



Semi-implicit direct forcing immersed boundary method for incompressible viscous thermal flow problems: A Schur complement approach

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 28 March 2018
Received in revised form 13 June 2018
Accepted 19 June 2018
Available online 11 July 2018

Keywords:

Immersed boundary method
Segregated pressure-velocity coupling
Schur complement
Distributed Lagrange multiplier

ABSTRACT

An extended immersed boundary method utilizing a semi-implicit direct forcing approach for the simulation of confined incompressible viscous thermal flow problems is presented. The method utilizes a Schur complement approach to enforce the kinematic constraints of no-slip and the corresponding thermal boundary conditions for immersed surfaces. The developed methodology can be straightforwardly adapted to any existing incompressible time marching solver based on a segregated pressure-velocity coupling. The method accurately meets the thermal and the no-slip boundary conditions on the surfaces of immersed bodies for the entire range of Rayleigh numbers $10^3 \leq Ra \leq 10^6$. Strategies for further increasing the computational efficiency of the developed approach are discussed. The method has been extensively verified by applying it for the simulation of a number of representative fully 3D confined natural convection steady and periodic flows. Complex dynamic phenomena typical of this kind of flow including vortical structures and convection cells and instability characteristics, were simulated and visualized and the results were found to compare favorably with results known from literature.

© 2018 Elsevier Ltd. All rights reserved.

1. Introduction

Accurate simulation of viscous flows in the presence of bodies of complex geometry is critical, both in pure science and in engineering. The traditional, and probably still the most popular, methodology for resolving the characteristics of viscous flows in the vicinity of immersed bodies utilizes a body-conformal grid precisely matching the discretized body surface with the adjacent flow domain (up to the discretization error). However, the ever-increasing complexity of the flow configurations studied seriously challenges the efficiency and precision of numerical methods based on the body-conformal approach, as the grid orthogonality and skewness may have a significant effect on the accuracy and the stability of the methods. The immersed boundary (IB) method, in which the body surface is determined by the location of a set of discrete Lagrangian points that do not necessarily coincide with the underlying Eulerian grid, comprises an attractive alternative to the body-conformal grid approach. The method, initially developed by Peskin [1] for the simulation of the blood flow in the mitral valve in the heart, has become very popular over the last three decades and comprises the basis for this rapidly developing field of computational science.

Accurately imposing a body-force field in the vicinity of an immersed surface to enforce the no-slip kinematic constraints is critical for any IB method. Among a number of existing methods, the present study focuses on the direct forcing approach, which enforces the desired value of velocity directly on the boundary without involving any dynamical process. The direct forcing approach, initially formulated by Mohd-Yusof [2], has gained popularity over the years due to its simple implementation and robustness. In the direct forcing approach, the Lagrangian body-force field can be calculated either explicitly or implicitly. The fully explicit formulation is typically implemented on the basis of the segregated pressure-velocity coupling utilized for the solution of incompressible Navier Stokes (NS) equations. In this case, the kinematic no-slip constraints are applied to the intermediate non-solenoidal velocity, which is further corrected to meet the divergence free constraint (see e.g. [2–4]). The explicit approach has been adopted in various engineering applications, as detailed in the comprehensive review of Mittal and Iaccarino [5]. Recently the approach has also been extended to thermal flow problems [6–9]. Explicit calculation of the Lagrangian forces can be straightforwardly implemented without requiring any modification of the existing time marching solvers, which explains its high popularity. Despite its evident attractiveness, the explicit treatment has two major drawbacks. First, the existence of local mass leakage through

E-mail address: yurifeld@bgu.ac.il

boundaries of the immersed body and second, the pointwise character of the calculation of Lagrangian forces and heat fluxes, which does not reflect their mutual interaction in a given time step. While the first limitation can be successfully remedied by simply decreasing the value of time step, the second one can seriously deteriorate the precision of the performed analysis, as it violates the intrinsic elliptic character of NS equations. In particular, the issue is of critical importance in confined natural convection unsteady flows characterized by low and moderate Rayleigh numbers, which are the focus of the present study.

The limitations of the explicit direct forcing IB method motivated further research aimed at improving the existing explicit formulations. Su et al. [10] proposed a new forcing procedure based on a solution of a banded linear system, coupling together all the Lagrangian markers. A similar approach was then implemented by Ren et al. [7,8], who implicitly evaluated all the Lagrangian forces and heat sources by assembling them into a single system of equations. The studies of Wang et al. [11] and Kempe et al. [12,13], who introduced multi-direct forcing (MDF) schemes based on iterative enhancement of Euler-Lagrange coupling, are also worth mentioning. Although the developed approaches succeeded in improving the accuracy of the imposition of no-slip and thermal constraints, non-negligible inaccuracies attributed to the explicit treatment of the diffusion term in the NS and energy equations still remain. Moreover, correction of velocity and temperature fields by the MDF schemes locally deteriorates the momentum and the thermal balances close to the immersed boundaries and can result in non-negligible inaccuracies in the estimation of local Nusselt numbers and drag and lift coefficients.

The fully implicit implementation of the Lagrangian forces and heat fluxes offers an alternative to the fully explicit direct forcing approach. The Lagrangian forces and heat fluxes implicitly embedded into the corresponding energy and momentum equations play the role of distributed Lagrange multipliers, which reflect the impact of the immersed body on the surrounding flow. Together with additional equations connecting between the corresponding Eulerian and Lagrangian temperature and velocity fields, they form the whole closed system of the coupled equations. Solution of the system provides high fidelity results for a broad range of Reynolds and Rayleigh numbers. Pioneering work in this field is due to Golowinski et al. [14], who applied the distributed Lagrange multiplier method (DLM) for the simulation of 2D flow around a moving disc. The method was then extended to the simulation of particulate flows [15–17] and to the simulation of fluid/flexible-body interactions [18]. Another approach was proposed by Taira and Colonius [19], who utilized the distributed Lagrange multiplier to simultaneously satisfy the divergence-free and no-slip kinematic constraints by the solution of the modified Poisson equation in the framework of the projection method. The developed approach was then applied in various engineering fields, including active and passive flow control [19,20], optimization of performance of a hot air balloon [21] and the dynamic interactions between rigid-body systems and incompressible viscous flows [22].

The latest theoretical developments of the fully coupled Lagrange multiplier approach are: due to the study of Kallemov et al. [23], who presented a novel IB formulation for modeling flows around fixed or moving rigid bodies that is suitable for a broad range of Reynolds numbers, including steady Stokes flow; the formulation developed by Stein et al. [24], who established the immersed boundary smooth extension (IBSE) method and demonstrated the superiority of its convergence for a wide spectrum of equations with both Dirichlet and Neumann boundary conditions; the work of Feldman and Gulberg [25], comprising an extended fully pressure-velocity-Lagrange multipliers coupled formulation of the IB method, capable of performing accurate linear stability analysis of incompressible viscous and thermal 2D and

axi-symmetric flows; and the recent study of Liska and Colonius [26] who proposed novel time integration schemes for the efficient solution of the discrete momentum equations coupled with the discrete divergence free and no-slip constraints. It is also worth mentioning the recent works of Bao et al. [27], who succeeded in substantially improving the volume conservation for fluid-structure interaction applications in periodic domains, as well as Stein et al. [28], who further extended the IBSE method to allow for the imposition of a divergence free constraint.

Although the fully implicit treatment of the Lagrangian forces and heat fluxes succeeds in accurately satisfying the kinematic constraints on the surface of an immersed body, it often includes non-trivial intermediate stages requiring substantial modification of time stepping codes which were originally developed without IB capability. In our present work we pursue a different concept, where the IB approach is not seen as a stand alone solver, but, rather, it comprises a methodology of enforcing boundary conditions. One of the main purposes of the present study is to demonstrate the ways of plugging the IB approach in any of many available open source CFD packages or in-house developed solvers in a modular fashion. To achieve this we apply a semi-implicit approach, which is applicable for the whole family of pressure-velocity segregated solvers based, for example, on SIMPLE or fractional step algorithms. The idea is to implicitly couple the Lagrangian forces with an intermediate non-solenoidal velocity field, further updated by the following correction step. The concept has already been successfully implemented for the investigation of isothermal viscous flows by Park et al. [29], who combined a block *LU* decomposition technique with Taylor series expansion to construct the implicit IB forcing in a recurrence form, and by Le et al. [30], who used the FISHPACK [31] library for the solution of modified Helmholtz and Poisson equations. The present study demonstrates how the developed methodology can be embedded into the generic incompressible NS solver based on the SIMPLE algorithm [32] in order to endow it with the IB capability. The solver is based on the algorithm employing the tensor product factorization (TPF) method [33] combined with the Thomas solver (TPT). In this case the kinematic constraints for temperature and velocity are satisfied by the machine zero precision. Although in the present formulation the no-slip constraint is precisely satisfied only for the intermediate non-solenoidal velocity field, it is shown that an inaccuracy accumulated in the corrected velocity field in the vicinity of an immersed surface is less than a discretization error. A number of representative fully 3D confined natural convection flows are simulated and the obtained results are favorably compared with the data available in the literature.

2. The theoretical background

2.1. Predictor-corrector approach with incorporated IB functionality

We start with a brief description of the standard predictor-corrector approach based on the SIMPLE algorithm of Patankar and Spalding [34] with incorporated IB capability. The flow is governed by the non-dimensional incompressible NS and energy equations, in which the buoyancy effects are introduced by applying the Boussinesq approximation:

$$\nabla \cdot \mathbf{u} = 0, \quad (1)$$

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\nabla p + \sqrt{\frac{Pr}{Ra}} \nabla^2 \mathbf{u} + \theta \vec{e}_z + \mathbf{f}, \quad (2)$$

$$\frac{\partial \theta}{\partial t} + (\mathbf{u} \cdot \nabla) \theta = \frac{1}{\sqrt{PrRa}} \nabla^2 \theta + q, \quad (3)$$

where $\mathbf{u} = (u, v, w)$, p , t , and θ are the non-dimensional velocity, pressure, time and temperature, respectively, and $\bar{\mathbf{e}}_z$ is a unit vector in the vertical (y) direction. In accordance with the Boussinesq approximation the density field is given by $\rho = \rho_0(1 - \beta(T - T_c))$, where ρ_0 is the density value corresponding to the thermal equilibrium. The equations are normalized by utilizing L , $U = \sqrt{g\beta\Delta T}$, $t = L/U$, and $P = \rho U^2$ for length, velocity, time, and pressure scales, respectively. The Rayleigh, Ra , and Prandtl, Pr , numbers are $Ra = \frac{g\beta}{\nu\alpha} \Delta T L^3$ and $Pr = \nu/\alpha$, respectively, where ν is the kinematic viscosity, α is the thermal diffusivity and ΔT is the temperature difference between the maximal and the minimal temperatures of the problem.

The approach is applied to the simulation of unsteady natural convection incompressible flow in the presence of an immersed body of irregular shape. The surface of the immersed body is determined by a set of discrete Lagrangian points, whose location does not necessarily coincide with the underlying Eulerian staggered grid, as shown in Fig. 1.

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, ΔX , should be approximately the same. Away from the body, non-uniform discretization can be used. The impact of the immersed body on the nearby flow is reflected by the force, \mathbf{f} , and the power, q , densities, each related to the corresponding virtual volume located within the shadowed region confined by internal and external concentric circles and confining every Lagrangian point (see Fig. 1). The densities enter as sources into the momentum, Eq. (2), and energy, Eq. (3), equations. The values of the sources are not known a priori and are part of the solution along with the velocity, pressure and temperature fields. To enforce the non-slip and thermal (temperature or heat flux) boundary conditions on the surface of the immersed body, the nearby velocity and temperature fields in the vicinity of every Lagrangian point (dotted circle) are interpolated to the location of the Lagrangian points.

