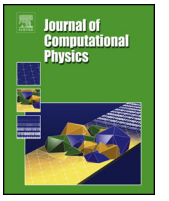
Journal of Computational Physics 487 (2023) 112148

Contents lists available at ScienceDirect

Journal of Computational Physics

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



# The immersed boundary method: A SIMPLE approach

Kirill Goncharuk, Oz Oshri, Yuri Feldman *

*Department of Mechanical Engineering, Ben-Gurion University of the Negev, P.O. Box 653, Beer-Sheva 84105, Israel*

ARTICLE INFO

*Article history:*
Received 8 September 2022
Received in revised form 29 January 2023
Accepted 11 April 2023
Available online 18 April 2023

*Keywords:*
Implicit immersed boundary method
Distributed Lagrange multiplier
Schur complement

ABSTRACT

A novel formulation of the direct forcing immersed boundary (IB) method, that treats it as an integral part of a SIMPLE method is presented for the simulation of incompressible flows. The incompressibility and no-slip kinematic constraints are treated implicitly as distributed Lagrange multipliers and are fully coupled with each other by combining them into a single Poisson-body forces system of equations that constitutes a regularized saddle point problem. A physically justified approximation of the system, resembling a technique typical of the penalty method, is carried out to facilitate the solution. By utilizing the Schur complement approach, the approximated system is decomposed into two separate systems of equations, allowing us to compute the values for the volumetric force and pressure corrections. The first system, which is characterized by a small (only $O(10)$) value of the condition number, is conveniently solved by the BiCgStab method [1], converging within 2-3 iterations, while the second system is addressed by the direct TPF solver [2], characterized by $O(N^{4/3})$ complexity. The entire methodology is designed to be highly portable, which facilitates the use of any available solver that was designed to simulate incompressible flows governed by the Helmholtz and Laplace operators but could not benefit from the immersed boundary formalism. The capabilities of the developed methodology applied to the simulation of representative shear- and buoyancy-driven confined flows developing in the presence of stationary immersed bodies are demonstrated. A further application of the developed approach to moving boundary and two-way coupled fluid-structure interaction problems is discussed.



## 1. Introduction

The immersed boundary (IB) method, initially introduced by Peskin [3] about half a century ago, may perhaps be regarded as the start of the rapidly growing field of computational science that facilitates the numerical simulation of a wide range of physical phenomena, including thermally driven flows, flows developing around solid bodies (either stationary or moving with prescribed kinematics), flows around elastic bodies and within elastic confinements, particulate flows, and two-phase flows, to name but a few. A full list of the application areas of the IB method is far more extensive, and thus, rather than enumerating them, we refer the interested reader to a number of recent review papers [4–7]. Over the years, the IB method has evolved into a number of branches, which can be classified into ghost-cell [8,9], cut-cell [10], sharp interface [11], cut-stencil [12], iso-geometric [13] IB methods, as well as the recently developed shifted boundary method [14,15].

* Corresponding author.
*E-mail address:* yurifeld@bgu.ac.il (Y. Feldman).

https://doi.org/10.1016/j.jcp.2023.112148
0021-9991/

While some of the above branches resemble the original IB method only vaguely (if at all), in the current paper we focus on the family of IB methods based on the same concepts as those elaborated in the study of [3]. The method, originally designed for the simulation of the fluid-structure interactions (FSI), is based on a number of fundamental concepts. First, two separate grids are defined—a Lagrangian grid for the body surface and a Eulerian grid for the surrounding fluid. Second, the impact of the body on the surrounding flow is reflected by the dynamical response of the body surface obtained by applying constitutive laws, for example, linear springs connecting the Lagrangian points of the surface. As a result, the surface of an immersed body exerts forces on the surrounding fluid, constituting dynamic constraints on the flow. To impose the same velocity on the solid boundary and the fluid adjacent to it, the kinematic constraint of no-slip should be imposed on the surface of the immersed body. Third, as the two grids are not necessarily superimposable, two adjoint operators, namely, the interpolation and the regularization operators, are introduced to convey information on the dynamic and kinematic constraints between the two grids. Both operators are implemented by utilizing discrete Dirac delta functions that satisfy a number of properties so as to provide conservation of the fluid dynamics laws [16]. Finally, the flow field is resolved outside and within the body, although for most configurations flow within the body is considered nonphysical. Again, we refer the interested reader to the excellent review of Mittal and Iaccarino [17], which provides a comprehensive picture of the developments in the field.

Unfortunately, the original IB method [3] cannot be straightforwardly applied to the simulation of the flow around rigid non-deformable immersed bodies. Simply assigning large enough values to the stiffness coefficients of the linear springs connecting the Lagrangian points of the body surface significantly increases the stiffness of the problem, leading to convergence problems and to inaccurate results. The remedy derives from the direct forcing approach, initially introduced by Mohd-Yusof and co-authors [18,19], that allows the kinematic constraint of no-slip to be directly imposed on the immersed boundary without making any assumptions about the dynamics of the system. We note that the earlier versions of the direct forcing approach had relied mostly on the explicit calculation of the forces exerted by the immersed body on the surrounding flow, as this approach requires the introduction of only minimal changes into the solvers of Navier-Stokes (NS) equations. Conceptually, the explicit formulation required applying the available solvers, which were not originally equipped with the IB functionality, in a predictor–corrector manner. The drawbacks of this approach derive from three main factors, namely, the need to proceed in very small time steps to avoid nonphysical flow leaks through the surface of the immersed body, the imposition of the no-slip constraint on the predicted, rather than on the corrected, flow field, and, most importantly, the very fact of violating the elliptic physics of the NS equations. In the original explicit formulation, each force is calculated locally for each Lagrangian point. As a result, the force value is not immediately affected by far distant velocities and, in turn, cannot affect the balance of momentum in far distant locations of the computational domain. The problem becomes critical for unsteady flows characterized by low, $O(10^2)$ and moderate $O(10^3)$ values of the Reynolds number and also in the simulation of dense particulate flows characterized by a large number of collisions between particles taking place in a short period of time.

To address the challenges posed by the explicit formulation of the direct forcing approach, research dedicated to developing its semi- or fully implicit counterparts has been conducted in recent decades. Worth mentioning in this context is the study of Wu and Shu [20], who developed an implicit IB formulation incorporated within the lattice-Boltzmann (LB) method. The authors successfully addressed the elliptic physics of the NS equations by coupling all the Lagrangian forces. At the same time, the calculation of the forces was based entirely on the predicted value of the velocity. In contrast, Taira and Colonius [21] developed a fully implicit IB method. The key idea was to formulate a saddle point problem in which the velocity field is coupled to both the pressure and the Lagrangian forces fields constituting distributed Lagrange multipliers (DLMs) to simultaneously impose incompressibility and no-slip constraints. The system was solved by applying the projection method. That study was then paired with the lattice Green's function to efficiently simulate external flows [22], and more recently it was extended to simulate FSI flows [23]. A similar saddle point formulation was also used for simulating flows around rigid bodies [24–27] and two-phase flows [28,29] and for addressing the FSI configurations of thick deformable bodies [30]. A slightly different semi-implicit formulation coupling Lagrangian forces and heat fluxes with a predicted (i.e. non-divergence free) velocity field was proposed in [31], and subsequently it was used for the investigation of $3D$ moving boundary flows [32] and natural convection confined flows developing around hot and cold cylinders [33,34].

The current study aims at extending the toolbox of available numerical IB frameworks by providing an alternative way of fully coupling between the flow properties and the Lagrangian forces introduced to impose the kinematic constraints of no-slip on the surface of an immersed body. The novel IB formulation described herein shares ideas with the study of Taira and Colonius [21] but at the same time gives a new perspective for the coupling of DLMs and primitive flow variables by adopting the well-known SIMPLE approach of Patankar and Spalding [35]. The key idea is to regard the entire IB method as a SIMPLE approach by coupling between the *corrections* of pressure, velocity and the Lagrangian forces and then utilizing them for imposing both incompressibility and no-slip constraints. As a novel aspect, the developed formulation is reduced to the solution of a Poisson-body forces system of equations constituting a *regularized* saddle point, that can be conveniently transformed into equivalent positive definite system [36]. We then propose a physically justified approximation of the system allowing for immense decrease in memory consumption compared to the previously developed numerical methodologies [31,32]. Additionally, we put an emphasis on the portability of the developed methodology to provide its convenient embedding into any available solvers of the Laplace and Helmholtz equations. A comprehensive verification study was next performed by applying the developed method to the simulation of representative $2D$ and $3D$ steady and supercritical flows. The study is summarized by presenting the efficiency characteristics of the developed methodology and

discussing the steps to be taken for further adapting it for the simulation of moving boundary and two-way coupled FSI configurations.

## 2. Theoretical background

### 2.1. Governing non-dimensional equations

We start with a brief formulation of the NS equations governing the incompressible shear- and thermally driven flows considered in the current study. According to the formalism of the IB method, additional volumetric body forces and heat fluxes are introduced into the momentum and energy equations, respectively, constituting the dynamic conditions imposed by the immersed body on the surrounding flow. The equations governing the kinematic constraints of no-slip and a prescribed temperature are then formulated on the surface of the immersed body to achieve closure of the overall system. In summary, the incompressible isothermal shear-driven flow is governed by a system of continuity and NS Eqs. (1):

$$\begin{cases} \nabla\cdot\boldsymbol{u}=0, \\ \dfrac{\partial\boldsymbol{u}}{\partial t}+(\boldsymbol{u}\cdot\nabla)\boldsymbol{u}=-\nabla p+\dfrac{1}{Re}\nabla^2\boldsymbol{u}+\boldsymbol{f}, \\ \boldsymbol{u}_s=\boldsymbol{U}^{\Gamma}, \end{cases} \tag{1}$$

the non-isothermal natural convection flow is governed by a system of continuity, NS and energy Eqs. (2):

$$\begin{cases} \nabla\cdot\boldsymbol{u}=0, \\ \frac{\partial\boldsymbol{u}}{\partial t}+(\boldsymbol{u}\cdot\nabla)\boldsymbol{u}=-\nabla p+\sqrt{\frac{Pr}{Ra}}\nabla^2\boldsymbol{u}+\theta\vec{e}_z+\boldsymbol{f}, \\ \frac{\partial\theta}{\partial t}+(\boldsymbol{u}\cdot\nabla)\theta=\frac{1}{\sqrt{PrRa}}\nabla^2\theta+q, \\ \boldsymbol{u}_s=\boldsymbol{U}^{\Gamma},\theta_s=\theta^{\Gamma}, \end{cases} \tag{2}$$

and the non-isothermal forced convection flow is governed by a system of continuity, NS and energy Eqs. (3):

$$\begin{cases} \nabla\cdot\boldsymbol{u}=0, \\ \frac{\partial\boldsymbol{u}}{\partial t}+(\boldsymbol{u}\cdot\nabla)\boldsymbol{u}=-\nabla p+\frac{1}{Re}\nabla^2\boldsymbol{u}+\frac{Ra}{PrRe^2}\theta\vec{e}_z+\boldsymbol{f}, \\ \frac{\partial\theta}{\partial t}+(\boldsymbol{u}\cdot\nabla)\theta=\frac{1}{PrRe}\nabla^2\theta+q, \\ \boldsymbol{u}_s=\boldsymbol{U}^{\Gamma},\theta_s=\theta^{\Gamma}. \end{cases} \tag{3}$$

The unknown functions introduced into the system of Eqs. (1) are the non-dimensional velocity vector field $\boldsymbol{u}=(u,v,w)$, the pressure field $p$, and the volumetric force field $\boldsymbol{f}$ exerted by the immersed body on the surrounding flow. $\boldsymbol{u}_s$ corresponds to the values of the velocity components interpolated from the surrounding field $\boldsymbol{u}$ to the surface of the immersed body, $\boldsymbol{U}^{\Gamma}$ is the value of the prescribed non-dimensional velocity of the surface of the immersed body, and $t$ is non-dimensional time. Note that for the rigid stationary immersed bodies considered in the current study, $\boldsymbol{U}^{\Gamma}=0$ on the entire surface of the body. Compared to the system of Eqs. (1), the system of Eqs. (2)-(3) include additional unknown functions $\theta$ and $q$ corresponding to the non-dimensional temperature and volumetric heat flux, respectively. $\vec{e}_z$ is a unit vector in the vertical ($z$) direction. The source term $\theta\vec{e}_z$ appearing in the right-hand side (RHS) of the momentum equation of both systems reflects the effects of buoyancy, as follows from the Boussinesq approximation; $\theta_s$ corresponds to the temperature values interpolated from the surrounding field $\theta$ to the surface of the immersed body; and $\theta^{\Gamma}$ is the value of the prescribed non-dimensional temperature of the surface of the immersed body.

The normalized system of Eqs. (1) is governed by a single non-dimensional parameter, namely, the Reynolds number, $Re=LU_0/\nu$ (where $\nu$ is the liquid kinematic viscosity), resulting from utilizing $L$, $U_0$, $L/U_0$, $\rho{U_0}^2$, and $\rho{U_0}^2/L$ scales for normalizing the length, velocity, time, pressure and force density, respectively. The normalized system of Eqs. (2) is governed by two non-dimensional parameters, namely, the Rayleigh number, $Ra=\frac{g\beta}{\nu\alpha}\Delta TL^3$ (where $g$ is the gravitational acceleration, $\beta$ is the liquid coefficient of thermal expansion, $\Delta T$ is the temperature difference between the maximal and the minimal temperatures of the liquid, and $\alpha$ is the liquid thermal diffusivity) and the Prandtl number, $Pr=\nu/\alpha$, where both numbers are obtained by utilizing $L$, $\sqrt{g\beta L\Delta T}$, $L/U$, $\rho U^2$, $\rho U^2/L$ and $\rho C_p\Delta T/(L/U)$ (where $C_p$ is the liquid specific heat capacity) scales for normalizing the length, velocity, time, pressure, and force and heat flux densities, respectively. The normalized system of Eqs. (3) is governed by three non-dimensional parameters, namely, the Reynolds number $Re$, the Prandtl number $Pr$, and the Rayleigh number $Ra$, as defined above, which were obtained by utilizing $L$, $U_0$, $L/U_0$, $\rho{U_0}^2$, $\rho{U_0}^2/L$, and $\rho C_p\Delta T/(L/U_0)$ scales for normalizing the length, velocity, time, pressure, and force and heat flux densities, respectively. For all the simulations presented in this study, a constant value of $Pr=0.7$, corresponding to air, was used.

### 2.2. Immersed boundary functionality built into the SIMPLE approach

The numerical solution of any system governed by incompressible continuity and NS equations begins with choosing a strategy for pressure-velocity coupling and with the temporal and spatial discretization of the corresponding differential operators. Systems incorporating the IB method functionality must also be provided with ways to calculate additional volumetric body forces and heat fluxes. We first give all the details of the steps taken to solve the system of Eqs. (1) and then continue the discussion by pointing out what needs to be done to extend the developed methodology for solving the systems of Eqs. (2) and (3).

The SIMPLE algorithm [35] (see also Appendix A for a brief description of the SIMPLE algorithm) was used for pressure-velocity coupling and was extended further with a built-in IB method functionality, leading to reformulating the momentum and kinematic constraint equations as:

$$\frac{3\boldsymbol{u}^* - 4\boldsymbol{u}^n + \boldsymbol{u}^{n-1}}{2\Delta t} = -\nabla p^n + \frac{1}{Re}\nabla^2\boldsymbol{u}^* + \boldsymbol{R}[\boldsymbol{F}^n] - (\boldsymbol{u}\cdot\nabla\boldsymbol{u})^n, \tag{4}$$

$$\frac{3\boldsymbol{u}'}{2\Delta t} = -\nabla p' + \boldsymbol{R}[\boldsymbol{F}'], \tag{5}$$

$$\boldsymbol{I}[\boldsymbol{u}^{n+1}] = \boldsymbol{U}^{\Gamma}, \tag{6}$$

where the standard second-order finite volume method [37] and the second-order backward finite difference were utilized for the spatial and temporal discretizations, respectively, and $\boldsymbol{u}'$, $p'$ and $\boldsymbol{F}'$ correspond to corrections of the velocity, pressure, and volumetric force fields, respectively. Compared to the original SIMPLE algorithm [35], the current formulation is extended by the $\boldsymbol{R}[\boldsymbol{F}^n]$ and $\boldsymbol{R}[\boldsymbol{F}']$ terms now included in the $RHS$ of Eq. (4) and Eq. (5), respectively, and by introducing an additional equation, Eq. (6), constituting a kinematic no-slip constraint on the surface of the immersed body. In this extended formulation, $\boldsymbol{R}$ and $\boldsymbol{I}$ correspond to two adjoint operators, namely, the regularization and interpolation operators, introduced to exchange information between the Lagrangian grid (given by a series of Lagrangian points, determining the surface of the immersed body) and the underlying Eulerian grid (typically constituting a uniform structured grid), on which the solution of discretized Eqs. (4) – (6) is obtained. The formal definition of the $\boldsymbol{R}$ and $\boldsymbol{I}$ operators smearing the Lagrangian forces over the underlying Eulerian grid and interpolating velocity components from the Eulerian grid to the Lagrangian points of the immersed body can vary, depending on the specific configuration. We determine the $\boldsymbol{R}$ and $\boldsymbol{I}$ operators by utilizing the discrete Dirac delta functions introduced in [16], as detailed in Appendix B.

Similarly to the original SIMPLE algorithm, $\boldsymbol{u}^*$ is the predicted velocity field obtained by the solution of Eq. (4) utilizing the pressure and volumetric force fields taken from the previous time step. For this reason, the predicted velocity field is non-solenoidal and also does not satisfy the kinematic constraint of no-slip on the surface of the immersed body, so that to proceed to the next time step all the flow fields should be corrected as $\boldsymbol{u}^{n+1} = \boldsymbol{u}^* + \boldsymbol{u}'$, $p^{n+1} = p^n + p'$, and $\boldsymbol{F}^{n+1} = \boldsymbol{F}^n + \boldsymbol{F}'$ to satisfy the fully constrained system of continuity and NS equations. The standard step of the original SIMPLE algorithm [35] is then replaced to convert the continuity equation into an equation for pressure and Lagrangian forces corrections. By taking the divergence of both sides of Eq. (5) and by using $\nabla\cdot\boldsymbol{u}^{n+1} = 0$, we obtain:

$$-\nabla^2 p' + \nabla\cdot\left(\boldsymbol{R}[\boldsymbol{F}']\right) = -\frac{3}{2\Delta t}\nabla\cdot\boldsymbol{u}^*. \tag{7}$$

Eq. (7), obtained in this way, constitutes a modified Poisson equation, which is unclosed, since it contains both $p'$ and $\boldsymbol{F}'$ as unknown fields. Closure is achieved by combining it with the kinematic constraint of no-slip, formulated by Eq. (6), which now requires us to express $\boldsymbol{u}^{n+1}$ in terms of $p'$ and $\boldsymbol{F}'$. The procedure is performed in two steps. In the first step, Eq. (5) is rewritten to express the velocity correction field as:

$$\boldsymbol{u}' = -\frac{2}{3}\Delta t\left(\nabla p' - \boldsymbol{R}[\boldsymbol{F}']\right). \tag{8}$$

In the second step, we exploit the linearity of the interpolation operator $\boldsymbol{I}$ (see Appendix B) with respect to the interpolated fields, so that $\boldsymbol{I}[\boldsymbol{u}^{n+1}] = \boldsymbol{I}[\boldsymbol{u}^*] + \boldsymbol{I}[\boldsymbol{u}']$ and $\boldsymbol{I}\left[-\nabla p' + \boldsymbol{R}[\boldsymbol{F}']\right] = -\boldsymbol{I}[\nabla p'] + \boldsymbol{I}\boldsymbol{R}[\boldsymbol{F}']$. We then reformulate Eq. (6) as:

$$-\boldsymbol{I}[\nabla p'] + \boldsymbol{I}\boldsymbol{R}[\boldsymbol{F}'] = \frac{3}{2}\frac{\boldsymbol{U}^{\Gamma} - \boldsymbol{I}[\boldsymbol{u}^*]}{\Delta t}. \tag{9}$$

Finally, the closed Poisson-body forces system of equations is formulated, and is written in block-matrix form as:

$$\begin{bmatrix} -\nabla^2(\ ) & \nabla\cdot(\boldsymbol{R}[\ ]) \\ \boldsymbol{I}[\nabla(\ )] & -\boldsymbol{I}\boldsymbol{R}[\ ] \end{bmatrix}\begin{bmatrix} p' \\ \boldsymbol{F}' \end{bmatrix} = \begin{bmatrix} -\frac{3}{2\Delta t}\nabla\cdot\boldsymbol{u}^* \\ \frac{3}{2}\frac{\boldsymbol{I}[\boldsymbol{u}^*]-\boldsymbol{U}^{\Gamma}}{\Delta t} \end{bmatrix}. \tag{10}$$

The sequence of operations of the developed algorithm, in order of their execution, is summarized in Table 1.

