An extension of the immersed boundary method based on the distributed Lagrange multiplier approach

Yuri Feldman *, Yosef Gulberg

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

ARTICLE INFO

ABSTRACT

An extended formulation of the immersed boundary method, which facilitates simulation of incompressible isothermal and natural convection flows around immersed bodies and which may be applied for linear stability analysis of the flows, is presented. The Lagrangian forces and heat sources are distributed on the fluid–structure interface. The method treats pressure, the Lagrangian forces, and heat sources as distributed Lagrange multipliers, thereby implicitly providing the kinematic constraints of no-slip and the corresponding thermal boundary conditions for immersed surfaces. Extensive verification of the developed method for both isothermal and natural convection 2D flows is provided. Strategies for adapting the developed approach to realistic 3D configurations are discussed.

1. Introduction

Since the immersed boundary (IB) method was first introduced by Peskin [1], the IB method and its modifications have become very popular numerical tools for describing the flow around moving or deformable bodies with complex surface geometry [2,3]. An arbitrary immersed object, whose geometry does not, in general, have to conform to an underlying spatial grid, is typically determined by a set of Lagrangian points. At the Lagrangian points, appropriate volumetric (or surface) forces are applied to enforce no-slip velocity boundary conditions on the body surface. These forces appear as additional unknown variables, whose values – along with those for the pressure and velocity fields – are obtained by solving the Navier Stokes (NS) equations. Since the location of the Lagrangian boundary points does not necessarily coincide with the underlying spatial discretization, interpolation and regularization operators must be defined to convey information to and from the body surface.

An accurate calculation of the Lagrangian forces, precisely enforcing the no-slip constraint on the surface of the immersed body, is the key issue in any IB formulation. Lagrangian forces acting on rigid bodies (as well as on bodies with a prescribed surface motion) can be treated explicitly or implicitly. Historically, explicit treatment of Lagrangian forces has received the most attention, giving rise to the direct forcing approach, introduced by Mohd-Yusof [4] and coauthors [5], and to the immersed interface method (IIM), introduced by Lee and LeVeque [6] and revisited by Linnick and Fasel [7]. The direct forcing approach has recently been extended to thermal flow problems, see e.g. [8–11], by adding an energy equation along with the appropriate volumetric heat sources at the Lagrangian points. The direct forcing approach is not a standalone solver; rather, it may be viewed as a feature that can be easily plugged into an existing time marching solver, typically developed for the solution of NS equations on structured grids in rectangular domains. The procedure does not require

---

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

any significant modifications to the existing time marching solver, which explains why the direct forcing approach is so popular. However, the direct forcing approach has a number of drawbacks. First, the no-slip condition is explicitly enforced on the intermediate non-solenoidal velocity field, whereas the divergence-free velocity field is calculated afterwards, after a projection–correction step. Second, it should be stressed that even if the NS equations are exactly solved by the projection method, resulting in a solenoidal velocity field on the Eulerian grid, the velocity interpolated to the Lagrangian points is not necessarily divergence free, which may result in a local mass leakage through the boundaries of the immersed body. Third, a pointwise local calculation of the Lagrangian forces and heat sources does not take into account their mutual interaction, which contradicts the elliptic character of the NS equations. A number of techniques have been developed in the past decade to improve the accuracy of the direct forcing approach. Worth mentioning here are the works of Ren et al. [9,10], who proposed an implicit evaluation of all the Lagrangian forces and heat sources by assembling them into a single system of equations. Another approach is due to Kempe et al. [12,13], who introduced additional iterations to enhance Euler–Lagrange coupling, thereby providing a substantially more accurate imposition of the boundary conditions on the immersed body surface.

A coupled scheme in which the momentum equations are implicitly coupled with the Lagrangian forces and heat sources and simultaneously solved as a whole system offers an alternative to the direct forcing approach. The closure of this new system is achieved by adding equations interpolating the Eulerian velocity and the temperature fields on the surface of the immersed body to enforce the prescribed boundary conditions. In this setup, the Lagrangian forces and heat sources distributed on the fluid–structure interface can be seen as distributed Lagrange multipliers, enforcing velocity and temperature constraints on the surface of the immersed body similarly to the pressure comprising distributed Lagrange multiplier that acts to enforce the solenoidal constraint on the velocity field. The power of the coupled Lagrange multiplier approach is that it can be straightforwardly adapted to various numerical methods and applications in fluid mechanics, providing accurate and physically substantiated results. Chronologically, the idea was first expressed in the distributed Lagrange multiplier method (DLM) of Glowinski et al. [14], who used a variational principle framework for discretization of the NS equations and applied it for the simulation of 2D flow around moving disc [14]. More recently, the method was successfully extended to the simulation of particulate flows [15–17] and to the simulation of fluid/flexible-body interactions [18]. An additional impact on the active development of the coupled Lagrange multiplier approach was due to the work of Taira and Colonius [19], who combined the coupled IB method with a projection approach to satisfy the divergence-free and no-slip kinematic constraints. That study was further successfully implemented for the investigation of steady blowing into separated flows behind low-aspect-ratio rectangular wings [20]; for prediction of the natural convection heat transfer and buoyancy of a hot air balloon [21]; for the simulation of rigid-particle-laden flows [22]; for investigation of the forces and unsteady flow structures associated with harmonic oscillations of an airfoil [23]; and recently for simulating the dynamic interactions between incompressible viscous flows and rigid-body systems [24]. The latest theoretical developments of the coupled Lagrange multiplier approach can be found in two recent studies: Kallemov et al. [25] developed a novel (IB) formulation for modeling flows around fixed or moving rigid bodies suitable for a broad range of Reynolds numbers, including steady Stokes flow, and Stein et al. [26] established immersed boundary smooth extension (IBSE) method, which demonstrates fourth- and third-order pointwise convergence for Dirichlet and Neumann problems, respectively.

The present paper reports on our ongoing effort aimed at extension of the coupled Lagrange multiplier approach to problems involving buoyancy-driven flows, steady-state non-Stokes flows and linear stability analysis of the flows in the presence of immersed bodies of arbitrary shape. To demonstrate the new capabilities of the coupled Lagrange multiplier approach, we utilize the previously developed fully pressure–velocity coupled direct solver (FPCD) [27] as a computational platform. The idea is similar to that established by Taira and Colonius [19], the only differences being that the pressure–velocity coupling is implemented by **LU**-decomposition of the full Stokes operator instead of by the projection approach. This allows us to formulate a full Jacobian operator to compute the steady-state solution and then to conduct a linear stability analysis by a shift-invert Arnoldi iteration. To the best of our knowledge, to date the only available approach embedding IB functionality into a linear stability analysis is that due to Giannetti and Luchini [28], who utilized an adjoint NS operator (in addition to the direct one) to couple between the immersed body and the surrounding isothermal flow. The present approach does not involve an adjoint NS operator, which is an advantage for computational efficiency.

The paper is organized as follows. In section 2, the numerical formulation of the developed methodology is presented. The section includes an introductory description of the previously developed FPCD solver (section 2.1), the concepts of IB formalism, based on the Lagrange multipliers approach (section 2.2), a detailed description of the time marching solver developed in this work (section 2.3), the steady-state solver (section 2.4) and the linear stability solver (section 2.5). An extended discussion of the pros and cons of the presented approach, including the general strategies for the further enhancement of the established methodology, is presented in section 2.6. Section 3 presents a detailed verification of all the developed solvers for incident and natural convection incompressible 2*D* flows. The final section presents a summary and the main conclusions of the study.

2. The numerical formulation

The developed numerical methodology, based on the implicit formulation of the IB method and a fully pressure–velocity coupled approach, incorporates three solvers: a time marching solver for the time integration of the NS equations; a steady-state solver based on the full Newton iteration; and a linear stability solver for calculating the necessary part of the whole

spectrum of the flow by utilizing the Arnoldi iteration method. All three solvers are based on the previously developed fully pressure–velocity coupled direct (FPCD) solver [29,27] briefly described here for the sake of completeness.

2.1. The FPCD solver

We consider the 2D NS equations for isothermal incompressible flow:

$$\nabla \cdot \boldsymbol{u} = 0, \tag{1a}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = -\nabla p + \frac{1}{Re}\nabla^2 \boldsymbol{u}, \tag{1b}$$

where $\boldsymbol{u}(u, v)$, p, and Re are the non-dimensionalized velocity vector, the pressure field, and the Reynolds number, respectively. By applying a second-order backward finite difference scheme for time discretization, Eqs. (1) can be rewritten as:

$$\nabla \cdot \boldsymbol{u}^{n+1} = 0, \tag{2a}$$

$$[\frac{1}{Re}\nabla^2 \boldsymbol{u} - \frac{3}{2\Delta t}\boldsymbol{u}]^{n+1} - \nabla p = [(\boldsymbol{u} \cdot \nabla)\boldsymbol{u} - \frac{2}{\Delta t}\boldsymbol{u}]^n + \frac{1}{2\Delta t}\boldsymbol{u}^{n-1}. \tag{2b}$$

Note that all the non-linear terms are taken from the previous time step and moved to the right hand side (RHS) of Eqs. (2). The system of vector Eqs. (2b) can be compactly rewritten in a block-matrix form as:

$$\begin{bmatrix} H_u & 0 & -\nabla_p^x \\ 0 & H_v & -\nabla_p^y \\ \nabla_u^x & \nabla_v^y & 0 \end{bmatrix} \begin{bmatrix} u^{n+1} \\ v^{n+1} \\ p \end{bmatrix} = \begin{bmatrix} RHS_u^{n-1,n} \\ RHS_v^{n-1,n} \\ 0 \end{bmatrix}, \tag{3}$$

where ∇^x and ∇^y are the first derivatives with respect to the x and y coordinates, respectively, $H = \frac{1}{Re}\Delta - 3I/2\Delta t$ are the corresponding Helmholtz operators acting on u and v velocity components, I is the identity operator, and Δ is the Laplacian operator. The lower indices correspond to the scalar fields on which an operator acts. The left hand side (LHS) of Eqs. (3), known as the Stokes operator, is further discetized with a standard staggered mesh second-order conservative finite-volume formulation [30]. Non-linear terms, moved to the RHS of Eqs. (3), are approximated by the conservative central differencing scheme to exclude the appearance of artificial viscosity (see Ref. [29] for the discretization details). Following Refs. [29,27], the fully pressure–velocity coupled solution of Eqs. (3) can be obtained by **LU**-factorization of the Stokes operator with a set of suitable boundary conditions for all the velocity components and a single Dirichlet reference point for the pressure field. The discrete Stokes operator remains unchanged during the solution, reducing the time integration of the NS equations to two backward substitutions at each time step. The efficiency of the above approach (see Ref. [29] for the characteristic computational times) is boosted by utilizing a modern multifrontal direct solver for sparse matrices (MUMPS[1]), exploiting the sparseness of the discrete Stokes operator at both **LU**-factorization and back substitution stages. The FPCD approach formulated in Eqs. (3) can be straightforwardly adjusted to the simulation of natural convection flows, with buoyancy effects being introduced by the Boussinesq approximation and governed by:

$$\nabla \cdot \boldsymbol{u} = 0, \tag{4a}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} = -\nabla p + Gr^{-0.5}\nabla^2 \boldsymbol{u} + \theta \overrightarrow{e_y}, \tag{4b}$$

$$\frac{\partial \theta}{\partial t} + (\boldsymbol{u} \cdot \nabla)\theta = Pr^{-1}Gr^{-0.5}\nabla^2 \theta, \tag{4c}$$

where $\boldsymbol{u}$, θ, and p correspond to the non-dimensionalized velocity, the temperature and the pressure fields respectively, Gr is the Grashof number, Pr is the Prandtl number, and $\overrightarrow{e_y}$ is the unit vector in the opposite direction to gravity. Discretizing the time by a second-order backward finite difference scheme leads to:
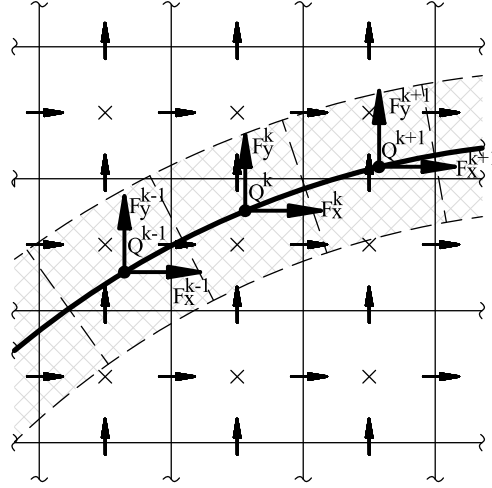
$$\nabla \cdot \boldsymbol{u}^{n+1} = 0, \tag{5a}$$

$$[Gr^{-0.5}\nabla^2 \boldsymbol{u} - \frac{3}{2\Delta t}\boldsymbol{u} + \theta \overrightarrow{e_y}]^{n+1} - \nabla p = [(\boldsymbol{u} \cdot \nabla)\boldsymbol{u} - \frac{2}{\Delta t}\boldsymbol{u}]^n + \frac{1}{2\Delta t}\boldsymbol{u}^{n-1}, \tag{5b}$$

$$[Pr^{-1}Gr^{-0.5}\nabla^2 \theta - \frac{3}{2\Delta t}\theta]^{n+1} = [(\boldsymbol{u} \cdot \nabla)\theta - \frac{2}{\Delta t}\theta]^n + \frac{1}{2\Delta t}\theta^{n-1}. \tag{5c}$$

Then, using the same notations as for Eqs. (3), the compact block-matrix form of the vector Eqs. (5) reads:

---

[1] http://mumps.enseeiht.fr/.