Next, two adjoint operators are defined to exchange information between Lagrangian points and the Eulerian grid: namely, an interpolation operator \mathbf{I} interpolating the values of Eulerian velocities $\mathbf{u}(\mathbf{x}_i)$ and temperatures $\theta(\mathbf{x}_i)$ to the nearby Lagrangian points \mathbf{X}^k ; and a regularization operator \mathbf{R} smearing the values of the Lagrangian force $\mathbf{F}^k(\mathbf{X}^k)$ and the power $Q^k(\mathbf{X}^k)$ densities to the nearby Eulerian grid:

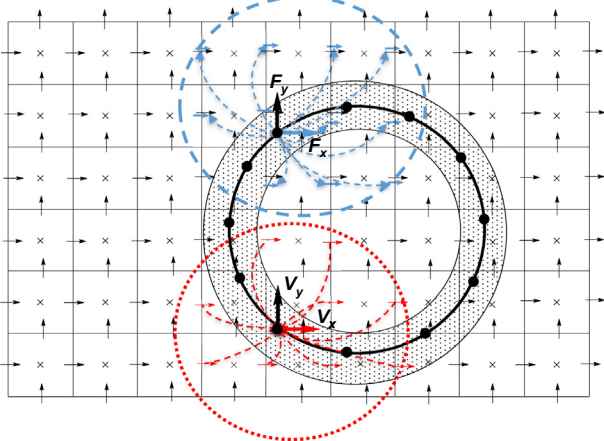


Fig. 1. A schematic description of the basic principles of the IB method on the staggered Eulerian grid. The circles drawn by dotted and dashed lines denote the vicinity of Lagrangian point in which the interpolation and regularization operators act. The filled annular pattern corresponds to a collection of virtual volumes surrounding Lagrangian points of the immersed body.

$$\mathbf{R}(\mathbf{F}^k(\mathbf{X}^k), Q^k(\mathbf{X}^k)) = \int_S (\mathbf{F}^k(\mathbf{X}^k), Q^k(\mathbf{X}^k)) \cdot \delta(\mathbf{x}_i - \mathbf{X}^k) dV_S^k, \quad (4a)$$

$$\mathbf{I}(\mathbf{u}(\mathbf{x}_i), \theta(\mathbf{x}_i)) = \int_{\Omega} (\mathbf{u}(\mathbf{x}_i), \theta(\mathbf{x}_i)) \cdot \delta(\mathbf{X}^k - \mathbf{x}_i) dV_{\Omega_i}, \quad (4b)$$

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_{Ω_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. Note that both interpolation and regularization operators use convolutions with the same discrete Dirac delta function δ of the form

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

introduced by Roma et al. in [35]. Here Δr is the cell width in the r direction. The chosen delta function, successfully utilized in a large number of previous studies (see e.g. [12,19,25,36]), has been specifically designed for performing calculations on the staggered grids and became popular due to its compact kernel (only three cells in each direction of the computational domain are involved). Following the SIMPLE method [34], the NS and energy equations, Eqs. (1)–(3), are discretized as:

$$\begin{aligned} \frac{3\theta^{n+1}}{2\Delta t} - \frac{1}{\sqrt{PrRa}} \mathbf{L}(\theta^{n+1}) - \mathbf{R}(Q^k(\mathbf{X}^k)) \\ = \frac{4\theta^n - \theta^{n-1}}{2\Delta t} - \mathbf{N}(\theta^n, \mathbf{u}^n) + \theta^n \bar{\mathbf{e}}_z, \end{aligned} \quad (6)$$

$$\frac{3\mathbf{u}^*}{2\Delta t} - \sqrt{\frac{Pr}{Ra}} \mathbf{L}(\mathbf{u}^*) - \mathbf{R}(\mathbf{F}^k(\mathbf{X}^k)) = \frac{4\mathbf{u}^n - \mathbf{u}^{n-1}}{2\Delta t} - \mathbf{N}(\mathbf{u}^n) + \theta^{n+1} \bar{\mathbf{e}}_z - \nabla p^n, \quad (7)$$

$$\Delta(\delta p) = \frac{3}{2\Delta t} \nabla \cdot \mathbf{u}^*, \quad (8)$$

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \frac{2\Delta t}{3} \nabla(\delta p), \quad p^{n+1} = p^n + \delta p, \quad (9)$$

where the second order backward finite difference scheme was utilized for discretizing the temporal derivatives, while a standard staggered mesh second-order conservative finite-volume formulation [37] was used for discretizing all the spatial derivatives. Operators \mathbf{N} and \mathbf{L} read for non-linear convective and linear Laplacian terms of the NS equations, respectively. For the confined natural convection flows which are the focus of the present study Eqs. (6)–(9) are solved with the Dirichlet (or Neumann) boundary conditions for the temperature field and non-slip boundary conditions for all velocity components. Neumann boundary conditions (zero value of derivative in the normal to the boundary direction) with a single reference Dirichlet point introduced anywhere inside the computational domain are used for the solution of the Poisson equation, Eq. (8), formulated for the pressure correction field. Following the formalism of the SIMPLE method the pressure field in the predictor, Eq. (7), is taken from the previous time step and the solution is first obtained for the non-solenoidal velocity field \mathbf{u}^* . Then, the pressure correction field is calculated by the solution of Eq. (8). Finally, Eq. (9) are used to correct the pressure and the velocity values at the next time step. Until now, nothing was said

about the terms $\mathbf{R}(Q^k(\mathbf{X}^k))$ and $\mathbf{R}(\mathbf{F}^k(\mathbf{X}^k))$ implicitly entering into the left-hand sides of the energy, Eq. (6), and the momentum, Eq. (7), equations. These terms are related to the regularized power, q , and force, \mathbf{f} , densities imposing temperature (or heat flux) and kinematic no-slip constraints, respectively, on the surface of an immersed body. Implicit treatment of the above terms implies introducing additional relationships to achieve closure of the overall system of Eqs. (6)–(9), as detailed in the next section.

2.2. Formalism of the semi-implicit direct forcing IB method

Closure of the system of Eqs. (6)–(9) is achieved by introducing additional relationships comprising thermal and kinematic no-slip constraints for the energy and momentum equations, respectively. The full system of equations extended with IB capability and allowing for calculation of temperature, θ^{n+1} , and intermediate velocity, \mathbf{u}^* , followed by the correction of velocity, \mathbf{u} , and the pressure, p , fields can be compactly written as:

$$\begin{bmatrix} \mathbf{H}_{\theta, \mathbf{u}} & \mathbf{R}(Q^k(\mathbf{X}^k), \mathbf{F}^k(\mathbf{X}^k)) \\ \mathbf{I}(\theta(\mathbf{x}_i), \mathbf{u}^*(\mathbf{x}_i)) & 0 \end{bmatrix} \begin{bmatrix} \theta^{n+1}, \mathbf{u}^* \\ Q, \mathbf{F} \end{bmatrix} = \begin{bmatrix} \mathbf{RHS}_{\theta, \mathbf{u}}^{n-1, n} \\ \theta_b, \mathbf{U}_b \end{bmatrix}, \quad (10a)$$

$$\Delta(\delta p) = \frac{3}{2\Delta t} \nabla \cdot \mathbf{u}^*, \quad (10b)$$

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \frac{2\Delta t}{3} \nabla(\delta p), \quad p^{n+1} = p^n + \delta p, \quad (10c)$$

where the first row of Eq. (10) is presented in a block-matrix form of either the energy (see Eq. (6)) or the momentum (see Eq. (7) equations), while $\mathbf{H}_\theta = \left(\frac{3}{2\Delta t}(\theta^{n+1}) - \frac{1}{\sqrt{\text{Pr}Ra}}\mathbf{L}(\theta^{n+1})\right)$ and $\mathbf{H}_\mathbf{u} = \left(\frac{3}{2\Delta t}(\mathbf{u}^*) - \sqrt{\frac{\text{Pr}}{Ra}}\mathbf{L}(\mathbf{u}^*)\right)$ reads for the modified Helmholtz operator acting on the temperature and the velocity fields, respectively, $\mathbf{RHS}_{\theta, \mathbf{u}}^{n-1, n}$ corresponds to the right hand sides of Eqs. (6) and (7), $\mathbf{R}(Q^k(\mathbf{X}^k), \mathbf{F}^k(\mathbf{X}^k))$ corresponds to the entries obtained by regularization of the power and force densities determined at Lagrangian points of the immersed body to the nearby locations of Eulerian grid, and $\mathbf{I}(\theta(\mathbf{x}_i), \mathbf{u}(\mathbf{x}_i))$ corresponds to the entries obtained by interpolation of Eulerian temperature and velocity fields to the Lagrangian points of the immersed body. Note that if discretized on a uniform staggered grid, $\mathbf{I}(\theta(\mathbf{x}_i), \mathbf{u}(\mathbf{x}_i)) = \mathbf{R}^T(Q^k(\mathbf{X}^k), \mathbf{F}^k(\mathbf{X}^k))$, although this property was not explicitly exploited while implementing the currently developed approach.

3. The implementation details

3.1. The Schur complement approach

Recalling that the current study aims to develop a general methodology to acquire the previously developed pressure-velocity segregated solvers of incompressible NS equations with the IB capability, we next present further details regarding implementation of a solution of the system of Eq. (10a). Utilizing the Schur complement approach, the system of Eq. (10a) is analytically split into two smaller equivalent systems:

$$[Q, \mathbf{F}] = [\mathbf{I}\mathbf{H}^{-1}\mathbf{R}]^{-1}[\mathbf{I}\mathbf{H}^{-1}\mathbf{RHS}_{\theta, \mathbf{u}}^{n-1, n} - \theta_b, \mathbf{U}_b], \quad (11a)$$

$$[\theta, \mathbf{u}^*] = \mathbf{H}^{-1}[\mathbf{RHS}_{\theta, \mathbf{u}}^{n-1, n} - \mathbf{R}[Q, \mathbf{F}]], \quad (11b)$$

solution of which is first performed for the distributed Lagrange multiplier terms Q and \mathbf{F} (Eq. (11a)), and then for the corresponding Eulerian temperature, θ , and intermediate velocity, \mathbf{u}^* , fields

(Eq. (11b)). A closer look at the structural characteristics of the matrices \mathbf{H} , \mathbf{I} , and \mathbf{R} gives rise to considerations regarding an efficient solution of Eqs. (11a) and (11b). Matrix \mathbf{H} , comprising the modified Helmholtz operator built for the corresponding fields of temperature and velocity components, is the largest $m \times m$ matrix, where m reads for the total number of unknowns on the Eulerian grid (typically $O(10^6 - 10^7)$ for realistic 3D problems). Matrix \mathbf{H} is typically a sparse matrix, and for the spatial and temporal discretizations utilized in the present study has non-zero entries arranged along only 7 diagonals. For other spatial discretizations which may be based on stencils of higher orders and also include additional non linear terms discretized in a semi-implicit manner, the matrix \mathbf{H} will include more non-zero entries, but will still remain sparse, making it possible to perform simulations on dense grids with a reasonable memory consumption. In particular, the maximum amount of memory required to store all the four \mathbf{H} matrices (one for the temperature and three for all the velocity components) with all auxiliary data did not exceed 6 Gb of the CPU RAM for the 300³ grid resolution. Matrices \mathbf{I} and \mathbf{R} containing entries related to the acquired IB capability are of dimensions $n \times m$ and $m \times n$, respectively, where n reads for the total number of Lagrangian points (typically $O(10^3 - 10^4)$ for realistic 3D problems) determining the surfaces of all the immersed bodies involved in the simulation. The amount of non-zero entries in any $n - th$ row or column of the matrices \mathbf{I} and \mathbf{R} , respectively, depends on a specific kernel of the discrete delta functions or on an order of interpolations utilized in the interpolation, \mathbf{I} , and the regularization, \mathbf{R} , operators, but on any account is not higher than $O(10^2)$. As a result, the matrices \mathbf{I} and \mathbf{R} are extremely sparse; they both are stored in compressed sparse row (CSR) format and their multiplication by any vector is further implemented by using standard routines from the Intel Math Kernel Library (MKL).

