Journal of Computational Physics 229 (2010) 7072–7101

Contents lists available at ScienceDirect

Journal of Computational Physics

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

Nested Cartesian grid method in incompressible viscous fluid flow

Yih-Ferng Peng [a,*], Rajat Mittal [b], Amalendu Sau [c,**], Robert R. Hwang [d]

[a] Department of Civil Engineering, National Chi Nan University, Puli 545, Taiwan
[b] Department of Mechanical and Aerospace Engineering, George Washington University, Washington, DC 20052, USA
[c] Department of Mechanical and Aerospace Engineering, Gyeongsang National University, Gyeongnam 660-701, Jinju, South Korea
[d] Institute of Physics, Academia Sinica, Nankang, Taipei 11529, Taiwan

ARTICLE INFO

Article history:
Received 15 January 2009
Received in revised form 16 April 2010
Accepted 28 May 2010
Available online 8 June 2010

Keywords:
Nested Cartesian grid
Fractional-step method
Immersed boundary method

ABSTRACT

In this work, the local grid refinement procedure is focused by using a nested Cartesian grid formulation. The method is developed for simulating unsteady viscous incompressible flows with complex immersed boundaries. A finite-volume formulation based on globally second-order accurate central-difference schemes is adopted here in conjunction with a two-step fractional-step procedure. The key aspects that needed to be considered in developing such a nested grid solver are proper imposition of interface conditions on the nested-block boundaries, and accurate discretization of the governing equations in cells that are with block-interface as a control-surface. The interpolation procedure adopted in the study allows systematic development of a discretization scheme that preserves global second-order spatial accuracy of the underlying solver, and as a result high efficiency/accuracy nested grid discretization method is developed. Herein the proposed nested grid method has been widely tested through effective simulation of four different classes of unsteady incompressible viscous flows, thereby demonstrating its performance in the solution of various complex flow–structure interactions. The numerical examples include a lid-driven cavity flow and Pearson vortex problems, flow past a circular cylinder symmetrically installed in a channel, flow past an elliptic cylinder at an angle of attack, and flow past two tandem circular cylinders of unequal diameters. For the numerical simulations of flows past bluff bodies an immersed boundary (IB) method has been implemented in which the solid object is represented by a distributed body force in the Navier–Stokes equations. The main advantages of the implemented immersed boundary method are that the simulations could be performed on a regular Cartesian grid and applied to multiple nested-block (Cartesian) structured grids without any difficulty. Through the numerical experiments the strength of the solver in effectively/accurately simulating various complex flows past different forms of immersed boundaries is extensively demonstrated, in which the nested Cartesian grid method was suitably combined together with the fractional-step algorithm to speed up the solution procedure.

1. Introduction

In the numerical simulations of complex physical flow phenomena the crucial requirement is the predictability, that is, the simulated results should remain faithful to the actual processes. To this point, the generation/accumulation of numerical errors during the computation is of special concern, since it introduces distortions that fundamentally alter the fidelity of the

* Corresponding author.
** Corresponding author.
E-mail addresses: yfpeng@ncnu.edu.tw (Y.-F. Peng), asau@gnu.ac.kr (A. Sau).

0021-9991/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2010.05.041

simulations. Errors resulting from a lack of spatial resolution are particularly deleterious. On the other hand, over-resolving is computationally expensive. Therefore, how to efficiently/effectively solve the governing partial differential equations (PDEs), representing the mathematical model of the physical problem concerned, becomes a subject of active research in the numerical analysis [1,2].

In general, there are two approaches to obtain accurate solution of PDEs. One such approach is to employ high-order numerical methods [3–5]. The high-order methods can achieve a given accuracy of the solutions with much coarser mesh than the low-order schemes. Thus, despite the larger bandwidth of the stiffness matrix, the high-order methods often prove to be more computationally efficient than low-order schemes. However, high-order methods can yield accurate results only for length scales that are larger than a few mesh cells. For problems with locally high gradient, they may not be able to accurately capture the local fine-scale behavior of the physical phenomenon. Therefore, the higher order methods are usually preferred for problems with smooth solutions. In addition, for many types of practical problems, there exists a limit beyond which it is quite difficult to improve the accuracy of the solutions by using higher order methods. Thus, an extremely attractive alternative is to resort to the second approach, that is, implementation of added mesh refinement in order to improve the solution through resolution of computational grids. Notably, the mesh refinement is desirable to improve spatial resolution both in uniform and non-uniform grids. In the uniform mesh refinement, the resulting grid evolution equation is generally less non-linear, which becomes very efficient in conjunction with the successive-over-relaxation (SOR). However, the uniform mesh refinement may not be perfect for applications in which the solution may need different resolutions at different regions. For the well-understood physical problems, a non-uniform mesh can therefore be designed to reflect the necessary resolution requirement. For example, for the boundary value problems, a fine resolution is typically required for regions near the boundaries. However, for the evolving interfacial flows or flow fields with complicated structures, the local refinement techniques are more preferred in order to locally increase mesh densities at the regions of interest, and thereby saving the computer resources. Nowadays, in general, the local mesh refinement plays an indispensable role in the efficient solution of both industrial and scientific problems. Over the past decades, intensive efforts have been made for the development of local refinement procedures [6–15]. Consequently, a large number of adaptive algorithms have been proposed, which allow use of multi-resolution approximation in the simulation and represent recent advances in the area. The strategies of adaptive mesh refinement are further classified into structured and unstructured approaches.