Fig. 1. Schematic representation of a staggered grid discretization of a two-dimensional computational domain D with a segment of an immersed boundary of a body B. A virtual shell, whose thickness is equal to the grid cell width, is shaded. The horizontal and vertical arrows ($\rightarrow$, $\uparrow$) represent the discrete u_i and v_i velocity locations, respectively. Pressure p_j and temperature T_j are applied at the center of each cell ($\times$). Lagrangian points $\boldsymbol{\xi}^k(\xi^k, \ \eta^k)$ along ∂B are shown as black circles $\bullet$ where volumetric boundary forces $\boldsymbol{F}^k = (F_x^k, \ F_y^k)$ and volumetric boundary heat sources Q^k are applied.

$$
\begin{bmatrix}
H_u & 0 & 0 & -\nabla_p^x \\
0 & H_v & \overrightarrow{e_y} & -\nabla_p^y \\
0 & 0 & H_\theta & 0 \\
\nabla_u^x & \nabla_v^y & 0 & 0
\end{bmatrix}
\begin{bmatrix}
u^{n+1} \\
v^{n+1} \\
\theta^{n+1} \\
p
\end{bmatrix}
=
\begin{bmatrix}
RHS_u^{n-1,n} \\
RHS_v^{n-1,n} \\
RHS_\theta^{n-1,n} \\
0
\end{bmatrix},
\tag{6}
$$

where $H_u = H_v = Gr^{-0.5}\Delta - 3I/2\Delta t$ are the Helmholtz operators for the scalar momentum equations, and $H_\theta = Pr^{-1}Gr^{-0.5}\Delta - 3I/2\Delta t$ is the Helmholtz operator for the energy equation. All the other notations and the spatial discretization are the same as in Eqs. (3). The discrete differential operators in the LHS of Eqs. (3) and (6) can contain different boundary conditions, and therefore for the general case $H_u \neq H_v$, and $\nabla_u^x \neq \nabla_p^x$, $\nabla_v^y \neq \nabla_p^y$.

2.2. The immersed boundary formalism

The IB method can be viewed as a "philosophy" for enforcing boundary conditions on the surface of an immersed body of an arbitrary shape. The boundary of an immersed body is typically preset by a series of Lagrangian points $\boldsymbol{X}^k$, whose location does not necessarily coincide with the underlying Eulerian grid. Each Lagrangian point is associated with the corresponding discrete volume ΔV^k, such that an ensemble of these volumes forms a thin shell (see Fig. 1). The boundary conditions are enforced by introducing additional functions in the form of volumetric forces, $\boldsymbol{F}^k$, and heat sources, Q^k, each associated with the corresponding volume ΔV^k. The values of the above functions are not known a priori and are an inherent part of the overall solution in the present implicit formulation. To exchange information between the Eulerian grid and the Lagrangian points, regularization R and interpolation I operators are defined:

$$
R(\boldsymbol{F}^k(\boldsymbol{X}^k), Q^k(\boldsymbol{X}^k)) = \int_S (\boldsymbol{F}^k(\boldsymbol{X}^k), Q^k(\boldsymbol{X}^k)) \cdot \delta(\boldsymbol{x}_i - \boldsymbol{X}^k) dV_S^k,
\tag{7a}
$$

$$
I(\boldsymbol{u}(\boldsymbol{x}_i), \theta(\boldsymbol{x}_i)) = \int_\Omega (\boldsymbol{u}(\boldsymbol{x}_i), \theta(\boldsymbol{x}_i)) \cdot \delta(\boldsymbol{X}^k - \boldsymbol{x}_i) dV_{\Omega i},
\tag{7b}
$$

where S corresponds to all the cells belonging to the immersed body surface, Ω corresponds to a group of flow domain cells located in the close vicinity of the immersed body surface, dV_S^k corresponds to the virtual volume surrounding each Lagrangian point k, and $dV_{\Omega i}$ is the volume of the corresponding cell of the Eulerian flow domain, whose velocity and temperature values are explicitly involved in enforcing the boundary conditions at point k of the immersed body. The purpose of the regularization operator R is to smear the volumetric forces, $\boldsymbol{F}^k$, and heat sources, Q^k, on the nearby computational domain by embedding them as sources into the corresponding momentum/energy equations. At the same time, the interpolation operator I imposes no-slip/thermal boundary conditions on the Lagrangian points of the body surface by adding the equations necessary to achieve closure of the overall system. Both operators use convolutions with the Dirac delta function δ to facilitate an exchange of information between the Lagrangian points of the body surface and the Eulerian grid. The discrete delta function introduced by Roma et al. [31] was used in the present study:

$$d(r) = \begin{cases} \frac{1}{6\Delta r}\left[5 - 3\frac{|r|}{\Delta r} - \sqrt{-3\left(1 - \frac{|r|}{\Delta r}\right)^2 + 1} \right] & \text{for } 0.5\Delta r \le |r| \le 1.5\Delta r, \\ \frac{1}{3\Delta r}\left[1 + \sqrt{-3\left(\frac{|r|}{\Delta r}\right)^2 + 1} \right] & \text{for } |r| \le 0.5\Delta r, \\ 0 & \text{otherwise,} \end{cases} \tag{8}$$

where Δr is the cell width in the r direction. The chosen delta function was specifically derived for use on staggered grids, and it has been successfully utilized in a number of previous studies [32,19,12]. The delta function involves only three cells in each computational direction, which is an advantage for computational efficiency. To provide the best accuracy, the method utilizes a uniform grid in the vicinity of the immersed body surface. In this region, the distance between the neighboring points of the immersed body surface Δl and the width of a grid cell should be approximately the same (i.e., $\Delta l \approx \Delta x = \Delta y$ and $dV_S^k \approx dV_{\Omega i}$). Away from the body, non-uniform discretization can be used. The general discrete forms of the regularization and interpolation operators for 2D geometry are governed by Eqs. (9):

$$(\boldsymbol{f}_i, q_i) = \Delta x \Delta y \sum_k (\boldsymbol{F}^k, Q^k) d(\epsilon^k - x_i) d(\eta^k - y_i), \tag{9a}$$

$$(\boldsymbol{U}^k, \Theta^k) = \Delta x \Delta y \sum_i (\boldsymbol{u}_i, \theta_i) d(x_i - \epsilon^k) d(y_i - \eta^k), \tag{9b}$$

where $\boldsymbol{f}_i$, q_i are the discrete volumetric force and heat source, respectively, defined on a staggered grid (x_i, y_i) and $\boldsymbol{U}^k$, Θ^k are the discrete boundary velocity and temperature, respectively, defined at the k-th Lagrangian point (ϵ^k, η^k). Following Peskin [1] and Beyer and LeVeque [33], we used the same delta functions for interpolation and regularization operators. The contour of the immersed body should not contain repeating Lagrangian points. In addition, if a certain Lagrangian point moves with velocity $\boldsymbol{U}$, its location should be updated at each time step by time integration of the corresponding velocity.

2.3. Implicit immersed boundary FPCD time stepper

The discrete pressure p appearing in Eqs. (2)–(6) does not actively participate in time propagation and therefore can be viewed as the Lagrange multiplier that constrains the solenoidal velocity field (see e.g. [34,19]). It is therefore reasonable to augment the existing Stokes operators (see Eqs. (3) and (6)) with the IB functionality by adding an additional set of Lagrange multipliers to enforce the appropriate boundary conditions at the Lagrangian points. Formally, the extended block-matrix form of the Stokes operator for 2D isothermal incompressible flow (see Eqs. (3)) is formulated as:

$$\begin{bmatrix} H_u & 0 & -\nabla_p^x & R_{F_x} & 0 \\ 0 & H_v & -\nabla_p^y & 0 & R_{F_y} \\ \nabla_u^x & \nabla_v^y & 0 & 0 & 0 \\ I_u & 0 & 0 & 0 & 0 \\ 0 & I_v & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u^{n+1} \\ v^{n+1} \\ p \\ F_x \\ F_y \end{bmatrix} = \begin{bmatrix} RHS_u^{n-1,n} \\ RHS_v^{n-1,n} \\ 0 \\ U_b \\ V_b \end{bmatrix}. \tag{10}$$

Here, the vertical and horizontal dashed lines separate between the original Stokes operator, located at the top left corner of the matrix, and the additional entries related to the embedded immersed boundary functionality. These additional entries can be formally divided into two types. The first type corresponds to the weights of the unknown non-dimensional volumetric forces, F_x and F_y, obtained by applying the regularization operator R, smearing the forces over the vicinity of the Lagrangian points. The second type corresponds to the weights of the Eulerian velocity components. To impose no-slip boundary conditions, the sum of the above weights, each multiplied by its Eulerian velocity component, should be equal to the velocities U_b and V_b of the corresponding Lagrangian points. In other words, entries of the second type are nothing more than the additional equations necessary to achieve closure of the whole system of Eqs. (10), after the unknowns F_x and F_y have been added. It should be noted that as a result of the utilization of the same Dirac delta functions in both the interpolation I and regularization R operators and the uniform staggered grid in the near vicinity of the immersed body surface, the interpolation and regularization operators are transposed to each other, $R_F = I_{\boldsymbol{u}}^T$. Note also that for all rigid stationary immersed bodies the values of U_b and V_b are all equal to zero and the extended Stokes operator in Eqs. (10) does not vary in time. As a result, **LU**-factorization of the extended Stokes operator should be performed only once at the beginning of the computational procedure. For moving/deforming bodies, the location of the Lagrangian points is updated at each time step, requiring modification of the extended Stokes operator (see Eqs. (10)) with its subsequent **LU**-factorization. The factorization can be efficiently performed on a massively parallel machine, taking advantage of the high scalability parallelization built-in into the MUMPS solver [35].

Using the same notations as for Eqs. (6) and (10), an extended immersed boundary formulation for the natural convection flow can be written as:

$$
\begin{bmatrix}
H_u & 0 & 0 & -\nabla_p^x & R_{F_x} & 0 & 0 \\
0 & H_v & \overrightarrow{e_y} & -\nabla_p^y & 0 & R_{F_y} & 0 \\
0 & 0 & H_\theta & 0 & 0 & 0 & R_Q \\
\nabla_u^x & \nabla_v^y & 0 & 0 & 0 & 0 & 0 \\
I_u & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & I_v & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & I_\theta & 0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
u^{n+1} \\ v^{n+1} \\ \theta^{n+1} \\ p \\ F_x \\ F_y \\ Q
\end{bmatrix}
=
\begin{bmatrix}
RHS_u^{n-1,n} \\ RHS_v^{n-1,n} \\ RHS_\theta^{n-1,n} \\ 0 \\ U_b \\ V_b \\ \Theta
\end{bmatrix}.
\tag{11}
$$

Similarly to the Eqs. (10), the original Stokes operator located at the top left corner of the block-matrix form is separated by the vertical and horizontal dashed lines from the immersed boundary entries. The R_Q entries correspond to the weights of the unknown non-dimensional volumetric heat sources smeared over the vicinity of the corresponding Lagrangian points by the regularization operator R, whereas the I_θ entries are the weights of the Eulerian temperatures, imposing Dirichlet boundary conditions at the neighboring Lagrangian points.