Recall next that the previously developed solver [32], or any other solver for the solution of original (without IB functionality) NS equations $[\mathbf{H}_{\theta, \mathbf{u}}][\theta, \mathbf{u}] = [\mathbf{RHS}_{\theta, \mathbf{u}}]$, based on the pressure-velocity segregated approach, is available. The existing solver can be straightforwardly exploited in a black box manner for computing the product of the inverse modified Helmholtz operator, \mathbf{H}^{-1} , by any given vector of dimension $m \times 1$ comprising an essential part of the presently developed algorithm for the solution of the system of Eqs. (11a) and (11b), as detailed in Table 1.

The present study focuses on the configurations characterized by the stationary surfaces of immersed bodies that make it possible to precompute the matrix $[\mathbf{I}\mathbf{H}^{-1}\mathbf{R}]$ (see step 4 – a in Table 1), which comprises the most time consuming step of the developed approach. The final solution of Eq. (11a) (see step 5-a in Table 1) can be either completed by a direct method, which will include LU factorization of the matrix $[\mathbf{I}\mathbf{H}^{-1}\mathbf{R}]$ performed once at the beginning of the process followed by elimination and back substitution stages, or, alternatively, by utilizing any of the available iterative algorithms in each time step. Both options have been implemented in the present study. Namely, the direct LU factorization of the matrix $[\mathbf{I}\mathbf{H}^{-1}\mathbf{R}]$ has been performed by utilizing an open source MUMPS package [38,39], while the iterative solution of Eqs. (11) was performed by the bi-conjugate gradient (BICG) method based on Krylov subspace iteration. Implementation of both direct and iterative algorithms for completing step 5-a of the developed approach increases its flexibility, making it possible to switch between the algorithms depending on the hardware configuration, as well as the structure and the size of the matrix $[\mathbf{I}\mathbf{H}^{-1}\mathbf{R}]$.¹ It is remarkable that, similarly to the explicit direct forcing method [2,12], the single time step integration implemented in the present

¹ See the next section for more extensive discussion.

Table 1

A detailed description of the major steps of the developed algorithm for solution of equations, Eqs. (11a) and (11b).

The solution of $[Q, \mathbf{F}] = [\mathbf{I}\mathbf{H}^{-1}\mathbf{R}]^{-1}[\mathbf{I}\mathbf{H}^{-1}\mathbf{R}\mathbf{H}\mathbf{S}_{\theta, \mathbf{u}}^{n-1, n} - \theta_b, \mathbf{U}_b]$, (Eq. (11a))		
1-a.	Calculation of $\mathbf{H}^{-1}\mathbf{R}\mathbf{H}\mathbf{S}_{\theta, \mathbf{u}}^{n-1, n}$:	Employs the original solver [32]. The result is a $m \times 1$ vector.
2-a.	Calculation of $\mathbf{I}\mathbf{H}^{-1}\mathbf{R}\mathbf{H}\mathbf{S}_{\theta, \mathbf{u}}^{n-1, n}$:	Employs matrix–vector multiplication of the matrix \mathbf{I} stored in compressed sparse row (CSR) format by the vector $\mathbf{H}^{-1}\mathbf{R}\mathbf{H}\mathbf{S}_{\theta, \mathbf{u}}^{n-1, n}$ obtained in 1-a. The result is a $n \times 1$ vector.
3-a.	Calculation of $\mathbf{I}\mathbf{H}^{-1}\mathbf{R}\mathbf{H}\mathbf{S}_{\theta, \mathbf{u}}^{n-1, n} - \theta_b, \mathbf{U}_b$:	Subtraction of two $n \times 1$ vectors. The result is a $n \times 1$ vector.
4-a.	Calculation of $[\mathbf{I}\mathbf{H}^{-1}\mathbf{R}]$:	The procedure is repeated n times for every column $[\mathbf{R}]_n$ of the matrix \mathbf{R} . The original solver [32] is employed first to calculate $\mathbf{H}^{-1}[\mathbf{R}]_n$, followed by matrix–vector multiplication $\mathbf{I}\mathbf{H}^{-1}[\mathbf{R}]_n$. The final result is stored in the column $[\mathbf{I}\mathbf{H}^{-1}\mathbf{R}]_n$ of the $n \times n$ matrix.
5-a.	Calculation of $[\mathbf{I}\mathbf{H}^{-1}\mathbf{R}]^{-1}[\mathbf{I}\mathbf{H}^{-1}\mathbf{R}\mathbf{H}\mathbf{S}_{\theta, \mathbf{u}}^{n-1, n} - \theta_b, \mathbf{U}_b]$:	Either by employing <i>LU</i> factorization of the small $n \times n$ matrix ^a or by any iterative (e.g. Krylov space or multigrid based) method ^b .
The solution of $[\theta, \mathbf{u}^*] = \mathbf{H}^{-1}[\mathbf{R}\mathbf{H}\mathbf{S}_{\theta, \mathbf{u}}^{n-1, n} - \mathbf{R}[Q, \mathbf{F}]]$, (Eq. (11b))		
1-b.	Calculation of $\mathbf{R}[Q, \mathbf{F}]$:	Employs matrix–vector multiplication of the matrix \mathbf{R} stored in compressed sparse row (CSR) format by the vectors $[Q, \mathbf{F}]$ calculated by the solution of Eq. (11a).
2-b.	Calculation of $[\mathbf{R}\mathbf{H}\mathbf{S}_{\theta, \mathbf{u}}^{n-1, n} - \mathbf{R}[Q, \mathbf{F}]]$:	Subtraction of two $m \times 1$ vectors. The result is a $m \times 1$ vector.
3-b.	Calculation of $\mathbf{H}^{-1}[\mathbf{R}\mathbf{H}\mathbf{S}_{\theta, \mathbf{u}}^{n-1, n} - \mathbf{R}[Q, \mathbf{F}]]$:	Employs the original solver [32]. The result is a $m \times 1$ vector.

^a We use an open source MUMPS package [38,39].^b We use the bi-conjugate gradient (BICG) method.

implicit formulation also involves dual employment of the original solution of the NS (without IB functionality), as follows from steps 1 – a and 3 – b in Table 1.

3.2. Optimization strategies

With the aim of developing a generalized methodology capable of solution of highly resolved 3D flows, much effort was made to further optimize the developed algorithm in terms of memory consumption and CPU time. The study yielded a number of important optimization strategies, as detailed in the following.

3.2.1. Optimization of memory consumption

Optimization of the memory consumption is closely related to the way the $n \times n$ matrix $[\mathbf{I}\mathbf{H}^{-1}\mathbf{R}]$ was built. In general, this is an indefinite and non-symmetric matrix, with a structure depending on a kernel of discrete Dirac delta functions utilized in the interpolation and regularization operators, and on a spatial distribution of Lagrangian points. Furthermore, all the tests performed in the framework of the present study revealed that the matrix $[\mathbf{I}\mathbf{H}^{-1}\mathbf{R}]$ is always full of non-zero entries with absolute values that are mostly very small ($O(10^{-16})$ and smaller). Storing all these non-zero entries is prohibitively expensive, so that it is necessary to set up a sparsing threshold. Recalling that the matrix $[\mathbf{I}\mathbf{H}^{-1}\mathbf{R}]$ is an operator determining the values of the Lagrangian forces and heat fluxes enforcing the kinematic no-slip and thermal constraints on the surfaces of immersed bodies, the threshold value satisfying the required precision can be adjusted manually for any specific flow configuration. In all the simulations performed in the present study the absolute threshold values were in the range of (10^{-16} – 10^{-25}). For this range the thermal and kinematic no-slip constraints were met with up to 6 decimal digits, which should be less than the discretization error of the second order finite volume method utilized in the present study.

Another two factors which have a significant impact on the overall memory consumption are the value of the time step, Δt , and the Rayleigh number, Ra . The larger the time step the higher is amount of entries of the matrix $[\mathbf{I}\mathbf{H}^{-1}\mathbf{R}]$ that are above the given threshold value and therefore have to be stored in the memory. It

was found that for a time step of ($O(10^{-2})$) the relative amount of non-zero entries of matrix $[\mathbf{I}\mathbf{H}^{-1}\mathbf{R}]$ was typically about 3% of the total $n \times n$ matrix size. Decreasing the time step by a factor of 10 while keeping the same threshold value resulted in a steeper (by an order of magnitude or even more) decrease in the percentage of non-zero entries. Apart from the memory directly involved in the storage of non-zero entries of the matrix $[\mathbf{I}\mathbf{H}^{-1}\mathbf{R}]$, the time step and the Ra values have a significant impact on the overall memory consumption when performing *LU* factorization of the matrix. In fact, the typical amount of memory required to store the factors of a single $[\mathbf{I}\mathbf{H}^{-1}\mathbf{R}]$ matrix² obtained on the 300^3 grid for the immersed body determined by approximately 2×10^5 points is about 40 Gb for a time step of ($O(10^{-2})$) and only about 4 Gb for a time step of ($O(10^{-3})$). Detailed information regarding memory consumption versus the values of Ra number is given in Section 4 for all the flow configurations considered in the study.

3.2.2. Optimization of CPU time

Optimization of CPU time was implemented on two levels. The first level is related to building the matrices \mathbf{I} and \mathbf{R} and precomputing the matrix $[\mathbf{I}\mathbf{H}^{-1}\mathbf{R}]$. Recalling that the originally developed solver [32] was accelerated by utilizing a multithreading parallelism based on the OpenMP approach, it was natural to use the same parallelism paradigm when extending the code with the IB capability. All the simulations were performed on a standard Linux server having 128 GB DDR3 shared memory and 2 Intel Xeon 12C processors, 24 threads each (48 threads in total). This also allowed us to exploit a built-in multithreading parallelism of the matrix–vector multiplication functions from the Intel MKL library. The typical times required for precomputing the matrices \mathbf{I} and \mathbf{R} when running the simulation on 48 threads varied from a number of seconds to several minutes for the surfaces of immersed bodies determined by $O(10^3)$ and $O(10^5)$ Lagrangian points, respectively. Much more significant deviation in computational times was observed when precomputing the matrices $[\mathbf{I}\mathbf{H}^{-1}\mathbf{R}]$ for all the velocity and temperature fields, which involved performing n time integrations

² For 3D simulation the factors of four of such matrices need to be stored.

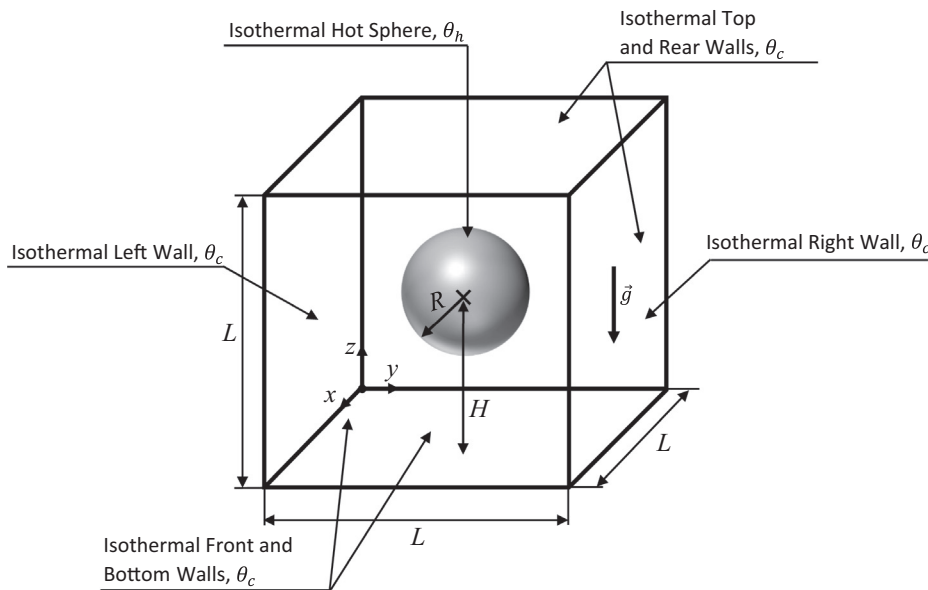


Fig. 2. Physical model of a hot sphere inside a cold cube characterized by $R = 0.2L$. The surface of the sphere is held at a constant hot temperature θ_h , while all the cube walls are held at a constant temperature θ_c . The position of the sphere along the cube centerline is determined by the non-dimensional parameter $\kappa = (H - 0.5L)/L$.

by the original solver for the three components of velocity and temperature. The typical times required at this stage varied from a number of hours to a number of days for 200^3 and 300^3 grids, respectively.