Notably, a representative structured grid approach is the adaptive Cartesian mesh refinement of Berger and Oliger [6]. Their formulation is established on regular Cartesian meshes and arranged hierarchically with different resolutions. In regard to the unstructured approach, Zhu and Zienkiewicz [8] reviewed state-of-the-art automatic mesh refinement strategies, and discuss its important role in the error estimation and automatic adaptation in finite-element analysis. Again, in the context of block-structured local refinement, important it may be to mention here the work of Minion [16] in which a second-order accurate cell-centered Godunov type projection method is used to solve incompressible Euler equations. On the other hand, Martin and Colella [17] present a solution algorithm for Euler equations that uses block-structured local refinements both in space and time. The method has subsequently been extended by Martin et al. [18] to compute three-dimensional incompressible viscous flows, in which a second-order semi-implicit Runge–Kutta scheme is used to compute the viscous terms. Notably, while solving flow interaction with visco-elastic structures, Griffith et al. [19] efficiently combined the Godunov type projection method (to compute the non-linear advection terms) of Minion [16] together with the semi-implicit second-order Runge–Kutta scheme (to calculate diffusion) as introduced in McCorquodale et al. [20]. Worth mentioning it may be here that, the projection methods are a class of fractional-step algorithm that first compute the velocity field (which is generally not divergence free) from the momentum equations without imposing the incompressibility constraint. The updated velocity is subsequently obtained by solving the pressure Poisson equation.

In this work, we focus on developing a structured multi-block Cartesian grid method for effectively simulating unsteady viscous incompressible flows with complex immersed boundaries. The strongest point for the proposed structured nested grid refinement approach is that the solution process of the resulting ‘uniform’ grid evolution equation is very efficient in conjunction with the line-SOR. Notably, the presently implemented alternating direction line-SOR method is one of the most widely used iterative methods for solving equations resulting from discretization on structured grids. The advantage of using the line-SOR, as a pre-conditioner, is that convergence rate becomes significantly faster than what is obtained with a simple point Jacobi method [21]. A further advantage here (of using line SOR) is that the procedure only requires the solution of tri-diagonal systems, and this can be accomplished with ease using the Thomas algorithm.

The idea of immersed boundary (IB) method was first introduced by Peskin [22], in which the solid object is represented by a distributed body force in the Navier–Stokes equations. Since Peskin [22] introduced the method, the numerous modifications and refinements have been proposed and a number of variants of this approach now exist [23,24]. In addition, there appear another class of method, usually referred to as "Cartesian grid method," which were originally developed for simulating inviscid flows with complex embedded solid boundaries on Cartesian grids [25]. These methods have subsequently been extended to simulate the unsteady viscous flows [26,21] and have capabilities similar to those of IB methods. Notably, the original IB method by Peskin [22] was developed for the coupled simulation of blood flow and the muscle contraction in a beating heart and is generally suitable for flows with immersed elastic boundaries. But its application to flows with rigid bodies poses problems, because the constitutive laws used for elastic boundaries are not generally well posed in the rigid limit. Subsequently, another method for the rigid body class is developed by Angot et al. [27] and Khadra et al. [28]. In this method, the entire flow is assumed to occur in a porous medium and is governed by the Navier–Stokes/Brinkman equations. These equations contain an extra force term (with respect to the classical Navier–Stokes equations) in the form $\vec{f} = (\mu/K)\vec{u}$, where K is the permeability of the medium having values infinity or zero for the fluid and the solid regions, respectively. The force activates only within the solid, driving the velocity field to zero. In practice, K is large (small) in fluid (solid) regions. Notably, such a method has conveniently been used to simulate flows past a single/pair of circular cylinder [29], and over a backward facing step [28].

In this work, we suitably combine ideas of the IB method of Peskin [22] and the VOF methodology [30] to simulate flows past single/multiple bluff bodies. Similar to the IB method, the presence of the solid object in the flow field is introduced in the form of an external body force density in the governing equations. Implementation of the force density is effected by making use of the volume-fraction field as introduced in the VOF method. To be precise, volume-fraction field determines what fraction of each computational cell is occupied by the solid phase. In cells where the volume-fraction is unity, the velocity is set equal to that of the solid body (which is zero here, since the immersed bodies are stationary in our case), and for cells where volume-fraction is between zero and one, the velocity is adjusted in proportion to the volume-fraction. On the other hand, for purely fluid cells this volume-fraction is zero, and there acts no body force. This way, in the computation, the entire flow field (including the solid bodies) is treated as a continuous fluid medium, and the body force acts on the fluid to satisfy the condition that the velocity of cells occupied by the solid body matches the velocity of the immersed surface. Primary advantage of our IB method is that the task of grid generation is greatly simplified here. On the other hand, generating body conformal structured/unstructured grid is often very cumbersome. While the unstructured grid approach is better suited for complex geometries, the grid quality can deteriorate with increasing complexity of the geometry. In contrast, in simulations with non-body conformal Cartesian grid, the grid complexity and the quality of results are not significantly affected due to complexity in the geometry. In the present Cartesian grid method with nested multiple blocks our objective is therefore to construct an efficient solution methodology that provides adequate local resolution even for complex immersed geometries and with minimum number of total grid points.

The current paper will therefore focus on describing these salient features of the numerical methodology, together with validating the accuracy, examining fidelity of the approach, and extensively demonstrating the capabilities of the proposed nested grid solver in some of the complex configurations like, lid-driven cavity flow, transient evolution of wake past a circular cylinder placed symmetrically in a channel, flow past an elliptic cylinder with an angle of attack, and flow control (e.g., suppression of vortex shedding) behind two tandem circular cylinders of different diameters.