In most thermal problems, precise estimation of the average $\overline{Nu}$ number is of significant practical importance and is particularly critical for the present implementation of the IB method, which relies on a uniform Cartesian grid. As a result, a further refining of the Eulerian grid adjacent to the immersed boundary for a more precise resolution of the thinnest boundary layers is not practical. An alternative way to obtain an accurate estimation the $\overline{Nu}$ number is to express the unknown Lagrangian non-dimensional volumetric heat sources in terms of the temperature gradients in the direction normal to the immersed boundary as:

$$
Q = \frac{1}{Pr\sqrt{Gr}\Delta x}\frac{\partial\theta}{\partial\boldsymbol{n}},
\tag{12}
$$

where $\Delta x = \Delta y$ is the dimension of the uniform Eulerian grid in the vicinity of the immersed surface. Following [9], the $\overline{Nu}$ value averaged over the surface of the immersed body reads:

$$
\overline{Nu} = \frac{1}{2}\left(\sum_{k=1}^{M}\frac{\partial\theta}{\partial\boldsymbol{n}}\Delta x\right)_k,
\tag{13}
$$

where the local $\frac{\partial\theta}{\partial\boldsymbol{n}}$ values at every point $1 \leq k \leq M$ of the immersed body are provided by the solution of Eqs. (11), reformulated in terms of the temperature gradients in the direction normal to the body surface. Following the same principle, the drag C_d and the lift C_l coefficients can be obtained by:

$$
(C_d, C_l) = -2\sum_{k=1}^{M}(F_{x_k}, F_{y_k})/\rho U_\infty d,
\tag{14}
$$

where F_{x_k} and F_{y_k} are an intrinsic part of the overall solution obtained at every point k of the immersed body and $\rho U_\infty d = 1$ for the presently used normalization.

The above immersed boundary formulation embedded into the FPCD time stepper can be seen as an extension of the algorithm recently developed by Taira and Colonius [19], who coupled unknown volumetric forces acting at the Lagrangian points with an intermediate non-solenoidal velocity field, which was then be further projected to the divergence free subspace by a projection–correction step. Based on the full pressure–velocity coupling, the present direct method does not require the projection–correction step.

2.4. Steady-state immersed boundary FPCD solver

A steady isothermal incompressible flow with an embedded immersed boundary functionality is governed by the following continuity and momentum equations:

$$
\nabla \cdot \boldsymbol{u} = 0,
\tag{15a}
$$

$$
(\boldsymbol{u}\cdot\nabla)\boldsymbol{u} + \nabla p - \frac{1}{Re}\nabla^2\boldsymbol{u} - R_{\boldsymbol{F}} = 0,
\tag{15b}
$$

$$
I(\boldsymbol{u}) - \boldsymbol{U}_b = 0,
\tag{15c}
$$

where $R_{\boldsymbol{F}}$ and $I(\boldsymbol{u})$ are additional entries resulting from applying the regularization R and interpolation I operators. Note that the steady-state formulation formally treats the flow around an immersed body in the same way as its unsteady analog given by Eqs. (10). All the differential operators of Eqs. (15) are subsequently discretized in space by the standard staggered grid second-order conservative finite-volume method (in the same way as in the corresponding unsteady formulation). All the additional entries related to the IB formulation are discretized by using discrete Dirac delta functions. The discretized Eqs. (15) summarized in a compact block-matrix form in Eqs. (16) are then solved by the Newton–Raphson method.

$$
\begin{bmatrix}
J_x & 0 & J_p & R_{F_x} & 0 \\
0 & J_y & J_p & 0 & R_{F_y} \\
J_u & J_v & 0 & 0 & 0 \\
I_u & 0 & 0 & 0 & 0 \\
0 & I_v & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
\delta(u) \\
\delta(v) \\
\delta(p) \\
\delta(F_x) \\
\delta(F_y)
\end{bmatrix}
= -
\begin{bmatrix}
Fn_x & - \sum_k R_{kF_x} \\
Fn_y & - \sum_k R_{kF_y} \\
Fn_p \\
\sum_i I_{iu_x} - U_{b_x} \\
\sum_i I_{iu_y} - U_{b_y}
\end{bmatrix},
\tag{16}
$$

where J_x, J_y, J_p, J_u, J_v entries of Jacobian $\boldsymbol{J}$ correspond to the discrete linearized terms of the original (without IB functionality) momentum and continuity equations, with the corresponding discrete right-hand sides Fn_x, Fn_y, Fn_p being calculated at the iteration n. The additional entries $R_{\boldsymbol{F}}$ and $I_{\boldsymbol{u}}$ of the Jacobian operator, related to the embedded IB formulation, are separated by the horizontal and vertical dashed lines. The IB entries also contribute to the RHS of Eqs. (16). The sums of smeared volumetric forces $\boldsymbol{F}_k$ and interpolated velocities $\boldsymbol{u}_i$, both calculated at iteration n, are added to the corresponding right hand sides of the momentum equations and to the complementary interpolation relations. Here, the indexes i and k represent the total number of Eulerian and Lagrangian points, respectively, participating in the summation.

The developed steady-state IB solver can be straightforwardly adjusted to the steady-state solution of the natural convection flow, governed by:

$$\nabla \cdot \boldsymbol{u} = 0, \tag{17a}$$

$$(\boldsymbol{u} \cdot \nabla)\boldsymbol{u} + \nabla p - Gr^{-0.5}\nabla^2 \boldsymbol{u} - \theta \overrightarrow{e_y} - R_{\boldsymbol{F}} = 0, \tag{17b}$$

$$(\boldsymbol{u} \cdot \nabla)\theta - Pr^{-1}Gr^{-0.5}\nabla^2\theta - R_Q = 0, \tag{17c}$$

$$I(\boldsymbol{u}) - \boldsymbol{U}_b = 0, \tag{17d}$$

$$I(\theta) - \Theta_b = 0, \tag{17e}$$

where the Boussinesq approximation is utilized for simulating the buoyancy effects, and again $R_{\boldsymbol{F}}$, R_Q, $I(\boldsymbol{u})$, $I(\theta)$ are the additional entries stemming from applying the regularization R and interpolation I operators. Utilizing the same spatial discretization and Dirac delta functions as for Eqs. (16), the discretized Eqs. (17) are solved by the Newton–Raphson method, whose compact block-matrix form reads:

$$
\begin{bmatrix}
J_x & 0 & 0 & J_p & R_{F_x} & 0 & 0 \\
0 & J_y & \overrightarrow{e_y} & J_p & 0 & R_{F_y} & 0 \\
0 & 0 & J_\theta & 0 & 0 & 0 & R_Q \\
J_u & J_v & 0 & 0 & 0 & 0 & 0 \\
I_u & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & I_v & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & I_\theta & 0 & 0 & 0 & 0
\end{bmatrix}
\begin{bmatrix}
\delta(u) \\
\delta(v) \\
\delta(\theta) \\
\delta(p) \\
\delta(F_x) \\
\delta(F_y) \\
\delta(Q)
\end{bmatrix}
= -
\begin{bmatrix}
Fn_x & - \sum_k R_{kF_x} \\
Fn_y & - \sum_k R_{kF_y} \\
Fn_\theta & - \sum_k R_{kQ} \\
Fn_p \\
\sum_i I_{iu_x} - U_{b_x} \\
\sum_i I_{iu_y} - U_{b_y} \\
\sum_i I_{i\theta} - \Theta_b
\end{bmatrix}.
\tag{18}
$$

The compact block-matrix form of Eqs. (18) bears a striking resemblance to that determined by Eqs. (16) (corresponding to the isothermal flow), the only exceptions being the additional entries related to the energy equations and to the Dirichlet temperature boundary conditions applied to the immersed surface. As was done in the time integration analysis, the volumetric heat sources Q can be expressed in terms of the normal temperature gradients $\frac{\partial \theta}{\partial \boldsymbol{n}}$ (see Eq. (12)) required for the precise estimation of the $\overline{Nu}$ value. Note that all the entries related to the IB functionality (i.e., I_u, I_v, I_θ, R_{Fx}, R_{Fy}, R_Q) are linear and therefore have the same form both in the Stokes operator (see Eqs. (10) and (11)) and in the corresponding Jacobian operator (see Eqs. (16) and (18)).

2.5. Linear stability immersed boundary FPCD solver

For the sake of conciseness, only equations for the linear stability analysis of the natural convection flow will be derived in this section. The equations for the linear stability of the isothermal flow can be obtained by a straight-forward omission of the energy equations and the temperature terms in the corresponding momentum equations. The linear stability eigenproblem is formulated by assuming infinitesimally small perturbations in the form of $\{\widetilde{\boldsymbol{u}}(x, y), \ \widetilde{\theta}(x, y), \ \widetilde{p}(x, y), \ \widetilde{\boldsymbol{F}}(x, y), \ \widetilde{Q}(x, y)\}e^{\lambda t}$ around the steady state flow $\boldsymbol{U}$, Θ, P, $\boldsymbol{F}$, Q, as follows:

$$\lambda\tilde{\boldsymbol{u}} = -(\boldsymbol{U} \cdot \nabla)\tilde{\boldsymbol{u}} - (\tilde{\boldsymbol{u}} \cdot \nabla)\boldsymbol{U} - \nabla\tilde{p} + Gr^{-0.5}\nabla^2\tilde{\boldsymbol{u}} - \tilde{\theta}\overrightarrow{e_y} + R_{\tilde{\boldsymbol{F}}}, \tag{19a}$$

$$\lambda\tilde{\theta} = -(\boldsymbol{U} \cdot \nabla)\tilde{\theta} - (\tilde{\boldsymbol{u}} \cdot \nabla)\Theta + Pr^{-1}Gr^{-0.5}\nabla^2\tilde{\theta} + R_{\tilde{Q}}, \tag{19b}$$

$$0 = \nabla \cdot \tilde{\boldsymbol{u}}, \tag{19c}$$

$$0 = I(\tilde{\boldsymbol{u}}), \tag{19d}$$

$$0 = I(\tilde{\theta}), \tag{19e}$$

or in a block-matrix form as:

$$
\lambda \boldsymbol{B} \begin{bmatrix} \tilde{\boldsymbol{u}} \\ \tilde{\theta} \\ \tilde{p} \\ \tilde{\boldsymbol{F}} \\ \tilde{Q} \end{bmatrix} = \boldsymbol{J} \begin{bmatrix} \tilde{\boldsymbol{u}} \\ \tilde{\theta} \\ \tilde{p} \\ \tilde{\boldsymbol{F}} \\ \tilde{Q} \end{bmatrix},
\tag{20}
$$

where $\boldsymbol{J}$ is the Jacobian matrix calculated from the RHS of Eqs. (19), and $\boldsymbol{B}$ is the diagonal matrix whose diagonal elements, corresponding to the values of $\tilde{\boldsymbol{u}}$, $\tilde{\theta}$ are equal to unity, and whose diagonal elements, corresponding to $\tilde{p}$, $\tilde{\boldsymbol{F}}$, $\tilde{Q}$, are equal to zero. Note that for Cartesian coordinates and the staggered uniform grid in the vicinity of immersed body surface, the discrete Jacobians, $\boldsymbol{J}$ of Eqs. (18) and (20) are the same. The generalized eigenproblem (20) cannot be directly transformed into a standard eigenproblem, since $det(\boldsymbol{B}) = 0$; instead it is solved in a shift-invert mode:

$$
(\boldsymbol{J} - \sigma \boldsymbol{B})^{-1} \boldsymbol{B} \begin{bmatrix} \tilde{\boldsymbol{u}} \\ \tilde{\theta} \\ \tilde{p} \\ \tilde{\boldsymbol{F}} \\ \tilde{Q} \end{bmatrix} = \mu \begin{bmatrix} \tilde{\boldsymbol{u}} \\ \tilde{\theta} \\ \tilde{p} \\ \tilde{\boldsymbol{F}} \\ \tilde{Q} \end{bmatrix}, \qquad \mu = \frac{1}{\lambda - \sigma}
\tag{21}
$$

The solution is based on a standard Arnoldi iteration implemented within an open source ARPACK package,[2] providing the dominant eigenvalue (i.e., the eigenvalue with the largest modulus). In a linear stability analysis, we are typically interested in finding the critical value of the control parameter (e.g., Gr_{cr} or Re_{cr} numbers) at which $Real(\lambda) = 0$ (to a prescribed precision), where λ is the leading eigenvalue. The dominant eigenvalue μ can be related to the leading eigenvalue λ (i.e., that of a zero real part) when the approach is applied to a shift-invert problem, where σ is a complex shift (see Eqs. (21)). To converge, the approach requires that the complex shift σ[3] be close to the λ value, whose imaginary part $Im(\lambda)$ corresponds to the critical angular oscillating frequency, ω_{cr}. The value of ω_{cr} is either known a priori (for benchmark problems) or can be estimated by a series of successive direct numerical simulations of the slightly bifurcated flow.

