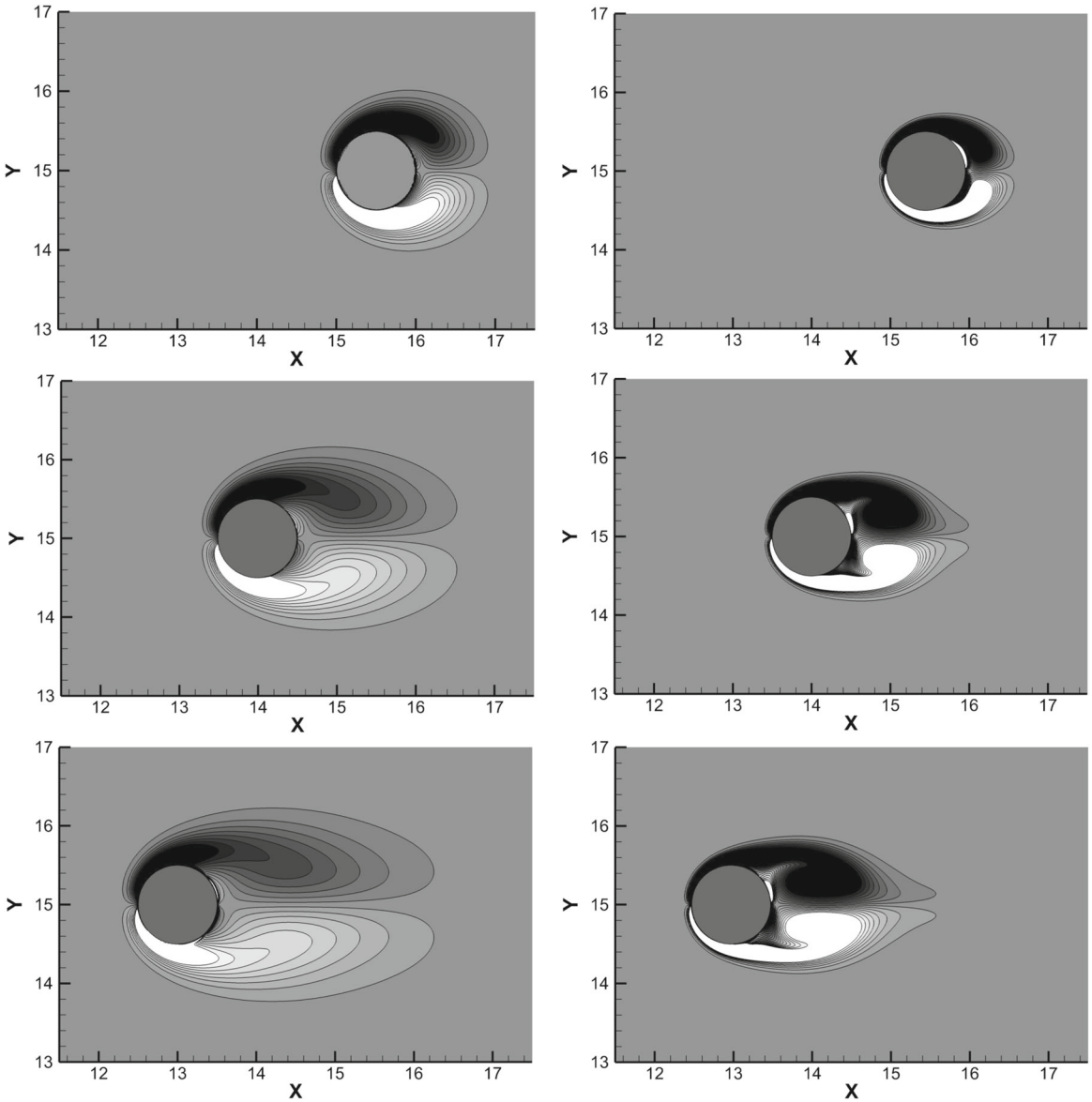


Fig. 20 Instantaneous vorticity fields around a moving circular cylinder at $Re = 40$ (left) and $Re = 200$ (right) shown at non-dimensional times of 1, 2.5, and 3.5 (from top to bottom). Contour levels ranging from -3 to 3 in increments of 0.4 were chosen

22 for times $t = 1.6$ (left) and $t = 3.2$ (right) exhibit satisfactory agreement with the corresponding numerical and experimental independent data.