2. Fractional-step method in nested grids

2.1. Fractional-step formulation

We now describe the underlying fractional-step method for a Cartesian mesh. The governing equations are the unsteady viscous incompressible Navier–Stokes equations written in terms of primitive variables. The equations are subsequently discretized on a Cartesian mesh using a cell-centered colocated (non-staggered) arrangement of variables $(\vec{u},p)$. The integral forms of the implemented dimensionless governing equations are:

The mass conservation equation

$$\int_{CS}\vec{u}\cdot\vec{n}\,dS=0, \tag{1}$$

and the momentum conservation equation

$$\frac{\partial}{\partial t}\int_{CV}\vec{u}\,dV+\int_{CS}\vec{u}(\vec{u}\cdot\vec{n})dS=-\int_{CS}p\,\vec{n}\,dS+\frac{1}{Re}\int_{CS}\nabla\vec{u}\cdot\vec{n}\,dS. \tag{2}$$

These equations are used as the starting point for deriving a second-order accurate finite-volume method. In the equations above CV and CS denote the control-volume and the control-surface, respectively, and $\vec{n}$ is a unit vector normal to the control-surface. A second-order accurate two-step fractional-step method is used for advancing the solutions in time. In this time-stepping scheme, the solutions are advanced from time level "n" to "$n + 1$" through an intermediate advection–diffusion step, where momentum equations without the pressure gradient terms are first advanced in time. A second-order Adams–Bashforth scheme is employed for the convective terms, and the diffusion terms are discretized using an implicit Crank–Nicolson scheme. This eliminates the viscous stability constraint which can be quite severe in simulation of such flows. At this stage, in addition to the cell-center velocities, which are denoted by $\vec{u}$, we also introduce face-center velocities $\vec{U}$. However, in a manner similar to a fully staggered arrangement, only the component normal to the cell-face is computed and stored (see Fig. 1). In our finite-volume discretization, the face-center velocity is used for computing the volume flux from each cell. The semi-discrete form of the advection–diffusion equation for each cell may, therefore, be written as:

$$\int_{CV}\frac{\vec{u}^{*}-\vec{u}^{n}}{\Delta t}dV=-\frac{1}{2}\int_{CS}\left[3\vec{u}^{n}\left(\vec{U}^{n}\cdot\vec{n}\right)-\vec{u}^{n-1}\left(\vec{U}^{n-1}\cdot\vec{n}\right)\right]dS+\frac{1}{2Re}\int_{CS}\left(\nabla\vec{u}^{*}+\nabla\vec{u}^{n}\right)\cdot\vec{n}\,dS, \tag{3}$$

where $\vec{u}$ is the intermediate cell-center velocity; CV and CS denote volume and surface of a cell, respectively. Following the advection–diffusion step, the intermediate face-center velocity $\vec{U}$ is computed by interpolating the intermediate cell-center velocity.

The advection–diffusion step is followed by the pressure-correction step

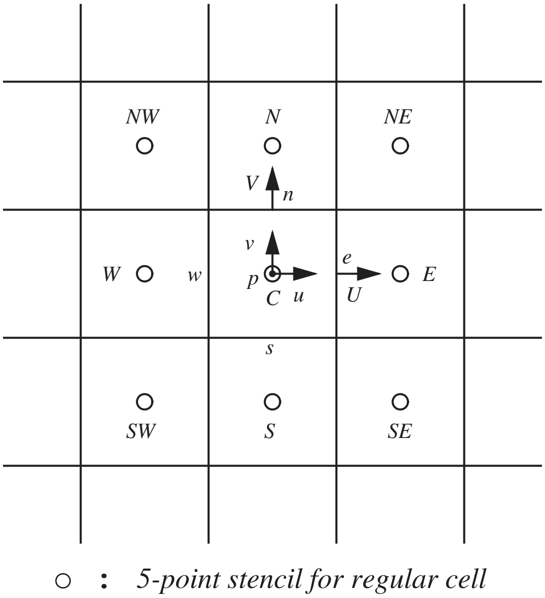

Fig. 1. Schematic of regular Cartesian cells, and the arrangement of cell-center and face-center velocities.

$$\int_{CV}\frac{\vec{u}^{n+1}-\vec{u}^{*}}{\Delta t}dV=-\int_{CV}\nabla p^{n+1}dV, \tag{4}$$

where we require that the final velocity field satisfy the integral mass conservation equation,

$$\int_{CS}\vec{U}^{n+1}\cdot\vec{n}\,dS=0. \tag{5}$$

This results to the following pressure equation

$$\int_{CS}(\nabla p^{n+1})\cdot\vec{n}\,dS=\frac{1}{\Delta t}\int_{CS}\left(\vec{U}^{*}\cdot\vec{n}\right)dS \tag{6}$$

which is the integral form of the Poisson equation. Note that, the pressure correction step is represented by the inviscid Eq. (4) and is well posed only if the velocity component normal to the boundary is specified. The velocity boundary condition consistent with Eq. (4) is $\vec{u}^{n+1}\cdot\vec{N}=\vec{v}^{n+1}\cdot N$, where $\vec{v}$ is the prescribed boundary velocity, and $\vec{N}$ is the unit normal to the boundary of the flow domain. It can easily be shown that $(\nabla p^{n+1})\cdot\vec{N}=0$, which we use as the boundary condition for Eq. (6). Once the pressure is obtained by solving this equation, both the cell-center (cc) and face-center (fc) velocities are updated separately as,

$$\vec{u}^{n+1}=\vec{u}^{*}-\Delta t\left(\nabla p^{n+1}\right)_{cc},\quad \vec{U}^{n+1}=\vec{U}^{*}-\Delta t\left(\nabla p^{n+1}\right)_{fc}. \tag{7}$$

For instance, with reference to Fig. 1, the x-direction pressure gradient at the cell center ‘c’ is computed as

$$(\partial p/\partial x)_C=(p_E-p_W)/2\Delta x, \tag{8}$$

whereas the same gradient on the east face is given by

$$(\partial p/\partial x)_e=(p_E-p_C)/\Delta x. \tag{9}$$

It follows however that $\vec{U}^{n+1}$ is not simply an interpolated form of the face-center velocities $\vec{u}^{n+1}$ [16]. In fact, the pressure equation Eq. (6) is discretized in terms of the pressure gradients on the cell faces and with the separate update of the face-center velocity, as shown in Eq. (7), ensuring exact satisfaction of Eq. (5).