The present linear stability approach extends the algorithm presented by Gelfgat [36], with an IB functionality. Theoretically, no specific restrictions are imposed either on the number of bodies or on their shape. However, the method requires that the body boundaries do not touch or intersect and that the minimal distance between neighboring bodies is at least the size of a single grid cell. The solution procedure is as follows. First, the steady-state solution is calculated by the Newton method for the given value of the control parameter (Gr or Re numbers). Then, the linear stability analysis is performed by utilizing a shift-invert Arnoldi iteration (see Eqs. (21)). The corresponding eigenvalue problem is solved by a secant method, providing a precise value for the critical control parameter. The overall process requires numerous solutions of large systems of linear equations, which should be performed at each step of the Newton method and while building the Krylov basis for the Arnoldi iteration. Typically, no more than ten iterations are required for the calculation of the steady-state solution (by the Newton method), while the shift-invert Arnoldi iteration needs $O(10^3$–$10^4)$ iterations to converge, thus comprising the key issue determining the computational efficiency of the whole process.

Next, to efficiently implement the product of the operator $(\boldsymbol{J} - \sigma \boldsymbol{B})^{-1} \boldsymbol{B}$ by the vector $[\tilde{\boldsymbol{u}}, \tilde{\theta}, \tilde{p}, \tilde{\boldsymbol{F}}, \tilde{Q}]^T$ required at each Arnoldi iteration step, we exploit the fact that the operator $(\boldsymbol{J} - \sigma \boldsymbol{B})^{-1} \boldsymbol{B}$ does not change during the building of the Krylov basis for the Arnoldi iteration (see Eqs. (21)). The product implementation is simply a solution $\boldsymbol{X}$ of the linear system $(\boldsymbol{J} - \sigma \boldsymbol{B}) \boldsymbol{X} = \boldsymbol{B}[\tilde{\boldsymbol{u}}, \tilde{\theta}, \tilde{p}, \tilde{\boldsymbol{F}}, \tilde{Q}]^T$. By utilizing the direct solver MUMPS, the **LU**-decomposition of the operator $(\boldsymbol{J} - \sigma \boldsymbol{B})$ is performed once at the beginning of the process, and then each vector of the Krylov basis is obtained by just two subsequent back substitutions, whose complexity is comparable to that of matrix–vector multiplication. Note also that the overall performance is additionally boosted by being a $(\boldsymbol{J} - \sigma \boldsymbol{B})$ sparse matrix. The advantage of the above approach over algorithms utilizing modern Krylov-subspace-based iteration methods (e.g., preconditioned GMRES and BiCGstab) for building the Krylov basis for the Arnoldi iteration was extensively discussed in [36] for natural convective flows in cavities. In the present study, we successfully extended the approach by embedding the IB functionality and applied it to a linear stability analysis of both open and confined flows.

2.6. Pros and cons of the developed method

The methodology underlying the present study is based on direct **LU**-factorization of the Stokes operator (for the time integration of the NS equations) or its Jacobian (for the computation of steady state flow fields and their subsequent linear stability analysis) – all extended by the distributed Lagrange multiplier introduced to provide the kinematic constraints of no-slip and the corresponding thermal boundary conditions for immersed surfaces. The resulting linear system of equations

---

[2] http://www.caam.rice.edu/software/ARPACK/.
[3] Typically σ is a pure imaginary number, since $Real(\lambda) \rightarrow 0$ at $Gr \approx Gr_{cr}$, or $Re \approx Re_{cr}$.

is of the saddle point type, the solution of which represents a significant challenge due to the indefiniteness and poor spectral properties of the system. The **LU**-decomposition of such systems is typically performed by multifrontal direct solvers, e.g., MUMPS, PARDISO[4] and codes from the Harwell Subroutine Library (HSL),[5] intelligently exploiting the sparseness of the discretized operator at the factorization and back substitution stages. The attractiveness of the direct solvers applied to the solution saddle point type system was reported by Perugia et al. [37] when performing analysis of magnetostatic fields and by Gelfgat and coauthors when conducting the linear stability analysis of convective flows in 2D cavities and performing three-dimensional stability calculations for axi-symmetric [29,38] and periodic [39] flows. While direct sparse solvers can be quite competitive for 2D (or quasi-3D) configurations, including those discretized on dense grids, they are not suitable for realistic three-dimensional problems. Recalling that the primary focus of the present paper is to demonstrate extension of the distributed Lagrange multiplier approach to new classes of problems (i.e., problems involving buoyancy-driven flows, steady non-Stokes flows and linear stability analysis), we elaborate only a set of 2*D* configurations demonstrating the viability of the developed method for up to 2000^2 grids.

Solution of saddle point systems arising from 3D Stokes operator should adopt iterative methods employing fractional-step or projection approaches. For excellent discussions regarding the methods specifically designed to solve this kind of problem, the book of Toselli and Widlund [40] (Chapter 9) and the review of Benzi et al. [41] should be consulted. Here, we restrict ourselves to presenting a brief discussion on the strategies that can be adopted for extension of the IB Lagrange multiplier approach to the solution of steady-state NS equations and to performing linear stability analysis in the presence of immersed bodies. As already mentioned, the family of fractional-step and projection methods, which typically benefit from definiteness and sparseness of the resulting operators, provides a natural choice for the flow configurations where the direct **LU**-factorization of the Stokes operator fails. To be more precise, the projection approach is, in itself, a block-**LU**-factorization of the same operator, the only difference being that at each fractional step the individual sub-operators retain their identity and each resulting system is solved sequentially [42]. This property of the projection approach was exploited by Taira and Colonius [19] when extending the projection method with IB functionality by utilizing the distributed Lagrange multiplier approach. Generally speaking, any time stepping scheme can be used for the linear stability (bifurcation) analysis without further modification. Steady state flow can be obtained by time integration until flow convergence is reached. Then, the steady state can be perturbed by variation of a control parameter until a transition to unsteadiness takes place. However, these techniques are often inefficient, since they require high computational effort and do not formally yield leading eigenvectors of the flow field. In contrast, any existing time-stepping code can be adapted to perform the linear stability analysis [43], providing a driver for inverse power or exponential methods that can be used to extract the values of growth rates, control parameters and the leading eigenvectors of the investigated flow. The inverse power method relies on the Stokes preconditioning providing the fastest convergence at Δt, which is 10–1000 times the Δt used for the time integration of NS equations [43]. As a result, the method cannot be utilized by the projection or fractional step approaches, which require small time increments to satisfy the divergence free constraint. This limitation can be remedied by utilizing the exponential power method, which can be easily implemented, is very reliable, and has been successfully used to compute leading eigenpairs in many problems of hydrodynamic stability [44–47]. The main drawback of the method is its slow convergence, which can, however, be substantially relaxed by parallelizing the code and running it on massively parallel HPC (High Performance Computing) platforms.

3. Results

3.1. Unsteady flow: periodic incident flow around two horizontally aligned circular cylinders

Verification of the developed implicit IB FPCD time stepper was first performed for simulation of the secondary instabilities in the flow around a tandem arrangement of two equal horizontally aligned cylinders of diameter d, as shown in Fig. 2. All the simulations were performed in a square computational domain of size 44 d in each direction. The two cylinders were centered in the vertical direction, while a distance equal to 15 d was set between the center of the forward cylinder and the inlet boundary of the domain. The computational domain was discretized by a non-uniform 1400×1400 mesh in the following manner: a square subregion of size 10 d in each direction, confining the pair as shown in Fig. 2, was discretized by a uniform 1000×1000 mesh with a grid step equal to $\Delta x = \Delta y = 0.01$. The mesh was built out of the square subregion (see Fig. 2) by gradually increasing Δx and Δy grid steps, which finally attain the values of $\Delta x \approx \Delta y \approx 0.22$ at all boundaries of the computational domain. Three configurations, each corresponding to a different center-to-center distance of $\kappa = L_x/d = [1.5, 2.3, 5]$ between the cylinders, were simulated for the value of $Re = U_\infty d/\nu = 200$, thus representing three different vortex shedding scenarios, as shown in Fig. 3. Here, ν is the kinematic viscosity of the fluid. The solutions were obtained with the following set of boundary conditions:

$$u_x(x = 0, y) = u_x(x, y = 0, y = 44d) = 1, \tag{22a}$$

$$u_y(x = 0, y) = u_y(x, y = 0, y = 44d) = 0, \tag{22b}$$
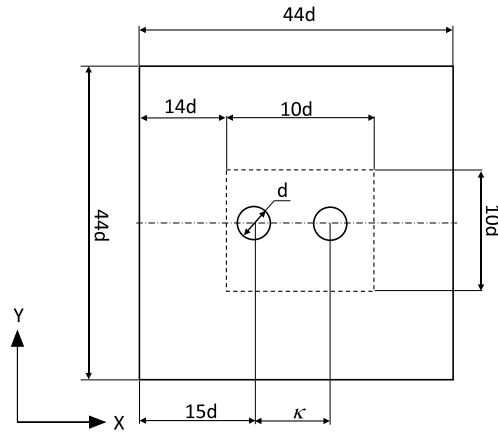
Fig. 2. Schematic representation of the geometrical model and discretization of the computational domain for the flow around two horizontally aligned cylinders arranged in tandem.

Table 1
Comparison between the values of oscillating frequencies f_{C_L} and f_{C_D} obtained for the lift and drag coefficients and the average value of the drag coefficient $\overline{C_D}$ measured on the downstream cylinder at $Re = 200$. All the reference values were obtained by digital scanning.

κ	1.5		2.3		5	
	Present	Ref. [49]	Present	Ref. [49]	Present	Ref. [49]
f_{C_L}	0.338	0.332	0.263	0.264	0.360	0.355
f_{C_D}	0.169	0.166	0.132	0.132	0.181	0.178
$\overline{C_D}$	−0.196	−0.194	−0.173	−0.172	0.423	0.401

$$p(x = 44d, y) = 0, \tag{22c}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{u}}{\partial x}(x = 44d, y) = 0. \tag{22d}$$

Note that Eq. (22d) determines the convective boundary condition at the outlet, allowing the vorticity to exit the domain freely [19]. The obtained results are in good qualitative (see Fig. 3) and quantitative (see Table 1) agreement with the corresponding flow characteristics obtained by Carmo et al. [48], who made a distinction between the three observed shedding scenarios, classifying them as SG (symmetric in the gap) for $\kappa = 1.5$, AG (alternating in the gap) for $\kappa = 2.3$, and WG (wake in the gap) for $\kappa = 5$. Note the interesting phenomenon of the drag inversion, extensively elaborated in [49] and characterized by a negative to positive change in the value of drag coefficient when the shedding regime changes from AG to WG.
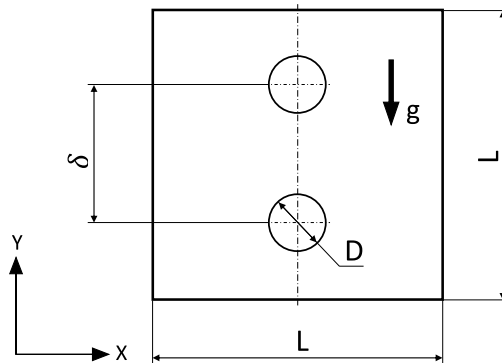
3.2. Unsteady flow: periodic natural convection flow around two vertically aligned circular cylinders

The next verification study of the developed time stepper was related to the simulation of unsteady natural convection flow around two cylinders confined by a square cavity (see Fig. 4). The ratio between the cylinder diameter, d, and the cavity side length, L, is equal to $d/L = 0.2$. The cylinders are aligned along the cavity's vertical centerline and are symmetrically distanced from the cavity's horizontal centerline. The distance δ between the cylinder centers, normalized by the cavity side length L, is equal to $\delta = 0.5$. Both cylinders are held at a constant hot temperature $\theta_H = 1$, whereas all the cavity boundaries are held at a constant cold temperature, $\theta_C = 0$. The force of gravity acts in the $-\hat{y}$ direction. Fig. 5 presents the flow characteristics of the periodic flow simulated at $Ra = 10^6$. A grid independence study was performed by comparing the velocity and temperature fields obtained on 400×400 and 500×500 grids. The maximum difference for all the flow characteristics obtained on the two grids did not exceed 0.5%, thereby successfully verifying the grid independence of the results. Figs. 5(a)–(b) present the time evolution of the amplitude, A ($A = \overline{Nu_l} - mean(\overline{Nu_l})$), where $mean(\overline{Nu_l})$ is the time averaged ($\overline{Nu_l}$) of the fluctuation of the average $\overline{Nu_l}$ number obtained for the lower cylinder and the corresponding frequency spectrum of A. Note that in agreement with the recent study of Park [50] the flow at $Ra = 10^6$ is governed by the single harmonics and its multipliers resulting from the flow non-linearity. The value of the angular frequency $\omega = 0.665$ is in a good agreement with the corresponding reference value, $\omega_{ref} = 0.656$, reported in [50].[6] Instantaneous streamlines and isotherms shown in Figs. 5(c)–(e) for the four representative times $[P_1, P_2, P_3, P_4]$, (see Fig. 5(a)) evenly distributed over the single oscillating period agree with the corresponding patterns reported in [50].