Note, it is assumed that a single correction of the predicted $p^n$, $\boldsymbol{u}^n$, and $\boldsymbol{F}^n$ fields at time step $n$ is sufficient to obtain the divergence-free velocity field $\boldsymbol{u}^{n+1}$ and the converged values of both the pressure field $p^{n+1}$ and the Lagrangian forces

**Table 1**
Sequence of operations for proceeding by a single time step.

| | |
|---|---|
| 1. | Guess or take from the previous time steps $p^n$, $\boldsymbol{u}^{n,n-1}$, and $\boldsymbol{F}^n$. |
| 2. | Solve the momentum Eq. (4) to obtain $\boldsymbol{u}^*$. |
| 3. | Solve the system of Eqs. (10) to *simultaneously* obtain $p'$ and $\boldsymbol{F}'$ fields. |
| 4. | Calculate $\boldsymbol{u}'$ by using Eq. (8). |
| 5. | Proceed to the next time step by assigning $p^{n+1} = p^n + p'$, $\boldsymbol{u}^{n+1} = \boldsymbol{u}^n + \boldsymbol{u}'$, $\boldsymbol{F}^{n+1} = \boldsymbol{F}^n + \boldsymbol{F}'$. |

$\boldsymbol{F}^{n+1}$, as follows from Step 5 of the developed algorithm (see Table 1). This assumption is reasonable for sufficiently small time steps $\Delta t \sim O(10^{-3})$, and initial conditions for the pressure and the velocity fields that would satisfy the momentum and the continuity equations (e.g., assigning zero values to all the flow fields). In fact, for all the configurations analyzed in the current study, the maximum value of $\nabla \cdot \boldsymbol{u}^{n+1}$ was less than $10^{-13}$, while the kinematic constraint of no-slip was met up to the value[1] of $10^{-6}$ immediately after the first correction in any given time step. At the same time for larger time steps or for the direct numerical simulation (DNS) of turbulent flows, it may be necessary to introduce underrelaxation factors along with internal iterations and to repeat steps 1–5 of the algorithm to meet the divergence-free and kinematic constraints of no-slip with the specified accuracy before proceeding to the next time step. For a more explicit discussion regarding the limitations that should be imposed on the time step and underrelaxation values to optimize the convergence of the SIMPLE-family algorithms, Refs. [38,39] should be consulted. It should also be noted that the currently developed methodology allowing us to obtain the pressure and volumetric force corrections in a coupled manner can be adapted with reasonable effort to any derivatives of the SIMPLE approach, e.g., the SIMPLEC, SIMPLER or PISO algorithms described in [40].

### 2.3. Solution of the momentum equation

Momentum equation Eq. (4) was solved by a solver based on an algorithm implementing the tensor product factorization (TPF) method combined with the Thomas solver [2]. The latter, which is a direct solver, is superior to the commonly used iterative BiCgStab [1] algorithm, when used for obtaining the matrix-vector product of the inverse Helmholtz operator; it has been extensively verified in our previous studies [41,31,32]. With emphasis on the portability of the developed methodology, the solver is utilized in this study in a black-box manner by modifying only the $RHS$ of Eq. (4). In practice, any other solver capable of giving the matrix-vector product of the inverse Helmholtz operator (either exact, e.g. [42], or approximate, e.g. [43]) can be utilized instead.

### 2.4. Solution of the Poisson-body forces system

The Poisson-body forces system presented in Eq. (10) is similar to that obtained in the study of [21], in the sense that it couples the pressure and the force correction fields, both treated as DLMs to simultaneously satisfy the incompressibility and no-slip constraints, respectively. Nonetheless, there is a major difference between the two approaches in that the system developed in this study contains an operator $\boldsymbol{IR}$ located in the right bottom corner of the system of Eq. (10) and acting on the force corrections $\boldsymbol{F}'$. Thus, the current formulation constitutes a regularized saddle point problem, with a convenient solution to this problem being the most important contribution of the developed methodology. The role of the $\boldsymbol{IR}$ operator will be discussed in detail below (we note here that the presence of this operator significantly facilitates the solution of the problem).

We also note that the specific aim of the current study is to address the most challenging configuration arising when simulating confined flows. In this case, the Neumann boundary conditions are imposed on the pressure (or pressure correction) fields at the solid boundaries. As a result, the Poisson-body forces system is, on the one hand, too poorly conditioned to be efficiently solved by using Krylov space iterative methods (e.g., BiCgStab [1]), and, on the other hand, too large to be handled by a sparse direct solver (e.g., MUMPS [44,45]) for realistic 3D configurations. An additional challenge is that the problem is stationary, so that the Tailor series expansion of the inverse Laplacian in $\Delta t$, as discussed in [43] for the simulation of general incompressible flows and further extended in [21] by coupling it with the IB method formalism, cannot be directly implemented. The solution to this problem lies in preconditioning the Laplacian with its approximate inverse, as was proposed by [43] and successfully implemented by [25]. Recalling, however, that in the framework of the current study the focus is on the portability of the developed IB method (i.e., the use of previously developed solvers that do not have the immersed boundary capability in a black-box manner), we propose an alternative way of solving the Poisson-body forces system of equations based on a physically justified approximation of the matrix-vector product of operator $\boldsymbol{IR}$.

#### 2.4.1. Approximation of the matrix-vector product of operator $\boldsymbol{IR}$

The impact of the $\boldsymbol{IR}$ operator can be quantified by obtaining the product of matrix $\boldsymbol{I}$ by matrix $\boldsymbol{R}$, each containing the "weights" determined by the distances between the corresponding locations of the Lagrangian and Eulerian grid points and

[1] Additional details for meeting the kinematic constraint of no-slip are given in section 2.4.2.

calculated by utilizing the discrete Dirac delta function [16]. Note that despite the fact that $\boldsymbol{I}$ and $\boldsymbol{R}$ operators are adjoint to each other, the two matrices are not reciprocal, so their product is not an identity matrix. In reality, however, it would be preferable to avoid explicit building and multiplication of the $\boldsymbol{I}$ and $\boldsymbol{R}$ matrices when calculating the product of $\boldsymbol{IR}$ by the vector of force corrections $\boldsymbol{F}'$; such a calculation could be very costly for realistic $3D$ configurations containing $O(10^5)$ or even more Lagrangian points. To develop a formally justified approximation of the above matrix-vector product, we next focus on the fact that if the same discrete Dirac delta functions are used in the interpolation $\boldsymbol{I}$ and regularization $\boldsymbol{R}$ operators, the sum of each row (or each column) of the matrix $\boldsymbol{IR}$ is equal to some constant $\lambda$. This property is a direct consequence of the so-called "sum of squares rule" imposed by [46] when constructing any discrete Dirac delta function:

$$\Sigma_i[\phi(r-i)]^2 = \lambda, \forall r, \tag{11}$$

where all the summations are performed for the integers $i$, such that $-\infty < i < +\infty$ and $\lambda$ is some constant. The need for Eq. (11) arises in situations where a quantity in a specific Lagrangian point is first regularized to the underlying Eulerian grid and the resulting field is interpolated back to the same Lagrangian point. The rule grantees that the result is independent of the position of the Lagrangian point relative to the Eulerian grid. For the currently used discrete Dirac delta function [16], this constant is equal to $\lambda = 0.5$.[2] Note, too, that the above rule is also preserved when function $\phi$ is utilized in $2D$ and $3D$ discrete Dirac delta functions. Therefore, the result of applying the operator $\boldsymbol{IR}$ to any uniform field is the field itself multiplied by a factor of 0.5, as follows straight forwardly from Eq. (11). Although the realistic force corrections vector $\boldsymbol{F}'$ is not uniform, it is still possible to use the above property and to approximate the product $\boldsymbol{IR} \times \boldsymbol{F}'$ based on the following observations:

- **Compact support of the *IR* operator.** As a result of the compact support of $\phi$ (only one and a half grid points in each direction), matrices $\boldsymbol{I}$ and $\boldsymbol{R}$ are extremely sparse, with a typical row containing only a few hundred non-zero entries, while the dimension of a typical row of the $\boldsymbol{IR}$ matrix is $O(10^5)$ for realistic $3D$ configurations. The matrix $\boldsymbol{IR}$, which couples between all the Lagrangian quantities, is also sparse, because only a very narrow neighborhood of the Eulerian grid is affected by the regularized Lagrangian quantities that are in close proximity to each other (since $\boldsymbol{R}$ is sparse) and, in turn, a very limited number of the Eulerian quantities simultaneously affect the given Lagrangian point by interpolation (since $\boldsymbol{I}$ is sparse).
- **Smoothing properties of the *IR* operator.** When calculated implicitly, the kinematic constraints suffer from spurious oscillations in the volumetric (or surface) forces (or heat fluxes). The observed inaccuracies stem from the ill-posed integral equation of the first kind for the surface stresses utilizing discrete smeared Dirac delta functions, as detailed in [23]. The same study proposed an efficient technique to filter out high frequencies and to converge the surface stresses and forces to physically correct parameters acting on the surface of the immersed body. Remarkably, an effect similar to filtering out the high-frequency oscillations can be achieved by applying the product of the $\boldsymbol{IR}$ operator to the vector of kinematic constraints. To demonstrate this effect, we analyzed the representative fields of the force density corrections $F'_x$ and $F'_y$ by taking a snapshot of the flow developing around a cylinder placed within the lid-driven cavity (one of the numerical experiments described below) at $Re = 100$ obtained on $100^2$ grid with $\Delta x = \Delta y = 10^{-2}$. The fields represented by black solid lines in Figs. 1 (a) and (b), suffer from spurious oscillations. In addition to the obtained $F'_x$ and $F'_y$ distributions, we present their smoothed counterparts (both indicated by solid red lines) obtained by calculating the $2\boldsymbol{IR} \times F'_x$ and $2\boldsymbol{IR} \times F'_y$ products.[3] These products determine the actual (physically correct) contribution of the Lagrangian quantities to the $\boldsymbol{IR} \times \boldsymbol{F}'$ term, entering Eq. (10), discretized on the Eulerian grid; i.e. due to the inclusion of the $\boldsymbol{IR}$ term into the currently developed IB formulation, the field actually contributed by the force density corrections is smooth (for sufficiently regular geometries) and can be approximately considered as piece-wise constant in the vicinity of each Lagrangian point for a reasonably dense grid resolution.

As follows from the above discussion, the action operator $\boldsymbol{IR}$ is local (i.e., affects only the quantities in a small number of neighboring Lagrangian points), and its product by the implicitly obtained field of kinematic constraints containing high-frequency oscillations can be approximated by replacing the original field with its smoothed counterpart. In practice, this allows us to replace the operator $\boldsymbol{IR}$ with a $\frac{1}{2}\mathbf{I}$ matrix (where $\mathbf{I}$ is the identity matrix) by assuming that the values of the smoothed Lagrangian quantities located in close proximity are approximately the same. Note that the obtained result can be seen as a generalization of the approximation recently proposed in [47,48] for the explicit formulation of the IB method. The approximate form of the Poisson-body forces system can thus be written as:

$$\begin{bmatrix} -\nabla^2(\ ) & \nabla\cdot(\boldsymbol{R}[\ ]) \\ \boldsymbol{I}[\nabla(\ )] & -\frac{1}{2}\mathbf{I} \end{bmatrix} \begin{bmatrix} p' \\ \boldsymbol{F}' \end{bmatrix} = \begin{bmatrix} -\frac{3}{2\Delta t}\nabla\cdot\boldsymbol{u}^* \\ \frac{3}{2}\frac{\boldsymbol{I}[\boldsymbol{u}^*]-\boldsymbol{U}^\Gamma}{\Delta t} \end{bmatrix}. \tag{12}$$

[2] See Refs. [46,16] for the full list of rules that should be imposed when constructing discrete Dirac delta function.

[3] The results were multiplied by a factor of 2 to make the visual comparisons easier.

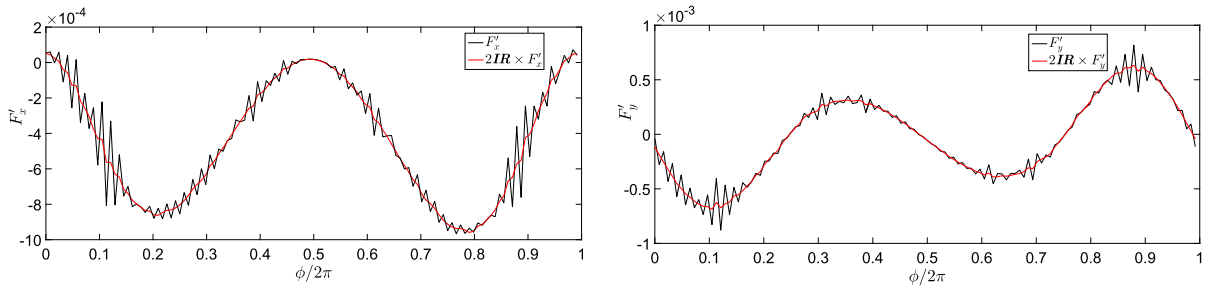



**Fig. 1.** The force correction density as a function of the angle on the surface of the cylinder immersed within lid-driven cavity flow: (a) in the $x$ direction; (b) in the $y$ direction. Black solid lines correspond to the original force correction density values, as obtained from the simulation. Red solid lines correspond to a processed force correction density field obtained by doubling the product of the $\boldsymbol{IR}$ operator by the original $F'_x$ and $F'_y$ fields. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

Note that the structure of the system of Eqs. (12) is typical of the systems solved by the penalty method. However, unlike the penalty method, which replaces the solution of an originally constrained problem with a series of solutions of unconstrained problems and for which the value of the constant multiplying the identity matrix is not obvious, the system of Eq. (12) is constrained, albeit approximated, and the $-\frac{1}{2}\mathbf{I}$ term has a physical basis. In all numerical experiments performed in the framework of the current study, the solution of both forms of the Poisson-body forces system, i.e., Eq. (10) and Eq. (12), yielded similar results, although the solution of the approximated system was characterized by much lower memory consumption and much faster convergence, as detailed below.

*2.4.2. Numerical procedure*

An important feature of the developed algorithm is that discretization of both $\nabla \cdot (\boldsymbol{R}[\,])$ and $\boldsymbol{I}[\nabla(\,)]$ operators is implemented implicitly, thereby constituting an additional challenge compared, for example, to implicit implementation of the interpolation $\boldsymbol{I}$ and regularization $\boldsymbol{R}$ operators commonly used in previous studies, see e.g. [21,25,31]. Implicit discretization of the $\nabla \cdot (\boldsymbol{R}[\,])$ and $\boldsymbol{I}[\nabla(\,)]$ operators can be implemented by using different strategies. The strategy implemented by us for simulating a 2D lid-driven cavity flow with an embedded cylinder, as appears in the complementary material, is based on the subtraction of two matrices. The key idea is to arrange the force weights contributed by each Lagrangian point into a single matrix, the first dimension of which is equal to the total number of Lagrangian points, and the second dimension, is equal to the total number of Eulerian cells. The divergence can then be conveniently obtained by subtracting the corresponding rows (in accordance with the global numbering of the velocity field) of the matrices so constructed. Note that the above procedure must be repeated for each velocity component. Utilizing the above strategy for a 3D configuration can be very costly in terms of memory consumption. The straightforward way to remedy the memory limitation, in this case, is to decompose the 3D domain into 2D sections and repeat the procedure described above separately for each section.

For the sake of conciseness, let us now denote the following: the Laplace operator acting on the pressure corrections field and taken with a negative sign as $-\nabla^2(\,) \equiv \boldsymbol{L}$; the divergence acting on the force corrections field regularized from the surface of the immersed body as $\nabla \cdot (\boldsymbol{R}[\,]) \equiv \boldsymbol{B}^T$; and the $-\boldsymbol{IR}$ operator acting on the regularized force corrections field and its approximation entering Eq. (12) as $-\boldsymbol{IR}[\,] \equiv \boldsymbol{C}$ and $-\frac{1}{2}\mathbf{I} \equiv \boldsymbol{C}$, respectively. It should be noted that $\nabla \cdot (\boldsymbol{R}[\,])$ and $\boldsymbol{I}[\nabla(\,)]$ are determined by transpose matrices when the same discrete Dirac delta functions are used by the regularization and interpolation operators. By following the above notation, the Poisson-body forces system can be reformulated as:

$$\begin{bmatrix} \boldsymbol{L} & \boldsymbol{B}^T \\ \boldsymbol{B} & \boldsymbol{C} \end{bmatrix} \begin{bmatrix} p' \\ \boldsymbol{F}' \end{bmatrix} = \begin{bmatrix} RHS_{p'} \\ RHS_{\boldsymbol{F}'} \end{bmatrix}. \tag{13}$$

For $2D$ configurations, the system of Eqs. (13) can be conveniently solved by utilizing a direct sparse solver, e.g., by performing $LU$ decomposition, as demonstrated by the MATLAB script provided in the complementary material. For $3D$ configurations, however, the above system is severely ill conditioned, and thus cannot be solved by the direct solver. Instead, we utilize a Schur complement approach to decompose the system of Eqs. (13) and to separately calculate the $\boldsymbol{F}'$ and $p'$ fields by:

$$\begin{cases} \boldsymbol{F}' = (\boldsymbol{B}\boldsymbol{L}^{-1}\boldsymbol{B}^T - \boldsymbol{C})^{-1}(\boldsymbol{B}\boldsymbol{L}^{-1}RHS_{p'} - RHS_{\boldsymbol{F}'}), \\ p' = \boldsymbol{L}^{-1}(RHS_{p'} - \boldsymbol{B}^T\boldsymbol{F}'). \end{cases} \tag{14}$$

It should be stressed that there is no need to explicitly calculate an inverse of any matrix at any stage of the solution. In fact, $\boldsymbol{B}\boldsymbol{L}^{-1}\boldsymbol{B}^T$ is built by solving a Laplace problem for each column of $\boldsymbol{B}^T$ and then immediately multiplying matrix $\boldsymbol{B}$ by the obtained vector. In this way, the building of intermediate large matrix $\boldsymbol{L}^{-1}\boldsymbol{B}^T$ is avoided [31]. Note that this most time-

consuming part of the algorithm can be pre-computed once at the beginning of the simulation.[4] This part of the algorithm is also "embarrassingly parallel" in the sense that the $\boldsymbol{BL}^{-1}\boldsymbol{B}^T$ product can be calculated independently for different columns of $\boldsymbol{B}^T$ on different servers without the need for the explicit use of MPI, so that the overall process speedup is linear if the same hardware is used. Although the resulting small matrix $\boldsymbol{BL}^{-1}\boldsymbol{B}^T$ should be collected in a single server at the end of the calculation, this procedure is performed only once before the time integration process starts. The products $\boldsymbol{BL}^{-1}RHS_{p'}$ and $\boldsymbol{L}^{-1}(RHS_{p'}-\boldsymbol{B}^T\boldsymbol{F}')$ are again calculated by solving the Laplace problem and by utilizing sparse matrix-vector multiplications, as detailed in [31,32]. In all the currently performed numerical experiments, the Laplace problem was solved by applying the TPF solver[5] [2].

We now discuss a strategy for solving a system of equations containing the $[(\boldsymbol{BL}^{-1}\boldsymbol{B}^T-\boldsymbol{C})]$ operator. According to our previous experience [31], the matrix $[(\boldsymbol{BL}^{-1}\boldsymbol{B}^T]$ is always full of non-zero entries, the vast majority of which are very small ($O(10^{-16})$ and smaller). Storing all these entries is prohibitively expensive for realistic $3D$ configurations. For this reason, a sparsing threshold was set such that values below it are not stored in the $[\boldsymbol{BL}^{-1}\boldsymbol{B}^T]$ matrix. The value of the sparsing threshold can vary for each problem and depends on the specific discrete Dirac delta functions utilized. As a rule of thumb, the sparsing threshold should be at least 4–5 orders of magnitude less than the absolute value of the maximal entry of the matrix $[\boldsymbol{BL}^{-1}\boldsymbol{B}^T]$. For a more formal choice of the sparsity threshold value, one should make sure that the error of meeting the kinematic constraint of no-slip is less than the discretization error of the currently used method. In fact, in all the numerical experiments performed in the current study, the threshold value for the solution of the system of Eq. (14) was set to $10^{-3}$. For this threshold, the kinematic constraint of no-slip was met up to $10^{-6}$ and $10^{-7}$ for $\Delta t=5\times10^{-3}$ and $\Delta t=10^{-3}$, respectively, so that it is indeed bounded by the discretization error of the currently utilized second-order (in time and in space) finite volume method.