The second level is related to the stage of solution of the system of Eq. (10a). As has been already mentioned, the solution of the system of equations was performed by both direct and iterative solvers. In the case of the direct solver (i.e. MUMPS [38,39]) the LU factorization should be performed once at the beginning of the process for the four $[\mathbf{I}\mathbf{H}^{-1}\mathbf{R}]$ matrices, corresponding to the three velocity components and temperature. Again, a built-in multithread parallelism of the solver was employed at this stage, and even for the most dense grids (consisting of 300^3 finite volumes) the overall process was completed within less than an hour. The limitations of the direct solver show up at the intrinsically sequential forward elimination and back substitution stages which do not enjoy multithread parallelism. While on the coarse and moderate grids (up to 200^3 finite volumes) the above drawback is not pronounced, on denser grids (300^3 finite volumes) it becomes critical and drastically deteriorates the overall performance of the computations involving a large number of time steps. In this case the remedy comes from utilizing an iterative solver which can be based either on a Krylov subspace iteration³ or on multigrid algorithms. We found, however, that the solver based on the BICG algorithm suffers from poor convergence on non-uniform grids stretched close to the boundaries of the computational domain.

In summary, none of the checked direct and iterative approaches can be considered as the ultimate strategy for solution of the system of Eq. (10a). In general, the optimized strategy can benefit from a combination of direct and iterative methods, and will always depend on the flow configuration considered. More details regarding the strategies applied to the analysis of the specific benchmark flows are provided in the following sections of the paper. It should also be noted that after the matrix $[\mathbf{I}\mathbf{H}^{-1}\mathbf{R}]$ has been built, the solution of Eq. (10a) comprises an independent algebraic problem completely detached from the original solver and physical model and, therefore, can be

separately solved by utilizing distributed memory parallelism, e.g. a hybrid or MPI based approach. Parallelized iterative solution of the system of Eq. (10a) by utilizing both hybrid and MPI paradigms remained out of the scope of the present study and will be the focus of future work.

4. Results and discussion

The results obtained by the developed method have been extensively verified against available benchmark data provided for both steady and periodic confined natural convection flows.

4.1. Natural convection from a hot sphere inside a cold cube

Buoyancy convection from a hot sphere placed inside a cold cube is considered. The physical model of the problem is shown in Fig. 2. The surface of the sphere is held at a constant hot temperature θ_h , while the cold temperature θ_c is preserved at all the cube walls. The cube edge length L is used for normalizing all the length scales of the problem. The hot sphere is of a constant radius $R = 0.2L$. The center of the sphere is attached to the vertical centerline of the cube and its position is determined by a non-dimensional parameter $\kappa = (H - 0.5L)/L$, where H is a non-dimensional distance between the center of the sphere and the bottom wall of the cube. The best accuracy of the IB method is achieved by providing a uniform distribution of the Lagrangian points over the surface of the sphere, implemented by the non-iterative method of Leopardi [40], so that each point is confined by a virtual surface of an equal area, as illustrated in Fig. 3. Defining the Nusselt number, Nu , as the ratio of convective to conductive heat fluxes from the given surface, its average value for the surface of the sphere is obtained by:

$$\overline{Nu} = \sqrt{PrRa\Delta x\overline{Q}}, \quad (12)$$

where the average heat flux \overline{Q} is an arithmetic mean of all the non-dimensional heat fluxes Q_k at each Lagrangian point k of the immersed surface, intrinsically calculated by the solution of the system of Eqs. (10). We next define an average value of the global Nusselt number \overline{Nu}_G as a measure of global heat flux through the surface of confining cubic enclosure as:

³ For example, presently employed BICG algorithm.

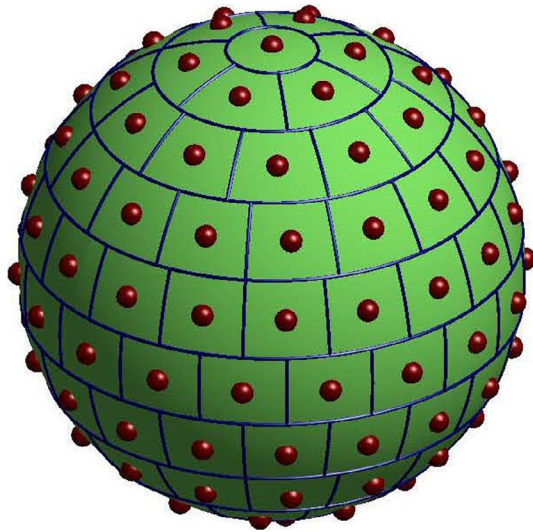


Fig. 3. Illustration of 100 Lagrangian points evenly distributed over the surface of a unit sphere by the non-iterative method of Leopardi [40].

$$\overline{Nu}_G = \frac{1}{N} \sum_{i=1}^N \frac{\partial \theta}{\partial \mathbf{m}} \quad (13)$$

where $\frac{\partial \theta}{\partial \mathbf{m}}$ is the temperature gradient averaged over the surface of a given edge of the cubic enclosure and N is the number of thermally un-insulated edges. The steady state flows were calculated for the values of $Ra = 10^5$ and $Ra = 10^6$ and $\kappa = \pm 0.25$. Convergence to steady state was assumed when the value of the L_∞ norm calculated for the difference between the flow characteristics obtained at two consecutive time steps was less than 10^{-6} for all the flow fields. Details regarding setup and characteristics of the numerical solution are given in Table 2. Although direct algorithms are known for their high memory requirements for performing LU factorization, the memory consumption for this specific configuration is quite modest, not exceeding 21 Gb of RAM for the most memory

consuming configuration. It is remarkable that the lower the Ra value, the higher is the memory consumption, which approximately scales as $Ra^{-0.16}$ when the threshold value for the matrix $[\mathbf{I}\mathbf{H}^{-1}\mathbf{R}]$ sparsing is equal to 10^{-22} . Recalling that the fullness of matrix $[\mathbf{I}\mathbf{H}^{-1}\mathbf{R}]$ reflects the intensity of mutual interactions between the Lagrangian forces and heat fluxes, the above observation is not surprising, since the intensity is higher for viscosity dominated flows. It is remarkable that for the moderate memory consumption typical of the present configuration the time step duration is almost independent of the fullness of matrix $[\mathbf{I}\mathbf{H}^{-1}\mathbf{R}]$, allowing for optimally exploiting advantages of the direct solver. Note also that if the sphere is positioned far enough (more than two grid cells) from the cube boundaries the characteristics of the numerical solution do not depend on the spatial location of the sphere inside. For this reason the characteristics in Table 2 are presented for only a single value of $\kappa = 0$.

Comparison between the obtained average \overline{Nu} and \overline{Nu}_G values and the corresponding values available in the literature is presented in Table 3. An acceptable agreement is observed for both $Ra = 10^5$ and $Ra = 10^6$ and the entire range of κ values, successfully verifying the present calculations. Natural convection flow is visualized by presenting vortical structures, determined by utilizing the λ_2 criterion proposed by Jeong and Hussain [41]. According to the authors, the outermost outer surface of the vortex can be revealed by connecting the same negative λ_2 values close to zero. Following the recent study in [42], the value of $\lambda_2 = -0.1$ was chosen for the visualization of convection cells. Fig. 4 presents the visualization vortical structures obtained for the values of $Ra = 10^5$ and $Ra = 10^6$ and $\kappa = 0.25; 0; -0.25$. As expected, the obtained fully 3D steady state flow is symmetric relative to the $X-Z$ and $Y-Z$ center planes, as well as relative to both main diagonal planes of the cube. For all the configurations a large circumferential convection cell is formed close to the top boundary of the cube. For $Ra = 10^5$ the cell is characterized by a nearly toroidal shape, whereas for $Ra = 10^6$ there is an entrainment of the bottom surface of the cell into the cell interior. The entrainment becomes more pronounced with decreasing the value of κ .

Table 2

Setup and characteristics of the numerical solution obtained on 200^3 grid for the simulation of the natural convection from a hot sphere placed inside a cold cube characterized by the value of $R/L = 0.2$, $\kappa = 0$.

Ra	Time step duration, s	RAM, Gb	Δt	Method of solution of Eq. (11a)	Number of time steps	Sparsing threshold for $[\mathbf{I}\mathbf{H}^{-1}\mathbf{R}]$
10^3	2.5	20.70	10^{-3}	Direct (LU)	$O(10^3)$	10^{-22}
10^4	2.4	15.04			$O(10^3)$	
10^5	2.3	9.734			$O(10^4)$	
10^6	2.2	6.892			$O(10^4)$	

Table 3

Comparison between the present and the previously published \overline{Nu} and \overline{Nu}_G values averaged over the surface of hot a sphere placed within a cold cube and the surface of the cold cube, respectively.

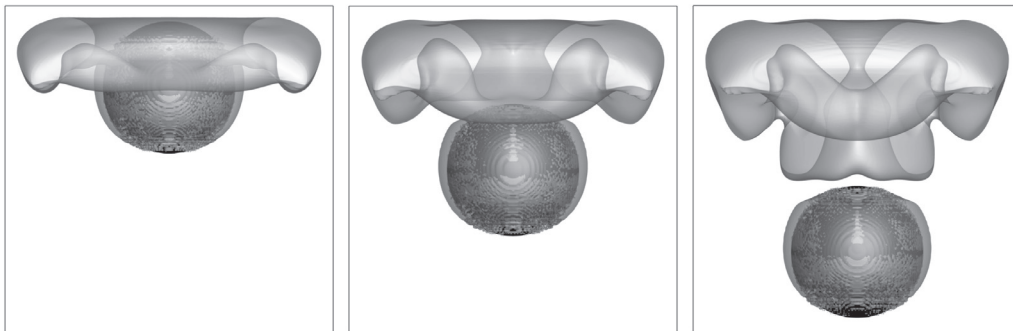
κ	\overline{Nu}						\overline{Nu}_G			
	$Ra = 10^5$			$Ra = 10^6$			$Ra = 10^5$		$Ra = 10^6$	
	Ref. [6]	Ref. [9]	Present	Ref. [6]	Ref. [9]	Present	Ref. [6]	Present	Ref. [6]	Present
-0.25	13.665	13.774	13.489	20.890	21.993	20.611	1.1995	1.1113	1.8372	1.7246
-0.2	12.931	13.058	12.768	20.631	21.862	20.517	1.1353	1.0507	1.8135	1.6680
-0.1	12.729	13.105	12.819	20.772	22.164	21.216	1.1111	1.0542	1.8230	1.7246
0	12.658	13.415	13.160	20.701	23.344	21.589	1.1044	1.0821	1.8174	1.7524
0.1	12.351	13.446	13.230	20.367	22.525	21.674	1.0802	1.0880	1.7816	1.7522
0.2	12.254	13.635	13.462	19.664	22.208	21.487	1.0619	1.1082	1.7227	1.7368
0.25	12.944	14.426	14.277	19.721	22.393	21.757	1.1193	1.1771	1.7222	1.7588



(a) Isosurfaces of $\lambda_2 = -0.1$, $Ra = 10^5$, $\kappa = 0.25; 0; -0.25$ - front view



(b) Isosurfaces of $\lambda_2 = -0.1$, $Ra = 10^5$, $\kappa = 0.25; 0; -0.25$ - top view



(c) Isosurfaces of $\lambda_2 = -0.1$, $Ra = 10^6$, $\kappa = 0.25; 0; -0.25$ - front view



(d) Isosurfaces of $\lambda_2 = -0.1$, $Ra = 10^6$, $\kappa = 0.25; 0; -0.25$ - top view

Fig. 4. Visualization of convection cells for the natural convection flow from a hot sphere placed inside a cold cube.