---

[6] Rescaled equivalent. Values reported in [50] were multiplied by the factor $\frac{1}{Pr\sqrt{Gr}}$ to fit the scaling adopted in this study.
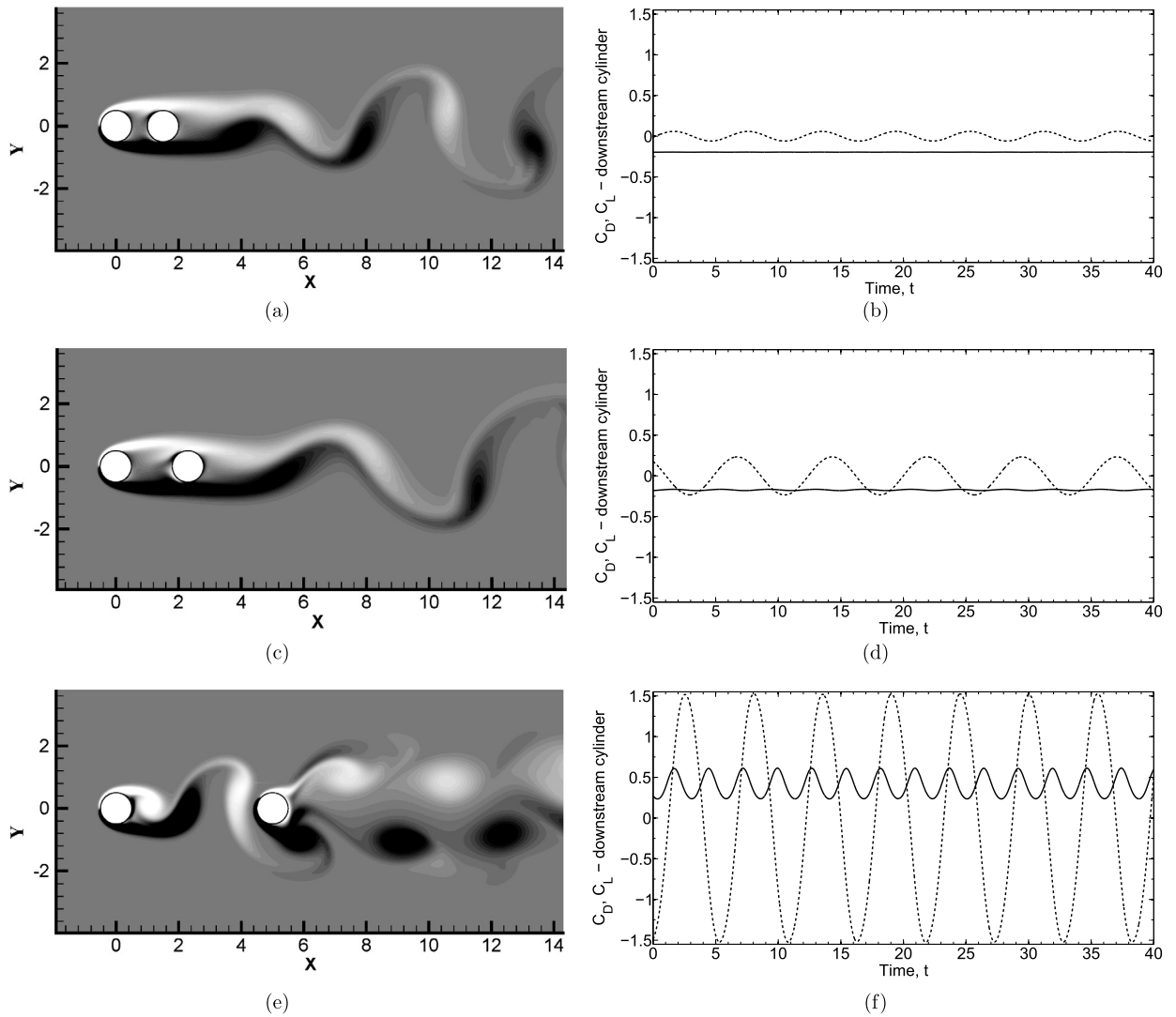
Fig. 3. Instantaneous vorticity contours, with drag coefficient, C_D (solid line) and lift coefficient C_L (dashed line), measured on the downstream cylinder at $Re = 200$ for: (a) $\kappa = 1.5$ – regime SG; (b) $\kappa = 2.3$ – regime AG; (c) $\kappa = 5$ – regime WG.

Fig. 4. Schematic representation of a geometrical model of the computational domain for the natural convection flow around vertically aligned cylinders confined by a square cavity.

(a)

(b)

(c)

(d)

(e)

(f)
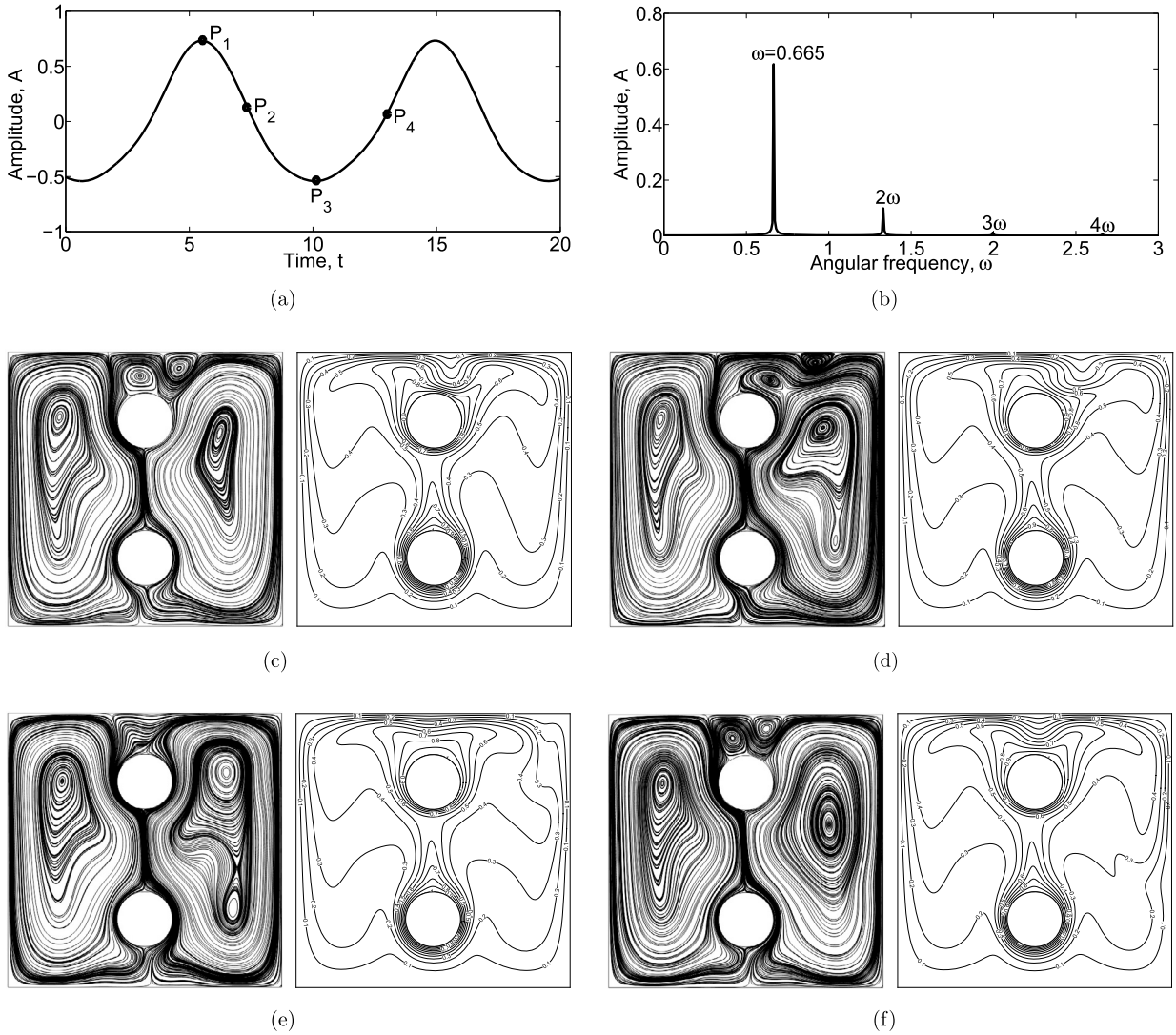
Fig. 5. Characteristics of periodic natural convection flow developing around two vertically aligned cylinders located inside a square cavity with all cold boundaries at $Ra = 10^6$ for $\delta = 0.5$: (a) time evolution of the fluctuation, A, of the Nusselt number $\overline{Nu_l}$ averaged over the surface of the lower cylinder ($A = \overline{Nu_l} - mean(\overline{Nu_l})$); (b) frequency spectrum of A; (c)–(f) instantaneous streamlines and isotherms at the selected time instances $[P_1, P_2, P_3, P_4]$.
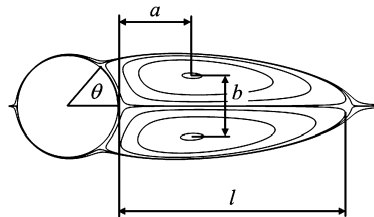
Fig. 6. Typical geometrical definitions of the steady state wake structure.

3.3. Steady-state flow: steady incident flow around a circular cylinder

This section presents a verification study of the fully implicit pressure–velocity coupled IB steady solver, based on the full Newton iteration, as defined by Eqs. (15)–(18). The characteristics of the wake structure, typical of isothermal steady state flow around a circular cylinder are defined in Fig. 6. The steady flow was simulated at $Re = 20$ and $Re = 40$, and the obtained results were compared with previously published data. The simulations were performed utilizing the same computational set up (including the geometry and the boundary conditions) as for the tandem arrangement of two cylinders
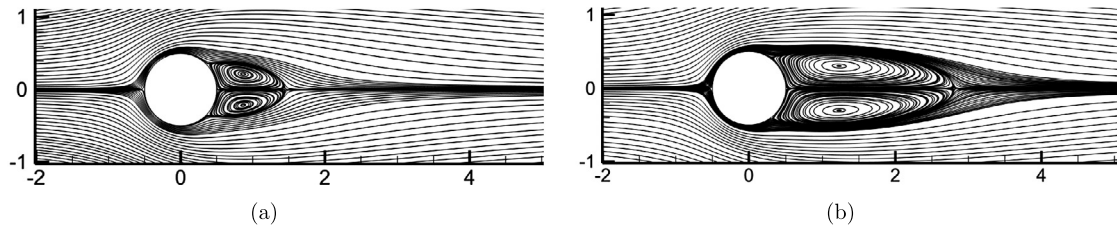
Fig. 7. Streamlines of the steady-state flow around a circular cylinder obtained for: (a) $Re = 20$; (b) $Re = 40$.

Table 2
Comparison of the wake characteristics and drag coefficients for steady-state flow over a cylinder for $Re = 20$ and $Re = 40$. Experimental results are denoted by a ⋆ symbol.

		l/d	a/d	b/d	θ	C_D
$Re = 20$	Present	0.95	0.37	0.43	42.9°	2.09
	Coutanceau and Bouard⋆ [51]	0.93	0.33	0.46	45.0°	–
	Taira and Colonius [19]	0.94	0.37	0.43	43.3°	2.06
	Linnick and Fasel [7]	0.93	0.36	0.43	43.5°	2.06
	Dennis and Chung [52]	0.94	–	–	43.7°	2.05
$Re = 40$	Present	2.13	0.76	0.59	53.3°	1.56
	Coutanceau and Bouard⋆ [51]	0.93	0.33	0.46	53.8°	–
	Taira and Colonius [19]	2.30	0.73	0.60	53.7°	1.54
	Linnick and Fasel [7]	2.28	0.72	0.60	53.6°	1.54
	Dennis and Chung [52]	2.35	–	–	53.8°	1.52

(see the previous section) by omitting the back cylinder. Figs. 7(a) and (b) demonstrate the typical steady flow patterns developing around a horizontal cylinder at $Re = 20$ and $Re = 40$, respectively. As expected, the flow is symmetric relative to the horizontal centerline with two recirculating bubbles, located behind the cylinders. The geometrical characteristics of the bubbles for the two different values of Re were compared with the literature data (see Table 2).