Table 9 summarises the comparison by listing the geometric centres of vortices $A - D$ at both time instants, together with the reference coordinates reported by [53]. The discrepancies are minor, indicating that the present solver captures the primary vortex dynamics and vortex formation with an accuracy comparable to simulations carried out on body-fitted grids and experiments.

Conclusion

The current study introduces a novel methodology for the simulation of incompressible flows around immersed bodies. The key idea lies in establishing a modified Vanka smoother that implicitly incorporates the direct forcing IBM formalism. The developed solver leverages the Schur complement decomposition of the extended

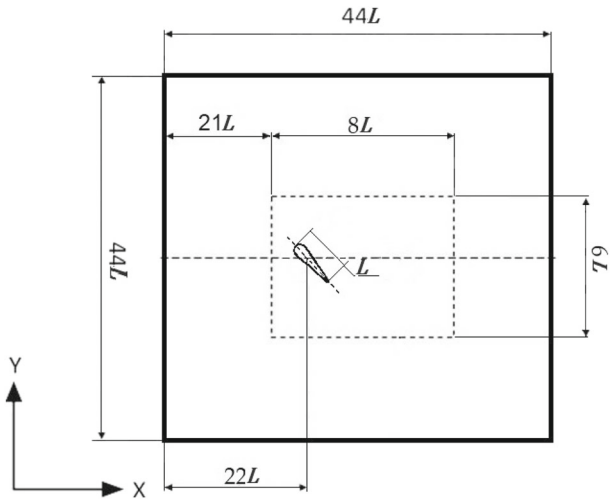


Fig. 21 Computational domain and grid setup for NACA0012 at the angle of attack $\alpha = 34°$

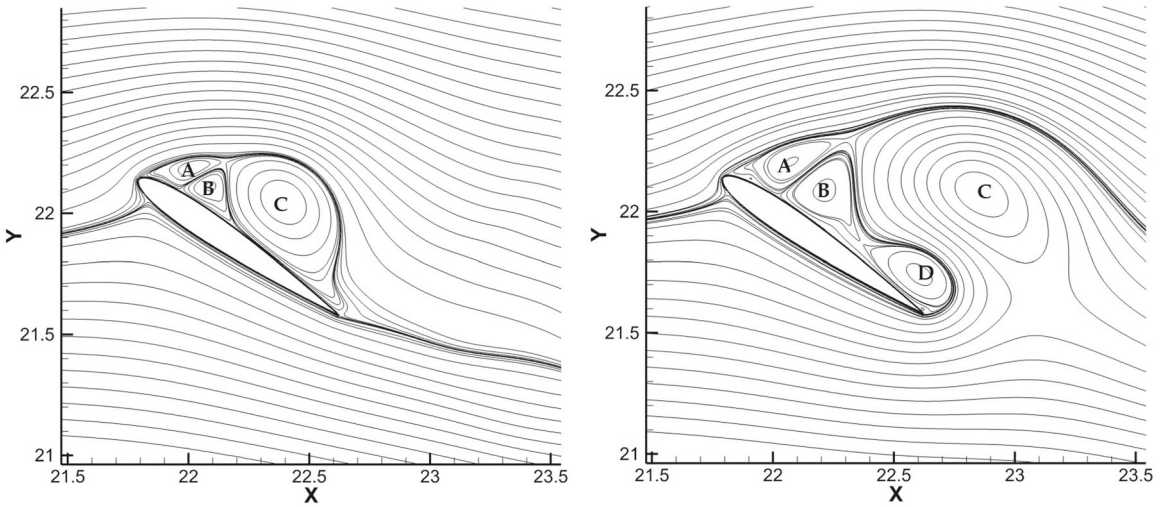


Fig. 22 Streamlines at $t = 1.6$ and $t = 3.2$ for an angle of attack $\alpha = 34°$. A, B, C and D mark the centres of the corresponding vortices

Table 9 Geometric-centre coordinates (x, y) of the dominant vortices at two instants, compared with the measurements of [53]