The value of κ is also directly correlated with the major radius of the toroidal convection cell. It can also be seen that for smaller values of κ the center of the convection cell acquires an elongated center-hollowed mushroom shape.

4.2. Natural convection from a hot horizontal cylinder in a cold cube

The natural convection flow from a hot horizontal cylinder of radius R , whose axis coincides with the spanwise centerline of a

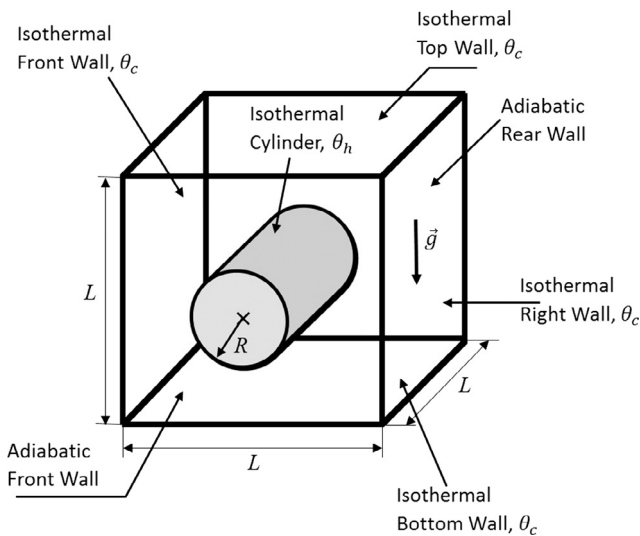


Fig. 5. Physical model of a hot cylinder inside a cold cube.

cold cube of edge L is considered. The cylinder extends across the whole width of the cube. The cylinder surface is held at a constant hot temperature, θ_h , while its edges are attached to the adiabatic walls of the cube. All other cube walls are held at a constant cold temperature, θ_c , and the gravity force acts downwards (see Fig. 5). In the present study the configurations with radii varying in the range of $0.1 \leq R/L \leq 0.4$ were investigated.

All the simulations were performed on 200^3 uniform grids with a time step of $\Delta t = 10^{-3}$. The convergence to steady state was assumed when the value of the L_∞ norm calculated for the difference between the flow characteristics obtained at two consecutive time steps was less than 10^{-6} for all the flow fields. Special treatment should be applied when interpolating Eulerian temperatures on the Lagrangian points of the cylinder attached to both adiabatic walls of the cube, and when regularizing the volumetric Lagrangian heat fluxes to the adjacent Eulerian grid. This is because of the symmetric two-sided character of the discrete Dirac delta function (see Eq. (5)) utilized in the present study. Therefore, just a naive employment of the above delta function in interpolation (or regularization) operators acting from only one side of the cube wall will lead to an accumulation of computational errors. The remedy comes from the fact that the adiabatic boundary condition is equivalent to a mirror symmetry distribution of the temperature field with respect to the thermally insulated wall. For this reason one should double the impact of interpolated temperatures and regularized volumetric heat fluxes when employing the delta function from one side of the adiabatic boundary. Note also that no special treatment is required for the velocity fields at the same Lagrangian points, as they are simply skipped when building the corresponding \mathbf{R} and \mathbf{I} matrices. This is due to the fact that the non-slip kinematic constraints are automatically enforced by the imposed non-slip boundary conditions on the cube's walls.

It should be noted that our simulations resulted in steady state flows for the entire range of R/L and Ra values. This result differs from the data recently published in [42], which reported quasi-periodic and periodic flows for $R/L = 0.3$ and $R/L = 0.4$, respectively, at $Ra = 10^6$. In order to exclude the dependence of the currently obtained solution on grid resolution and time step, additional simulations were performed for the above configurations for which the values of maximal grid size and time step were reduced by a factor of 2. Insignificant deviations (no more than 0.5% for all the flow variables) were found between the newly obtained and the original steady state flows, which favorably ver-

ifies the independence of our results on grid size and time step values. The apparent reason for the discrepancy observed between the current and the previously published [42] results can be attributed to the non-linear physics of the considered system, which can simultaneously exhibit a number of different states at the same value of Ra number. Performing a global linear stability analysis which would formally prove the existence and character of the bifurcated flow reported in [42] remained out of the scope of the present study. In the following we will put forward a number of arguments supporting the possibility that the currently observed and the previously reported flows belong to different branches. With all these caveats an acceptable agreement is observed when comparing the present and the previously reported \overline{Nu} and \overline{Nu}_C values [42] averaged over the surface of a hot cylinder and the surface of the cold cube, respectively,⁴ for the entire range of R/L and Ra values, as summarized in Tables 4 and 5, respectively. A spatial distribution of temperature isosurfaces corresponding to the steady state flows calculated for the entire range of R/L and Ra values is shown in Fig. 6. For all the configurations and for the entire range of Ra numbers the temperature distribution is symmetric relative to the $X-Z$ and $Y-Z$ planes. As expected, all the configurations are characterized by a linear temperature distribution with almost concentric isosurfaces for a small value of $Ra = 10^3$. With an increase in the Ra values, the flow convection becomes more dominant and the temperature isosurfaces acquire irregular shapes. For $R/L = 0.1$ and $R/L = 0.2$ the shape of temperature isosurfaces elongates towards the top of the cube. The shape elongation is not uniform and is more pronounced close to the isothermal boundaries, which is a result of lower vertical velocities in these regions. This trend, however, is not observed for configurations with higher R/L ratios. To understand the reason for this difference we visualize convection cells inherent to each configuration for the highest $Ra = 10^6$ value by plotting the λ_2 criterion (see Fig. 7). It can be seen that the configurations with $R/L = 0.1$ and $R/L = 0.2$ are characterized by two major convection cells which are symmetric relative to the $Y-Z$ plane and occupy the whole upper part of the cube. This is in contrast to configurations with higher R/L ratios (0.3 and 0.4), both having four major symmetric discrete convection cells located close to the top boundary of the cube.

We next focus on the differences between the present and the reported [42] time averaged spatial distributions of convection cells obtained for $Ra = 10^6$ and $R/L = 0.4$. Both distributions are symmetric relative to the $X-Z$ and $Y-Z$ planes. The currently obtained distribution is characterized by four major discrete convection cells appearing close to the cube top boundary. In contrast, the distribution reported in [42] has only two major discrete convection cells occupying almost the entire region adjacent to the top boundary of the cube. The above argument supports a possibility that the two flows belong to different branches, which can explain the observed qualitative and qualitative differences between the corresponding flow characteristics. The qualitative discrepancies between the present and the reported [42] time averaged spatial distributions of convection cells obtained for $Ra = 10^6$ and $R/L = 0.3$ are not as significant as for $R/L = 0.4$ and, apparently, can be attributed to the differences between the implemented numerical approaches. An accurately performed grid and time independence analysis further corroborates the currently obtained results.

Table 6 presents details regarding characteristics of the numerical solution as a function of the Ra and R/L values. As has been already observed for the configuration of a hot sphere inside a cold cube, the memory consumption varies inversely with the Ra value.

⁴ The \overline{Nu} and \overline{Nu}_C values corresponding to unsteady flows reported in [42] were averaged over time.

Table 4Comparison between the present and the previously published \overline{Nu} values averaged over the surface of a hot cylinder placed within a cold cube.

R/L	$Ra = 10^4$		$Ra = 10^5$		$Ra = 10^6$	
	Ref. [42]	Present	Ref. [42]	Present	Ref. [42]	Present
0.1	6.2493	6.4880	11.138	11.662	18.326	19.250
0.2	5.1184	5.1500	7.2271	7.5800	13.361	13.937
0.3	5.8084	5.7304	6.4790	6.5169	11.272	11.401
0.4	8.7030	8.5544	8.7030	8.7643	10.716	10.832

Table 5Comparison between the present and the previously published \overline{Nu}_c values averaged over the surface of the cold cube.

R/L	$Ra = 10^4$		$Ra = 10^5$		$Ra = 10^6$	
	Ref. [42]	Present	Ref. [42]	Present	Ref. [42]	Present
0.1	1.0201	1.0208	1.8099	1.8360	2.9945	3.0348
0.2	1.6161	1.6188	2.3766	2.3814	4.3985	4.3677
0.3	2.6216	2.9091	2.9726	3.0702	5.1956	5.3844
0.4	5.1919	5.3928	5.2651	5.5131	6.6106	6.8313

In the present case it scales as $Ra^{-0.16}$, $Ra^{-0.15}$, $Ra^{-0.12}$ and $Ra^{-0.09}$ for $R/L = 0.1$, $R/L = 0.2$, $R/L = 0.3$, and $R/L = 0.4$, respectively. At the same time, for the given Ra value the memory consumption is linearly proportional to the number of Lagrangian points characterizing the surface of the immersed body. Regarding the computational times, no clear trend was observed as a function of the Ra and R/L values, while the average time required for performing a single time step is equal to 3.4 s.

4.3. Differentially heated spherical shell

We consider the natural convection flow in a spherical shell formed by two concentric spheres,⁵ as shown in Fig. 8. The internal sphere of radius R_i is held at a constant hot temperature θ_H , while the external sphere of radius R_o is held at a constant cold temperature θ_C . The gravity force acts downwards. The difference between the temperatures of the internal and external spheres, $\theta_H - \theta_C$, and the difference between the external and internal radii, $L = R_o - R_i$, are used to scale the temperature and length fields. An additional parameter, $\phi = R_i/R_o$ is introduced to define the ratio between internal, R_i , and external, R_o , radii. Hence, the geometrical dimensions of the spherical gap are not constant and are determined by a non-dimensional distance between the external and internal radii (which is always equal to unity) and the value of parameter ϕ . According to our previous research [43], a grid consisting of no less than 6×10^6 finite volumes is required to accurately capture both steady and unsteady flow phenomena for this kind of flow. For this reason all the simulations presented in this section were performed on a 300^3 uniform grid.

We start our analysis with an investigation of the steady state flows developing inside the spherical shell characterized by the value of $\phi = 0.5$ for the range of $10^4 \leq Ra \leq 10^6$. Fig. 9 presents 3D iso-surfaces of temperature distribution, along with the temperature contours and pathlines of the steady flow observed at the $X - Z$ cross section.

It can be clearly recognized that the obtained steady state flow is axi-symmetric for all Ra values, in agreement with the previously published studies [9,44–46]. As expected the

dominance of convective heat transfer increases with increasing Ra , which is reflected by a prevalence of thin thermal and velocity boundary layers near the surface of the hot sphere and higher temperature gradients inside the hot thermal plume located at the top region of the gap. It can also be observed that the center of the crescent convection cell typical of these kinds of flow is shifted upwards as the Ra number increases. As can be seen from Table 7, the values of the \overline{Nu} number averaged over the surface of the outer sphere by

$$\overline{Nu} = \sqrt{PrRa\Delta x\overline{Q}}/\phi, \quad (14)$$

are in good agreement with the corresponding \overline{Nu} values previously reported in the literature for the entire range of Ra values. It should be noted here that there is a small (no more than 2%) discrepancy between the average \overline{Nu} values calculated for internal and external spheres. This discrepancy is typical of numerical methods which do not directly resolve the thinnest boundary layers near the solid surfaces, such as IB and lattice Boltzman methods. For example, a discrepancy of about the same magnitude was reported by Gallegos and Málaga [47], who investigated the natural convection flow in eccentric spherical shells by the lattice Boltzman method. It is believed, however, that the observed discrepancy does not lead to incorrect conservation of the overall heat flux through the boundaries of the sphere (otherwise the steady state regime would not have been reached), rather, it only appears while postprocessing the obtained results and can be further minimized by increasing the number of Lagrangian points. Similarly to the previously discussed configurations the memory consumption of the present flow also depends inversely on the Ra value (see Table 8). At the same time the memory demand is quite modest and does not exceed 21 Gb of RAM for the most memory consuming case characterized by the value of $Ra = 10^3$.