After the matrix $[\boldsymbol{BL}^{-1}\boldsymbol{B}^T]$ has been built, the operator $[\boldsymbol{BL}^{-1}\boldsymbol{B}^T-\boldsymbol{C}]$ can be calculated by simply subtracting the two matrices. A very important observation here is that a non-zero matrix $\boldsymbol{C}$ allows for stabilization of the above system. Further approximation of $\boldsymbol{C}$ as $\boldsymbol{C}=-\frac{1}{2}\boldsymbol{I}$ allows to significantly increase the main diagonal of the operator $[(\boldsymbol{BL}^{-1}\boldsymbol{B}^T-\boldsymbol{C})]$, which turns it into well-conditioned matrix, characterized by a small (only $O(10)$) value of the condition number. As a result, the system of equations can be conveniently solved by applying the BiCgStab [1] method, which converges in no more than 3 iterations to the value of $l_\infty\le10^{-9}$ of the infinity norm of the error. Another alternative for the solution of the above system is to apply the MUMPS [44,45] direct solver for performing the $LU$ decomposition of the $[\boldsymbol{BL}^{-1}\boldsymbol{B}^T-\boldsymbol{C}]$ matrix. The $LU$ factors can be then stored on the computer hard disk and read when necessary by employing the MUMPS functionality. As a result of the intrinsically sequential nature of the back substitution procedure, as implemented in MUMPS, this strategy has been found attractive only for moderate grid resolutions, not exceeding 200 grid points in each direction.

Summarizing the above discussion, we note that the approximation $\boldsymbol{C}=-\frac{1}{2}\boldsymbol{I}$ both provided an accurate meeting of all no-slip kinematic constraints and resulted in adequate sparsity of both the $[\boldsymbol{BL}^{-1}\boldsymbol{B}^T-\boldsymbol{C}]$ matrix and its $LU$ factors. In addition, the first equation of the system of Eqs. (14) can be conveniently solved by applying the BiCgStab method, which demonstrates rapid convergence when the above approximation of matrix $\boldsymbol{C}$ is utilized. This is in contrast to the exact implementation $\boldsymbol{C}=\boldsymbol{IR}$, which, on the one hand, resulted in very high memory consumption when storing the corresponding $LU$ factors required by the direct solver and, on the other hand, suffered from a poor convergence of the iterative solution performed by the BiCgStab method.

### 2.5. Simulation of natural convection flow

In this section, we briefly describe the procedure that must be followed to solve a system of equations Eqs. (2) and (3). We detail the steps that must be taken to solve the system of Eqs. (2). The system of Eqs. (3) is solved analogously. Following the same procedure as that performed for the NS equations governing isothermal flow (see Eq. (1)), Eqs. (2) governing natural convection flow may be rewritten in the semi-discrete form as:

$$\frac{3\theta^{n+1}-4\theta^n+\theta^{n-1}}{2\Delta t}=\frac{1}{\sqrt{PrRa}}\nabla^2\theta^{n+1}+\boldsymbol{R}(Q^{n+1})-(\boldsymbol{u}\cdot\nabla\theta)^n, \tag{15}$$

$$\boldsymbol{I}[\theta^{n+1}]=\Theta^{\Gamma}, \tag{16}$$

$$\frac{3\boldsymbol{u}^*-4\boldsymbol{u}^n+\boldsymbol{u}^{n-1}}{2\Delta t}=-\nabla p^n+\sqrt{\frac{Pr}{Ra}}\nabla^2\boldsymbol{u}^*+\boldsymbol{R}[\boldsymbol{F}^n]+\theta^{n+1}\vec{e}_z-(\boldsymbol{u}\cdot\nabla\boldsymbol{u})^n, \tag{17}$$

$$\frac{3\boldsymbol{u}'}{2\Delta t}=-\nabla p'+\boldsymbol{R}[\boldsymbol{F}'], \tag{18}$$

$$\boldsymbol{I}[\boldsymbol{u}^{n+1}]=\boldsymbol{U}^{\Gamma}, \tag{19}$$

where Eqs. (15) and (16) govern for the energy equation and the kinematic constraint of the temperature prescribed on the surface of the immersed body, and Eqs. (17)–(19) govern the momentum equation in the SIMPLE [35] formulation and the

[4] The pre-computing can be done for stationary geometries. For moving bodies, different strategies can be adopted, see e.g. [32]).

[5] Any other Laplace equation solver (exact or approximate, direct or iterative) can be used instead.

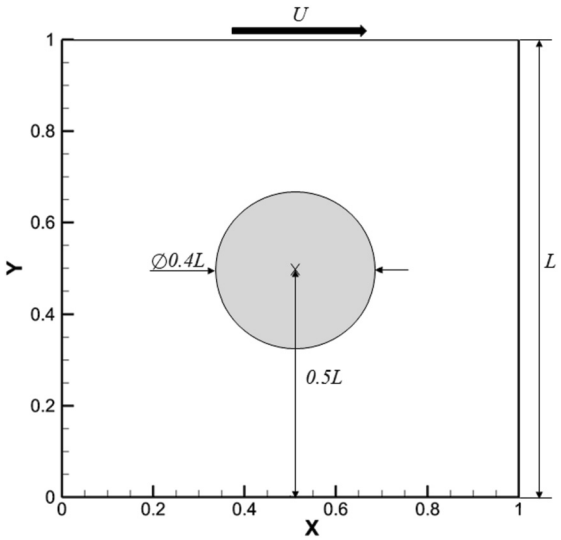



**Fig. 2.** $2D$ lid-driven cavity with an embedded cylinder in the center – physical model.

kinematic constraint of no-slip on the surface of the immersed body. Note that utilizing the above temporal discretization allows us to decouple the energy and the momentum equations and to solve them separately in each time step. Similarly to the isothermal configuration, the standard second-order finite volume method [37] and the second-order backward finite difference were utilized for the spatial and temporal discretizations, respectively. Eqs. (15) and (16) can be written in a compact block-matrix form as:

$$\begin{bmatrix} \boldsymbol{H} & \boldsymbol{R} \\ \boldsymbol{I} & 0 \end{bmatrix} \begin{bmatrix} \theta^{n+1} \\ Q \end{bmatrix} = \begin{bmatrix} RHS^{n-1,n} \\ \Theta^{\Gamma} \end{bmatrix}, \tag{20}$$

where $\boldsymbol{H} = (\frac{3}{2\Delta t}(\theta^{n+1}) - \frac{1}{\sqrt{PrRa}}\nabla^2(\theta^{n+1}))$ is the Helmholtz operator. The saddle node problem governed by Eq. (20) is then solved by applying Schur complement decomposition, leading to:

$$\begin{cases} Q = [\boldsymbol{I}\boldsymbol{H}^{-1}\boldsymbol{R}]^{-1}[\boldsymbol{I}\boldsymbol{H}^{-1}RHS^{n-1,n} - \Theta^{\Gamma}], \\ \theta = \boldsymbol{H}^{-1}[RHS^{n-1,n} - \boldsymbol{R}[Q]]. \end{cases} \tag{21}$$

As described above, the TPF method combined with the Thomas solver [2] is utilized for obtaining all the required matrix-vector products of the inverse Helmholtz operator and for pre-computing the $[\boldsymbol{I}\boldsymbol{H}^{-1}\boldsymbol{R}]$ matrix. In this case, the sparsing threshold was set to $10^{-10}$ when filling the $[\boldsymbol{I}\boldsymbol{H}^{-1}\boldsymbol{R}]$ matrix. In all the currently performed numerical experiments, it was verified that the infinity norm of the error obtained when enforcing the kinematic constraint of the temperature prescribed on the surface of the immersed body was $l_\infty \leq 10^{-8}$, which is bounded by the temporal and the spatial discretization errors of the applied finite volume method. The subsequent solution steps employed $LU$ factorization of the pre-computed $[\boldsymbol{I}\boldsymbol{H}^{-1}\boldsymbol{R}]$ matrix, which was performed by utilizing the MUMPS solver. For a more detailed description of all the stages involved in solving the system of Eqs. (20), our previous study [31] should be consulted. After the temperature field $\theta^{n+1}$ had been obtained by the solution of the system of Eqs. (20), it was then used when building the $RHS$ of Eq. (17). In the subsequent step, Eqs. (17)–(19) were solved in the same sequence as that used when solving Eqs. (4)–(6), as described above.

## 3. Results and discussion

### 3.1. $2D$ lid-driven cavity with an embedded cylinder

The flow developing within square lid-driven cavity of side $L$ with a cylinder of diameter $0.4L$ placed in the geometrical center of the cavity is examined (see Fig. 2). The top lid of the cavity moves with constant velocity $U$ in the $x$ direction, while all other cavity boundaries and the cylinder are stationary. Table 2 summarizes the verification study of the developed $2D$ solver applied to the simulation of lid-driven cavity flow (without an embedded cylinder) at $Re = 1000$. The obtained flow characteristics are compared with the data available in the literature. The criterion for reaching the steady-state solution was $l_\infty \leq 10^{-5}$ of the infinity norm calculated for the relative deviation between two consecutive time steps for all the flow variables. It can be seen that the obtained values of the maximal and minimal velocities and their spatial locations are in excellent agreement with the independent results, which successfully verifies the developed $2D$ solver.

We next present the results of the grid independence study for the lid-driven cavity with an embedded cylinder of diameter $0.4L$ placed in the cavity center. The simulations were performed for three different $Re$ values on three different

**Table 2**
Comparison of the flow characteristics obtained for the flow within 2D lid-driven cavity, $Re = 1000$.

| | Grid | $u_{min}$ | $y_{min}$ | $v_{max}$ | $x_{max}$ | $v_{min}$ | $x_{min}$ |
|---|---|---|---|---|---|---|---|
| Ref. [49] | $128^2$ | -0.3653 | 0.1744 | 0.3547 | 0.1506 | -0.4824 | 0.9037 |
| Ref. [50] | $129^2$ | -0.3829 | 0.1719 | 0.3710 | 0.1563 | -0.5155 | 0.9063 |
| Ref. [51] | $256^2$ | -0.3883 | 0.1698 | 0.3768 | 0.1564 | -0.5270 | 0.9088 |
| Current study | $512^2$ | -0.3865 | 0.1719 | 0.3749 | 0.1582 | -0.5247 | 0.9082 |

**Table 3**
Results of a grid independence study performed for $2D$ lid driven-cavity flow with a cylinder of diameter $D = 0.4$ embedded in the center at three different $Re$ values.

| | Re=1000 | | | Re=5000 | | | Re=8000 | | |
|---|---|---|---|---|---|---|---|---|---|
| Grid | $u_{min}$ | $v_{min}$ | $v_{max}$ | $u_{min}$ | $v_{min}$ | $v_{max}$ | $u_{min}$ | $v_{min}$ | $v_{max}$ |
| $128^2$ | -0.3382 | -0.4568 | 0.3253 | -0.4057 | -0.5142 | 0.3913 | -0.4017 | -0.5002 | 0.3920 |
| $256^2$ | -0.3434 | -0.4638 | 0.3297 | -0.4290 | -0.5426 | 0.4125 | -0.4358 | -0.5435 | 0.4238 |
| $512^2$ | -0.3448 | -0.4651 | 0.3309 | -0.4357 | -0.5508 | 0.4189 | -0.4469 | -0.5570 | 0.4336 |

**Table 4**
Comparison of the oscillation frequencies $f$ obtained for different $Re$ values.

| | $Re = 8600$ | $Re = 8700$ | $Re = 8800$ |
|---|---|---|---|
| Current | 0.491 | 0.489 | 0.489 |
| Ref. [51] | 0.519 | 0.519 | 0.519 |
| Rel. Dev. [%] | 5.395 | 5.780 | 5.780 |

grids, with grid resolution being doubled each time. It can be seen that the relative deviations between the corresponding extreme values of all the velocity components obtained on $256^2$ and $512^2$ grids do not exceed 0.5%, 1.5% and 2.5% for the values of $Re = 1000, 5000$, and 8000, respectively, which successfully verifies the grid independence of the results obtained on both grids. See Table 3.

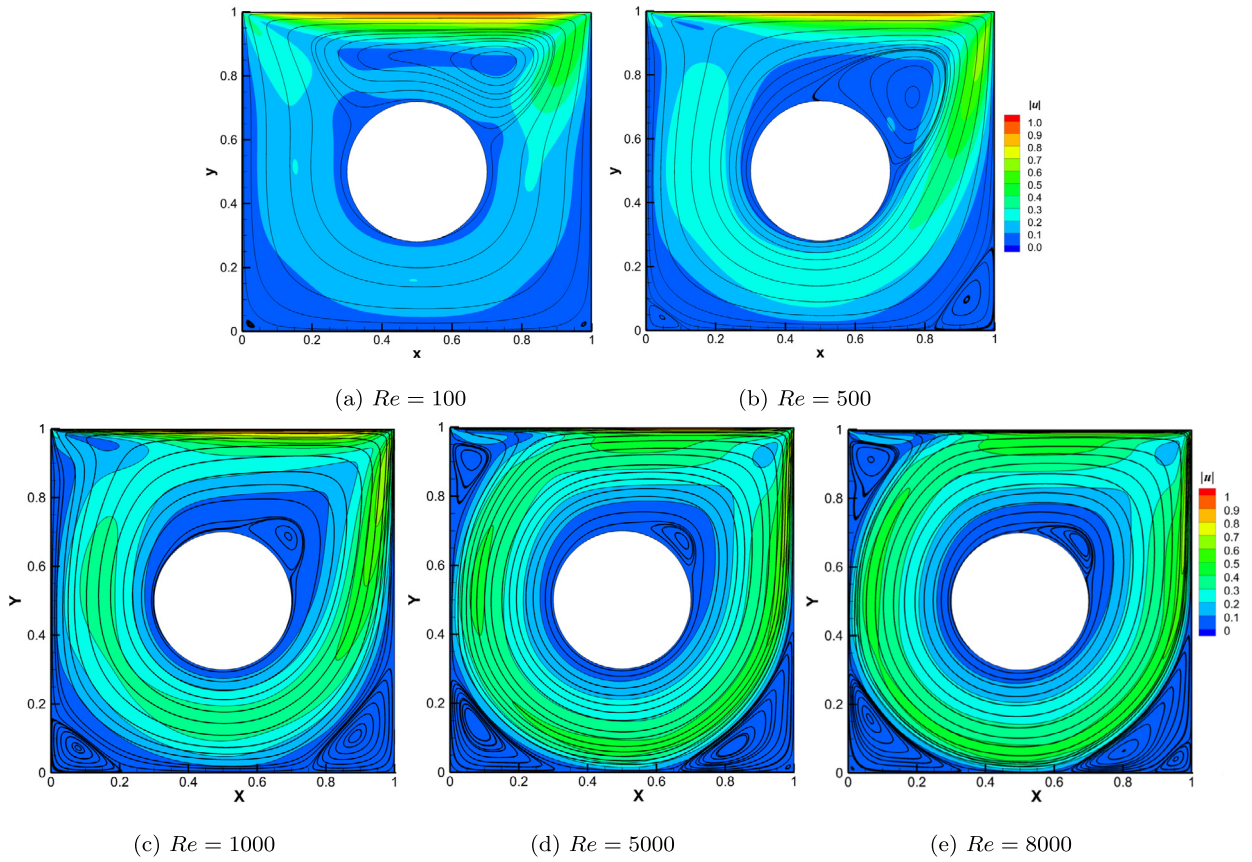
Based on the above analysis, all the time consuming simulations of supercritical flow regimes were performed on a uniform $256^2$ grid, while less time consuming simulations of steady state flows, were conducted on a $512^2$ grid.

To prevent ill-conditioning, the space between the Lagrangian points on the cylinder surface was almost the same (to within 5%) as that of the Eulerian grid. The obtained characteristics of the steady-state flow are presented in Fig. 3 in terms of streamlines and contours of the velocity magnitude for the values of $Re = 100, 500, 1000, 5000$ and 8000. Both flow characteristics are similar qualitatively and quantitatively to the corresponding data reported in Refs. [51,52].

Thereafter, we calculated the vorticity fields $\boldsymbol{\omega} = \nabla \times \boldsymbol{u}$ for $Re = 1000, 5000$ and 8000. Those vorticity fields, presented in Fig. 4, are in good qualitative agreement with the corresponding data reported in [51,53]. We confirmed the presence of a number of secondary eddies at the top left corner and both bottom corners of the cavity. Unfortunately, it was impossible to conduct a quantitative comparison of extreme vorticity values, since the data presented in [51] is not grid independent. In addition, we performed a quantitative comparison between the values of the $u_x$ and $u_y$ velocity components along the vertical and horizontal centerlines of the cavity, respectively, as obtained in this study for the value of $Re = 1000$, and the corresponding values reported in [52,54], which were acquired by photogrammetry. The data shown in Fig. 5 shows excellent agreement between our results and the independent results of both studies. The work performed in this section is summarized by comparison between our results and the independent results obtained in Ref. [51] for supercritical flow regimes. Fig. 6 presents the time evolutions of the $u_x$ velocity component and the corresponding frequency spectra measured at the control point (0.9,0.1) for the values of $Re = 8600, 8700$, and 8800. The supercritical flow is governed by single oscillating harmonics, indicating that the flow undergoes Hopf bifurcation during the transition from subcritical to supercritical flow regimes. The multipliers of the main harmonics, resulting from the non-linearity of the supercritical flow, are clearly recognizable. Remarkably, there was a small decrease in the oscillating frequency value when moving from $Re = 8600$ to $Re = 8700$. A further increase to $Re = 8800$ did not result in any variation of the oscillating frequency value of the main harmonics. Surprisingly, there were large differences between our results and the values of the $u_x$ signals acquired at the control point (0.9,0.1) reported in [51]. Despite this unexplained observation, acceptable agreement (no more than 6% discrepancy) was obtained between the two sets of results with respect to the frequency spectra values (see Table 4), as expected for slightly supercritical flows; this agreement successfully verifies the developed solver.

### *3.2. 2D laminar mixed convection flow in a lid-driven cavity with an embedded cylinder*

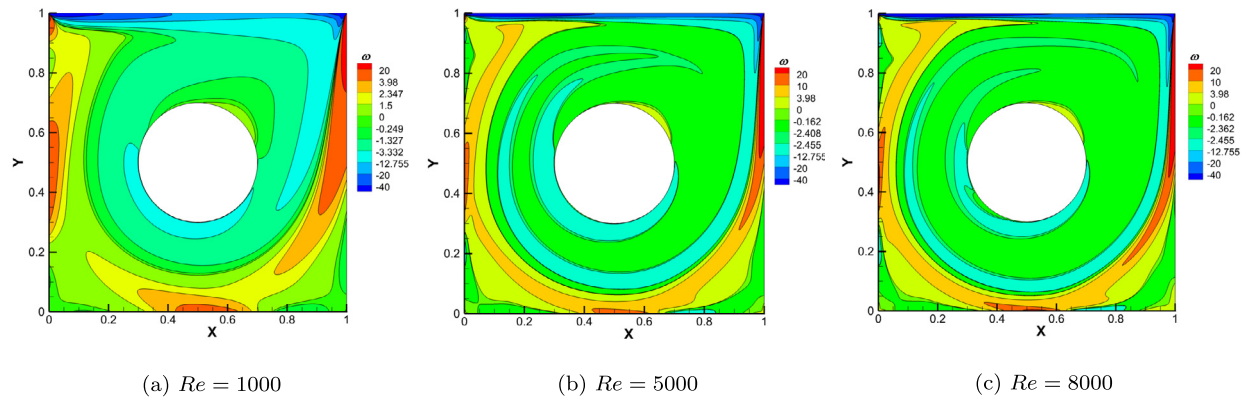
Mixed convection flow in a square lid-driven cavity of side $L$ with an embedded circular cylinder of diameter $0.4L$ placed in the center of the cavity is considered (see Fig. 7). The top lid of the cavity moves with a constant horizontal velocity




(a) $Re = 100$ (b) $Re = 500$

(c) $Re = 1000$ (d) $Re = 5000$ (e) $Re = 8000$

**Fig. 3.** Streamlines and contours of the velocity magnitude obtained for five different $Re$ values, $Re = 100, 500, 1000, 5000, 8000$.