Good quantitative agreement was observed between all the wake characteristics simulated in this study and those reported in the literature, thus verifying the developed steady-state solver for isothermal incompressible flows.

3.4. Steady-state flow: steady natural convection confined flow

Simulation of natural convection confined flow was the focus of the next verification study. A configuration comprising an isothermal hot circular cylinder located at the center of a square cavity with all isothermal cold boundaries was chosen. Distributions of the isotherm contours and the stream function for three different R/L ratios, $R/L = 0.1, 0.2, 0.3$, as shown in Fig. 8, agree with the corresponding data reported by Seta [53]. Table 3 presents a quantitative comparison of the $\overline{Nu}$ numbers, averaged over the cylinder surface, and of the absolute maximum values of the stream function $|\Psi_{max}|$ with the corresponding literature values. Acceptable agreement was found between our values and those reported in the literature for the entire range of Ra and R/L ratio values, thus verifying the developed steady-state solver applied to the simulation of natural convection flows.

3.5. Linear stability analysis: incident flow around a circular cylinder

The developed IB FPCD linear stability solver was verified by conducting a linear stability analysis of the incident base flow around a circular cylinder. The base flow was calculated by the steady-state solver, which was verified in the previous section. The same computational domain and grid resolution as for the case of steady-state flow analysis around a circular cylinder were utilized. Perturbation values of all the flow fields were set to zero at all the boundaries. Figs. 9(a) and 9(b) present the contours of vorticity of the real and imaginary parts of the leading eigenmode, respectively. Both patterns are symmetric relative to the $Y = 0$ line and are characterized by alternating minima and maxima values in the streamwise direction. Contours of the vorticity of the imaginary part of the leading eigenmode (Fig. 9(b)) bear a striking resemblance to the corresponding pattern reported by Barkley [56]. Figs. 9(c) and 9(d) present a quantitative comparison between the obtained and reference [56] values for the frequency and growth rate, calculated in the range of $30 \leq Re \leq 180$ by the linear stability analysis. It can be seen that both quantities are in good agreement for the entire range of Re values, thereby successfully verifying the developed linear stability solver.

3.6. Linear stability analysis: incident flow around two horizontally aligned circular cylinders

Fig. 10 presents values of the critical Reynolds number, Re_{cr} and the critical oscillating frequency, f_{cr} as a function non-dimensional distance between the cylinder centers, l/d. The simulations were performed by utilizing the same setup (size of the computational domain and grid resolution) as for the analysis of unsteady flow, detailed in Section 3.1. As can
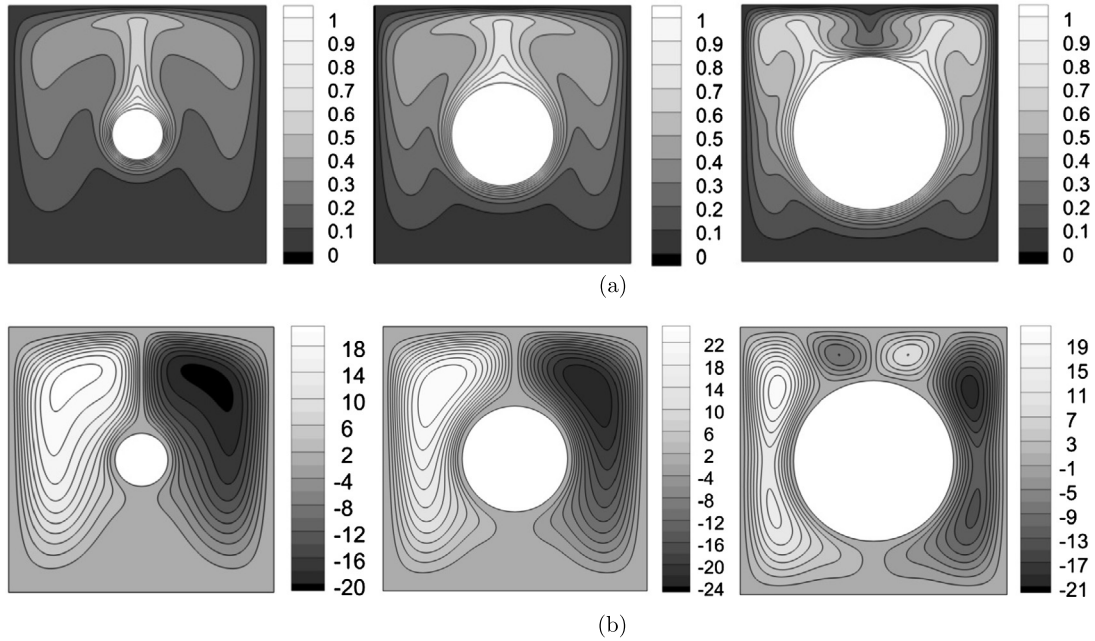
Fig. 8. Flow characteristics, obtained for $Ra = 10^6$, and $R/L = 0.1, 0.2, 0.3$: (a) isotherm contours; (b) stream function.

Table 3
Comparison of the $\overline{Nu}$ number averaged over the cylinder surface and of the maximum absolute values of the stream function (rescaled equivalent, the calculated values of stream function were multiplied by the $Pr\sqrt{Gr}$ factor). All the results for the current study were obtained on a 400×400 grid.

		$Ra = 10^4$		$Ra = 10^5$		$Ra = 10^6$	
		$\overline{Nu}$	$|\Psi_{max}|$	$\overline{Nu}$	$|\Psi_{max}|$	$\overline{Nu}$	$|\Psi_{max}|$
$R/L = 0.1$	Present	2.083	1.768	3.803	10.05	6.146	20.78
	Seta [53]	2.206	1.743	3.987	10.11	6.542	21.05
	Moukalled and Acharya [54]	2.071	1.73	3.825	10.15	6.107	25.35
	Shu and Zhu [55]	2.08	1.71	3.79	9.93	6.11	20.98
$R/L = 0.2$	Present	0.95	0.997	0.43	8.271	8.949	23.92
	Seta [53]	3.461	0.981	5.253	8.267	9.547	24.23
	Moukalled and Acharya [54]	3.331	1.02	5.08	8.38	9.374	24.07
	Shu and Zhu [55]	3.24	0.97	4.86	8.10	8.90	24.13
$R/L = 0.3$	Present	5.402	0.494	6.246	5.046	11.967	20.23
	Seta [53]	5.832	0.486	6.685	5.023	12.87	20.33
	Moukalled and Acharya [54]	5.826	0.50	6.107	5.10	11.62	21.30
	Shu and Zhu [55]	5.40	0.49	6.21	5.10	12.00	20.46

be seen from Table 4, the obtained Re_{cr} values agree with the independent results reported by Carmo et al. [49] for the entire range of l/d values. Note the non-homogeneity of the $Re - l/d$ functionality for the investigated range of l/d values. The non-homogeneity can be explained by the existence of three different vortex shedding regimes SG, AG, and WG. It is noteworthy that the $f_{cr} - l/d$ functionality, shown in Fig. 10(b), exhibits a different trend. The value decays continuously for the range of $1.5 \le l/d < 4$. Thereafter, the trend is reversed, and the value of f_{cr} grows continuously, attaining a maximum at $l/d \approx 4.9$. Finally, a rapid decrease of the f_{cr} value is observed at $l/d = 5$.

Additional evidence for the existence of three different vortex shedding regimes for the incident flow around two horizontally aligned cylinders as a function of the distance between the cylinder centers is provided by the contours of the corresponding leading eigenvectors obtained for $Re = Re_{cr}$. In fact, both the real and imaginary parts of the leading eigenvectors of vorticity exhibit different patterns for the three different values of l/d distance, as shown in Fig. 11. As expected, the largest differences in the perturbation fields are observed in the intermediate region between the cylinders and in the wake in the close vicinity of the trailing cylinder. Further away from the trailing cylinder, all the patterns corresponding to the leading eigenvectors of vorticity attain similar sign-alternating petal structures, symmetrically aligned along the horizontal centerline.

(a)

(b)

(c)

(d)
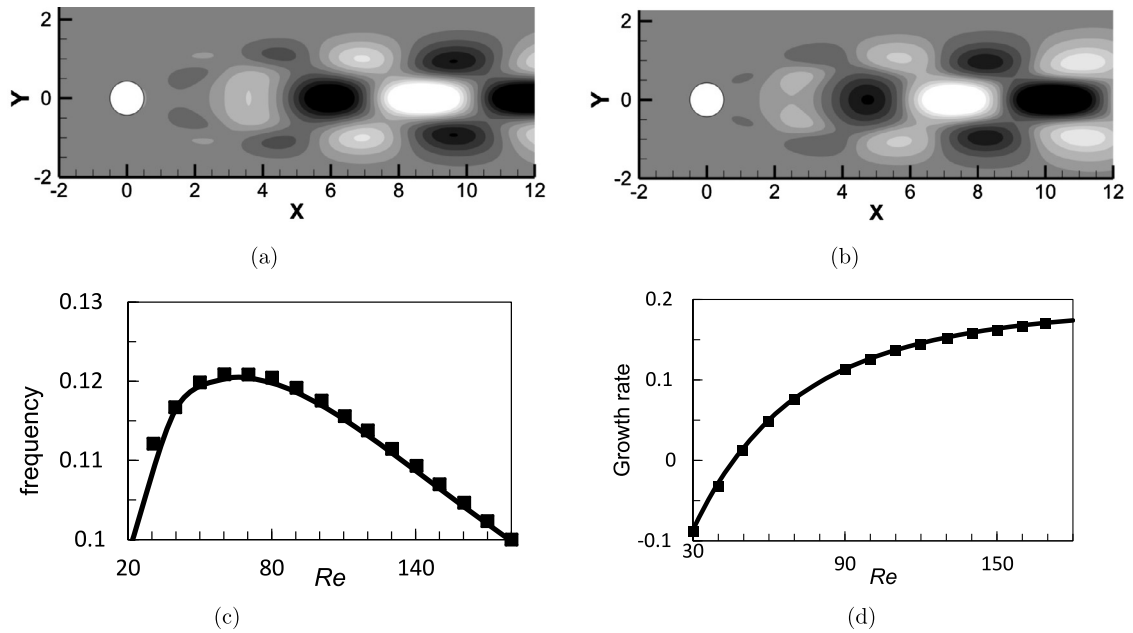
Fig. 9. Results of the linear stability analysis of incident flow over the cylinder at $Re = 100$: (a) real part of the leading eigenmode of vorticity; (b) imaginary part of the leading eigenmode of vorticity; (c) comparison of the critical frequency values (bold line) obtained in this study with the reference values (black squares, scanned from [56]) as a function of the Re number; (d) comparison of the growth rate values (bold line) obtained in this study with the reference values (black squares, scanned from [56]) as a function of the Re number.


(a)

(b)
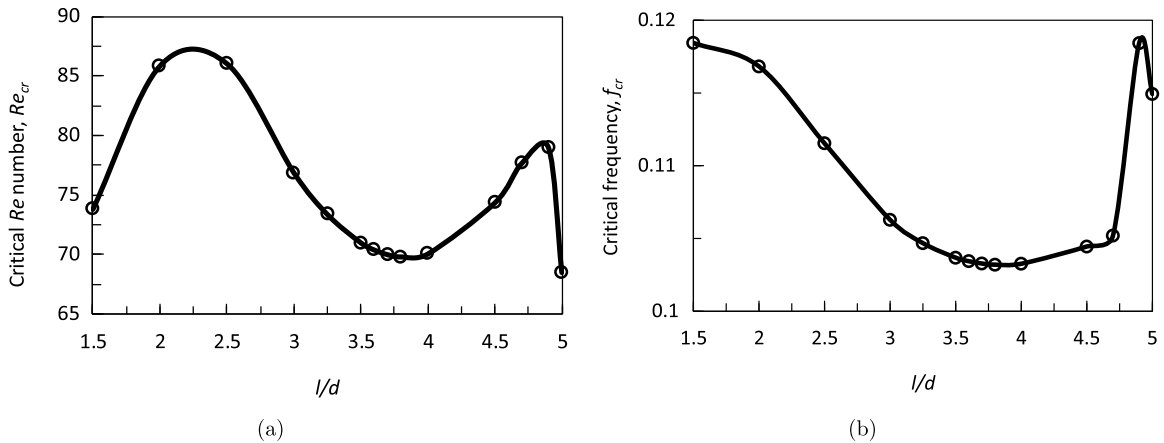
Fig. 10. Values of critical Reynolds number, Re_cr and critical frequency, f_{cr} versus non-dimensional distance, l/d, obtained for the flow around two circular cylinders in tandem.