To prove that the developed method can also accurately simulate unsteady fluid dynamics, the natural convection inside a concentric spherical shell characterized by the value of $\phi = 0.714$ is simulated at $Ra = 4.6 \times 10^4$ and $Pr = 0.7$. At these values, the natural convection flow developing inside the above configuration exhibits complex periodic multi cellular dynamics in the form of either pulsating or travelling waves, as detailed in a number of previous studies [48–50]. Fig. 10a presents a snapshot of the distribution of radial velocity v_r on the midrange $D = (D_i + D_o)/2$ spherical

⁵ Both shells are located at the center of a cube with all no-slip cold boundaries which is not shown in Fig. 8 to avoid clumsiness. In all the simulations the minimal distance between the external sphere and the cube walls was at least 10 grid cells, in order to avoid high velocity and temperature gradients outside the external sphere.

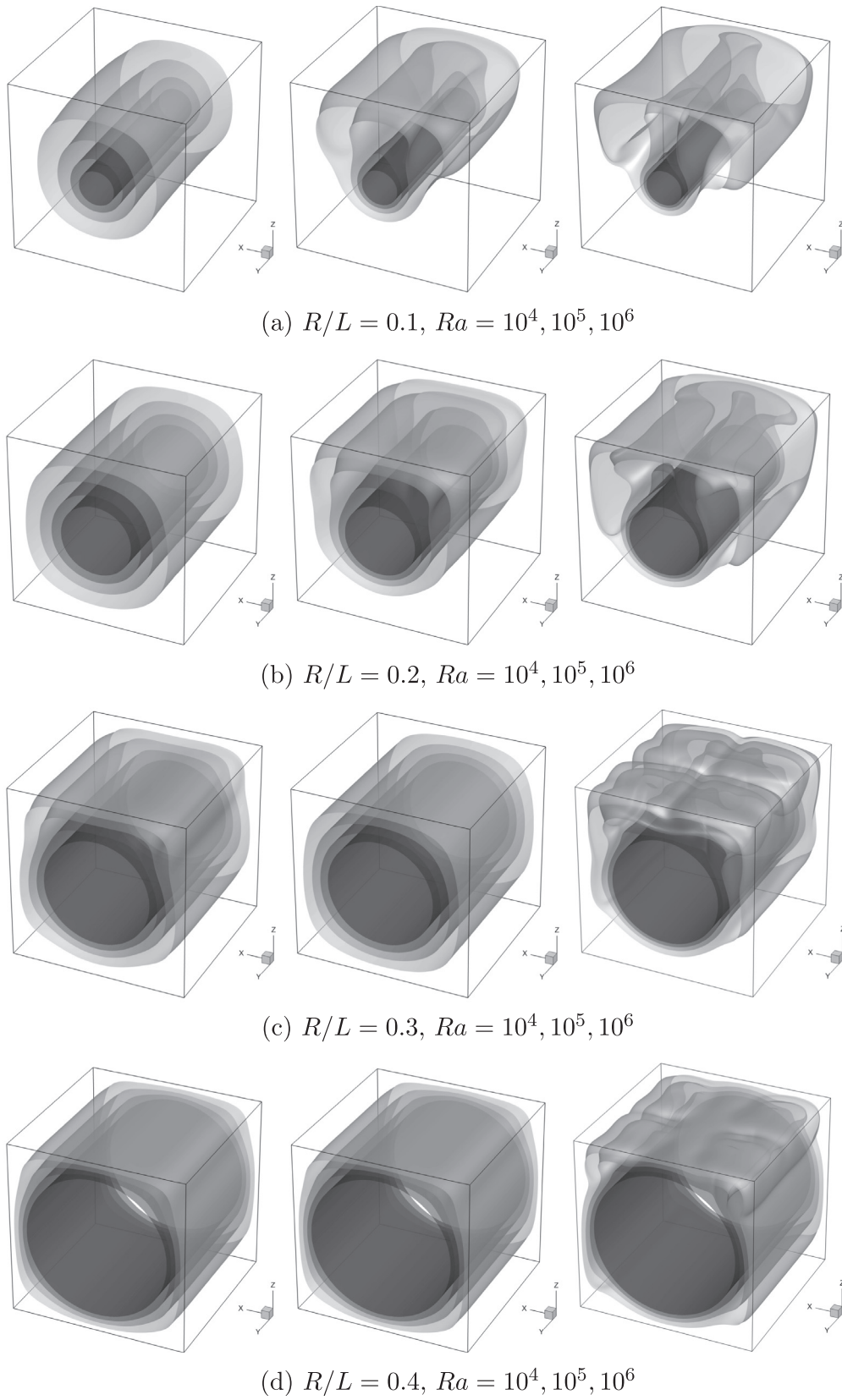
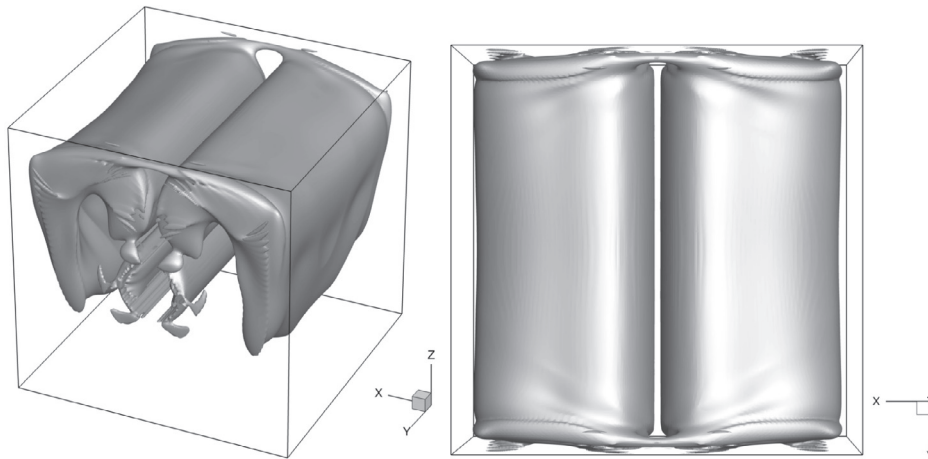
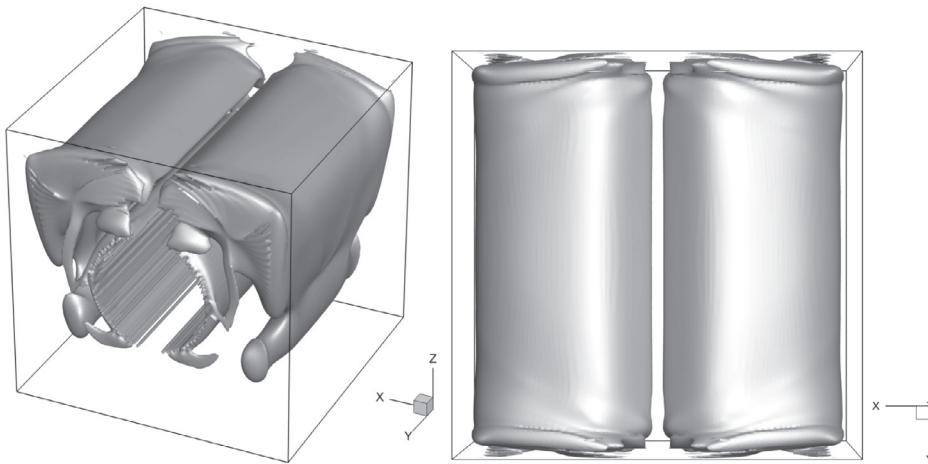
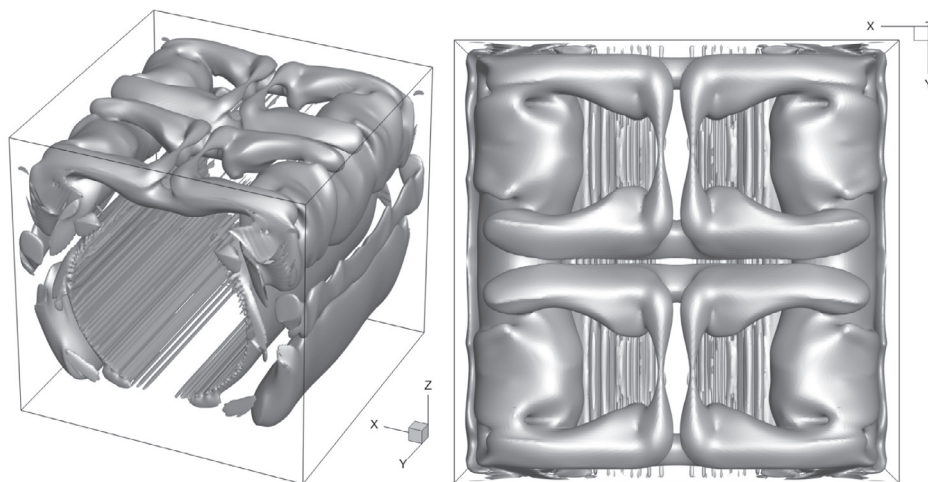


Fig. 6. Spatial distribution of temperature isosurfaces for the values of $\theta = 0, 0.25, 0.5, 0.75$.

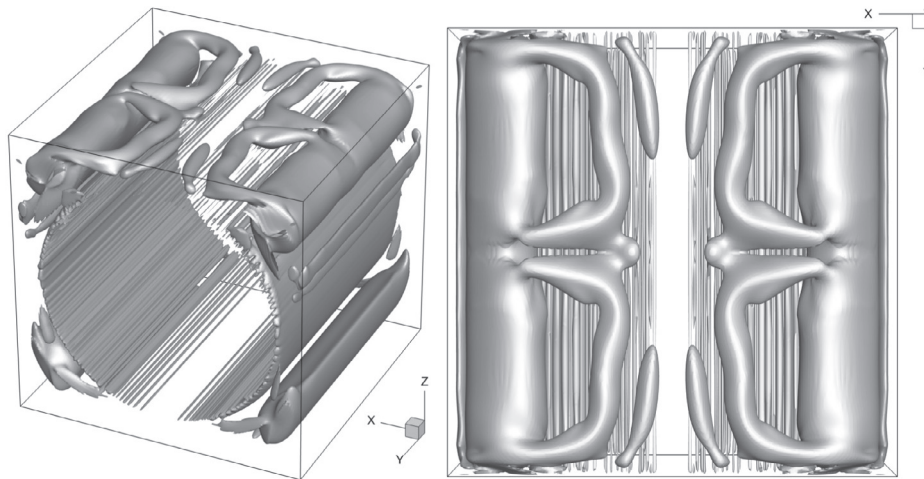
surface. The distribution is characterized by 10 clearly distinguished discrete convection cells which have been formed as a result of symmetry breaking Hopf bifurcation. This is in agreement

with the results obtained by the linear stability analysis of Travnikov et al. [50], which predict the highest growth rate for the wave number $m = 10$ at $Ra_{cr} = 3.9563 \times 10^4$. Fig. 10b presents the time

(a) Isosurfaces of $\lambda_2 = -0.1, R/L = 0.1$ (b) Isosurfaces of $\lambda_2 = -0.1, R/L = 0.2$ (c) Isosurfaces of $\lambda_2 = -0.1, R/L = 0.3$ **Fig. 7.** Visualization of convection cells for the natural convection flow from a hot cylinder placed inside a cold cube.

evolution of temperature acquired at point $(0.01262, -2.1212, 2.1165)$ located on the midrange spherical surface on the meridian circle along which the maximal v_r component variations are

observed. The Fourier transform of the temperature evolution yields the value of angular frequency of the bifurcated flow $\omega = 0.389$. Note that this value is in excellent agreement with



(d) Isosurfaces of $\lambda_2 = -0.1, R/L = 0.4$

Fig. 7 (continued)

Table 6

Setup and characteristics of the numerical solution obtained on a 200^3 grid for the simulation of the natural convection from a hot cylinder placed inside a cold cube.