(a) $Re = 1000$ (b) $Re = 5000$ (c) $Re = 8000$

**Fig. 4.** Vorticity contours obtained for three different $Re$ values.

$U$, while all other walls are stationary. In addition, the bottom and the top walls of the cavity are held at constant hot $\theta_H$ and cold $\theta_C$ temperatures, respectively, while both vertical walls are thermally insulated. Gravity acts in the negative direction of the $y$ axis. The cylinder surface is either held at constant cold temperature $\theta_C$ or thermally insulated. Starting with performing a grid independence study, we conducted the simulations on $50^2$, $100^2$ and $200^2$ grids for the isothermal cold cylinder. The grid convergence of the results was verified in terms of comparison of the values of the $\overline{Nu}$ number averaged over the surface of the cylinder, as presented in Table 5. The results were obtained for four different values of the Richardson, $Ri = Ra/PrRe^2$, number, $Ri = 0.01, 1, 5$, and 10. It can be seen that the maximal deviation of the $\overline{Nu}$ values obtained on $100^2$ and $200^2$ grids does not exceed 0.4%, which successfully verifies the grid independence of the results. Therefore, all the results presented in this section for a cylinder with an isothermal surface were obtained on a $200^2$ grid. For the adiabatic cylinder configuration, whose simulation requires an explicit calculation of the temperature gradient on the surface of the cylinder [55], we used even denser $300^2$ grids. The contours of the temperature obtained for the isothermal

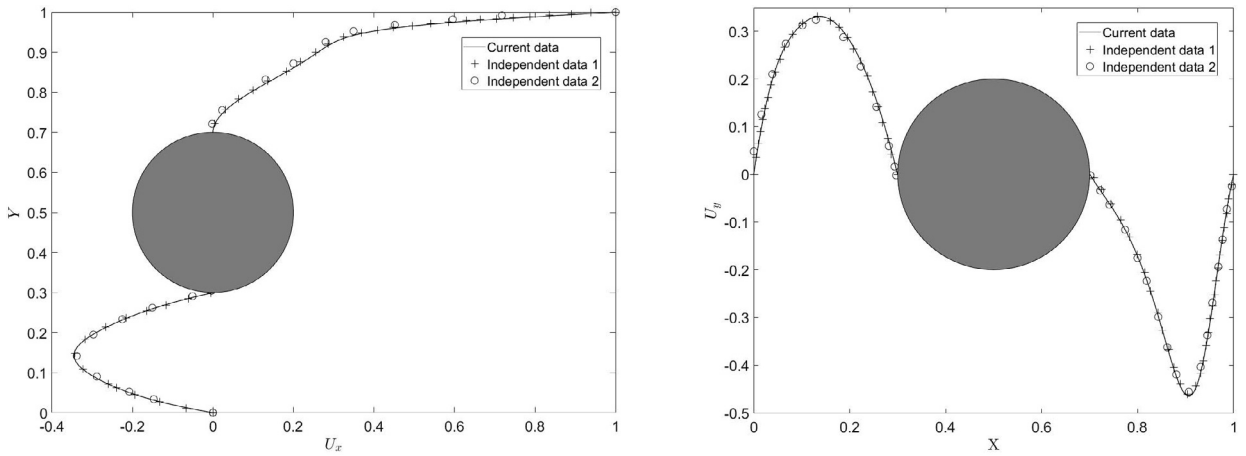



**Fig. 5.** Velocity components measured along the cavity centerlines, $Re = 1000$: (a) $u_x$ velocity component measured along the vertical centerline; (b) $u_y$ velocity component measured along the horizontal centerline. Solid line corresponds to the currently obtained data, ∘ symbols correspond to the data reported in [52], and + symbols correspond to the data reported in [54].

**Table 5**
Results of the grid independence study.

| | $\overline{Nu}$ | | |
|---|---|---|---|
| $Ri/Grid$ | $50^2$ | $100^2$ | $200^2$ |
| 0.01 | 2.9251 | 2.9340 | 2.9381 |
| 1 | 3.4707 | 3.4920 | 3.5026 |
| 5 | 4.7184 | 4.7099 | 4.7061 |
| 10 | 5.0906 | 5.0688 | 5.0641 |

**Table 6**
Comparison between the $\overline{Nu}$ values averaged over the lower surface of a lid driven cavity calculated for $Re = 100$.

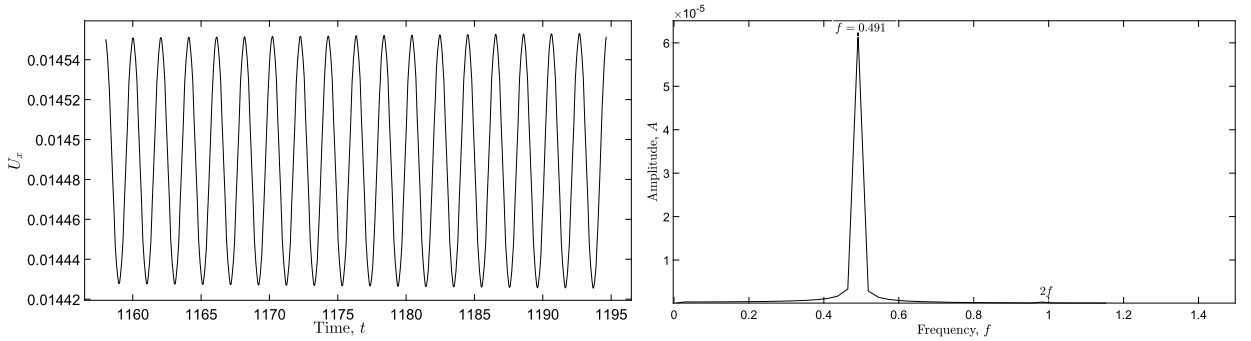
| | $\overline{Nu}$ | | | |
|---|---|---|---|---|
| $Ri$ | Isothermal | Isothermal, Ref. [56] | Adiabatic | Adiabatic, Ref. [56] |
| 0.01 | 2.9381 | 2.92 | 2.2650 | 2.24 |
| 1 | 3.5026 | 3.5 | 3.3669 | 3.33 |
| 5 | 4.7061 | 4.69 | 4.0988 | 4.06 |
| 10 | 5.0641 | 5.04 | 4.4866 | 4.50 |

and the adiabatic cylinders presented in Figs. 8 and 9, respectively, are in good qualitative agreement with the data available in the literature [56] for the entire range of the Richardson numbers. As expected, the configuration with adiabatic cylinder is characterized by a more uniform temperature distribution over the cavity compared to that with the isothermal cold cylinder. A quantitative comparison between the currently obtained and the independent [56] $\overline{Nu}$ values averaged over the lower surface of a lid-driven cavity is summarized in Table 6. The results demonstrate a high degree of agreement for both isothermal and adiabatic cylinders for the entire range of $Ri$ numbers, with a maximum deviation of 1.3%.

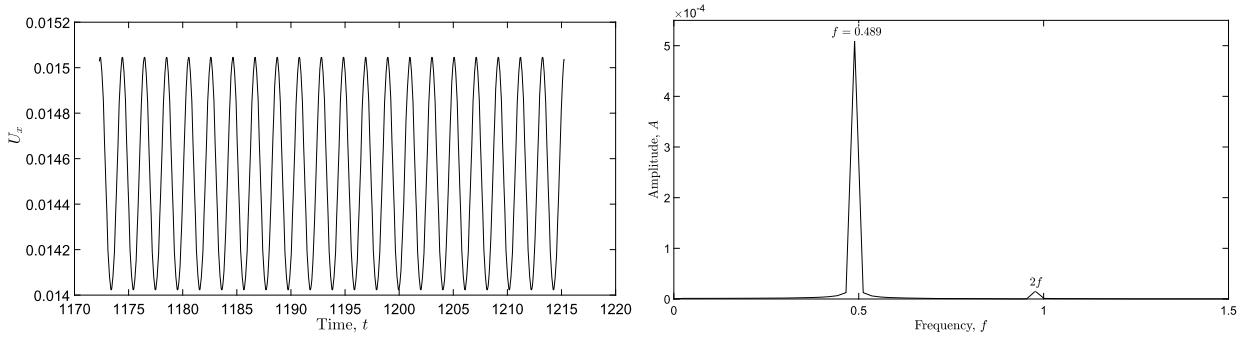
In what follows, we focus on the local flow characteristics obtained for a lid-driven cavity with an embedded adiabatic cylinder. A comparison between the currently obtained and the independent values of the local $Nu$ values calculated along the bottom wall of the cavity and the temperature values measured on the cylinder surface for three different $Ri$ values is presented in Figs. 10 (a) and (b), respectively. The results are in acceptable qualitative agreement with the corresponding independent data and successfully capture the trends of the heat flux enhancement through the right half of the cavity bottom over the entire range of $Ri$ values (see Fig. 10 (a)) as well as the temperature distribution polarization along the cylinder surface as the $Ri$ number decreases (see Fig. 10 (b)).

### *3.3. 3D lid-driven cavity with an embedded sphere*

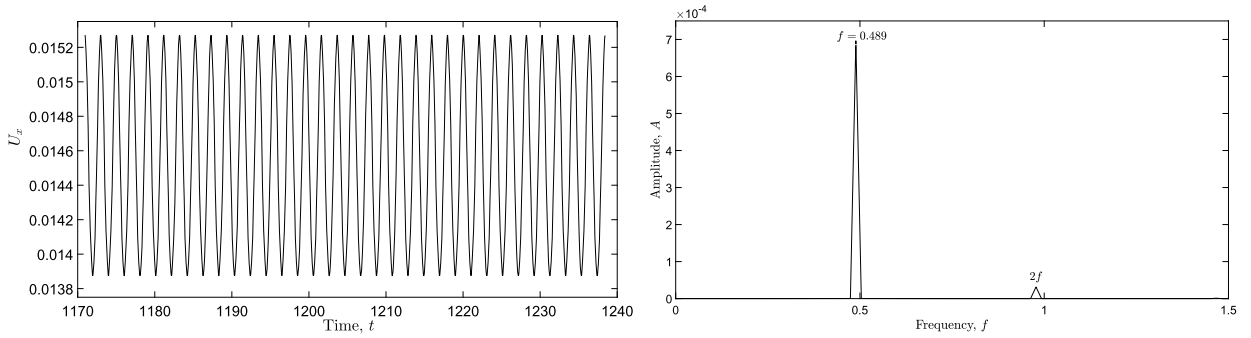
In this and the following sections we present the $3D$ results obtained when analyzing shear- and thermally driven confined flows around immersed bodies of spherical shape. To achieve the best accuracy of the IB method, all the spheres were built of equispaced Lagrangian points characterized by a distance equal to that of the grid size of the corresponding




(a) Time evolution and spectra, $Re = 8600$




(b) Time evolution and spectra, $Re = 8700$




(c) Time evolution and spectra, $Re = 8800$

**Fig. 6.** Time evolution and spectra of the $u_x$ velocity component measured at the control point (0.9, 0.1) for different $Re$ values.

Eulerian grid. The mapping of the spherical surfaces was implemented by utilizing the non-iterative method of Leopardi [59]. The flow within a cubic lid-driven cavity of side $L$ with a sphere of diameter $\alpha L$ placed in the geometric center of the cavity was investigated (see Fig. 11). The top lid of the cavity moves with a constant velocity $U$ in the $x$ direction, while all the other boundaries and the sphere are stationary. All the simulations were performed on a $200^3$ grid.

### 3.3.1. *Sphere of diameter* $0.25L$

To ensure the accuracy of our results, we performed a grid independence study for the flow within the given configuration. We conducted simulations on grids of sizes $100^3$, $150^3$, and $200^3$ at Reynolds numbers of 1, 100, and 400. The values of the $u_x$ and $u_y$ velocity components along the vertical and horizontal centerlines are presented in Table 7. Our analysis showed that the velocities obtained on the $150^3$ and $200^3$ grids were in agreement, to within 3 decimal digits, for most points, with maximum relative deviations of 2.5% occurring in regions of low velocity. These results demonstrate the grid independence of the obtained results, and we therefore conducted all further analysis using a $200^3$ grid.

The contours of the $u_x$ velocity component in the mid-cross section of the cavity for three different $Re$ numbers are presented in Fig. 12. It may be seen that the distribution of the $u_x$ velocity component maintains symmetry about the vertical centerline for all $Re$ values. As expected, the velocity gradients increase in the vicinity of the moving lid as the $Re$

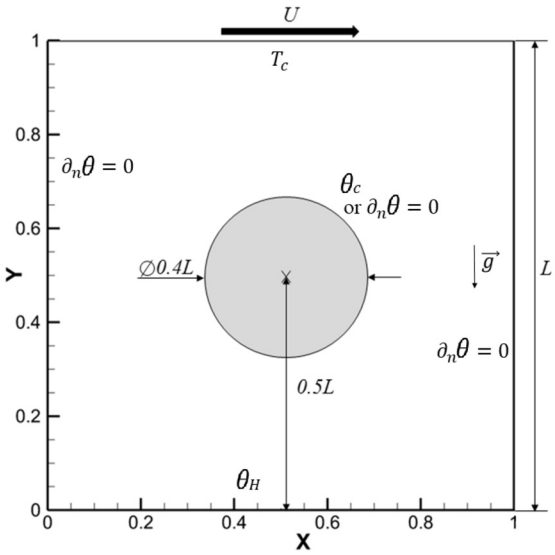
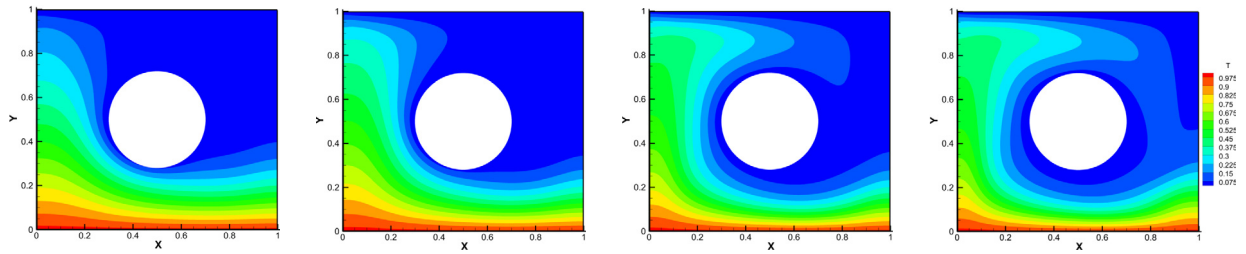
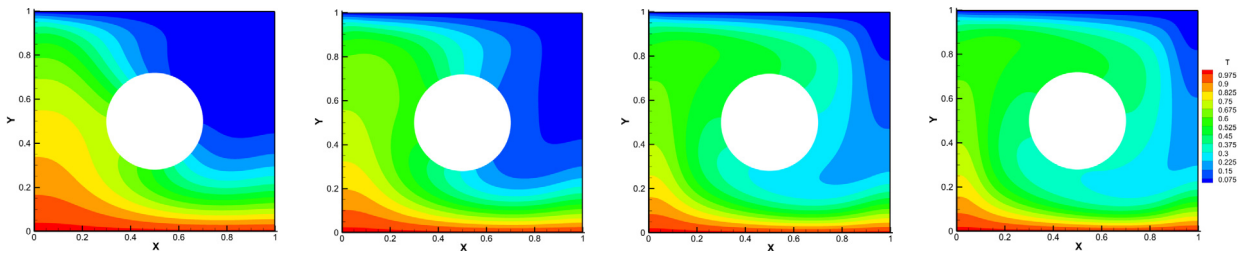



**Fig. 7.** Laminar mixed convection flow in a lid driven cavity with an embedded cylinder – physical model.



(a) $Ri = 0.01$ (b) $Ri = 1$ (c) $Ri = 5$ (d) $Ri = 10$

**Fig. 8.** Temperature distribution obtained for the isothermal cylinder.



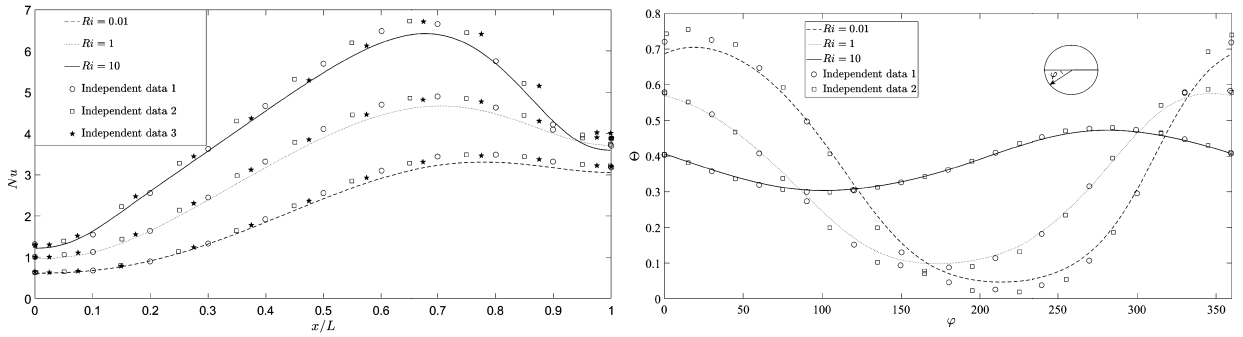
(a) $Ri = 0.01$ (b) $Ri = 1$ (c) $Ri = 5$ (d) $Ri = 10$

**Fig. 9.** Temperature distribution obtained for the adiabatic cylinder.

values increase. These contours are in good qualitative agreement with the data reported in reference [60]. Table 8 presents a quantitative comparison of the values of $u_x$ and $u_z$ obtained along the vertical and horizontal centerlines, respectively, for three different values of $Re = 1, 100$, and 400 with the corresponding data reported in [60], which was acquired by photogrammetry. It can be seen that for the vast majority of measurements the results are in good agreement, with a maximal relative deviation of 6%. For a small number of control points, characterized by a close-to-zero value of $u_z$, the relative deviation between the corresponding values is close to 10%.

### *3.3.2. Sphere of diameter* $0.4L$

The steady state and supercritical flows were then investigated for the $3D$ lid-driven cavity flow in a cube with a sphere of diameter $0.4L$ placed in the center of the cube. The steady-state flow was simulated for $Re = 100, 400, 1000$ and 1500 values. The obtained flow characteristics are presented in Fig. 13 in terms of the iso-contours of the $u_x$, $u_y$ and $u_z$ velocity components in the mid-cross-section of the cavity for the values of $Re = 100, 400$ and 1000. The calculated iso-




(a) Local Nusselt number at $y = 0$ (b) Temperature along the cylinder surface

**Fig. 10.** Comparison of the local $Nu$ and the temperature values along the surface of adiabatic cylinder placed in the center of a lid-driven cavity obtained for three different Richardson numbers $Ri = 0.01$, $Ri = 1$ and $Ri = 10$. Dashed, dotted and solid lines correspond to the currently obtained data, ★ symbols correspond to the data reported in [56] ∘ symbols correspond to the data reported in [57], and □ symbols correspond to the data reported in [58].

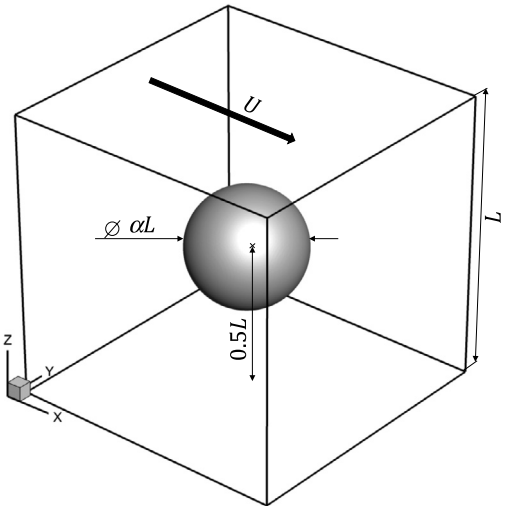
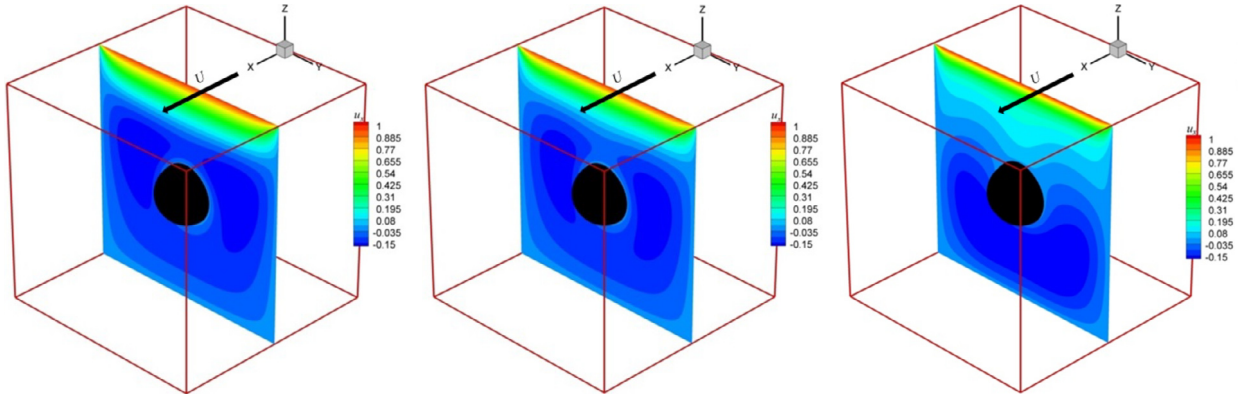



**Fig. 11.** Cubic lid-driven cavity with an embedded sphere in the center – physical model.




(a) $Re = 1$ (b) $Re = 100$ (c) $Re = 400$

**Fig. 12.** Contours of $u_x$ for the lid driven cavity flow with an embedded in the center sphere of diameter $D = 0.25L$.