2.2. The nested grid methodology

In this work, a local grid refinement technique is adopted by introducing multiple nested blocks. A detailed sketch of the flow domain, together with the inside coarse/fine grid area, is presented in Fig. 2(a) and (b). For simplicity, we describe our nested-block technique for two-dimensional (2D) flows with a two-block domain. However, the present nested-block technique can be extended for 3D flows and with higher multi-block domains without much hindrance. The coarse and fine grid domains in Fig. 2(a) and (b) are denoted by D1 and D2, respectively. Again, for simplicity, the fine grid length is defined

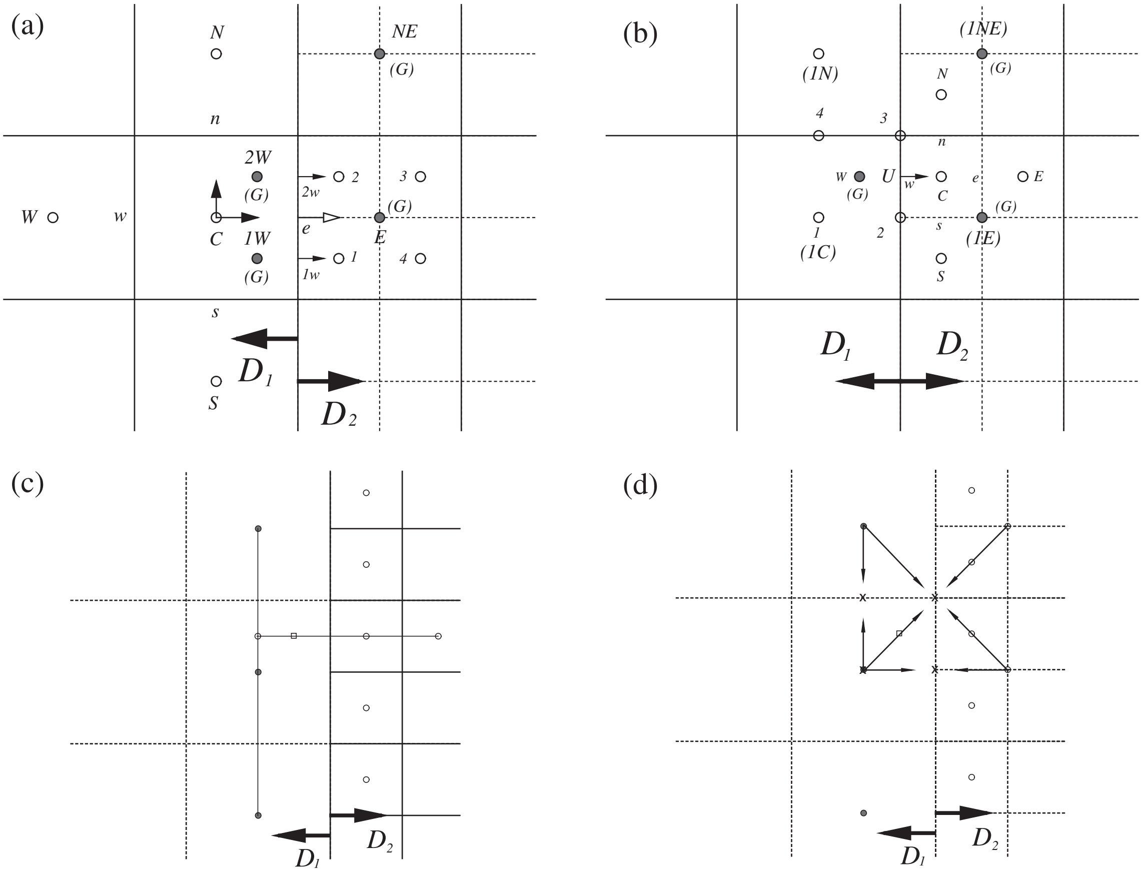

Fig. 2. Schematic plots of refinement cells. (a) Coarse cell. (b) Fine cell. (c and d) Adopted interpolation procedures.

as half the coarse grid length. The approach adopted here is to effectively solve the flow within D1 and D2 domains simultaneously in combination with the fractional-step procedure. Notably, while the basic concept of local grid refinement used in our study appears relatively similar to the earlier cell-center based projection methods (e.g., [19]), here the imposition of interface conditions on the nested-block boundaries and subsequent discretization of the governing equations in cells that are with block-interface remained quite different. Moreover, the immersed boundary method employed in our study is also different from that of Griffith et al. [19]. For clarity, we now describe interpolation relationships for ‘ghost cell’ of fine cells with fine/coarse domain interface of both the methods. As shown in Fig. 2(c), the cell centered adoptive projection method use shifted coarse grid stencils (solid circles) to get intermediate value (left most open circle), then perform final interpolation to get ‘ghost cell’ values (open square). Note that to perform interpolation for the horizontal coarse–fine interface, the cell-centered adoptive projection method need to shift the coarse stencil. While in the present study, as noted from Fig. 2(d), the ‘ghost cell’ value (open square) used is the average of the four corner values (values at cross symbols around the open square), in which the corresponding cross symbol values are specified following Eq. (17). Although there appears variety of interpolations for ‘ghost cell’ of fine cells with fine/coarse domain interface, the interpolation adopted here allows systematic development of a very convenient high efficiency nested grid immersed boundary method.