Vortex	$t = 1.6$		$t = 3.2$	
	Present	Ref. [53]	Present	Ref. [53]
A	(−0.092, 0.166)	(−0.110, 0.162)	(−0.056, 0.217)	(−0.040, 0.219)
B	(0.013, 0.144)	(−0.021, 0.126)	(0.132, 0.192)	(0.133, 0.169)
C	(0.296, 0.262)	(0.220, 0.268)	(0.665, 0.545)	(0.608, 0.464)
D	–	–	(0.655, 0.113)	(0.720, 0.064)

Vanka-type operator, exploiting its symmetrical structure and fully diagonal part of the matrix attributed to velocity corrections to optimize computational performance. This decomposition efficiently reduces the operator size to approximately one-third of its original size without compromising numerical stability or accuracy. The method's accuracy stems from its fully coupled treatment of all terms, including pressure-velocity coupling, non-linear convection terms and Lagrangian forces, thereby eliminating splitting errors inherent in segregated approaches such as fractional step, projection, or SIMPLE methods. Furthermore, this monolithic approach avoids the need to explicitly prescribe pressure boundary conditions, which is an aspect that can pose challenges in segregated methods, particularly in Dirichlet condition cases [3].

Extensive verification studies were conducted across a range of canonical flows, including flow between two concentric rotating cylinders; 2D and 3D lid-driven cavity flows with an immersed cylinder and sphere placed at the cavity center, respectively; flow around single and tandem stationary cylinders; flow developing following the spontaneous start of a cylinder; and flow past a NACA0012 airfoil. Throughout these tests, the method exhibited good agreement with both numerical and experimental data across the entire range of Reynolds numbers and geometric configurations examined. The solver consistently achieved solution times comparable to semi-implicit approaches while offering improved accuracy and stability at CFL numbers up to 0.5. This capability of handling relatively high CFL numbers is particularly advantageous for simulating flows with moving boundaries, where precise resolution of fluid-structure interaction is crucial.

Future work will focus on extending the methodology to three-dimensional configurations through domain decomposition techniques. This extension will leverage parallel computing architectures to handle the increased computational demands of 3D simulations while maintaining the method's favorable stability characteristics. Beyond direct time integration, the developed solver serves as a foundation for linear stability analysis by functioning as a time-stepper that approximates the Jacobian's exponential, enabling matrix-free implementations of Newton-Krylov methods and Arnoldi iterations. This approach overcomes limitations of existing IBM implementations, where neither explicit nor semi-implicit formulations can provide the robust and rapid convergence required for stability analysis. The extension to fully three-dimensional flows will significantly broaden the scope of fluid-structure interaction problems amenable to systematic stability investigation.

Acknowledgements This research was partially supported by the Israel Science Foundation (grant no. 950/22) and by the Israel Ministry of Energy and Infrastructure (grant No. 315-222-11-049).

Funding Open access funding provided by Ben-Gurion University.

Declarations

Author Contribution K.G. : analysis and related examples, programming, validation, writing. M.K.: analysis, programming, validation. O.O.: supervision, validation, writing. Y.F.: conceptaluaziation, supervision, software, validation, writing.

Data Availability No datasets were generated or analysed during the current study.

Competing interests The authors declare no competing interests.

A Interpolation and Regularization Operators

The immersed boundary method (IBM) relies on two key operators to couple Lagrangian dynamics with the Eulerian flow field: the interpolation operator **I** and the regularization operator **R**. In the classical IBM formulation [1], the surface of the immersed body is represented by a series of Lagrangian markers that typically do not coincide with the underlying Eulerian grid, where the Navier–Stokes equations are solved. Consequently, **I** interpolates Eulerian fields (e.g., velocity) onto the Lagrangian points, while **R** smears Lagrangian quantities back to the Eulerian grid.

These operators are constructed using discrete approximations of the Dirac delta function, which serve a dual role: (1) they provide information transfer between non-conforming grids, and (2) they regularize sharp or even singular source terms, such as the force required to enforce no-slip boundary conditions, thereby ensuring stability and numerical consistency. Throughout this study, we use the three-point discrete delta function developed by Roma et al. [28], which has compact support and satisfies key conservation and smoothness

properties on staggered, uniformly spaced grids:

$$\delta(r) = \begin{cases} \frac{1}{6\Delta r}\left[5 - 3\frac{|r|}{\Delta r} - \sqrt{-3\left(1 - \frac{|r|}{\Delta r}\right)^2 + 1}\right] & \text{for } 0.5\Delta r \le |r| \le 1.5\Delta r, \\ \frac{1}{3\Delta r}\left[1 + \sqrt{-3\left(\frac{|r|}{\Delta r}\right)^2 + 1}\right] & \text{for } |r| \le 0.5\Delta r, \\ 0 & \text{otherwise,} \end{cases} \tag{15}$$

This kernel provides a favorable balance between computational locality and accuracy but assumes evenly spaced grids and marker distributions. Deviation from this assumption may lead to increased stiffness or degraded interpolation accuracy.