Table 4
Values of the critical Reynolds number, Re_{cr} versus the non-dimensional distance, l/d obtained for the flow around two circular cylinders in tandem. Comparison of the presently obtained and reference values as scanned from Carmo et al. [49].

l/d	Present	Ref. [49]
2.5	86.55	85.41
3.0	77.19	76.89
3.5	71.35	71.07
3.6	70.76	70.41
3.7	70.72	70.01
3.8	70.68	69.83
4.0	70.06	70.07
4.3	73.10	74.35
4.7	78.71	77.66
5.0	67.80	68.52
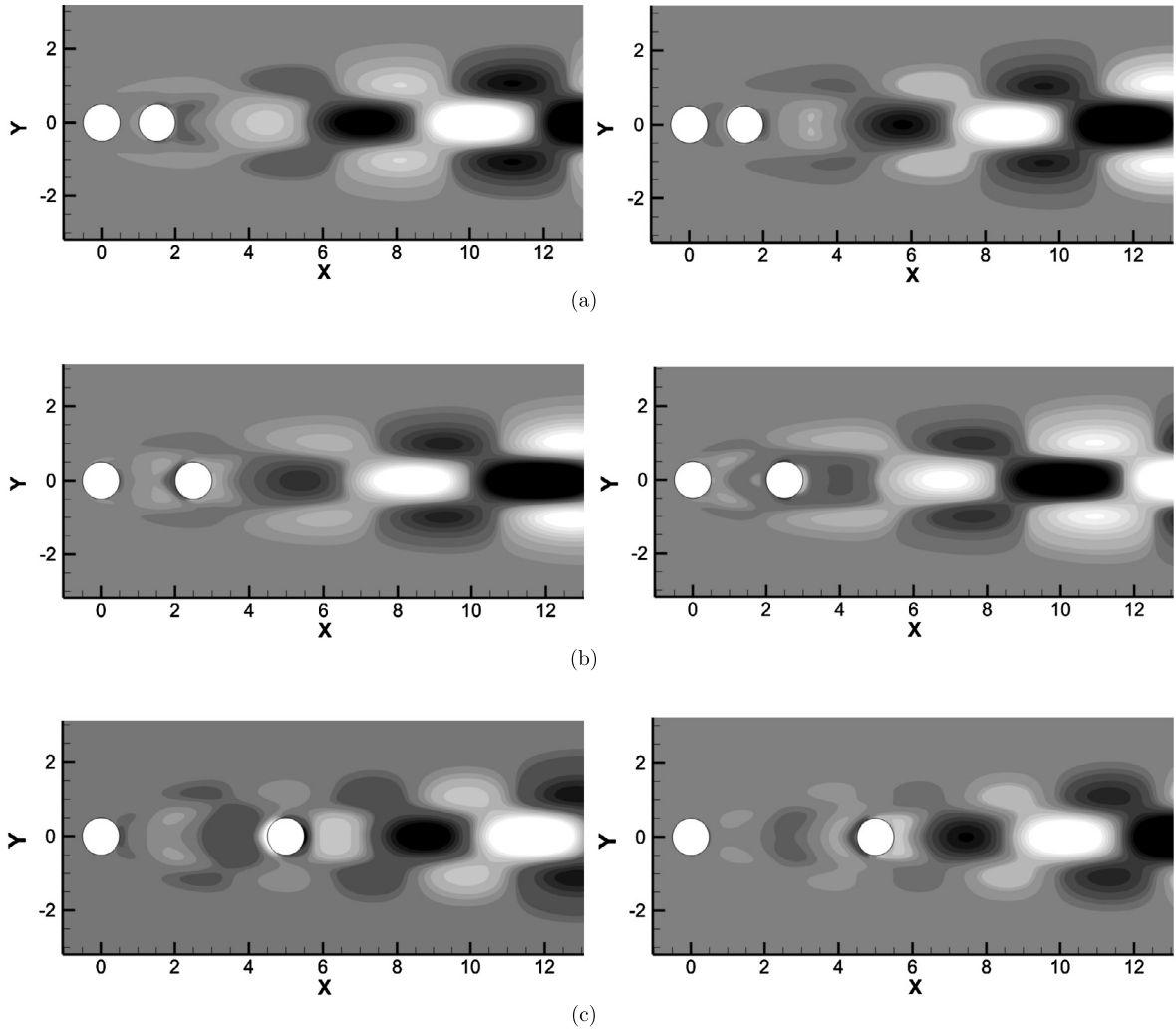
Fig. 11. Results of a linear stability analysis of the incident flow over the two horizontally aligned cylinders at $Re = Re_{cr}$. Left and right columns correspond to the contours of real and imaginary parts of the vorticity of the leading eigenmode, respectively, for: (a) $l/d = 1.5$ and $Re_{cr} = 73.1$; (b) $l/d = 2.5$ and $Re_{cr} = 86.86$; (c) $l/d = 5$ and $Re_{cr} = 68.51$.

Table 5
Grid convergence for the critical Ra_{cr} and ω_{cr} values, $\delta = 0.5$.

Grid	$Ra_{cr} \times 10^5$	ω_{cr}
400×400	5.073	0.2816
600×600	5.035	0.2842
800×800	5.026	0.2855
1000×1000	5.022	0.2866
1200×1200	5.018	0.2871
1400×1400	5.015	0.2873
1600×1600	5.012	0.2874
1800×1800	5.011	0.2875
2000×2000	5.011	0.2875

### 3.7. Linear stability analysis: confined natural convection flow

This section focuses on the linear stability analysis of the confined natural convection flow around two hot vertically aligned circular cylinders located inside a square cavity with all cold boundaries. For this configuration, unsteady periodic flow was observed at $Ra = 10^6$ (see section 3.2), thereby providing the lower value of Ra_{cr} for the first Hopf bifurcation. The contours of the absolute values of the leading eigenmode of the temperature $|\theta'|$ and the x- and y-velocity components, $|u'|$ and $|v'|$, obtained at $Ra_{cr} = 5.011 \times 10^5$ on a 1800×1800 grid are shown in Fig. 12. For all quantities, the region characterized by the highest amplitude of perturbation can be clearly recognized above the top cylinder. All amplitude
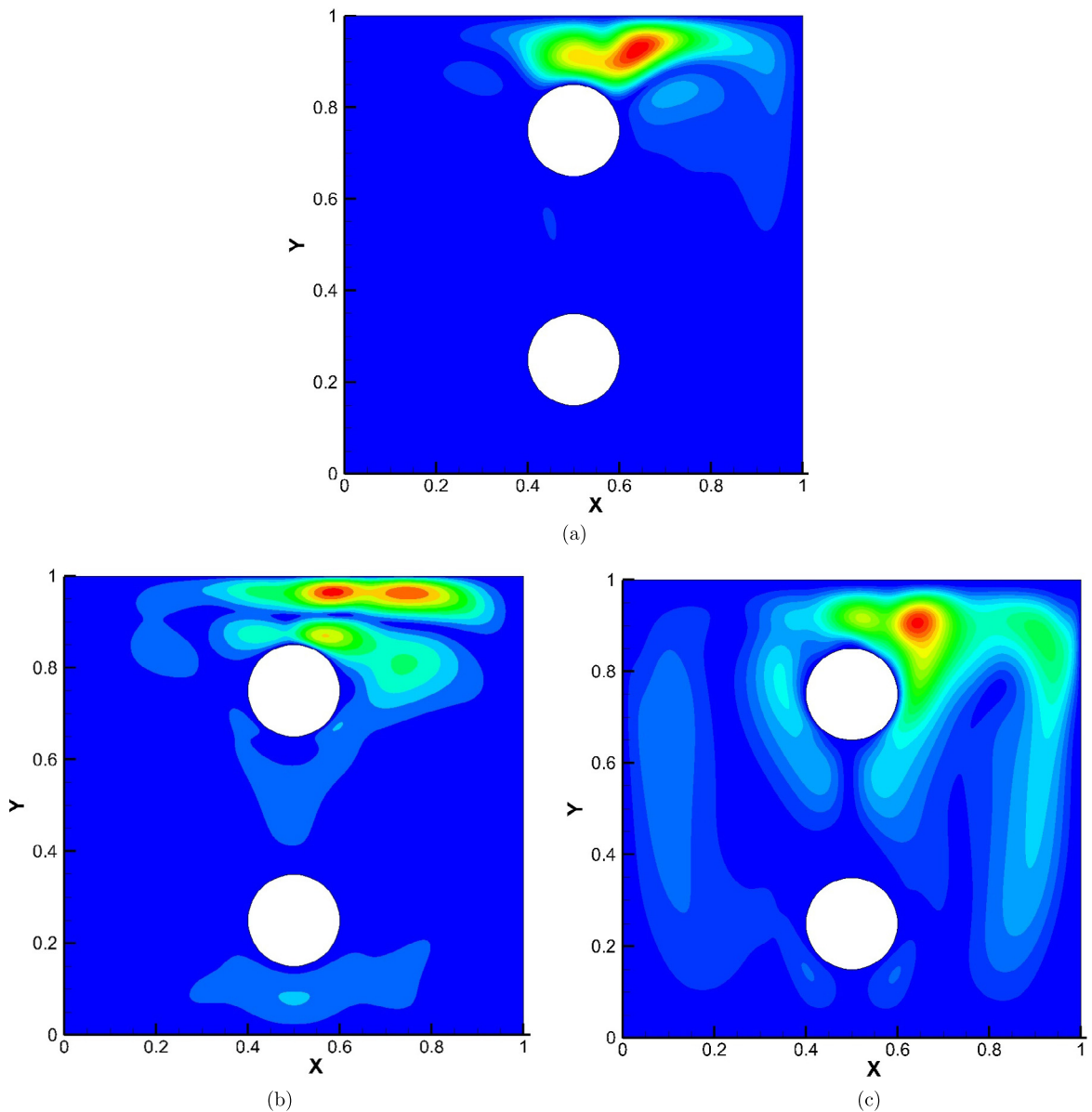
Fig. 12. Contours of absolute values of the leading eigenmode obtained at $Ra_{cr} = 5.011 \times 10^5$ and $\delta = 0.5$ on a 1800×1800 grid for: (a) temperature, $|\theta'|$; (b) x-velocity component, $|u'|$; y-velocity component, $|v'|$).

distributions are not symmetric and are biased to the right (but could also be biased to the left with the same probability) relative to the vertical centerline. The obtained patterns are consistent with the structure of supercritical flow at $Ra = 10^6$ (see Fig. 5), clearly indicating the interaction between two counter rotating vortices as the primary source of the observed instability. A summary of the grid convergence study with respect to the obtained values of the critical Rayleigh number, Ra_{cr}, and the oscillating frequency ω_{cr} is presented in Table 5. It can be seen that for the 2000×2000 grid the results have converged up to the fourth decimal place, thereby verifying the independence of the obtained Ra_{cr} and ω_{cr} values on the resolution of the grid. Fig. 13 presents the requirements for CPU time and computer memory, respectively, for the **LU**-decomposition of the real and complex Jacobian matrices involved in the Newton and in the shift and invert Arnoldi iterations. All the statistics were acquired for the presently discussed confined natural convection flow. The calculations were performed on a standard Unix server equipped with 4 $AMD\ Opteron^{TM}$ processors (64 cores in total) and 256 GB shared RAM. We used the default MUMPS settings which activate openMP parallelization on 32 processors (half of the overall computational capacity) for performing **LU**-decomposition. It can be seen that the CPU time required for the **LU**-factorization of both real and complex Jacobian matrices scales approximately as $N^{2.5}$, where N is the number of grid points in a single spatial direction. The RAM requirement scales as $N^{2.2}$ when performing **LU** factorization of both real and complex Jacobian
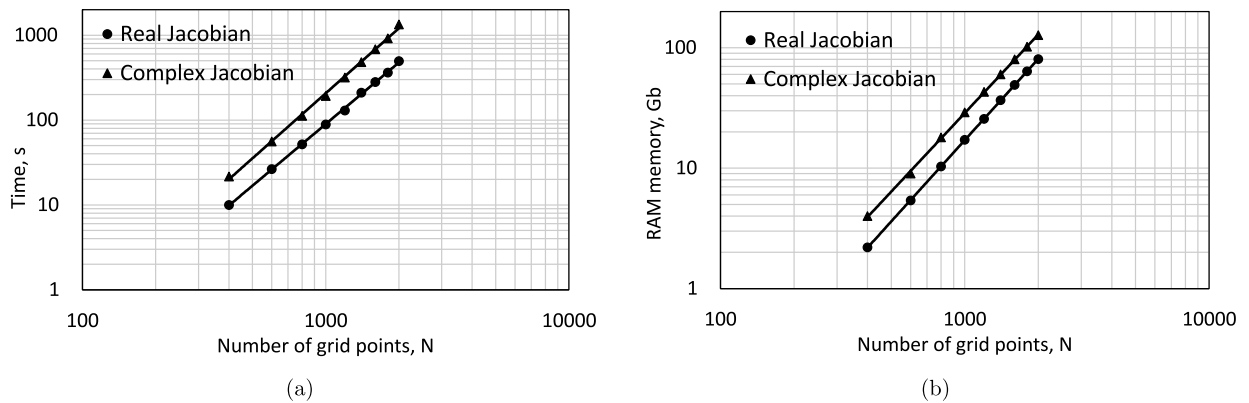
Fig. 13. Scaling characteristics of **LU**-decomposition versus number of grid points in terms of: (a) consumed time in seconds; (b) consumed RAM in gigabytes.

matrices. It is remarkable that similar trends for CPU time and memory consumptions were reported in [36], were the MUMPS solver was used for the linear stability analysis of confined natural convection 2D flows but only on a single processor. It may thus be concluded, that the openMP parallelization applied to the **LU-**decomposition of large indefinite matrices has a small impact on its scaling characteristics. It should also be noted that for all the linear stability calculations the number of Arnoldi iterations varied between 5000 and 7500 and was almost independent of the grid resolution.