2.2.1. The advection–diffusion step

Notably, the presently derived nested-block formulation is extended from a single block algorithm. To start with, in the advection–diffusion step, the scalar form of discretized u^* and v^* equation [Eq. (3)] may be written as:

$$\begin{aligned}\frac{u^*-u^n}{\Delta t}\Delta x\Delta y = &-\frac{1}{2}\{3[(u_e^nU_e^n-u_w^nU_w^n)\Delta y+(u_n^nV_n^n-u_s^nV_s^n)\Delta x]\\&-\left[\left(u_e^{n-1}U_e^{n-1}-u_w^{n-1}U_w^{n-1}\right)\Delta y+\left(u_n^{n-1}V_n^{n-1}-u_s^{n-1}V_s^{n-1}\right)\Delta x\right]\Big\}\\&+\frac{1}{2Re}\left\{\left[\left(\frac{\partial u^*}{\partial x}\right)_e-\left(\frac{\partial u^*}{\partial x}\right)_w\right]\Delta y+\left[\left(\frac{\partial u^*}{\partial y}\right)_n-\left(\frac{\partial u^*}{\partial y}\right)_s\right]\Delta x\right.\\&\left.+\left[\left(\frac{\partial u^n}{\partial x}\right)_e-\left(\frac{\partial u^n}{\partial x}\right)_w\right]\Delta y+\left[\left(\frac{\partial u^n}{\partial y}\right)_n-\left(\frac{\partial u^n}{\partial y}\right)_s\right]\Delta x\right\},\end{aligned}$$

$$
\begin{aligned}
\frac{v^{*}-v^{n}}{\Delta t}\Delta x\Delta y = & -\frac{1}{2}\{3\left[(v_e^nU_e^n - v_w^nU_w^n)\Delta y + (v_n^nV_n^n - v_s^nV_s^n)\Delta x\right] \\
& - \left[\left(v_e^{n-1}U_e^{n-1} - v_w^{n-1}U_w^{n-1}\right)\Delta y + \left(v_n^{n-1}V_n^{n-1} - v_s^{n-1}V_s^{n-1}\right)\Delta x\right]\} \\
& + \frac{1}{2Re}\left\{\left[\left(\frac{\partial v^{*}}{\partial x}\right)_e - \left(\frac{\partial v^{*}}{\partial x}\right)_w\right]\Delta y + \left[\left(\frac{\partial v^{*}}{\partial y}\right)_n - \left(\frac{\partial v^{*}}{\partial y}\right)_s\right]\Delta x\right. \\
& \left. + \left[\left(\frac{\partial v^{n}}{\partial x}\right)_e - \left(\frac{\partial v^{n}}{\partial x}\right)_w\right]\Delta y + \left[\left(\frac{\partial v^{n}}{\partial y}\right)_n - \left(\frac{\partial v^{n}}{\partial y}\right)_s\right]\Delta x\right\},
\end{aligned} \quad (10)
$$

where u and v represent the x- and y-direction velocities, respectively, at the cell-center. U_e, U_w, V_n, and V_s are the introduced face-center velocities.

Notably, Eq. (10) is the discretized form of advection–diffusion equation for a general cell. We now describe how these equations are implemented for cells without domain interface, cells (D1) with domain interface, and cells (D2) with domain interfaces in sequence.

(i) *Cells without domain interface*

For the control volumes without domain interface, Eq. (10) is exactly the same as Eq. (3) of a single block domain, which has been outlined in the previous section. In such a situation, by using the second-order accurate central-difference method for spatial terms, the cell-center velocities and their gradients at the cell faces are computed from the adjacent velocities, i.e.,

$$u_e = (u_E + u_C)/2, \quad u_w = (u_W + u_C)/2, \quad u_n = (u_N + u_C)/2, \quad u_s = (u_S + u_C)/2, \quad (11)$$

$$v_e = (v_E + v_C)/2, \quad v_w = (v_W + v_C)/2, \quad v_n = (v_N + v_C)/2, \quad v_s = (v_S + v_C)/2, \quad (12)$$

and

$$
\begin{aligned}
&\left(\frac{\partial u}{\partial x}\right)_e - \left(\frac{\partial u}{\partial x}\right)_w = \frac{(u_E - 2u_C + u_W)}{\Delta x}, \quad \left(\frac{\partial v}{\partial x}\right)_e - \left(\frac{\partial v}{\partial x}\right)_w = \frac{(v_E - 2v_C + v_W)}{\Delta x}, \\
&\left(\frac{\partial u}{\partial y}\right)_n - \left(\frac{\partial u}{\partial y}\right)_s = \frac{(u_N - 2u_C + u_S)}{\Delta y}, \quad \left(\frac{\partial v}{\partial y}\right)_n - \left(\frac{\partial v}{\partial y}\right)_s = \frac{(v_N - 2v_C + v_S)}{\Delta y},
\end{aligned} \quad (13)
$$

(ii) *D1 cells with domain interface*

For those D1 cells with domain interface, as shown in Fig. 2(a), since u_E and v_E are not defined in the computational domain, the cell-center velocities at the east surface can not be computed by using Eqs. (11) and (12) directly. Here we need to compute the velocities at the ghost cell by introducing appropriate interpolations. In our study, a simple way devoted to calculate these interfacial ghost velocities is obtained by using average velocity of the occupied D2 cells instead, i.e.,

$$u_E = (u_1 + u_2 + u_3 + u_4)/4, \quad \text{and} \quad v_E = (v_1 + v_2 + v_3 + v_4)/4 \quad (14)$$

where subscript E denotes the ghost cell with respect to D1 (E being the eastern cell), as shown in Fig. 2(a), in which control volume E is occupied by four (1, 2, 3, and 4) D2 cells. Notably, in view of Eq. (14), the general form of the cell-surface velocities [Eqs. (11) and (12)] and their second-order accuracy could be preserved.