The formal definitions of the regularization and interpolation operators are given by:

$$\boldsymbol{R}(\boldsymbol{F}^k) = \int_S (\boldsymbol{F}^k(\boldsymbol{X}^k)) \cdot \delta(\boldsymbol{x}_i - \boldsymbol{X}^k), dV_S^k, \tag{16a}$$

$$\boldsymbol{I}\left(\boldsymbol{u}(\boldsymbol{x}_i)\right) = \int_\Omega \left(\boldsymbol{u}(\boldsymbol{x}_i)\right) \cdot \delta(\boldsymbol{X}^k - \boldsymbol{x}i), dV_{\Omega_i}, \tag{16b}$$

where dV_S^k and dV_{Ω_i} denote the control volumes around the k^{th} Lagrangian marker and the i^{th} Eulerian cell, respectively.

In some cases, particularly for moving bodies, delta functions can introduce nonphysical oscillations in the computed forces. These oscillations appear over time with small amplitudes that depend on the coupling strategy and discretization scheme. To suppress spurious force oscillations, we replace the Roma et al. delta function [28] with the integrated smoothed delta function [29] in this specific case. This alternative kernel improves the stability of the interpolation and regularization steps by more smoothly distributing the discontinuous forcing introduced at early times, when the velocity mismatch between the fluid and structure is initially large.

References

1. Peskin, C.S.: Flow patterns around heart valves: A numerical method. J. Comput. Phys. **10**(2), 252–271 (1972). https://doi.org/10.1016/0021-9991(72)90065-4
2. Verzicco, R.: Immersed boundary methods: Historical perspective and future outlook. Annu. Rev. Fluid Mech. **55**, 129–155 (2023)
3. Gresho, P.M., Sani, R.L.: On pressure boundary conditions for the incompressible Navier-Stokes equations†. Int. J. Numer. Meth. Fluids **7**(10), 1111–1145 (1987). https://doi.org/10.1002/fld.1650071008. (**First published: October 1987; Based on an invited lecture; 441 citations (as of original note)**)
4. Mohd-Yusof, J.: Combined immersed-boundary/b-spline methods for simulations of flow in complex geometries. Center for Turbulence Research, Annual Research Briefs, 317–327 (1997)
5. Fadlun, E.A., Verzicco, R., Orlandi, P., Mohd-Yusof, J.: Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations. J. Comput. Phys. **161**(1), 35–60 (2000)
6. Uhlmann, M.: An immersed boundary method with direct forcing for the simulation of particulate flows. J. Comput. Phys. **209**, 448–476 (2005)
7. Le, D.V., Khoo, B.C., Lim, K.M.: An implicit-forcing immersed boundary method for simulating viscous flows in irregular domains. Comput. Methods Appl. Mech. Eng. **197**, 2119–2130 (2008)
8. Glowinski, R., Pan, T.W., Périaux, J.: Distributed Lagrange multiplier methods for incompressible viscous flow around moving rigid bodies. Comput. Methods Appl. Mech. Eng. **151**, 181–194 (1998)
9. Taira, K., Colonius, T.: The immersed boundary method: a projection approach. J. Comput. Phys. **225**, 3121–3133 (2007)
10. Goncharuk, K., Oshri, O., Feldman, Y.: The immersed boundary method: A simple approach. J. Comput. Phys. **487**, 112148 (2023)
11. Yovel, R., Treister, E., Feldman, Y.: The immersed boundary method: an accelerated simple approach for moving bodies. arXiv (2025). https://doi.org/10.48550/arXiv.2501.15314, arXiv:2501.15314 [physics.flu-dyn]
12. Kallemov, B., Bhalla, A., Griffith, B., Donev, A.: An immersed boundary method for rigid bodies. Commun. Appl. Math. Comput. Sci. **11**, 79–141 (2016)
13. Feldman, Y., Gulberg, Y.: An extension of the immersed boundary method based on the distributed lagrange multiplier approach. J. Comput. Phys. **322**, 248–266 (2016)
14. Gulberg, Y., Feldman, Y.: Flow control through the use of heterogeneous porous media: "smart" passive thermo-insulating materials. Int. J. Therm. Sci. **110**, 369–382 (2016). https://doi.org/10.1016/j.ijthermalsci.2016.07.008
15. Goza, A., Colonius, T.: A strongly-coupled immersed-boundary formulation for thin elastic structures. J. Comput. Phys. **336**, 401–411 (2017)
16. Stein, D., Guy, R., Thomases, B.: Immersed boundary smooth extension (ibse): a high-order method for solving incompressible flows in arbitrary smooth domains. J. Comput. Phys. **335**, 155–178 (2017)
17. Vanka, S.: Block-implicit multigrid solution of navier–stokes equations in primitive variables. J. Comput. Phys. **65**, 138–158 (1986)