**Table 7**
Summary of grid independence study. The data corresponds to the values of $u_x$ and $u_z$ velocity components collected along the vertical and horizontal centerlines, respectively.

| $Re = 1$ | | | | $Re = 100$ | | | | $Re = 400$ | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $z/_{Grid}$ | $u_x$ | | | $z/_{Grid}$ | $u_x$ | | | $z/_{Grid}$ | $u_x$ | | |
| | $100^3$ | $150^3$ | $200^3$ | | $100^3$ | $150^3$ | $200^3$ | | $100^3$ | $150^3$ | $200^3$ |
| 0.95195 | 0.68862 | 0.68973 | 0.68966 | 0.95423 | 0.65728 | 0.65821 | 0.65804 | 0.95423 | 0.47726 | 0.47887 | 0.49609 |
| 0.90160 | 0.40895 | 0.41050 | 0.41049 | 0.90389 | 0.35729 | 0.35845 | 0.35815 | 0.90389 | 0.23154 | 0.23234 | 0.23759 |
| 0.85355 | 0.19340 | 0.19505 | 0.19508 | 0.85355 | 0.15682 | 0.15802 | 0.15789 | 0.85355 | 0.15959 | 0.16010 | 0.16018 |
| 0.80320 | 0.02546 | 0.02672 | 0.02666 | 0.80320 | 0.03200 | 0.03318 | 0.03315 | 0.80320 | 0.12652 | 0.12702 | 0.12721 |
| 0.75057 | -0.08568 | -0.08551 | -0.08581 | 0.75286 | -0.04202 | -0.04134 | -0.04143 | 0.75286 | 0.09885 | 0.09964 | 0.10109 |
| 0.70023 | -0.12603 | -0.12825 | -0.12912 | 0.70252 | -0.07351 | -0.07446 | -0.07490 | 0.70252 | 0.06920 | 0.07071 | 0.07254 |
| 0.30206 | -0.09859 | -0.10144 | -0.10209 | 0.30206 | -0.10661 | -0.11032 | -0.11114 | 0.30206 | -0.16812 | -0.17320 | -0.17455 |
| 0.25172 | -0.11808 | -0.11924 | -0.11953 | 0.25172 | -0.12815 | -0.13005 | -0.13048 | 0.25172 | -0.21185 | -0.21502 | -0.21603 |
| 0.20366 | -0.11799 | -0.11825 | -0.11831 | 0.20137 | -0.12750 | -0.12829 | -0.12851 | 0.19908 | -0.21639 | -0.21862 | -0.21813 |
| 0.14874 | -0.10383 | -0.10357 | -0.10357 | 0.15103 | -0.11317 | -0.11332 | -0.11341 | 0.15103 | -0.18980 | -0.19112 | -0.19332 |
| 0.10069 | -0.08148 | -0.08109 | -0.08106 | 0.10069 | -0.08799 | -0.08783 | -0.08785 | 0.10069 | -0.14754 | -0.14822 | -0.14852 |
| 0.05263 | -0.04934 | -0.04902 | -0.04900 | 0.05263 | -0.05335 | -0.05315 | -0.05315 | 0.05034 | -0.08970 | -0.08992 | -0.08680 |
| $Re = 1$ | | | | $Re = 100$ | | | | $Re = 400$ | | | |
| $x/_{Grid}$ | $u_z$ | | | $x/_{Grid}$ | $u_z$ | | | $x/_{Grid}$ | $u_z$ | | |
| | $100^3$ | $150^3$ | $200^3$ | | $100^3$ | $150^3$ | $200^3$ | | $100^3$ | $150^3$ | $200^3$ |
| 0.05721 | 0.09030 | 0.09007 | 0.09016 | 0.05734 | 0.08944 | 0.08938 | 0.08955 | 0.05023 | 0.13483 | 0.13599 | 0.13675 |
| 0.10755 | 0.14447 | 0.14431 | 0.14440 | 0.10550 | 0.13530 | 0.13536 | 0.13545 | 0.10046 | 0.19602 | 0.19755 | 0.19826 |
| 0.15789 | 0.17390 | 0.17393 | 0.17406 | 0.15596 | 0.16021 | 0.16042 | 0.16056 | 0.14840 | 0.21368 | 0.21496 | 0.21555 |
| 0.20824 | 0.17969 | 0.18010 | 0.18027 | 0.20642 | 0.16593 | 0.16644 | 0.16663 | 0.20091 | 0.20982 | 0.21089 | 0.21137 |
| 0.25858 | 0.16333 | 0.16445 | 0.16476 | 0.25688 | 0.15397 | 0.15516 | 0.15546 | 0.24886 | 0.19194 | 0.19321 | 0.19371 |
| 0.30892 | 0.12253 | 0.12544 | 0.12612 | 0.30734 | 0.11927 | 0.12226 | 0.12294 | 0.29909 | 0.15367 | 0.15645 | 0.15724 |
| 0.70709 | -0.13910 | -0.14129 | -0.14170 | 0.70872 | -0.15334 | -0.15570 | -0.15625 | 0.70091 | -0.06086 | -0.06351 | -0.06393 |
| 0.75744 | -0.17145 | -0.17234 | -0.17255 | 0.75917 | -0.20512 | -0.20677 | -0.20726 | 0.75114 | -0.17000 | -0.17226 | -0.17276 |
| 0.80778 | -0.18095 | -0.18129 | -0.18141 | 0.80963 | -0.22841 | -0.22958 | -0.23000 | 0.80137 | -0.28543 | -0.28791 | -0.28882 |
| 0.85812 | -0.16777 | -0.16781 | -0.16790 | 0.86009 | -0.21376 | -0.21446 | -0.21477 | 0.84932 | -0.35992 | -0.36274 | -0.36419 |
| 0.90389 | -0.13477 | -0.13459 | -0.13466 | 0.90826 | -0.16230 | -0.16266 | -0.16284 | 0.90183 | -0.32061 | -0.32236 | -0.32367 |
| 0.95652 | -0.07144 | -0.07132 | -0.07127 | 0.96101 | -0.07428 | -0.07436 | -0.07442 | 0.94977 | -0.16749 | -0.16868 | -0.16906 |

contours bear a striking resemblance (both qualitatively and quantitatively) to the corresponding independent data available in literature [51]. The same trends were observed with respect to the symmetry preserving of all the velocity components relative to the vertical centerline and to the presence of numerous secondary eddies in the bottom region of the cavity. The acceptable quantitative agreement between the maximal values of the velocity components obtained in this study and those obtained in Ref. [51] as summarized in Table 9, verifies the correctness of our steady-state results.

The characteristics of the supercritical flow developing for the values of $Re = 1790, 1810, 1820$ and 1850 were then compared with the corresponding results reported in [51]. The time evolutions of the $v_z$ velocity component collected at the control point $(0.85, 0.5, 0.5)$ for all the simulated periodic flows, along with the corresponding frequency spectra, are shown in Fig. 14. The main harmonic and its multipliers typical of non-linear supercritical flow regimes can be clearly recognized. The values of the obtained main harmonics for all the $Re$ values are close to each other and are in good agreement with the corresponding values reported in [51]. The time evolutions of signals obtained in this study are, however, characterized by higher amplitudes and a more pronounced impact of the multiplied harmonics—characteristics that were apparently under-resolved in [51] as a result of insufficient grid resolution (the calculations were performed on a $80^3$ grid).

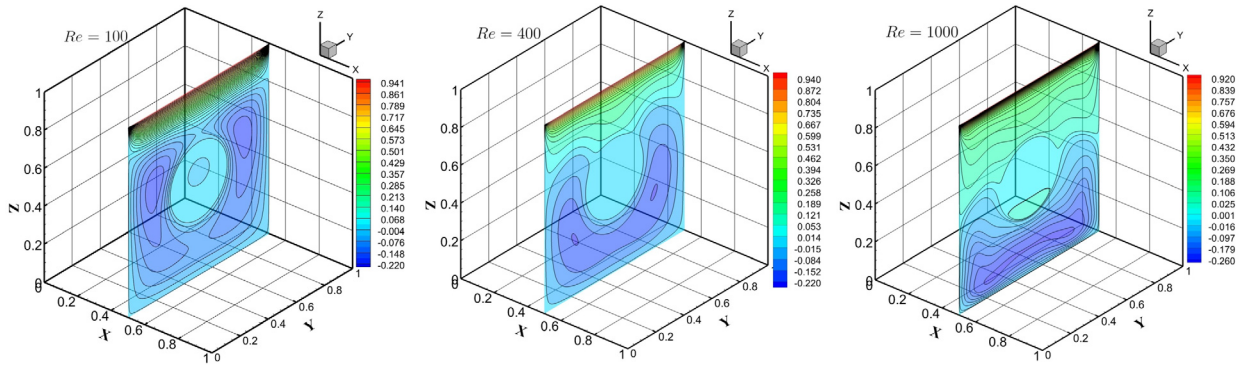
The phase plots obtained by plotting the time evolutions of $u_z$ versus $u_x$ collected at the control point (0.85, 0.5, 0.5) for different $Re$ values are shown in Fig. 15. There is good agreement between our data and the corresponding data reported in [51] in terms of extreme values of the measured $u_z$ and $u_x$ velocities for the entire range of $Re$ values. Nonetheless, the shape of our phase plot curves is more complicated than that in [51] which, again, is a consequence of the more pronounced impact of the multiplied harmonics characterizing our simulated flow fields.

### *3.4. Diagonally 3D lid-driven cavity with an embedded sphere*

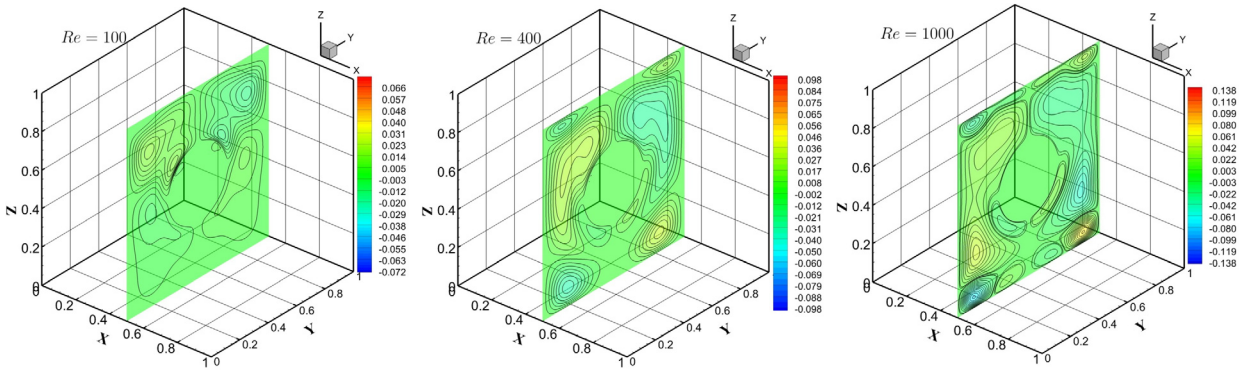
We examine the configuration of a cubic diagonally lid-driven cavity of side $L$ in which the lid moves along the main diagonal with constant velocity $U(cos(\frac{\pi}{4}), sin(\frac{\pi}{4}), 0)$ and in which a sphere of diameter $L/4$ is placed in the cavity center (see Fig. 16). The diagonally lid-driven cavity has attracted interest due to the complex separation flows developing within it, in both the subcritical and supercritical regimes (see e.g. [61], [62]). We currently extend the state of the art on the above complex flow by investigating the characteristics of the subcritical flow regime developing within the diagonally lid-driven cavity hosting a stationary no-slip sphere at its center. All the simulations were performed on a $200^3$ grid.

**Table 8**
Comparison between the values of $u_x$ and $u_z$ velocities currently calculated along the vertical and horizontal centerlines and the corresponding data reported in [60]. The values are compared in terms of relative deviation between the results.

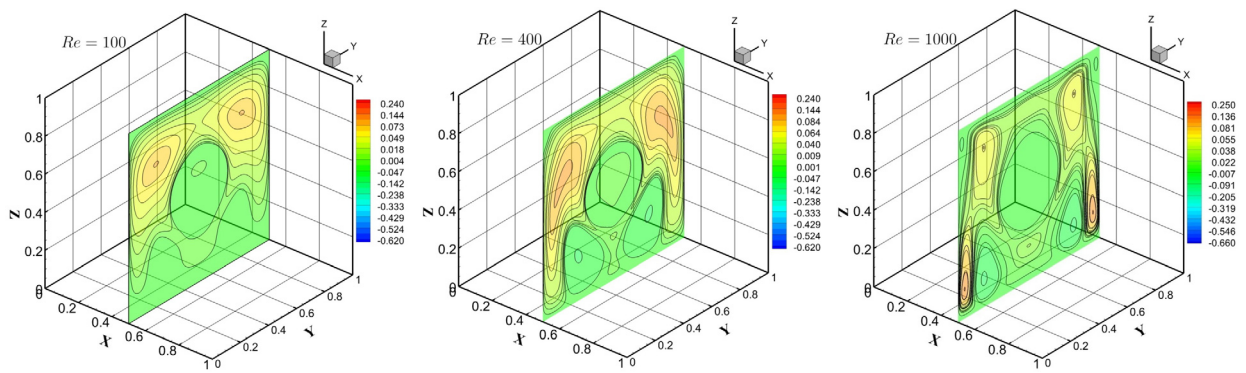
| $Re = 1$ | | | | | | $Re = 100$ | | | | | | $Re = 400$ | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $Rel.Dev.[\%]$ | | | $Rel.Dev.[\%]$ | | | $Rel.Dev.[\%]$ | | | $Rel.Dev.[\%]$ | | | $Rel.Dev.[\%]$ | | | $Rel.Dev.[\%]$ | | |
| $z$ | $u_x$ [60] | $u_x$ | $x$ | $u_z$ [60] | $u_z$ | $z$ | $u_x$ [60] | $u_x$ | $x$ | $u_z$ [60] | $u_z$ | $z$ | $u_x$ [60] | $u_x$ | $x$ | $u_z$ [60] | $u_z$ |
| 0.95195 | 2.117 | 0.68966 | 0.05721 | 1.016 | 0.09016 | 0.95423 | 2.407 | 0.65804 | 0.05734 | 0.345 | 0.08955 | 0.95423 | 2.672 | 0.49609 | 0.05023 | 2.075 | 0.13675 |
| 0.90160 | 3.559 | 0.41049 | 0.10755 | 3.004 | 0.14440 | 0.90389 | 2.659 | 0.35815 | 0.10550 | 0.325 | 0.13545 | 0.90389 | 0.797 | 0.23759 | 0.10046 | 0.414 | 0.19826 |
| 0.85355 | 4.798 | 0.18508 | 0.15789 | 1.232 | 0.17406 | 0.85355 | 4.126 | 0.15789 | 0.15596 | 1.193 | 0.16056 | 0.85355 | 1.434 | 0.16018 | 0.14840 | 2.976 | 0.21555 |
| 0.80320 | 3.826 | 0.01666 | 0.20824 | 2.821 | 0.18027 | 0.80320 | 8.772 | 0.03015 | 0.20642 | 0.250 | 0.16663 | 0.80320 | 2.533 | 0.12721 | 0.20091 | 2.848 | 0.21137 |
| 0.75057 | 3.998 | -0.08581 | 0.25858 | 1.386 | 0.16476 | 0.75286 | 10.702 | -0.04143 | 0.25688 | 1.380 | 0.15546 | 0.75286 | 1.861 | 0.10109 | 0.24886 | 0.414 | 0.19371 |
| 0.70023 | 2.528 | -0.12912 | 0.30892 | 0.208 | 0.12612 | 0.70252 | 2.010 | -0.07490 | 0.30734 | 1.346 | 0.12294 | 0.70252 | 2.210 | 0.07254 | 0.29909 | 0.417 | 0.15724 |
| 0.30206 | 3.620 | -0.10209 | 0.70709 | 1.491 | -0.14170 | 0.30206 | 5.068 | -0.11114 | 0.70872 | 1.874 | -0.15625 | 0.30206 | 4.300 | -0.17455 | 0.70091 | 9.094 | -0.06293 |
| 0.25172 | 2.359 | -0.11953 | 0.75744 | 0.537 | -0.17255 | 0.25172 | 5.078 | -0.13048 | 0.75917 | 3.943 | -0.20726 | 0.25172 | 1.486 | -0.21603 | 0.75114 | 3.306 | -0.17276 |
| 0.20366 | 2.514 | -0.11831 | 0.80778 | 0.346 | -0.18141 | 0.20137 | 3.621 | -0.12851 | 0.80963 | 1.503 | -0.23000 | 0.19908 | 0.338 | -0.21813 | 0.80137 | 0.624 | -0.28882 |
| 0.14874 | 3.842 | -0.10357 | 0.85812 | 0.509 | -0.16790 | 0.15103 | 2.922 | -0.11341 | 0.86009 | 3.042 | -0.21477 | 0.15103 | 0.616 | -0.19332 | 0.84932 | 0.095 | -0.36419 |
| 0.10069 | 4.454 | -0.08106 | 0.90389 | 6.537 | -0.13466 | 0.10069 | 0.794 | -0.08785 | 0.90826 | 5.849 | -0.16284 | 0.10069 | 3.228 | -0.14852 | 0.90183 | 1.728 | -0.32367 |
| 0.05263 | 7.412 | -0.04900 | 0.95652 | 10.807 | -0.06927 | 0.05263 | 3.564 | -0.05315 | 0.96101 | 4.680 | -0.07442 | 0.05034 | 2.453 | -0.08680 | 0.94977 | 3.895 | -0.16906 |




(a) Iso-contours of $u_x$ velocity




(b) Iso-contours of $u_y$ velocity

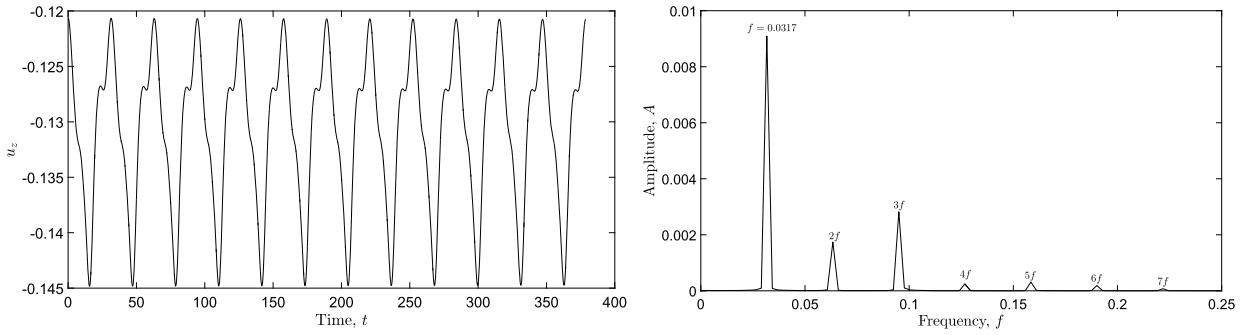



(c) Iso-contours of $u_z$ velocity

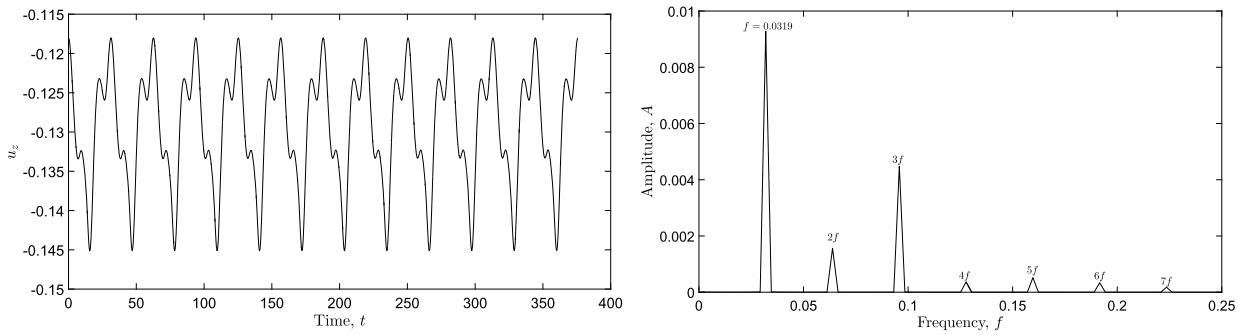
**Fig. 13.** Iso-contours of the velocity components in a mid cross-section of the cavity obtained for three different *Re* values.