Note that, it is the block-interface processing which plays the essential part of the present nested-block formulation. In the general advection–diffusion equation (Eq. (10)), except u_e and v_e, which have been described in the previous paragraph, there are some other terms such as U_e, $(\partial u/\partial x)_e$, and $(\partial\ v/\partial x)_e$ located at the block-interface. Here U_e, $(\partial u/\partial x)_e$, and $(\partial v/\partial x)_e$ are used for computing the volume flux and the momentum flux, respectively, from D1 to D2 cells. Although there are varied interpolations available, we calculate the volume and momentum fluxes from D1 to D2 cells as follows:

$$U_e \times \Delta y_1 = U_{1w} \times \Delta y_2 + U_{2w} \times \Delta y_2 \quad (15)$$

$$
\begin{aligned}
&\left(\frac{\partial u}{\partial x}\right)_e \times \Delta y_1 = \left(\frac{\partial u}{\partial x}\right)_{1w} \times \Delta y_2 + \left(\frac{\partial u}{\partial x}\right)_{2w} \times \Delta y_2 = \frac{u_1 - u_{1W}}{\Delta x_2} \times \Delta y_2 + \frac{u_2 - u_{2W}}{\Delta x_2} \times \Delta y_2, \\
&\left(\frac{\partial v}{\partial x}\right)_e \times \Delta y_1 = \left(\frac{\partial v}{\partial x}\right)_{1w} \times \Delta y_2 + \left(\frac{\partial v}{\partial x}\right)_{2w} \times \Delta y_2 = \frac{v_1 - v_{1W}}{\Delta x_2} \times \Delta y_2 + \frac{v_2 - v_{2W}}{\Delta x_2} \times \Delta y_2,
\end{aligned} \quad (16)
$$

where Δy_1 and Δy_2 denote the vertical lengths of D1 and D2 cells, respectively. It may be carefully noted that the use of Eqs. (15) and (16) ensures mass and the momentum diffusion fluxes through the D1 and D2 domain-interface and vice versa is consistent.

(iii) *D2 cells with domain interface*

For the D2 cells with domain interface, as shown in Fig. 2(b), since u_W, and v_W are not defined in the computational domain, the general form of the cell-face velocities [Eq. (12)] cannot be applied directly, and we need to compute these terms by introducing second-order accurate interpolations. Similar to the ghost D1 cells, these ghost velocities are obtained by using average value of adjacent velocities, i.e.,

$$u_W = u_G = (u_1 + u_2 + u_3 + u_4)/4, \quad \text{and} \quad v_W = v_G = (v_1 + v_2 + v_3 + v_4)/4, \tag{17}$$

where $u_1 = u_{1C}$, $u_2 = (u_{1C} + u_{1E})/2$, $u_3 = (u_{1C} + u_{1E} + u_{1NE} + u_{1N})/4$, $u_4 = (u_{1C} + u_{1N})/2$, and $v_1 = v_{1C}$, $v_2 = (v_{1C} + v_{1E})/2$, $v_3 = (v_{1C} + v_{1E} + v_{1NE} + v_{1N})/4$, $v_4 = (v_{1C} + v_{1N})/2$, in which the subscripts *1C, 1E, 1N*, and *1NE* denote variables corresponding to D1, D1-eastern, D1-northern, and D1-north-east cells, respectively, as shown in Fig. 2 (b). Here, u_{1E}, v_{1E}, u_{1NE}, and v_{1NE} are not defined in our computational domains, and these velocities are obtained again from Eq. (14). In other words, coarse–fine interfacial velocities are now obtained by interpolating the corresponding nearby D1 cell values.

2.2.2. Computing U^* and V^*

Following the advection–diffusion step, the intermediate face-center velocities U^* and V^* are computed by interpolating the intermediate cell-center velocity.

For fine-fine or coarse–coarse non-inter-domain cell-face we use

$$U^*_{i,j} = \left(u^*_{i,j} + u^*_{i+1,j}\right)/2, \quad \text{and} \quad V^*_{i,j} = \left(v^*_{i,j} + v^*_{i+1,j}\right)/2. \tag{18}$$

At the fine-coarse domain interface, as shown in Fig. 2(a), U_e is obtained from Eq. (15) with

$$U_e \times \Delta y_1 = U_{1w} \times \Delta y_2 + U_{2w} \times \Delta y_2.$$

U_{1w} and U_{2w} are computed by interpolating the intermediate cell-center velocity, i.e.,

$$U_{1w} = (u_{1W} + u_1)/2 \quad \text{and} \quad U_{2w} = (u_{2W} + u_2)/2, \tag{19}$$

where subscripts 1*W* and 2*W* corresponds to those of ghost D2 cells. Consequently, u_{1W} and u_{2W} are obtained again from the ghost cell equation [Eq. (17)].

2.2.3. Solving pressure

The scalar form of the pressure discretization equation (Eq. (6)) can be written as

$$\left[\left(\frac{\partial p^{n+1}}{\partial x}\right)_e - \left(\frac{\partial p^{n+1}}{\partial x}\right)_w\right]\Delta y + \left[\left(\frac{\partial p^{n+1}}{\partial y}\right)_n - \left(\frac{\partial p^{n+1}}{\partial y}\right)_s\right]\Delta x = \frac{1}{\Delta t}\left[(U^*_e - U^*_w)\Delta y + (V^*_n - V^*_s)\Delta x\right]. \tag{20}$$

Notably, for cells without domain interface, using the second-order accurate central-difference method, pressure gradients at cell surfaces are computed from

$$\begin{aligned} &(\partial p/\partial x)_e = (p_E - p_C)/\Delta x, \quad (\partial p/\partial x)_w = (p_C - p_W)/\Delta x, \\ &(\partial p/\partial y)_n = (p_N - p_C)/\Delta y, \quad (\partial p/\partial y)_s = (p_C - p_S)/\Delta y. \end{aligned} \tag{21}$$