18. Vanka, S.: A calculation procedure for three-dimensional steady recirculating flows using multigrid methods. Comp. Meth. Appl. Mech. Eng. **55**, 321–338 (1986)
19. Guy, R., Philip, B.: A multigrid method for a model of the implicit immersed boundary equations. Commun. Comput. Phys **12**, 378–400 (2012)
20. Tuckerman, L.S., Barkley, D.: Bifurcation analysis for timesteppers. Numerical Methods for Bifurcation Problems and Large-Scale Dynamical Systems **119**, 453–466 (2000). https://doi.org/10.1007/978-1-4612-1282-1_20
21. Feldman, Y.: Oscillatory instability of 2d natural convection flow in a square enclosure with a tandem of vertically aligned cylinders. Fluid Dyn. Res. **50**, 051410 (2018)
22. Giannetti, F., Luchini, P.: Structural sensitivity of the first instability of the cylinder wake. J. Fluid Mech. **581**, 167–197 (2007)
23. Tchoufag, J., Fabre, D., Magnaudet, J.: Global linear stability analysis of the wake and path of buoyancy-driven disks and thin cylinders. J. Fluid Mech. **740**, 278–311 (2014)
24. Tirri, A., Nitti, A., Sierra-Ausin, J., Giannetti, F., Tullio, M.D.: Linear stability analysis of fluid-structure interaction problems with an immersed boundary method. J. Fluids Struct. **117**, 103830 (2023)
25. Banks, J.W., Henshaw, W.D., Schwendeman, D.W., Tang, Q.: A stable partitioned fsi algorithm for rigid bodies and incompressible flow in three dimensions. J. Comput. Phys. **373**, 445–492 (2018)
26. Frederic, A., Jean-Christophe, R., Xavier, G.: A domain decomposition matrix-free method for global linear stability. Computers & Fluids **66**, 63–84 (2012)
27. Peskin, C.: The immersed boundary method. Acta Numerica, 479–517 (2002)
28. Roma, A., Peskin, C., Berger, M.: An adaptive version of the immersed boundary method. J. Comput. Phys. **153**, 509–537 (1999)
29. Yang, X., Zhang, X., Li, Z., He, G.-W.: A smoothing technique for discrete delta functions with application to immersed boundary method in moving boundary simulations. J. Comput. Phys. **228**(20), 7821–7836 (2009). https://doi.org/10.1016/j.jcp.2009.07.023
30. Guy, R., Philip, B., Griffith, B.: Geometric multigrid for an implicit-time immersed boundary method. Advances Comput. Math. **41**, 635–662 (2015)
31. Amestoy, P.R., Duff, I.S., Koster, J., L'Excellent, J.-Y.: A fully asynchronous multifrontal solver using distributed dynamic scheduling. SIAM J. Matrix Anal. Appl. **23**(1), 15–41 (2001)
32. Amestoy, P.R., Guermouche, A., L'Excellent, J.-Y., Pralet, S.: Hybrid scheduling for the parallel solution of linear systems. Parallel Comput. **32**(2), 136–156 (2006)
33. Patankar, S.: Numerical Heat Transfer and Fluid Flow. CRC press, (2018)
34. Intel Corporation: Intel Math Kernel Library, Version 2024.0. (2024). Available from https://software.intel.com/content/www/us/en/develop/tools/oneapi/components/onemkl.html
35. Zhu, J., Holmedal, L.E., Wang, H., Myrhaug, D.: Flow patterns in a steady lid-driven rectangular cavity with an embedded circular cylinder. Fluid Dyn. Res. **52**(6), 065508 (2020)
36. Cai, S.-G., Ouahsine, A., Favier, J., Hoarau, Y.: Moving immersed boundary method. Int. J. Numer. Meth. Fluids **85**(5), 288–323 (2017)