**Table 9**
Comparison between extreme values of the velocity components obtained for different *Re* values.

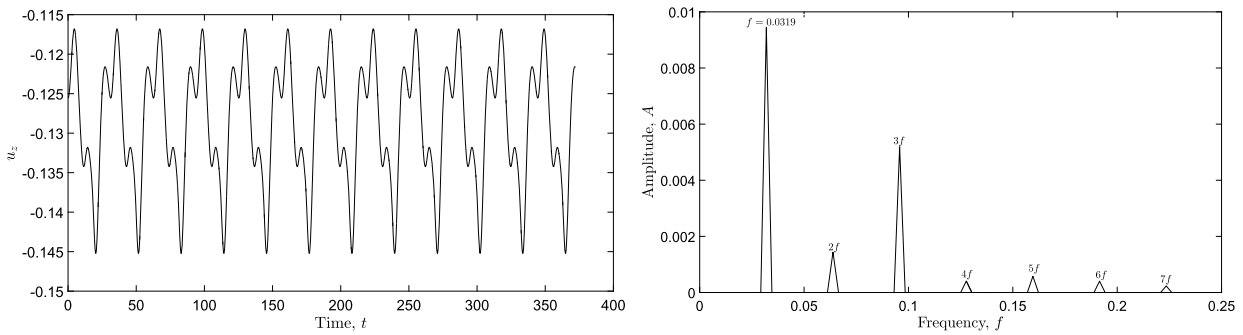
| *Re* | $u_{xmin}$ | | $u_{ymin}$ | | $u_{ymax}$ | | $u_{zmin}$ | | $u_{zmax}$ | |
|---|---|---|---|---|---|---|---|---|---|---|
| | Ref. [51] | Current | Ref. [51] | Current | Ref. [51] | Current | Ref. [51] | Current | Ref. [51] | Current |
| 100 | -0.22311 | -0.22421 | -0.07437 | -0.07598 | 0.07415 | 0.07603 | -0.50933 | -0.51036 | 0.23871 | 0.34058 |
| 400 | -0.22724 | -0.22795 | -0.09827 | -0.09699 | 0.09827 | 0.09699 | -0.62632 | -0.63243 | 0.24277 | 0.28287 |
| 1000 | -0.26324 | -0.26667 | -0.13936 | -0.13423 | 0.13936 | 0.13423 | -0.66759 | -0.67588 | 0.24654 | 0.24962 |
| 1500 | -0.28349 | -0.28920 | -0.18533 | 0.14843 | 0.18533 | 0.14843 | -0.66857 | -0.69072 | 0.24565 | 0.25013 |



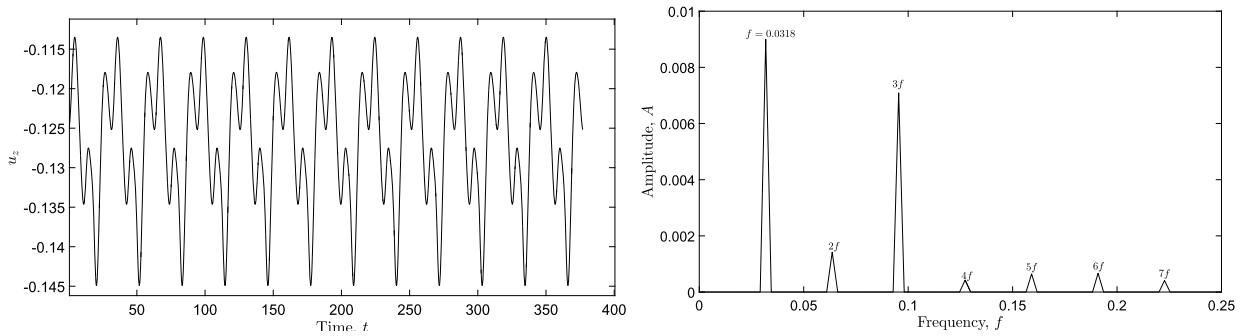

(a) Time evolution and spectra, $Re = 1790$




(b) Time evolution and spectra, $Re = 1810$




(c) Time evolution and spectra, $Re = 1820$




(d) Time evolution and spectra, $Re = 1850$

**Fig. 14.** Time evolution and spectra of the $u_z$ velocity component, as measured at control point (0.85, 0.5, 0.5) for different $Re$ values.

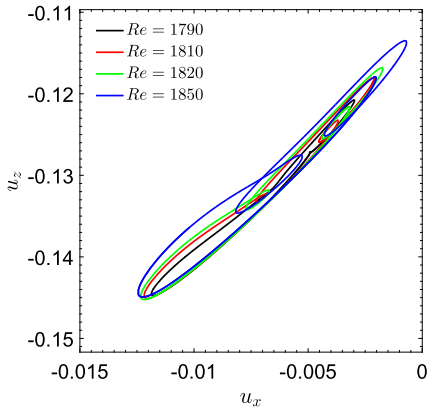



**Fig. 15.** Phase plots of the $u_z$ velocity component versus the $u_x$ velocity component, as measured at the control point (0.85, 0.5, 0.5) for different $Re$ values.

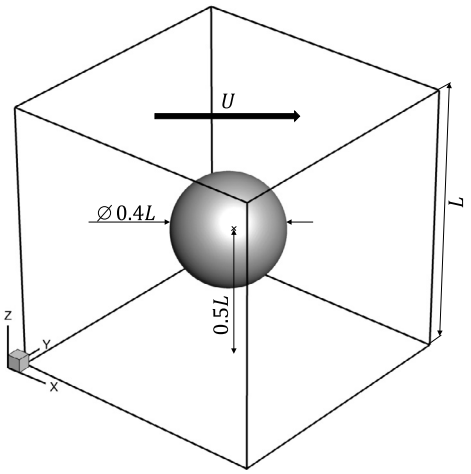
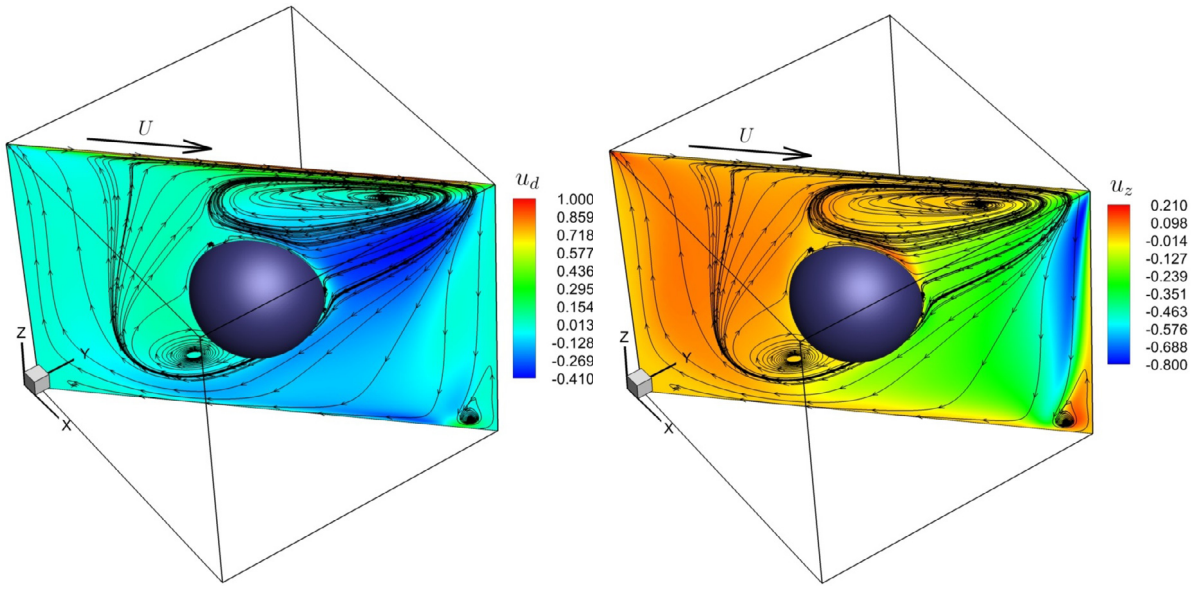



**Fig. 16.** Cubic diagonally lid-driven cavity with an embedded sphere in the center – physical model.

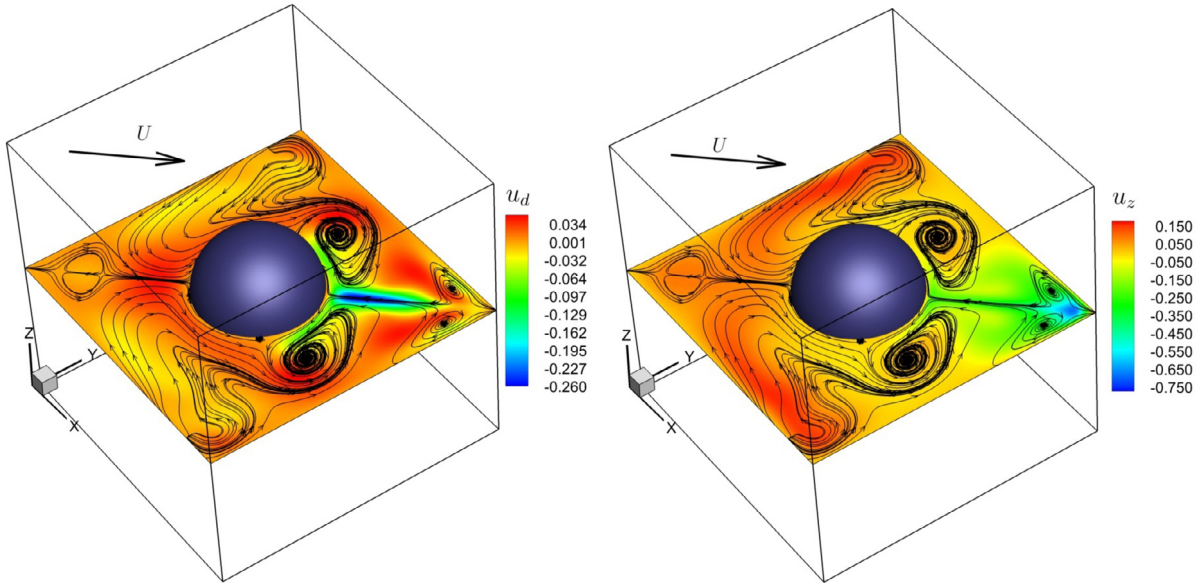
The iso-contours of the $u_d = u_x cos(\frac{\pi}{4}) + u_y sin(\frac{\pi}{4})$ and $u_z$ velocity components with superimposed flow pathlines projected on the main diagonal and horizontal cross-sections are shown in Fig. 17. The steady-state flow was calculated for $Re = 2800$. The steady-state flow preserves the symmetry relatively to the main diagonal plane. Remarkably, in contrast to the "classical" lid-driven cavity configuration, embedding the sphere in the center of the cavity suppresses the transition to unsteadiness. We note that without the embedded spheres the transition to unsteadiness in "classical" and diagonally lid-driven cavities sets in at $Re \approx 1920$ and $Re \approx 2300$, respectively [63,62,64]; in contrast to the spheres placed in the center of the cavities, supercritical flow is already observed at $Re = 1810$ for the "classical" lid-driven cavity, whereas the flow remains steady even at $Re = 2800$ for the diagonally lid-driven cavity. The observed phenomenon can be explained by the fact that in the diagonally lid-driven cavity the embedded sphere splits the primary vortex into two counter-rotating vortices. As a result, the interaction between the primary and the secondary vortices, constituting the primary source of the instability onset [64], weakens and, consequently, the transition to instability sets in at higher $Re$ values. We conclude this section by summarizing in Table 10 the values of all the velocity components along the vertical centerline of the cavity, which may be useful for comparison with independently obtained results in the future.

### 3.5. Hot sphere within a cold cubic container

The natural convection flow developing around a hot sphere of diameter $D = 0.4L$ placed inside a cold cubic container of side $L$ (see Fig. 18) is examined. The center of the sphere is placed on the vertical centerline at height $H$ from the cavity bottom. The gravity force acts in the $-\hat{z}$ direction. We now define a normalized distance between the sphere and the cubic container centers as $\kappa = (H - 0.5L)/L$. To quantify the heat transfer rate from the surface of the hot sphere and from the surfaces of the cold cubic cavity, we also define two values of the Nusselt number, $\overline{Nu} = \overline{PrRa}\Delta x\overline{Q}$ and $\overline{Nu}_G = \frac{1}{N}\sum_{i=1}^{N}\overline{\frac{\partial\theta}{\partial \mathbf{n}}}$, averaged over the surfaces of the sphere and the cubic container, respectively. Here $\overline{Q}$ is simply the arithmetic mean of all the heat fluxes $Q_k$ at each Lagrangian point $k$ of the sphere obtained directly by the solution of Eqs. (20), and $N$ corresponds to the total number of thermally non-insulated faces of the cubical container ($N = 6$ for




(a) Projection of the flow pathlines on the diagonal cross-section superimposed on contours of the $u_d$ and $u_z$ velocity components.




(b) Projection of the flow pathlines on the horizontal cross-section superimposed on contours of the $u_d$ and $u_z$ velocity components.

**Fig. 17.** Characteristics of the steady-state flow typical of a diagonally lid-driven cavity with an embedded sphere obtained for $Re = 2800$.

the configuration under consideration). The values $\overline{Nu}$ and $\overline{Nu}_G$ calculated for the steady-state natural convection flow developing at $Ra = 10^5$ and $Ra = 10^6$ values and their comparison with the corresponding results available in the literature are summarized in Table 11.

As can be seen from the Table 11, there is acceptable agreement (a maximal value of the relative deviation does not exceed 9%) between the currently obtained values and the independent data for the entire range of $Ra$ and $\kappa$ values. For additional verification of the conservation of the global heat flux of the system, we compared the values of $\overline{Nu}$ and $N/\pi D^2 \overline{Nu}_G$, corresponding to the average area specific heat fluxes from the surfaces of the hot sphere and the cold cubic container, respectively. The relative deviation between the values was less than 0.5% for the entire range of $Ra$ and $\kappa$ values,

**Table 10**
Numerical values of all the velocity components obtained along the vertical centerline for different $Re$ values.

| $z$ | $Re = 2400$ | | $Re = 2600$ | | $Re = 2800$ | |
|---|---|---|---|---|---|---|
| | $u_x, u_y$ | $u_z$ | $u_x, u_y$ | $u_z$ | $u_x, u_y$ | $u_z$ |
| 0.002489 | -1.3611E-02 | -5.5481E-04 | -1.3732E-02 | -5.9578E-04 | -1.3919E-02 | -6.3865E-04 |
| 0.061991 | -1.0740E-01 | -6.7994E-02 | -1.0350E-01 | -6.9455E-02 | -1.0054E-01 | -7.0909E-02 |
| 0.121493 | -1.1728E-01 | -1.3213E-01 | -1.1542E-01 | -1.3371E-01 | -1.1363E-01 | -1.3460E-01 |
| 0.180996 | -1.0329E-01 | -1.1703E-01 | -1.0665E-01 | -1.2232E-01 | -1.1002E-01 | -1.2723E-01 |
| 0.240499 | -3.8603E-02 | -3.5437E-02 | -3.9955E-02 | -3.7081E-02 | -4.1536E-02 | -3.8880E-02 |
| 0.759502 | -8.9426E-02 | 3.9756E-02 | -9.1101E-02 | 3.2964E-02 | -9.1534E-02 | 2.5465E-02 |
| 0.819004 | -3.3817E-02 | 2.7711E-02 | -3.6154E-02 | 2.0284E-02 | -3.8896E-02 | 1.4138E-02 |
| 0.878507 | 2.5904E-02 | 3.5641E-02 | 2.0932E-02 | 2.9248E-02 | 1.5168E-02 | 2.3410E-02 |
| 0.938009 | 8.6466E-02 | 2.9329E-02 | 8.4158E-02 | 2.8013E-02 | 8.1903E-02 | 2.6226E-02 |
| 0.997511 | 6.5758E-01 | 2.2253E-04 | 6.5574E-01 | 2.2055E-04 | 6.5399E-01 | 2.1611E-04 |

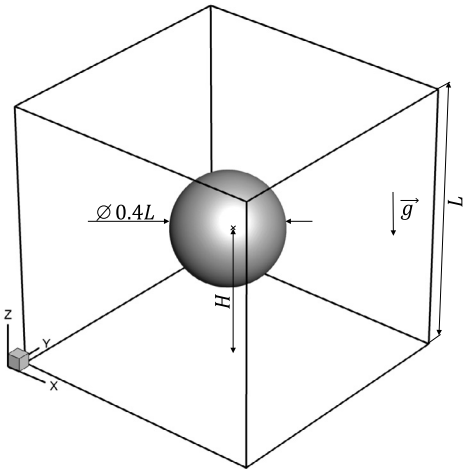


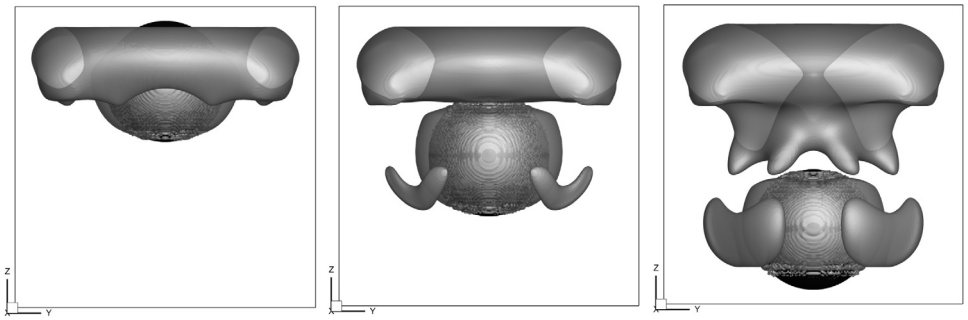

**Fig. 18.** Hot sphere placed inside a cold cubic container along its vertical centerline – physical model.

**Table 11**
Comparison between the values obtained in this study and those in the literature for $\overline{Nu}$ and $\overline{Nu}_G$ averaged over the surfaces of a hot sphere and a cold cube, respectively. The values are compared in terms of relative deviation between the results.

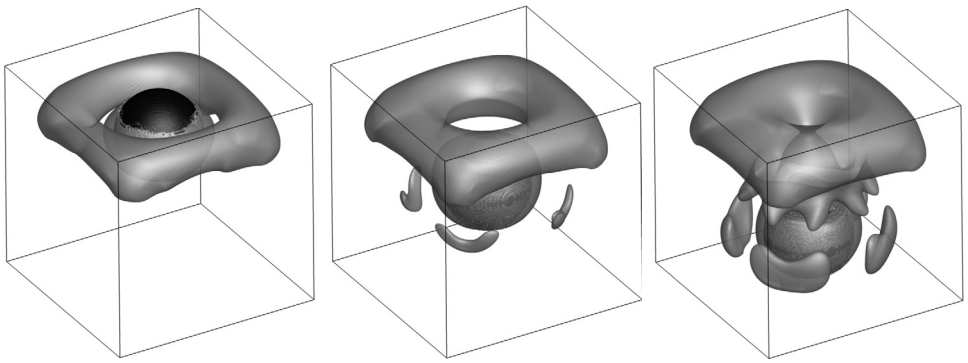
| | $\overline{Nu}$ | | | | | | | | $\overline{Nu}_G$ | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $Ra = 10^5$ | | | | $Ra = 10^6$ | | | | $Ra = 10^5$ | | | $Ra = 10^6$ | | |
| | $Rel.Dev.[\%]$ | | | | $Rel.Dev.[\%]$ | | | | $Rel.Dev.[\%]$ | | | $Rel.Dev.[\%]$ | | |
| $\kappa$ | Ref. [65] | Ref. [41] | Ref. [31] | Present | Ref. [65] | Ref. [41] | Ref. [31] | Present | Ref. [65] | Ref. [31] | Present | Ref. [65] | Ref. [31] | Present |
| -0.25 | 4.674 | 3.913 | 5.902 | 14.335 | 4.288 | 0.765 | 5.567 | 21.826 | 0.217 | 7.152 | 1.1969 | 0.602 | 5.563 | 1.8262 |
| -0.2 | 4.307 | 3.367 | 5.513 | 13.513 | 4.256 | 1.457 | 4.785 | 21.548 | 0.632 | 6.886 | 1.1284 | 0.582 | 7.488 | 1.8030 |
| -0.1 | 4.091 | 1.258 | 3.413 | 13.272 | 4.360 | 2.049 | 2.316 | 21.719 | 0.235 | 4.890 | 1.1084 | 0.341 | 5.075 | 1.8168 |
| 0 | 4.062 | 1.675 | 0.258 | 13.194 | 4.282 | 7.939 | 0.176 | 21.627 | 0.272 | 1.752 | 1.1014 | 0.520 | 3.075 | 1.8080 |
| 0.1 | 3.838 | 4.687 | 3.005 | 12.844 | 4.101 | 6.060 | 2.053 | 21.238 | 0.690 | 1.417 | 1.0728 | 0.327 | 1.329 | 1.7758 |
| 0.2 | 4.236 | 6.557 | 5.205 | 12.796 | 4.181 | 8.216 | 4.702 | 20.522 | 0.655 | 3.677 | 1.0689 | 0.379 | 1.200 | 1.7162 |
| 0.25 | 4.289 | 6.670 | 5.568 | 13.524 | 4.341 | 8.620 | 5.535 | 20.616 | 0.903 | 4.214 | 1.1295 | 0.099 | 2.024 | 1.7239 |

which proves the accurate preservation of the total thermal balance between the emanating surface of the hot sphere and the absorbing surfaces of the cold cubic container.