For D1 cells with domain interface, as shown in Fig. 2(a), the pressure gradient term at the interface, $(\partial p/\partial x)_e$, need to be processed rather than computed by using Eq. (21) directly. In a manner similar to the momentum diffusions processing, the pressure gradient is thereby obtained from the diffusion consistency equation,

$$\left(\frac{\partial p}{\partial x}\right)_e \times \Delta y_1 = \left(\frac{\partial p}{\partial x}\right)_{1w} \times \Delta y_2 + \left(\frac{\partial p}{\partial x}\right)_{2w} \times \Delta y_2 = \frac{p_1 - p_{1W}}{\Delta x_2} \times \Delta y_2 + \frac{p_2 - p_{2W}}{\Delta x_2} \times \Delta y_2. \tag{22}$$

For those D2 cells with domain interface (Fig. 2(b)), since the ghost cell pressure p_W is not defined in the computational domain, the pressure gradient at the west surface cannot be computed by using Eq. (21) directly. Therefore, we needed to compute the ghost pressure p_W. In such a situation, similar to the advection–diffusion processing, the implemented interpolations are listed below:

$$p_W = (p_1 + p_2 + p_3 + p_4)/4, \tag{23}$$

where $p_1 = p_{1C}$, $p_2 = (p_{1C} + p_{1E})/2$, $p_3 = (p_{1C} + p_{1E} + p_{1NE} + p_{1N})/4$, and $p_4 = (p_{1C} + p_{1N})/2$. Note that, p_{1E}, and p_{1NE} are not defined in our computational domains, and these pressures once again are obtained from the ghost sell equation

$$p_{1E} = (p_C + p_E + p_S + p_{SE})/4. \tag{24}$$

The subscript 1*E* here refers to the ghost cell with respect to the D1 cells (1*E* being the eastern cell), as shown in Fig. 2(b), for which the control volume is occupied by the four D2 cells (herein denoted by the subscripts *C, E, S*, and *SE*).

2.2.4. Velocity correction

Once the pressure is obtained by solving pressure equation, both the cell-center and face-center velocities are updated separately as:

$$\begin{aligned} &u^{n+1} = u^* - \Delta t(\partial p^{n+1}/\partial x)_{cc} \quad \text{and} \quad v^{n+1} = v^* - \Delta t(\partial p^{n+1}/\partial y)_{cc}, \\ &U^{n+1} = U^* - \Delta t(\partial p^{n+1}/\partial x)_{fc}, \quad \text{and} V^{n+1} = V^* - \Delta t(\partial p^{n+1}/\partial y)_{fc}. \end{aligned}$$

At this point it may be appropriate to describe how exactly the cell-center velocities be corrected for different cases, such as, for cells without domain interface, in D1 cells with domain interface, in D2 cells with domain interface; and how face-center velocities be corrected at the non-domain interface and for the domain interfaces in sequence.

(i) *Cell-center velocity correction for cells without domain interface*

For instance, with reference to Fig. 1, the velocity correction at the cell center is now conducted as

$$u^{n+1} = u^* - \Delta t \times (\partial p/\partial x)_C = u^* - \Delta t \times (p_E - p_W)/(2\Delta x),$$
$$v^{n+1} = v^* - \Delta t \times (\partial p/\partial y)_C = v^* - \Delta t \times (p_N - p_S)/(2\Delta y). \quad (25)$$

(ii) *Cell-center velocity correction in D1 cells with domain interface*

Notably, for those D1 cells with domain interface, as shown in Fig. 2 (a), since p_E is not defined in the computational domain, the cell-center velocity cannot be corrected by using Eq. (25) directly. In such a situation we first need to compute p_E by introducing an appropriate interpolation. Here the ghost pressure used is the averaged occupied D2 pressures [Eq. (24)], i.e.,

$$p_E = (p_1 + p_2 + p_3 + p_4)/4.$$

(iii) *Cell-center velocity correction in D2 cells with domain interface*

For D2 cells with domain interface, as shown in Fig. 2(b), since p_W is not defined in the computational domain, the pressure gradient at the east surface can not be computed by using Eq. (25), and we therefore need to compute p_W first. Similar to those described in the previous paragraph, the ghost pressure, as obtained from Eq. (23), now becomes,

$$p_W = (p_1 + p_2 + p_3 + p_4)/4.$$

(iv) *Velocity correction at non-inter-domain face-center*

For the fine-fine or the coarse–coarse non-inter-domain face-center we implement the general form of velocity correction at the face-center as:

$$U^{n+1} = U^* - \Delta t \times (\partial p^{n+1}/\partial x)_e = U^* - \Delta t \times (p_E - p_C)/\Delta x,$$
$$V^{n+1} = V^* - \Delta t \times (\partial p^{n+1}/\partial y)_n = V^* - \Delta t \times (p_N - p_C)/\Delta y. \quad (26)$$

(v) *Velocity correction at domain interface*

At the fine-coarse cell interface, as shown in Fig. 2(a), the coarse cell face-center velocity, U_e^{n+1}, is obtained from the consistency equation [Eq. (15)], i.e.,

$$U_e^{n+1} \times \Delta y_1 = U_{1w}^{n+1} \times \Delta y_2 + U_{2w}^{n+1} \times \Delta y_2$$

where $U_{1,w}^{n+1}$ and $U_{2,w}^{n+1}$ are corrected by Eq. (25) using the fine mesh domain ghost velocity formulation Eq. (17).

2.3. Immersed boundary method

In this study, a simple concept of immersed boundary (IB) method is adopted for the numerical simulation of bluff body wake flows by introducing a distributed body force in the Navier–Stokes equations. The governing equations for the flow are solved everywhere in the computational domain including cells which are occupied by the solid body. The distributed body force is then determined at every time step of the iteration by adjusting the value of velocity in those solid cells to match the prescribed velocity of the solid body (whenever relevant). Notably, a similar process has been adopted in Ravoux et al. [29], and the authors referred it as the 'embedding method'. The underlying logic of the present IB method, as described below, is however slightly different from that of Ravoux et al. [29].