Ra	R/L	Time step duration, s	RAM, Gb	Δt	Method of solution of Eq. (11a)	Number of time steps	Threshold for the filling of $[\mathbf{H}^{-1}\mathbf{R}]$
10^4	0.1	3.4	29.5	10^{-3}	Direct (LU)	$O(10^3)$	10^{-25}
	0.2	3.2	50.3			$O(10^3)$	
	0.3	7.5	68.0			$O(10^3)$	
	0.4	5.2	92.1			$O(10^3)$	
10^5	0.1	3.2	19.1	10^{-3}	Direct (LU)	$O(10^4)$	10^{-25}
	0.2	2.9	32.5			$O(10^3)$	
	0.3	3.5	48.7			$O(10^3)$	
	0.4	4.6	68.6			$O(10^3)$	
10^6	0.1	2.4	14.1	10^{-3}	Direct (LU)	$O(10^4)$	10^{-25}
	0.2	2.8	25.5			$O(10^4)$	
	0.3	3.2	40.6			$O(10^4)$	
	0.4	4.3	62.1			$O(10^4)$	

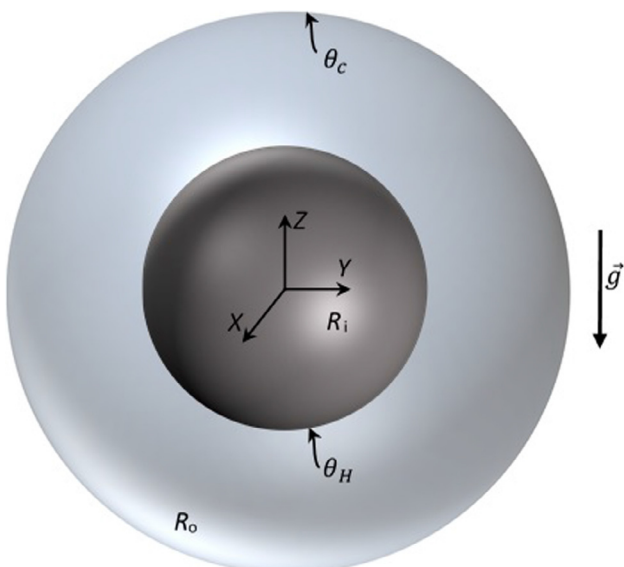


Fig. 8. Physical model of the differentially heated concentric spherical shell.

the critical frequency value ω_{cr} reported in [50].⁶ It is remarkable that the average Nusselt number \overline{Nu} is even more sensitive to the flow oscillations, and oscillates with an angular frequency twice that of the temperature (and the velocity) fields (see Fig. 10c), in agreement with the recent study of Scurtu et al. [48]. Taking into account that rotating wave is a pattern characterized by a rigid rotation yielding constant values of the global Nusselt number, the presently obtained oscillating behaviour of Nu indicates that the observed instability is of pulsating mode.

The advantages of utilizing an iterative BICG algorithm for the solution of Eq. (11a) for a large problem solved on a 300^3 grid can be seen from the setup characteristics summarized in Table 9. In fact, despite the high resolution and the large amount of Lagrangian points determining the surface of two immersed bodies (more than 3×10^5) the simulation only required 10 Gb of RAM. This is comparable with the memory required when simulating convection flow from a single sphere in a cube on a 200^3 grid (about 2×10^4 Lagrangian points) and approximately 6 time less than maximal amount memory consumed when simulating convection

⁶ The ω_{cr} reported in [50] was renormalized by dividing it by $\sqrt{\frac{Ra}{Pr}}$ to fit the scaling of the present study.

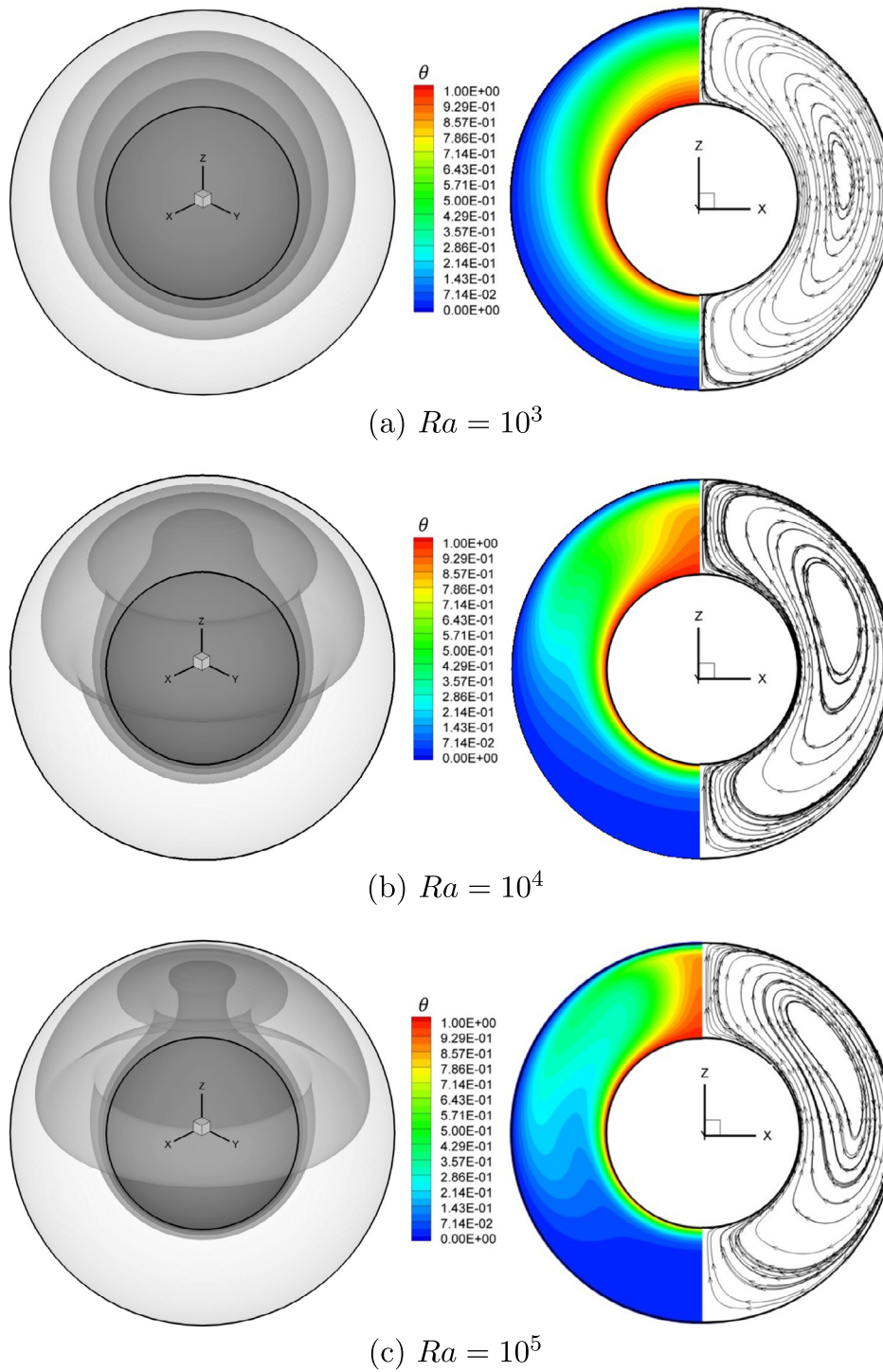


Fig. 9. Flow characteristics of steady state natural convection flow in a spherical gap characterized by the value of $\phi = 0.5$. Left column corresponds to 3D isosurfaces of temperature, $\theta = 1, 0.75, 0.5, 0.25, 0$; right column corresponds to the contours of temperature and path lines observed at $Z - X$ cross-section.

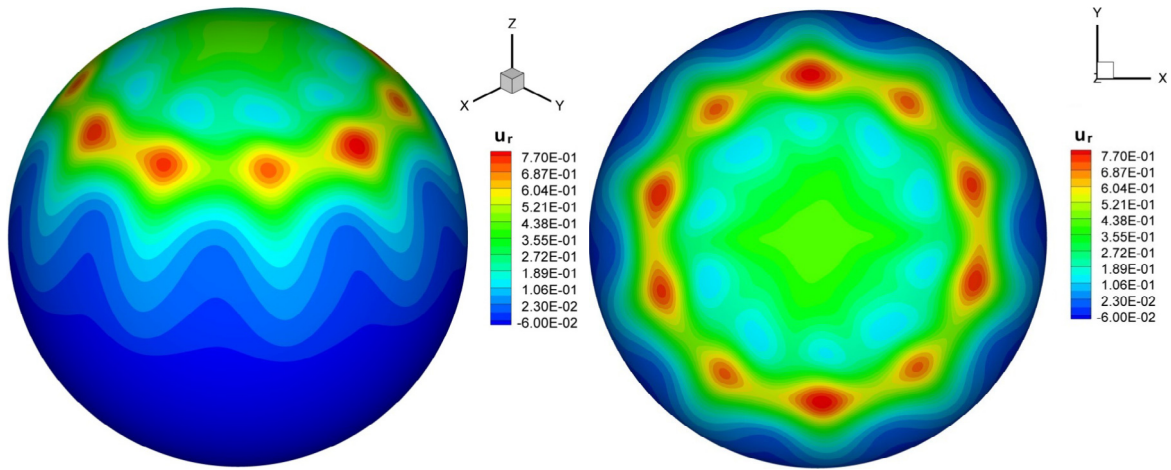
Table 7
The Nu number values averaged over the surface of an external sphere.

ϕ	Ra	Present	Ref. [9]	Ref. [44]	Ref. [45]	Ref. [46]
0.5	10^3	1.1088	1.120	1.104	1.0990	1.1310
	10^4	1.9648	1.987	1.9665	1.9730	1.9495
	10^5	3.4649	3.452	3.4012	3.4890	3.4648

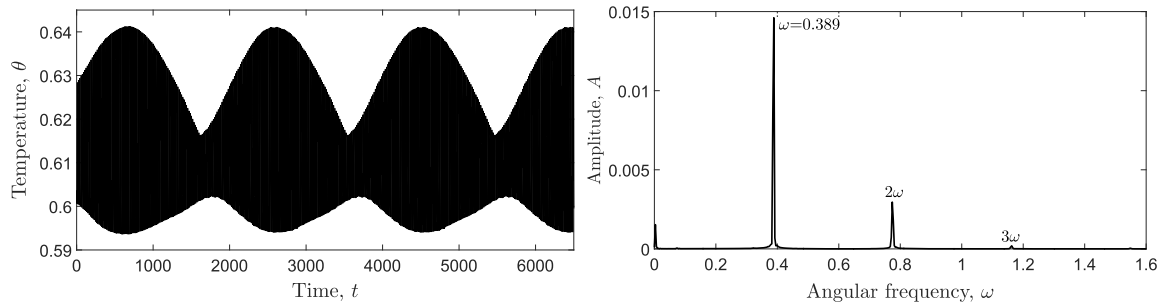
Table 8

Setup and characteristics of the numerical solution obtained on a 200^3 grid for the simulation of the natural convection inside a differentially heated spherical shell, characterized by the value of $\phi = 0.5$.