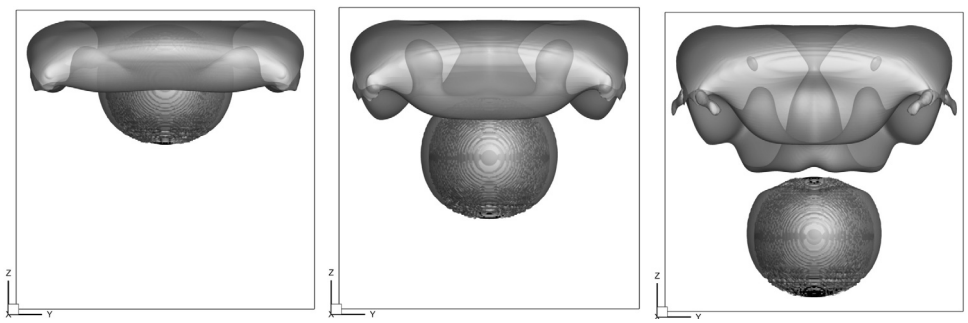
We now present the qualitative characteristics of the simulated steady-state flow in terms of the iso-surfaces of the $\lambda_2$ criterion [66] corresponding to the outermost surface of the vortical structures evolving in the flow. The value of $\lambda_2 = -0.1$ was chosen for visualization purposes in accordance with the works of [67,31]. The iso-surfaces presented in Fig. 19 are in good agreement with the corresponding data presented in [31]. Noticeably, in agreement with the observations made in [31], the flow vortical structures are symmetric relative to the $X - Z$ and $Y - Z$ centerplanes and to both main diagonal planes of the cubic container. A large circumferential convection cell located close to the top boundary of the cube is clearly recognizable for the entire range of $Ra$ and $\kappa$ values. For negative $\kappa$ values, the convection cell acquires a mushroom-like shape, while for positive values of $\kappa$ the convective cell acquires a torus shape, as was reported in [31].



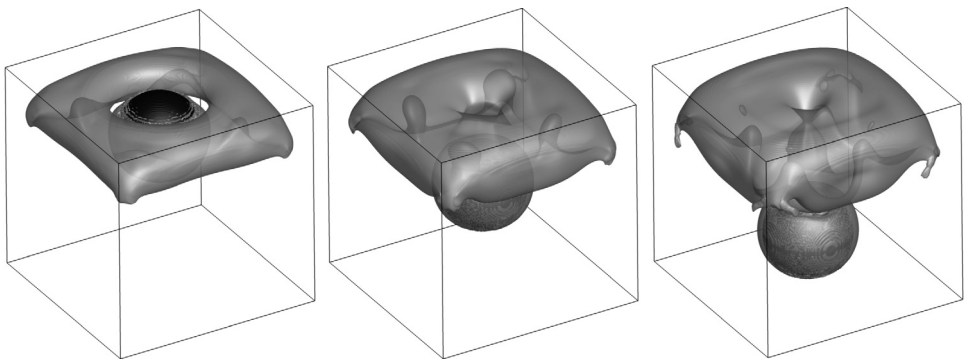
(a) Front view of $\lambda_2 = -0.1$ iso-surfaces obtained for $Ra = 10^5$, and $\kappa = 0.25, 0, -0.25$.



(b) Isometric view of $\lambda_2 = -0.1$ iso-surfaces obtained for $Ra = 10^5$, and $\kappa = 0.25, 0, -0.25$.



(c) Front view of $\lambda_2 = -0.1$ iso-surfaces obtained for $Ra = 10^6$, and $\kappa = 0.25, 0, -0.25$.



(d) Isometric view of $\lambda_2 = -0.1$ iso-surfaces obtained for $Ra = 10^6$, and $\kappa = 0.25, 0, -0.25$.

**Fig. 19.** Iso-surfaces of the $\lambda_2$ criterion.

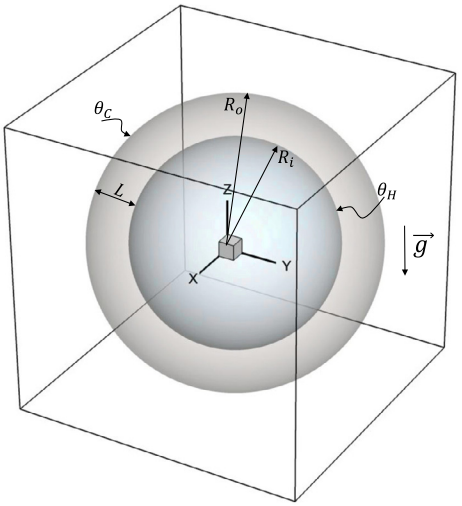



**Fig. 20.** Hot internal sphere of radius $R_i$ placed within a cold outer sphere of radius $R_o$ – physical model. The two spheres are placed within a cold cubic container in accordance with the formalism of the IB method.

### 3.6. Natural convection between two concentric spheres

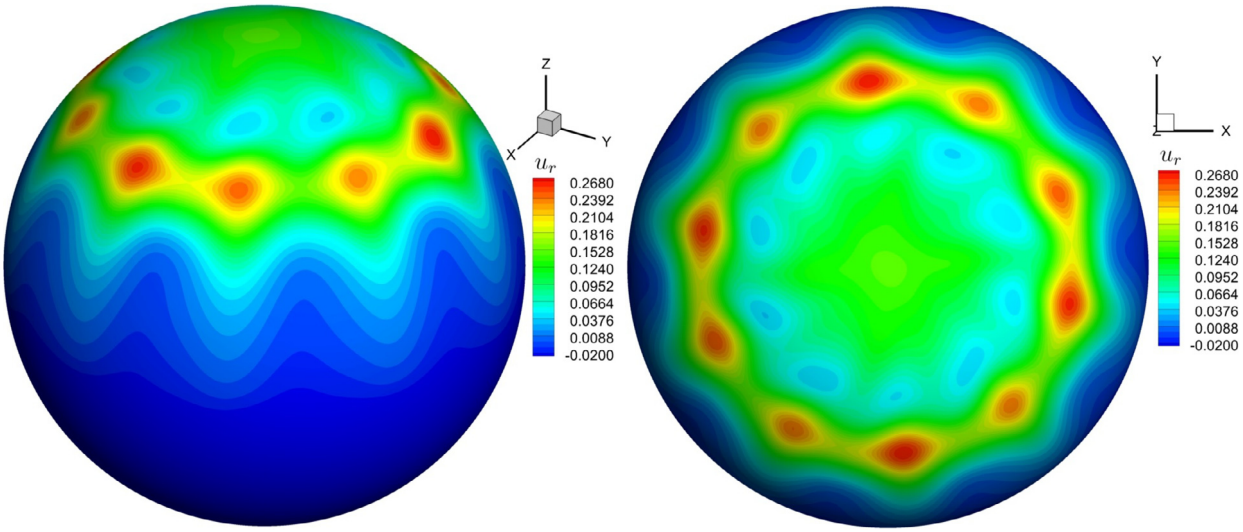
The natural convection within a spherical shell confined by hot and cold spheres of radii $R_i$ and $R_o$, respectively, is examined. The surfaces of the hot and cold spheres are held at constant temperatures $\theta_c$ and $\theta_h$, respectively. According to the formalism of the IB method, the simulations were conducted on a uniform structured grid, and therefore both spheres were placed at the center of the cubic cavity, although the domain of interest is restricted to the region confined by the two spheres (see Fig. 20). The container walls were held at a constant cold temperature $\theta_c$. The distance between the surfaces of the outer and inner spheres, $L$, was used to normalize the length scales. An additional non-dimensional parameter determining the geometry of the configuration under consideration was $\phi = R_i/R_o$. The dimensions of the cubic container were chosen in such a way that at least 10 grid points separated the container walls from the surface of the external sphere, so as to facilitate an adequate adjustment of the flow between them.

We then set the values of $Ra = 4.6 \times 10^4$ to enable us to perform an analysis of the slightly supercritical flow developing within the spherical gap characterized by the value of $\phi = 0.774$. The bifurcated flow regime is characterized by complex multi-cellular dynamics in the form of either pulsating or traveling waves, as detailed in [68–70,31]. Figs. 21 (a) and (b) present a snapshot of the contours of the radial velocity $u_r$ interpolated onto the mid-sphere surface between the hot and the cold spheres. The obtained results are in excellent agreement with those reported in [70], which used linear stability analysis to predict that the traveling wave instability sets in at $Ra_{cr} = 3.9562 \times 10^4$ and is characterized by a wave number of $m = 10$ of the leading eigenmode. In that study, the critical value of the angular frequency for this regime was found to be $\omega_{cr} = 0.389$.[6] This value is in excellent agreement with our results, which for the slightly supercritical flow simulated for $Ra = 4.6 \times 10^4$ predicted a traveling wave solution characterized by a wave number of $m = 10$ and a value of $w_{cr} = 0.387$ for the angular frequency. Note that the value of the angular frequency of the flow can be obtained simply by dividing by 2 the value of the angular frequency of the time evolution of the Nusselt number, $\overline{Nu}_o$ averaged over the surface of the outer sphere, as presented in Figs. 21 (c) and (d) (see e.g. [31,68]).

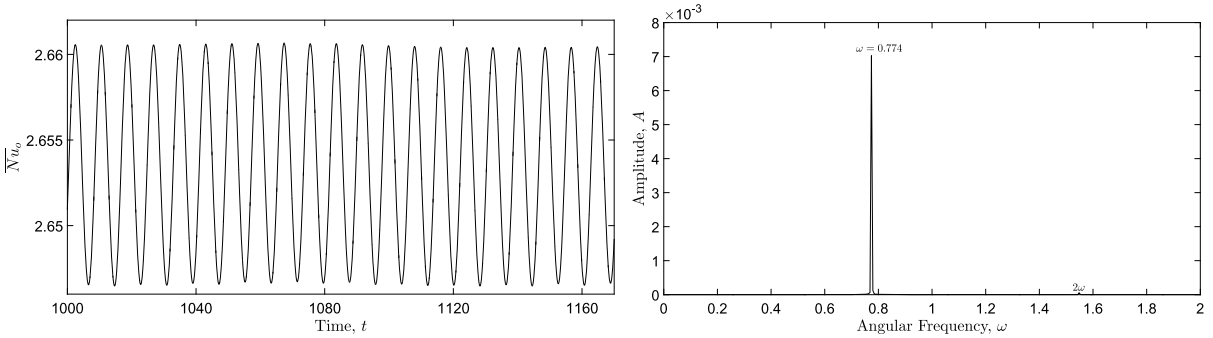
### 3.7. Memory consumption and time step and scaling

To further evaluate the efficiency characteristics of the solver developed in this study, we examined the way in which the memory consumption and the duration of a single time step scale with the total number of Lagrangian points, $N_s$ and with the total number of grid points, $N_g$. All the characteristics were obtained on a standard Linux server having 128 GB DDR3 shared memory and 2 Intel Xeon 12C processors, 24 threads each (48 threads in total). Shared memory openMP parallelization was used. The data was acquired for the grid resolutions starting from $50^3$ and ending with $300^3$ grid points when simulating natural convection flow developing between two concentric spheres for the values of $Ra = 4.6 \times 10^4$ and $\phi = 0.714$ (see Figs. 22-(a) and (b)), and the $3D$ lid-driven cavity flow with a sphere of diameter $0.4L$ placed in the cavity center for $Re = 400$ (see Figs. 22-(c) and (d)). For the first configuration, the number of Lagrangian points varied from 10,039 points for the coarsest grid resolution to 371,121 points for the finest grid resolution of the Eulerian grid. For the second configuration, the number of Lagrangian points varied from 491 to 17,675 points for the coarsest and the finest grid

[6] The value of $\omega_{cr} = 92.5$ reported in [70] was divided by $(Pr/Ra)^{0.5}$ for fitting the scaling of the current work.




(a) Snapshot of the $v_r$ velocity component interpolated onto the mid-sphere surface, $R = (R_i + R_o)/2$: isometric and top views.




(b) Time evolution and frequency spectra of the Nusselt number, $\overline{Nu_o}$ averaged over the surface of the outer sphere.

**Fig. 21.** Characteristics of the periodic natural convection flow developing between two concentric spheres for the values of $Ra = 4.6 \times 10^4$ and $\phi = 0.714$.

resolutions, respectively. In both cases the number of Lagrangian points was dictated by the necessity to preserve the same spacing for the Eulerian and Lagrangian grids.

For natural convection flow, the memory consumption and the time step duration scale as $N_s^{1.09}$ and $N_s^{1.37}$, respectively, with the total number of Eulerian points, $N_s$, as follows from Fig. 22-(a). Maximum values of about 17 Gb RAM and 6 s of memory consumption and duration of the time step, respectively, were obtained for the finest $300^3$ grid resolution. Remarkably, the memory consumption and the time step duration scale as $N_g^{0.72}$ and $N_g^{0.91}$, respectively, with the total number of grid points, as follows from Fig. 22 (b). The obtained result is not surprising, since for the equally spaced grids the total number of Eulerian grid points scales as $\Delta x^{1.5}$ to the total number of Lagrangian grid points, in keeping with the corresponding exponent values of $N_s$ and $N_g$. Note that the same trend is also observed for the lid-driven cavity configuration, for which the ratio between the corresponding exponents of $N_s$ and $N_g$ is also maintained at 1.5 (see Figs. 22-(c) and (d)). At the same time, there are two clearly observed differences between the corresponding efficiency characteristics typical of the two configurations. First, for the natural convection flow, the slopes characterizing the power law functionality of the memory consumption and the time step duration are different, while for the lid-driven cavity flow the two slopes are nearly the same. Second, the exponent values characterizing the time step duration scaling typical of the natural convection configuration are lower compared to the corresponding values obtained for the lid-driven cavity flow. The first difference can apparently be attributed to the fact that for the natural convection flow the time duration step consists of the sum of the times required for the solution of both the energy and the NS equations, while for the lid-driven cavity flow the time duration step includes only the time required for the solution of the NS equations. The second difference can be explained by the fact that the natural convection configuration is characterized by an order of magnitude higher number of Lagrangian points, $N_s$, and is thus advantageous for strong scaling by the currently utilized openMP parallelization. In general, the upper limit of the exponent value characterizing the scalability of the developed approach as a function of the Eulerian and

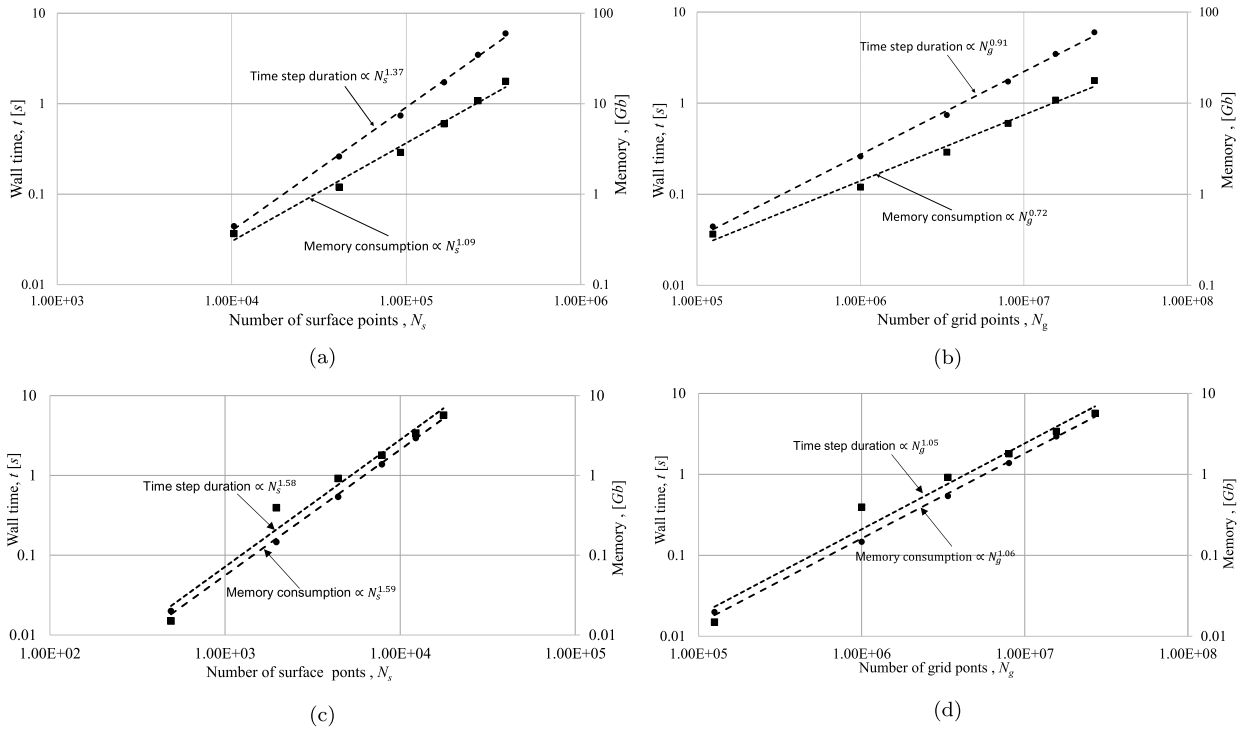



**Fig. 22.** Memory consumption and duration of a single time step as a function of the number of points on the surfaces of both spheres, $N_s$, and the total number of grid points, $N_g$ obtained when simulating: natural convection flow developing between two concentric spheres for the values of $Ra = 4.6 \times 10^4$ and $\phi = 0.714$ (a-b); or lid-driven cavity flow within a cube with an embedded sphere of diameter $0.4L$ placed at the center of the cavity for the value of $Re = 400$ (c-d).

Lagrangian grid points is limited to 1.6, which is close to the $O(N^{4/3})$ complexity of the original solver [2], not upgraded with the IB capability; this indicates the high efficiency of the currently developed approach.

### 3.8. Future potential of the developed methodology

Adapting the developed IB methodology to the simulation of incompressible flows in the presence of non-stationary immersed bodies (e.g., moving boundary and two-way coupled FSI configurations) is a promising direction that has yet to be developed. The key idea is to exploit the fact that the Poisson-body forces system of Eqs. (13) constitutes a regularized saddle point problem in which the non-zero matrix $\boldsymbol{C}$ and its inverse $\boldsymbol{C}^{-1}$ can be approximated as $\boldsymbol{C} = -\frac{1}{2}\mathbf{I}$ and $\boldsymbol{C}^{-1} = -2\mathbf{I}$, respectively. As a result, the order in the Schur complement decomposition of the system of Eqs. (13) can be switched to proceed first with finding the pressure corrections field and then to continue with obtaining a solution for the Lagrangian force corrections field as:

$$\begin{cases} p' = (\boldsymbol{L} + 2\boldsymbol{B}^T\boldsymbol{B})^{-1}(RHS_{p'} + 2\boldsymbol{B}^T RHS_{\boldsymbol{F}'}), \\ \boldsymbol{F}' = 2(\boldsymbol{B}p' - RHS_{\boldsymbol{F}'}). \end{cases} \tag{22}$$

Note that for the above formulation the matrix $(\boldsymbol{L} + 2\boldsymbol{B}^T\boldsymbol{B})$ can be rebuilt very rapidly, as only $\boldsymbol{B}^T$ and $\boldsymbol{B}$ need to be updated and multiplied in each time step. The updating of the above matrices is "embarrassingly" parallel and can be conveniently implemented by employing either distributed or shared parallelization, while the multiplication of the two matrices can be implemented by employing the standard routines of any available linear algebra (e.g. LAPACK) library. The drawbackof the above formulation derives from the fact that in this case the matrix-vector product of the operator $(\boldsymbol{L} + 2\boldsymbol{B}^T\boldsymbol{B})^{-1}$ cannot be implemented by the available solvers of Laplace or Helmholts equations, which implies the necessity to develop a solver specifically dedicated to the solution of system (13). This development will require an in-depth investigation of the structure and properties of the above operator and is currently under investigation by us.