4. Summary and conclusions

An approach demonstrating an extension of the IB method based on the distributed Lagrange multiplier approach in the context of time integration of buoyancy flows, calculation steady non-Stokes flows and linear stability analysis of confined and open flows around immersed bodies was presented. The new capabilities of the method were demonstrated by utilizing the fully pressure–velocity coupled direct solver (FPCD) [27] as a computational platform. The developed method facilitated an efficient linear stability analysis of the incompressible flows in the presence of a variety of arbitrarily oriented immersed bodies of various shapes. The method was extensively verified for both isothermal and natural convection flows. The independence of the obtained results on the resolution of the computational grid was established by an acceptable quantitative agreement with independent numerical and experimental results available in the literature and from the current convergence study.

The specific choice of a direct solver (FPCD) [27] as a computational platform implies **LU**-decomposition of a large indefinite system (known also as a Stokes operator). For this reason, the present study was restricted to $2D$ incompressible flows. It was shown that the method is viable for a grid resolution up to 2000×2000, which makes it attractive for the most kinds of 2D and 3D flows with periodic boundary conditions. The new methodology can also be efficiently applied to the mesoscale linear stability analysis of quasi-$3D$ flows in porous media and to the analysis of two-phase flows in which the curvature of the interface between the two phases plays the role of Lagrange multipliers (both applications are covered in our current research). An extended discussion regarding strategies for adapting the developed methodology to realistic 3D flows was presented. In light of increasing interest in the distributed Lagrange multiplier approach, we believe that the present study will contribute to the further extension of the presented methodology to various applications in fluid dynamics.

References

[1] C.S. Peskin, Flow patterns around heart valves: a numerical method, J. Comput. Phys. 10 (1972) 252–271.
[2] C.S. Peskin, The immersed boundary method, Acta Numer. 11 (2002) 479–517.
[3] R. Mittal, G. Iaccarino, The immersed boundary method, Annu. Rev. Fluid Mech. 37 (2005) 239–261.
[4] J. Mohd-Yusof, Combined immersed-boundary/b-spline methods for simulations of flow in complex geometries, in: Annual Research Briefs, Center for Turbulence Research, 1997, pp. 317–327.
[5] A.E. 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.
[6] L. Lee, R. LeVeque, An immersed interface method for incompressible Navier–Stokes equations, SIAM J. Sci. Comput. 25 (2003) 832–856.
[7] N.N. Linnick, H.H. Fasel, A high-order immersed interface method for simulating unsteady incompressible flows on irregular domains, J. Comput. Phys. 204 (2005) 157–192.
[8] 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.
[9] W.W. Ren, C. Shu, W.M. Yang, Boundary condition-enforced immersed boundary method for thermal flow problems with Dirichlet temperature condition and its applications, Comput. Fluids 57 (2012) 40–51.
[10] W. Ren, C. Shu, W. Yang, An efficient immersed boundary method for thermal flow problems with heat flux boundary conditions, Int. J. Heat Mass Transf. 64 (2013) 694–705.

[11] Y. Gulberg, Y. Feldman, On laminar natural convection inside multi-layered spherical shells, Int. J. Heat Mass Transf. 91 (2015) 908–921.
[12] T. Kempe, J. Fröhlich, An improved immersed boundary method with direct forcing for the simulation of particle laden flows, J. Comput. Phys. 231 (2012) 3663–3684.
[13] T. Kempe, M. Lennartz, S. Schwarz, J. Fröhlich, Imposing the free-slip condition with a continuous forcing immersed boundary method, J. Comput. Phys. 282 (2015) 183–209.
[14] R. Glowinski, T. Pan, J. Périaux, Distributed Lagrange multiplier methods for incompressible viscous flow around moving rigid bodies, Comput. Methods Appl. Mech. Eng. 151 (1998) 181–194.
[15] R. Glowinski, T. Pan, T. Hesla, D. Joseph, A. Periaux, A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: application to particulate flow, J. Comput. Phys. 169 (2001) 363–426.
[16] Z. Yu, N. Phan-Thien, Y. Fan, R. Tanner, Viscoelastic mobility problem of a system of particles, J. Non-Newton. Fluid Mech. 104 (2002) 87–124.
[17] Z. Yu, N. Phan-Thien, R. Tanner, Dynamical simulation of sphere motion in a vertical tube, J. Fluid Mech. 518 (2004) 61–93.
[18] Z. Yu, A DLM/FD method for fluid/flexible-body interactions, J. Comput. Phys. 207 (2005) 1–27.
[19] K. Taira, T. Colonius, The immersed boundary method: a projection approach, J. Comput. Phys. 225 (2007) 3121–3133.
[20] K. Taira, T. Colonius, Effect of tip vortices in low-Reynolds-number poststall flow control, AIAA J. 47 (2009) 749–756.
[21] A. Samanta, D. Appelo, T. Colonius, J. Nott, J. Hall, Computational modeling and experiments of natural convection for a titan Montgolfiere, AIAA J. 48 (2010) 1007–1016.
[22] S.G. Yiantsios, On the distributed Lagrange multiplier/fictitious domain method for rigid-particle-laden flows: a proposition for an alternative formulation of the Lagrange multipliers, Int. J. Numer. Methods Fluids 70 (2012) 1027–1047.
[23] J. Choi, T. Colonius, D.R. Williams, Surging and plunging oscillations of an airfoil at low Reynolds number, J. Fluid Mech. 763 (2015) 237–253.
[24] C. Wang, J.D. Eldredge, Strongly coupled dynamics of fluids and rigid-body systems with the immersed boundary projection method, J. Comput. Phys. 295 (2015) 87–113.
[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] D. Stein, R. Guy, B. Thomases, Immersed boundary smooth extension: a high-order method for solving PDE on arbitrary smooth domains using Fourier spectral methods, J. Comput. Phys. 304 (2016) 252–274.
[27] Y. Feldman, A.Y. Gelfgat, On pressure–velocity coupled time-integration of incompressible Navier–Stokes equations using direct inversion of stokes operator or accelerated multigrid technique, Comput. Struct. 87 (2009) 710–720.
[28] F. Giannetti, P. Luchini, Structural sensitivity of the first instability of the cylinder wake, J. Fluid Mech. 581 (2007) 167–197.
[29] A.Y. Gelfgat, Three-dimensional instability of axisymmetric flows: solution of benchmark problems by a low-order finite volume method, Int. J. Numer. Methods Fluids 54 (2007) 269–294.
[30] S.V. Patankar, Numerical Heat Transfer and Fluid Flow, McGraw–Hill, 1980.
[31] A. Roma, C.S. Peskin, M.J. Berger, An adaptive version of the immersed boundary method, J. Comput. Phys. 153 (1999) 509–534.
[32] M. Uhlmann, An immersed boundary method with direct forcing for the simulation of particulate flows, J. Comput. Phys. 209 (2005) 448–476.
[33] R.P. Beyer, R.J. LeVeque, Analysis of a one-dimensional model for the immersed boundary method, SIAM J. Numer. Anal. 29 (1992) 332–364.
[34] P.M. Gresho, R.L. Sani, On pressure boundary conditions for the incompressible Navier–Stokes equations, Int. J. Numer. Methods Fluids 7 (1987) 1111–1145.
[35] S.L. Xiaoye, J.W. Demmel, SuperLU DIST: a scalable distributed-memory sparse direct solver for unsymmetric linear systems, ACM Trans. Math. Softw. 29 (2003) 110–140.
[36] A.Y. Gelfgat, Stability of convective flows in cavities: solution of benchmark problems by a low-order finite volume method, Int. J. Numer. Methods Fluids 53 (2007) 485–506.
[37] Z. Perugia, A field-based mixed formulation for the two-dimensional magnetostatic problem, SIAM J. Numer. Anal. 34 (1997) 2382–2391.
[38] J.N. Sørensen, A.Y. Gelfgat, I.V. Naumov, R.F. Mikkelsen, Experimental and numerical results on three-dimensional instabilities in a rotating disk–tall cylinder flow, Phys. Fluids 21 (2009) 054102.
[39] A.Y. Gelfgat, S. Molokov, Quasi-two-dimensional convection in a 3D laterally heated box in a strong magnetic field normal to main circulation, Phys. Fluids 23 (2011) 034101.
[40] A. Toselli, O.B. Widlund, Domain Decomposition Methods: Algorithms and Theory, Springer Ser. Comput. Math., vol. 34, Springer, 2004.
[41] M. Benzi, G.H. Golub, J. Liesen, Numerical solution of saddle point problems, Acta Numer. 14 (2005) 1–137.
[42] J.B. Perot, An analysis of the fractional step method, J. Comput. Phys. 108 (1993) 51–58.
[43] L. Tuckerman, D. Barkley, Bifurcation Analysis for Timesteppers, Springer, New York, 1999.
[44] D. Barkley, L.S. Tuckerman, Stability analysis of perturbed plane Couette flow, Phys. Fluids 11 (1999) 1187–1195.
[45] D. Barkley, M.G.M. Gomez, R. Henderson, Three-dimensional stability analysis of flow over a backward facing step, J. Fluid Mech. 473 (2002) 167–190.
[46] P.S. Marcus, L.S. Tuckerman, Numerical simulation of spherical Couette flow. Part I. Numerical methods and steady states, J. Fluid Mech. 185 (1987) 1–30.
[47] P.S. Marcus, L.S. Tuckerman, Numerical simulation of spherical Couette flow. Part II. Transitions, J. Fluid Mech. 185 (1987) 31–65.
[48] B.C. Carmo, J.R. Meneghini, S.J. Sherwin, Secondary instabilities in the flow around two circular cylinders in tandem, J. Fluid Mech. 644 (2010) 395–431.
[49] B.S. Carmo, J.R. Meneghini, S.J. Sherwin, Possible states in the flow around two circular cylinders in tandem with separations in the vicinity of the drag inversion spacing, Phys. Fluids 22 (2010) 054101.
[50] Y. Park, M. Ha, C. Choi, J. Park, Natural convection in a square enclosure with two inner circular cylinders positioned at different vertical locations, Int. J. Heat Mass Transf. 77 (2014) 501–518.
[51] N. Coutanceau, R. Bouard, Experimental determination of the main features of the viscous flow in the wake of a circular cylinder in uniform translation. Part 1. Steady flow, J. Fluid Mech. 79 (1977) 231–256.
[52] S.C. Dennis, G. Chang, Numerical solutions for steady flow past a circular cylinder at Reynolds number up to 100, J. Fluid Mech. 42 (1970) 471–489.
[53] T. Seta, Implicit temperature-correction-based immersed-boundary thermal lattice Boltzmann method for the simulation of natural convection, Phys. Rev. E (2013) 063304.
[54] F. Moukalled, S. Acharya, Natural convection in the annulus between concentric horizontal circular and square cylinders, J. Thermophys. Heat Transf. 10 (1996) 524–531.
[55] C. Shu, Y.D. Zhu, Efficient computation of natural convection in a concentric annulus between an outer square cylinder and an inner circular cylinder, Int. J. Numer. Methods Fluids 38 (2002) 429–445.
[56] D. Barkley, Linear analysis of the cylinder wake mean flow, Europhys. Lett. 75 (2006) 750–756.