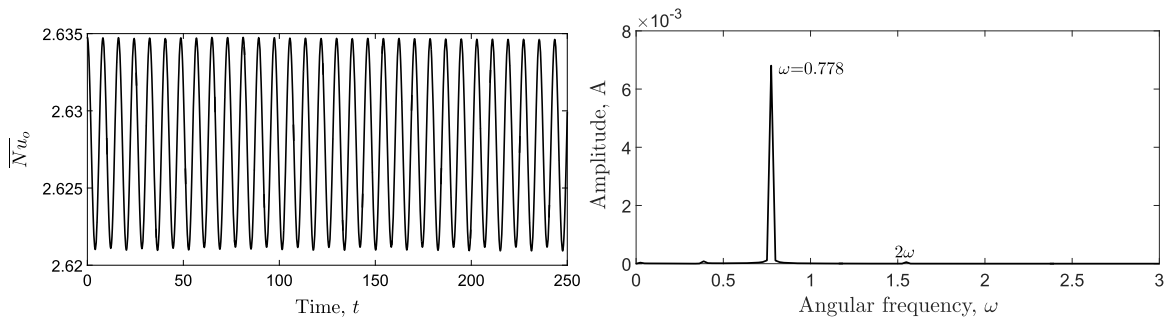
Ra	Time step duration, s	RAM, Gb	Δt	Method of solution of Eq. (11a)	Number of time steps	Sparsing threshold for $[IH^{-1}R]$
10^3	2.5	20.70	10^{-3}	Direct (LU)	$O(10^4)$	10^{-16}
10^4	2.4	15.04			$O(10^4)$	
10^5	2.3	9.734			$O(10^5)$	



(a) Snapshot of the radial velocity v_r on the midrange $D = (D_i + D_o)/2$ spherical surface



(b) Time evolution and Fourier spectrum of the temperature acquired at point $(0.01262, -2.1212, 2.1165)$



(c) Time evolution and Fourier spectrum of the Nusselt number averaged over the surface of the outer sphere, \overline{Nu}_o

Fig. 10. Flow characteristics of unsteady natural convection flow exhibiting instability of pulsating mode obtained for $Ra = 4.6 \times 10^4$.

Table 9
Setup characteristics of the numerical solution obtained on a 300^3 grid for the simulation of the natural convection inside a differentially heated spherical shell characterized by the value of $\phi = 0.714$.

ϕ	Time step duration, s	RAM, Gb	Δt	Method of solution of Eq. (11a)	Number of time steps	Threshold for the filling of $[\mathbf{IH}^{-1}\mathbf{R}]$
0.714	15.1	10.1	5×10^{-3}	BICG algorithm	$O(10^5)$	10^{-25}

flow from a horizontal cylinder in a cube on a 200^3 grid (about 1×10^5 Lagrangian points).⁷ The relatively long duration of a time step (15.1 s) is not surprising and is the result of both high grid resolution and a low value of convergence criterion for the BICG algorithm, which was set to $\xi = 10^{-7}$.

5. Conclusions

A novel formulation of the IB method based on a semi-implicit coupling between the Lagrangian force and heat flux densities and the Eulerian velocity and temperature fields has been developed and successfully verified for a number of confined 3D natural convection flows in the presence of immersed bodies of various geometries. The obtained results were favorably compared with available data for both steady and periodic flows. The existence of qualitative differences between the obtained and previously published study for the flow around a horizontal cylinder characterized by high R/L ratios and a value of $Ra = 10^6$ was attributed to the flow non-linearity, which can exhibit different states at the same Ra values.

The developed approach can be seen as a generic methodology for acquiring existing time marching solvers of NS equations based on a segregated pressure-velocity coupling (e.g. SIMPLE, fractional step, projection methods and their derivatives) with the IB functionality. The implemented semi-implicit coupling between the Lagrangian force and heat flux densities and Eulerian velocity and temperature fields has proved to be a highly efficient and high fidelity methodology for simulating confined natural convection flows in the presence of immersed bodies at moderate Ra values. The high efficiency of the methodology stems from the Schur complement decoupling of the governing equations, which allows for precomputing the most time consuming operations of the algorithm.

Performance of both direct and iterative methods was investigated for the time marching stage of the algorithm. It was found that the time marching stage of the approach utilizing the direct method based on LU decomposition is attractive for the problems solved on moderate (up to 200^3) grid resolutions. On the other hand, as a result of the poor scalability of the elimination and back substitution stages, the performance of the direct method based solvers deteriorates if high resolution grids are utilized. It was demonstrated that replacing a direct LU factorization by an iterative BICG algorithm at the time marching stage of the algorithm provides a promising alternative if the solution is performed on high resolution grids.

High memory consumption can be seen as a major bottleneck of the developed approach. While the above limitation is not expected to manifest itself if the program will run on a standard Linux server equipped with 128 Gb RAM or more, some adjustments are to be made if the developed approach is expected to be used on a standard PC. Summarizing the tests made for the benchmark flows discussed in the paper, the memory demand can be controlled by modifying sparsing threshold when building the matrix $[\mathbf{IH}^{-1}\mathbf{R}]$ which will still remain small enough to

precisely meet all kinematic constraints of the problem. A reasonable amount of required memory can be achieved by either decreasing the time step for flows characterized by the low values of Re and Ra numbers or by both increasing the grid resolution and decreasing the time step if the flows under consideration are characterized by the high values of Re and Ra numbers. It should be stressed again that for high resolution grids only iterative solvers have to be used for the solution of Eq. (11b).

Although the present study focused on a demonstration of the capabilities of the developed algorithm for stationary immersed bodies, it should be noted that the developed methodology can also be straightforwardly extended to configurations with periodically moving immersed bodies. Such an extension is of high relevance in the simulation of the undulatory motion of tiny sea creatures, as well as in various biomedical applications, and will be the focus of our future work. An emphasis will be put on parallelizing the time marching stage of the algorithm by employing hybrid and distributed memory paradigms. Recalling that the time marching stage comprises a stand alone part completely detached from the original solver and physical model, it is expected that the above task can be accomplished with a reasonable programming effort.

Conflict of interest

None.

Acknowledgement

The study was funded by GIF-German Israel Foundation, under contract number: I-2471-405.10/2017.

References

- [1] C.S. Peskin, Flow patterns around heart valves: a numerical method, *J. Comput. Phys.* 10 (1972) 252–271.
- [2] J. Mohd-Yusof, Combined immersed-boundary/b-spline methods for simulations of flow in complex geometries, *Center Turbul. Res., Ann. Res. Briefs* (1997) 317–327.
- [3] A. Faldun, E.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.
- [4] D.V. Le, B.C. Khoo, K.M. Lim, An implicit-forcing immersed boundary method for simulating viscous flows in irregular domains, *Comput. Meth. Appl. Mech. Engin.* 197 (2008) 2119–2130.
- [5] R. Mittal, G. Iaccarino, The immersed boundary method, *Ann. Rev. Fluid Mech.* 37 (2005) 239–261.
- [6] 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 Transfer* 53 (2010) 3143–3155.
- [7] 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. Fluid.* 57 (2012) 40–51.
- [8] W. Ren, C. Shu, W. Yang, An efficient immersed boundary method for thermal flow problems with heat flux boundary conditions, *Int. J. Heat Mass Transfer* 64 (2013) 694–705.
- [9] Y. Gulberg, Y. Feldman, On laminar natural convection inside multi-layered spherical shells, *Int. J. Heat Mass Transfer* 91 (2015) 908–921.
- [10] S.W. Su, M.C. Lai, C.A. Lin, An implicit-forcing immersed boundary method for simulating viscous flows in irregular domains, *Comput. Fluid.* 36 (2007) 313–324.
- [11] Z. Wang, J. Fan, K. Luo, Combined multi-direct forcing and immersed boundary method for simulating flows with moving particles, *Int. J. Multiphase Flow* 34 (2008) 283–302.
- [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.

⁷ Both simulations are based on direct LU decomposition utilized for the solution of Eq. (11a).

- [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. Meth. 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] 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.
- [21] A. Samanta, D. Appelo, T. Colonius, J. Nott, J. Hall, Computational modeling and experiments of natural convection for a titan mongolfiere, *AIAA J.* 48 (2010) 1007–1016.
- [22] 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.
- [23] B. Kallemov, A. Bhalla, B. Griffith, A. Donev, An immersed boundary method for rigid bodies, *Comm. App. Math. Comp. Sci.* 11 (2016) 79–141.
- [24] 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.
- [25] 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.
- [26] 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.
- [27] Y. Bao, A. Donev, B. Griffith, D. McQueen, C. Peskin, An Immersed Boundary method with divergence-free velocity interpolation and force spreading, *J. Comput. Phys.* 347 (2017) 183–206.
- [28] 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.
- [29] H. Park, X. Pan, C. Lee, J.-I. Choi, A pre-conditioned implicit direct forcing based immersed boundary method for incompressible viscous flows, *J. Comput. Phys.* 314 (2016) 774–799.
- [30] D. Le, B. Khoo, K. Lim, An implicit-forcing immersed boundary method for simulating viscous flows in irregular domains, *Comput. Meth. Appl. Mech. Eng.* 197 (2008) 2119–2130.
- [31] J. Adams, R. Swarztrauber, R. Sweet, FISHPACK: Efficient FORTRAN Subprograms for the Solution of Separable Elliptic Partial Differential Equations, 1999 <<http://www.scd.ucar.edu/css/software/fishpack/>>.
- [32] 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.
- [33] R.E. Lynch, J.R. Rice, D.H. Thomas, Direct solution of partial differential equations by tensor product methods, *Numer. Math.* 6 (1964) 185–199.
- [34] 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.
- [35] A. Roma, C.S. Peskin, M.J. Berger, An adaptive version of the immersed boundary method, *J. Comput. Phys.* 153 (1999) 509–534.
- [36] M. Uhlmann, An immersed boundary method with direct forcing for the simulation of particulate flows, *J. Comput. Phys.* 209 (2005) 448–476.
- [37] S.V. Patankar, *Numerical Heat Transfer and Fluid Flow*, McGraw-Hill, 1980.
- [38] P. Amestoy, I. Duff, J. LéXcellent, J. Koster, Multifrontal parallel distributed symmetric and unsymmetric solvers, *Comput. Meth. Appl. Mech. Eng.* 184 (1998) 501–520.
- [39] 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.
- [40] P. Leopardi, A partition of the unit sphere into regions of equal area and small diameter, *Electron. Trans. Numer. Anal.* 25 (2006) 309–327.
- [41] J. Jeong, F. Hussain, On the identification of a vortex, *J. Fluid Mech.* 285 (1995) 69–94.
- [42] Y. Seo, J. Doo, M. Ha, Three-dimensional flow instability of natural convection induced by variation in radius of inner circular cylinder inside cubic enclosure, *Int. J. Heat Mass Transfer* 95 (2016) 566–578.
- [43] Y. Feldman, T. Colonius, On a transitional and turbulent natural convection in spherical shells, *Int. J. Heat Mass Transfer* 64 (2013) 514–525.
- [44] Y. Feldman, T. Colonius, M. Pauken, J. Hall, J. Jones, Simulation and cryogenic experiments of natural convection for the Titan Montgolfiere, *AIAA J.* 50 (2012) 2483–2491.
- [45] H.S. Chu, T.S. Lee, Transient natural convection heat transfer between concentric spheres, *Int. J. Heat Mass Transfer* 36 (1993) 3159–3170.
- [46] A.A. Dehghan, K. Masih, Numerical simulation of buoyancy-induced turbulent flow between two concentric isothermal spheres, *Heat Transfer Eng.* 31 (2010) 33–44.
- [47] A.D. Gallegos, C. Malaga, Natural convection in eccentric spherical annuli, *Euro. J. Mech. - B/Fluids* 65 (2017) 464–471.
- [48] N. Scurtu, B. Fütterer, C. Egbers, Three-dimensional natural convection in spherical annuli, *J. Phys.: Conf. Ser.* 137 (2008) 1–9, 012017.
- [49] Y. Feldman, T. Colonius, On a transitional and turbulent natural convection in spherical shells, *Int. J. Heat Mass Transf.* 64 (2013) 514–525.
- [50] 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 Transfer* 80 (2015) 266–273.