## 4. Conclusions

In this study, a new implicit formulation of the IB method, based on the SIMPLE algorithm adapted for coupling between velocity, pressure, and Lagrangian forces was developed. In this methodology, the pressure and Lagrangian forces fill the role of the DLMs imposing the incompressibility and the kinematic no-slip constraints, respectively. The developed methodology

**Table A.12**
Sequence of operations for proceeding by a single time step.

| | |
|---|---|
| 1. | Initialize the velocity $\boldsymbol{u}^{n,n-1}$ and pressure $p^n$ fields with initial values or use them from the previous time steps. |
| 2. | Calculate the intermediate velocity field by solving the momentum equation and the values of the velocity and pressure fields from the previous time steps: $\frac{3\boldsymbol{u}^*-4\boldsymbol{u}^n+\boldsymbol{u}^{n-1}}{2\Delta t} = -\nabla p^n + \frac{1}{Re}\nabla^2\boldsymbol{u}^* - (\boldsymbol{u}\cdot\nabla\boldsymbol{u})^n$. |
| 3. | Calculate the pressure correction term by employing the continuity equation and solving the Poisson equation: $\nabla^2 p' = \frac{3}{2\Delta t}\nabla\cdot\boldsymbol{u}^*$. |
| 4. | Calculate the velocity correction field: $\boldsymbol{u}' = -\frac{2}{3}\Delta t\nabla p'$. |
| 5. | Proceed to the next time step by assigning $p^{n+1} = p^n + p'$, $\boldsymbol{u}^{n+1} = \boldsymbol{u}^* + \boldsymbol{u}'$. |

formulates the Poisson-body force system of equations constituting the regularized saddle point problem, the solution of which yields coupled fields of pressure and Lagrangian forces corrections. The corrections are then used to obtain the velocity field, which simultaneously meets the incompressibility and the no-slip constraints.

A physically justified approximation of the Poisson-body force system of equations was developed. On the one hand, the structure of the approximated Poisson-body force system of equations resembles that obtained when applying the penalty method. On the other hand, in contrast to the penalty method, which relaxes a constrained problem by solving a series of unconstrained problems with a gradually decreasing penalty parameter, the approximated Poisson-body force system is an a priori-constrained, albeit approximated, system. The approximated Poisson-body force system was solved by a direct solver for the $2D$ configuration and by an iterative BiCgStab solver for a number of realistic $3D$ configurations. The BiCGStab algorithm was applied to the Schur complement of the top left block of the Poisson-body force system constituting the Laplace operator. The very structure of the Schur complement, provided a high convergence rate (no more than three iterations at any time step) of the BiCGStab algorithm.

The methodology developed in this study extends the state-of-the-art toolbox of available semi- and fully implicit IB methods. The method was successfully verified by applying it to the analysis of a number of representative $2D$ and $3D$ shear- and thermally driven confined steady-state and supercritical flows. The efficiency of the method was quantified by means of scaling its time step and memory consumption characteristics. In light of the fact that the Poisson-body forces system of equations constitutes a regularized saddle point problem, the potential of the developed methodology in its further adaptation to the simulation of moving boundary and two-way coupled FSI configurations is discussed, and a number of promising directions worthy of further research are opened.

## CRediT authorship contribution statement

Conceptualization: Yuri Feldman, Oz Oshri, Kirill Goncharuk; Methodology: Yuri Feldman, Kirill Goncharuk; Software: Yuri Feldman, Kirill Goncharuk; Validation: Kirill Goncharuk; Data Curation: Yuri Feldman, Oz Oshri; Writing – Original Draft: Kirill Goncharuk; Writing – Review & Editing: Yuri Feldman, Oz Oshri; Project administration: Yuri Feldman, Oz Oshri; Funding acquisition: Yuri Feldman, Oz Oshri.

## Declaration of competing interest

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

## Data availability

No data was used for the research described in the article.

## Acknowledgement

This research was partially supported by the Israel Science Foundation (grant No. 950/22).

## Appendix A. Brief description of the SIMPLE algorithm

The SIMPLE algorithm [37] is a widely used numerical method for solving incompressible NS equations. The algorithm is based on a pressure-correction approach, in which the velocity and pressure fields are iteratively corrected until a converged solution is obtained. The SIMPLE algorithm has been successfully applied to a wide range of problems in fluid dynamics. The sequence of operations of the algorithm for proceeding by a single time step in order of execution is summarized in Table 1. Note that a number of iterations can be required to provide a divergence-free velocity field up to a given precision prior to proceeding to the next time step. In this case, the Steps 2-5 should be reiterated, while Step 2 uses the updated pressure field to recalculate the predicted velocity $\boldsymbol{u}^*$ and this, in turn, is used in Step 3 to recalculate the pressure correction field $p'$. See Table A.12.

## Appendix B. Interpolation and regularization operators

Following the formalism of the originally developed IB method [3], the surface of the immersed body is determined by a set of Lagrangian points not necessarily coinciding with an underlying Eulerian grid on which the continuity, NS, and energy equations are discretized and solved. For this reason, two adjoint operators, namely, the interpolation operator $\boldsymbol{I}$ and the regularization operator $\boldsymbol{R}$, are introduced. The interpolation operator $\boldsymbol{I}$ interpolates the Eulerian velocities and temperatures on the Lagrangian points, whereas the regularization operator $\boldsymbol{R}$ smears the Lagrangian volumetric forces and heat fluxes to the neighboring Eulerian grid. The interpolated velocities and temperatures are then used in the equations responsible for imposing the kinematic constraints of no-slip and prescribed temperature, respectively. The smeared volumetric forces and heat fluxes are, in turn, introduced as sources into the corresponding NS and energy equations.

Both the interpolation and the regularization operators used in this study utilize the discrete Dirac delta function proposed by Roma et al. [16]:

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

where $\Delta r$ is the cell width in the $r$ direction. This delta function has a compact support (only three grid cells in each direction), and obeys five basic rules (see Ref. [16] for additional details), which provide proper smoothness and conservation of the smeared/interpolated fields and makes the delta function applicable for the use on staggered grids. However, these rules can be met exactly only when the delta function is applied to equally spaced Eulerian grids and Lagrangian markers, while violation of these rules may result in increased stiffness of the interpolation and regularization operators.

The regularization and interpolation operators are defined as convolutions of the corresponding smeared or interpolated function with two- or three-dimensional discrete Dirac delta functions:

$$\boldsymbol{R}(\boldsymbol{F}^k(\boldsymbol{X}^k), Q^k(\boldsymbol{X}^k)) = \int_S (\boldsymbol{F}^k(\boldsymbol{X}^k), Q^k(\boldsymbol{X}^k)) \cdot \delta(\boldsymbol{x}_i - \boldsymbol{X}^k) dV_S^k, \tag{B.2a}$$

$$\boldsymbol{I}(\boldsymbol{u}(\boldsymbol{x}_i), \theta(\boldsymbol{x}_i)) = \int_\Omega (\boldsymbol{u}(\boldsymbol{x}_i), \theta(\boldsymbol{x}_i)) \cdot \delta(\boldsymbol{X}^k - \boldsymbol{x}_i) dV_{\Omega_i}, \tag{B.2b}$$

where $dV_S^k$ corresponds to the finite volume of the thin shell surrounding the $k^{th}$ Lagrangian point of the immersed body surface and $dV_{\Omega_i}$ corresponds to the $i^{th}$ finite volume of the Eulerian grid. Note that to provide the best accuracy of the results, the two finite volumes should be close to each other.

## Appendix C. Supplementary material

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.jcp.2023.112148.

## References

[1] W. Press, S. Teukolsky, W. Vetterling, B. Flannery, Numerical Recipes in Fortran 77: The Art of Scientific Computing, Cambridge University Press, 1989.
[2] H. Vitoshkin, A.Y. Gelfgat, On direct inverse of Stokes, Helmholtz and Laplacian operators in view of time-stepper-based Newton and Arnoldi solvers in incompressible CFD, Commun. Comput. Phys. 14 (2013) 1103–1119.
[3] C.S. Peskin, Flow patterns around heart valves: a numerical method, J. Comput. Phys. 10 (1972) 252–271.
[4] F. Sotiropoulos, X. Yang, Immersed boundary methods for simulating fluid-structure interaction, Prog. Aerosp. Sci. 65 (2014) 1–21.
[5] W. Hang, F. Tian, Recent trends and progress in the immersed boundary method, Proc. Inst. Mech. Eng., Part C, J. Mech. Eng. Sci. 223 (2019) 7617–7636.
[6] W. Kim, H. Choi, Immersed boundary methods for fluid-structure interaction: a review, Int. J. Heat Fluid Flow 75 (2019) 301–3–09.
[7] W. Xiao, H. Zhang, K. Luo, C. Mao, J. Fan, Immersed boundary method for multiphase transport phenomena, Rev. Chem. Eng. 38 (2022) 363–405.
[8] Y. Tseng, J. Ferziger, A ghost-cell immersed boundary method for flow in complex geometry, J. Comput. Phys. 192 (2003) 593–623.
[9] J. Lee, D. You, An implicit ghost-cell immersed boundary method for simulations of moving body problems with control of spurious force oscillations, J. Comput. Phys. 233 (2013) 295–314.
[10] D.M. Ingram, D.M. Causon, C.G. Mingham, Developments in cartesian cut cell methods, Math. Comput. Simul. 61 (2003) 561–572.
[11] R. Mittal, H. Dong, M. Bozkurttas, F.M. Najjar, A. Vargas, A. von Loebbecke, A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries, J. Comput. Phys. 227 (2008) 4825–4852.
[12] L. Duan, H. Wang, X. Zhong, A high-order cut-cell method for numerical simulation of hypersonic boundary-layer instability with surface roughness, J. Comput. Phys. 229 (2010) 7207–7237.
[13] D. Schillinger, L. Dede, M.A. Scott, J.A. Evans, M. Borden, E. Rank, T.J.R. Hughes, An isogeometric design-through-analysis methodology based on adaptive hierarchical refinement of NURBS, immersed boundary methods, and T-spline CAD surfaces, Comput. Methods Appl. Mech. Eng. 249 (2012) 116–150.
[14] A. Main, G. Scovazzi, The shifted boundary method for embedded domain computations. Part I: Poisson and Stokes problems, J. Comput. Phys. 372 (2018) 972–995.

[15] A. Main, G. Scovazzi, The shifted boundary method for embedded domain computations. Part II: Linear advection-diffusion and incompressible Navier-Stokes equations, J. Comput. Phys. 372 (2018) 996–1026.
[16] A. Roma, C.S. Peskin, M.J. Berger, An adaptive version of the immersed boundary method, J. Comput. Phys. 153 (1999) 509–534.
[17] R. Mittal, G. Iaccarino, The immersed boundary method, Annu. Rev. Fluid Mech. 37 (2005) 239–261.
[18] J. Mohd-Yusof, Combined Immersed-Boundary/B-Spline Methods for Simulations of Flow in Complex Geometries, Center for Turbulence Research, Annual Research Briefs, 1997, pp. 317–327.
[19] E.A. Faldun, R. Verzicco, P. Orlandi, J. Mohd-Yusof, Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations, J. Comput. Phys. 161 (2000) 35–60.
[20] J. Wu, C. Shu, Implicit velocity correction-based immersed boundary-lattice Boltzmann method and its applications, J. Comput. Phys. 228 (2009) 1963–1979.
[21] K. Taira, T. Colonius, The immersed boundary method: a projection approach, J. Comput. Phys. 225 (2007) 3121–3133.
[22] S. Liska, T. Colonius, A fast immersed boundary method for external incompressible viscous flows using lattice Green's functions, J. Comput. Phys. 331 (2016) 257–279.
[23] A. Goza, T. Colonius, A strongly-coupled immersed-boundary formulation for thin elastic structures, J. Comput. Phys. 336 (2017) 401–411.
[24] D. Stein, R. Guy, B. Thomases, Immersed Boundary Smooth Extension (IBSE): a high-order method for solving incompressible flows in arbitrary smooth domains, J. Comput. Phys. 335 (2017) 155–178.
[25] B. Kallemov, A. Bhalla, B. Griffith, A. Donev, An immersed boundary method for rigid bodies, Commun. Appl. Math. Comput. Sci. 11 (2016) 79–141.
[26] Y. Feldman, Y. Gulbeg, An extension of the immersed boundary method based on the distributed Lagrange multiplier approach, J. Comput. Phys. 322 (2016) 248–266.
[27] Y. Feldman, Oscillatory instability of 2D natural convection flow in a square enclosure with a tandem of vertically aligned cylinders, Fluid Dyn. Res. 50 (2018) 051410.
[28] M. Huang, M. Shelley, D. Stein, A stable and accurate scheme for solving the Stefan problem coupled with natural convection using the immersed boundary smooth extension method, J. Comput. Phys. 432 (2021) 110162.
[29] A. Spizzichino, S. Goldring, Y. Feldman, The immersed boundary method: application to two-phase immiscible flows, Commun. Comput. Phys. 25 (2019) 107–134.
[30] J. Eldredge, A method of immersed layers on Cartesian grids, with application to incompressible flows, J. Comput. Phys. 448 (2022) 110716.
[31] Y. Feldman, Semi-implicit direct forcing immersed boundary method for incompressible viscous thermal flow problems: a Schur complement approach, Int. J. Heat Mass Transf. 127 (2018) 1267–1283.
[32] R. Sela, E. Zemach, Y. Feldman, A semi-implicit direct forcing immersed boundary method for periodically moving immersed bodies: a Schur complement approach, Comput. Methods Appl. Mech. Eng. 373 (2021) 113498.
[33] A. Spizzichino, E. Zemach, Y. Feldman, Oscillatory instability of a 3d natural convection flow around a tandem of cold and hot vertically aligned cylinders placed inside a cold cubic enclosure, Int. J. Heat Mass Transf. 141 (2019) 327–345.
[34] E. Zemach, A. Spizzichino, Y. Feldman, Instability characteristics of a highly separated natural convection flow: configuration of a tandem of cold and hot horizontally oriented cylinders placed within a cold cubic enclosure, Int. J. Therm. Sci. 159 (2021) 106606.
[35] S.V. Patankar, D.B. Spalding, A calculation procedure for heat, mass and momentum in three-dimensional parabolic flows, Int. J. Heat Mass Transf. 15 (1972) 1787–1806.
[36] M. Benzi, G.H. Golub, J. Liesen, Numerical solution of saddle point problems, Acta Numer. 14 (2005) 1–137.
[37] S.V. Patankar, Numerical Heat Transfer and Fluid Flow, McGraw-Hill, 1980.
[38] J. Van Doormal, G. Raithby, Enhancements of the simple method for predicting incompressible fluid flows, Numer. Heat Transf. 7 (1984) 147–163.
[39] W. Zeng, W. Tao, A comparison study of the convergence characteristics and robusteness for four variants of SIMPLE-family at fine grids, Eng. Comput. 20 (2003) 320–340.
[40] T.J. Chung, Computational Fluid Dynamics, Cambridge University Press, 2010.
[41] Y. Gulberg, Y. Feldman, On laminar natural convection inside multi-layered spherical shells, Int. J. Heat Mass Transf. 91 (2015) 908–921.
[42] H.G. Weller, G. Tabor, H. Jasak, C. Fureby, A tensorial approach to computational continuum mechanics using object-oriented techniques, Comput. Phys. 12 (1998) 620–631.
[43] J. Perot, An analysis of the fractional step method, J. Comput. Phys. 108 (1993) 51–58.
[44] P. Amestoy, I. Duff, J. LÉxcellent, J. Koster, Multifrontal parallel distributed symmetric and unsymmetric solvers, Comput. Methods Appl. Mech. Eng. 184 (1998) 501–520.
[45] P. Amestoy, I. Duff, J. LÉxcellent, J. Koster, A fully asynchronous multifrontal solver using distributed dynamic scheduling, SIAM J. Matrix Anal. Appl. 23 (2001) 15–41.
[46] C.S. Peskin, The immersed boundary method, Acta Numer. 11 (2002) 479–517.
[47] S. Gsell, U. D'Ortona, J. Favier, Explicit and viscosity-independent immersed-boundary scheme for the lattice Boltzmann method, Phys. Rev. E (2019) 033306.
[48] S. Gsell, J. Favier, Direct-forcing immersed-boundary method: a simple correction preventing boundary slip error, J. Comput. Phys. 435 (2021) 110265.
[49] V. Babu, S. Korpela, Numerical solutions of the incompressible, three-dimensional Navier-Stokes equations, Comput. Fluids 23 (1994) 675–691.
[50] U. Ghia, K. Ghia, C. Shin, Hig-re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method, J. Comput. Phys. 48 (1982) 387–411.
[51] B. Souayeh, M. Alam, F. Hammami, N. Hdhiri, E. Yasin, Predicting the unsteady states of 2D and 3D lid-driven cavities induced by a centrally located circle and sphere, Fluid Dyn. Res. 52 (2020) 025507.
[52] J. Zhu, L.E. Holmedal, H. Wang, D. Myrhaug, Flow patterns in a steady lid-driven rectangular cavity with an embedded circular cylinder, Fluid Dyn. Res. 52 (2020) 065508.
[53] S.-G. Cai, A. Ouahsine, J. Favier, Y. Hoarau, H. Smaoui, Immersed boundary method for the simulation of lid-driven cavity flow with an embedded cylinder, in: VI International Conference on Computational Methods for Coupled Problems in Science and Engineering, CIMNE, 2015, pp. 1130–1137.
[54] S.-G. Cai, A. Ouahsine, J. Favier, Y. Hoarau, Moving immersed boundary method, Int. J. Numer. Methods Fluids 85 (2017) 288–323.
[55] C. Santarelli, T. Kempe, J. Fröhlich, Immersed boundary methods for heat transfer, Int. J. Numer. Methods Heat Fluid Flow 26 (2016) 504–514.
[56] K. Khanafer, S. Aithal, Laminar mixed convection flow and heat transfer characteristics in a lid driven cavity with a circular cylinder, Int. J. Heat Mass Transf. 66 (2013) 200–209.
[57] T. Huang, H. Zhao, H. Chen, Y. Yao, P. Yu, A hybrid cartesian-meshless method for the simulation of thermal flows with complex immersed objects, Phys. Fluids 34 (2022) 103318.
[58] Y. Wang, C. Shu, L. Yang, Boundary condition-enforced immersed boundary-lattice Boltzmann flux solver for thermal flows with Neumann boundary conditions, J. Comput. Phys. 306 (2016) 237–252.
[59] P. Leopardi, A partition of the unit sphere into regions of equal area and small diameter, Electron. Trans. Numer. Anal. 25 (2006) 309–327.
[60] D. Young, Y. Lin, C. Fan, C. Chiu, The method of fundamental solutions for solving incompressible Navier–Stokes problems, Eng. Anal. Bound. Elem. 33 (2009) 1031–1044.

[61] A. Povitsky, Three-dimensional flow in cavity at yaw, Nonlinear Anal., Theory Methods Appl. 63 (2005) e1573–e1584.
[62] Y. Feldman, Theoretical analysis of three-dimensional bifurcated flow inside a diagonally lid-driven cavity, Theor. Comput. Fluid Dyn. 29 (2015) 245–261.
[63] Y. Feldman, A.Y. Gelfgat, Oscillatory instability of a three-dimensional lid-driven flow in a cube, Phys. Fluids 22 (2010) 093602.
[64] A. Gelfgat, Linear instability of the lid-driven flow in a cubic cavity, Theor. Comput. Fluid Dyn. 33 (2019) 59–82.
[65] H.S. Yoon, D.H. Yu, M.Y. Ha, Y.G. Park, Three-dimensional natural convection in an enclosure with a sphere at different vertical locations, Int. J. Heat Mass Transf. 53 (2010) 3143–3155.
[66] J. Jeong, F. Hussain, On the identification of a vortex, J. Fluid Mech. 285 (1995) 69–94.
[67] Y.M. Seo, J.H. Doo, M.Y. Ha, Three-dimensional flow instability of natural convection induced by variation in radius of inner circular cylinder inside cubic enclosure, Int. J. Heat Mass Transf. 95 (2016) 566–578.
[68] N. Scurtu, B. Futterer, C. Egbers, Three-dimensional natural convection in spherical annuli, J. Phys. Conf. Ser. 137 (012017) (2008) 1–9.
[69] Y. Feldman, T. Colonius, On a transitional and turbulent natural convection in spherical shells, Int. J. Heat Mass Transf. 64 (2013) 514–525.
[70] V. Travnikov, K. Eckert, S. Odenbach, Linear stability analysis of the convective flow in a spherical gap with $\eta$ = 0.714, Int. J. Heat Mass Transf. 80 (2015) 266–273.