37. Leopardi, P.: A partition of the unit sphere into regions of equal area and small diameter. Electron. Trans. Numer. Anal. **25**, 309–327 (2006)
38. Young, D.L., Lin, Y.C., Fan, C.M., Chiu, C.L.: The method of fundamental solutions for solving incompressible navier–stokes problems. Eng. Anal. Boundary Elem. **33**, 1031–1044 (2009)
39. Coutanceau, M., Bouard, R.: Experimental determination of the main features of the viscous flow in the wake of a circular cylinder in uniform translation. part 1. steady flow. J. Fluid Mech. **79**(2), 231–256 (1977)
40. Tritton, D.J.: Experiments on the flow past a circular cylinder at low reynolds number. J. Fluid Mech. **6**, 547–567 (1959)
41. Dennis, S.C.R., Chang, G.: Numerical solutions for steady flow past a circular cylinder at reynolds number up to 100. J. Fluid Mech. **42**(3), 471–489 (1970)
42. Linnick, M.N., Fasel, H.F.: A high-order immersed interface method for simulating unsteady incompressible flows on irregular domains. J. Comput. Phys. **204**(1), 157–192 (2005). https://doi.org/10.1016/j.jcp.2004.09.017
43. Belov, A., Martinelli, L., Jameson, A.: A New Implicit Algorithm with Multigrid for Unsteady Incompressible Flow Calculations. In: 33rd Aerospace Sciences Meeting and Exhibit, Reno, NV (1995)
44. Liu, C., Zheng, X., Sung, C.H.: Preconditioned multigrid methods for unsteady incompressible flows. J. Comput. Phys. **139**, 35–57 (1998)
45. Lai, M.-C., Peskin, C.S.: An immersed boundary method with formal second-order accuracy and reduced numerical viscosity. J. Comput. Phys. **160**(2), 705–719 (2000). https://doi.org/10.1006/jcph.2000.6483
46. Roshko, A.: Report 1191: On the development of turbulent wakes from vortex streets. Technical report, NACA (1954)
47. Carmo, B.S., Meneghini, J.R., Sherwin, S.J.: Possible states in the flow around two circular cylinders in tandem with separations in the vicinity of the drag inversion spacing. Phys. Fluids **22**(5), 054101 (2010). https://doi.org/10.1063/1.3420111
48. Feldman, Y., Gulberg, Y.: An extension of the immersed boundary method based on the distributed lagrange multiplier approach. J. Comput. Phys. **322**, 248–266 (2016). https://doi.org/10.1016/j.jcp.2016.06.039
49. Bar-Lev, M., Yang, H.T.: Initial flow field over an impulsively started circular cylinder. J. Fluid Mech. **72**(4), 625–647 (1997). https://doi.org/10.1017/S0022112077001617
50. George, A., Huang, L.C., Tang, W.-P., Wu, Y.D.: Numerical simulation of unsteady incompressible flow (re $\leq$ 9500) on the curvilinear half-staggered mesh. SIAM J. Sci. Comput. **21**(6), 2331–2351 (2000)
51. George, A., Tang, W.-P., Wu, Y.D.: Multilevel one-way dissection factorization. SIAM J. Matrix Anal. Appl. **22**(3), 752–771 (2001). https://doi.org/10.1137/S0895479898332564
52. Bouchon, F., Dubois, T., James, N.: A second-order cut-cell method for the numerical simulation of 2d flows past obstacles. Computers & Fluids **65**, 80–91 (2012)

53. Daube, O., Loc, T.P., Monnet, P., Coutanceau, M.: Ecoulement instationnaire décollé d'un fluide incompressible autour d'un profil: une comparaison théorie-expérience. In: AGARD CP 386, Paper 3 (1985)