In this work, the presence of a solid object within the flow domain is represented by a locally active distributed body force in the Navier–Stokes equations, i.e.,

$$\frac{\partial \vec{u}}{\partial t} = -(\vec{u} \cdot \nabla)\vec{u} - \nabla p + \frac{1}{Re}\Delta \vec{u} + \vec{f}.$$

Here, the distributed force is defined as $\vec{f} \equiv \left((\vec{u} \cdot \nabla)\vec{u} + \nabla p - \frac{1}{Re}\Delta \vec{u}\right) \times \phi$, in which ϕ represents the volume-fraction of the solid body [30] within a cell. Note that, $\phi = 1$ for a structure cell (cell occupied entirely by the solid body), $\phi = 0$ for a fluid cell (cell occupied entirely by fluid), and $0 < \phi < 1$ for an interface cell (cell partially occupied by solid body and partially by fluid), respectively. As a consequence, the body force which acts on the fluid satisfy the condition that the velocity in cells occupied by the solid body should match the prescribed velocity of the solid body.

This IB method is subsequently implemented to the solver together with the fractional-step procedure. Although there are other IB methods available, such as those of Lai and Peskin [24] and Ye et al. [21], the present method is not only easy to use, but also found to be quite suitable for our nested grid formulation.

3. Numerical experiments

In order to demonstrate the performance of the proposed nested-block formulation, herewith we extensively examine several of our computed complex flow phenomena of recent interest, and validate them with the well established results available in the literature.

3.1. Lid-driven cavity flow and pearson vortex problems

The test cases we consider in this sub-section are lid-driven cavity flow and Pearson vortex problems. In the first case steady incompressible lid-driven flow in a square cavity is computed both by using the ordinary single-block method and the presently derived nested-block formulation. The two-dimensional lid-driven cavity flow at a Reynolds number $Re = 1000$ is considered here. Extensive simulations of the flow on various uniform meshes (the uniform grid system is referred to as the *one-block grid*, and denoted by B1 hereafter) of grid sizes 40 × 40 (this coarse grid G1 with $\Delta x = \Delta y = 0.025$ is referred as grid1), 80 × 80 (the finer grid G2 with $\Delta x = \Delta y = 0.0125$ is referred as grid2), and 160 × 160 (the finest grid G3 with $\Delta x = \Delta y = 0.00625$ is referred as grid3) are specially conducted for the relative comparison of the present nested two-block results. For convenience, the uniform mesh systems of grid sizes 40 × 40, 80 × 80, and 160 × 160 are named as B1G1, B1G2, and B1G3, respectively, while the related two-block grid systems are denoted by B2G2, B2G3, etc. The block numbers, grid sizes, and the refined grid region of the B2G3 grid system are shown in Fig. 3. Notably, the B2G3 grid system (Fig. 3) consists of two blocks. In the central area of $(0.05 \leqslant x \leqslant 0.95) \times (0.05 \leqslant y \leqslant 0.95)$ the grid length scale is the same as those of the G2 grids, while in the outlying area of $x \leqslant 0.05$ or $x \geqslant 0.95$ and $y \leqslant 0.05$ or $y \geqslant 0.95$, the grid length scale is the same as those of the G3 grids. For clarity, in Fig. 3 the simulated vorticity contours for the cavity flow problem (for $Re = 1000$) with two single block grids, G2, G3, and two nested-blocks grids, B2G2, B3G3, are plotted and superimposed, and notably, the depicted vorticity contour plots are seen to coincide exactly, and they remained quite smoothly distributed across the block-interfaces. This (Fig. 3), first, clearly shows the present nested-block formulation works quite well for such a problem. For the sake of enhanced clarity, in Fig. 4(a) we present the computed corresponding u velocity profile along the cavity centerline $x = 0.5$. In absence of suitable analytical solution to the problem, Ghia's [31] results are adopted here, as the benchmark data, to validate our simulated findings. It is important to note that, in general, the numerical results obtained from B1G2, B1G3, B2G3 grid systems, and those of Ghia's are very consistent [as demonstrated in Fig. 4(a)]. Notably, Fig. 4(b) presents a more focused view of the velocity component u between $y = 0.05$ and 0.35, where biggest reversed u velocity occurs. To be explicit [and as one may find from Fig. 4(b)], the maximum difference between the two solutions of u component (profile) obtained with the nested B2G3 grid and the finest uniform grid B1G3 is about 0.002; while such difference of u between B1G2 and B1G3 grids appears to be 0.01. Fig. 4(c) displays the computed corresponding v velocity profile along the centerline $y = 0.5$. Again, it is very encouraging to note [from Fig. 4(c)] that the presently simulated results, as obtained from B1G2, B1G3, B2G3, and those of Ghia's [31] are quite similar. In Fig. 4(d) we extract a more detailed view of the velocity component v between $x = 0.84$ and 0.97, where biggest negative v velocity occurs. Notably, the maximum difference (Fig. 4(d)) between the two solutions of v component, as obtained with the nested B2G3 grid and with the finest uniform grid B1G3, is about 0.003; while such difference of v, as obtained from B1G2 and B1G3 grids, appeared close to 0.02. This clearly displays the efficiency of the presently developed nested grid method. In addition, notably, our simulated results for u (Fig. 4(a) and (b)) and v (Fig. 4(c) and (d)) are also found to agree quite well with the recently published results of Chiu et al. [32]. Therefore, based on observations made from Figs. 4 and 5, appropriate it may be to comment here that, the consistencies of our computational results obtained from the presently adopted nested-block system B2G3 and those of Ghia's [31] and Chiu et al. [32] are quite encouraging. Since we used a very refined grid around the outlying area (Fig. 4(a)–(d)) magnificent

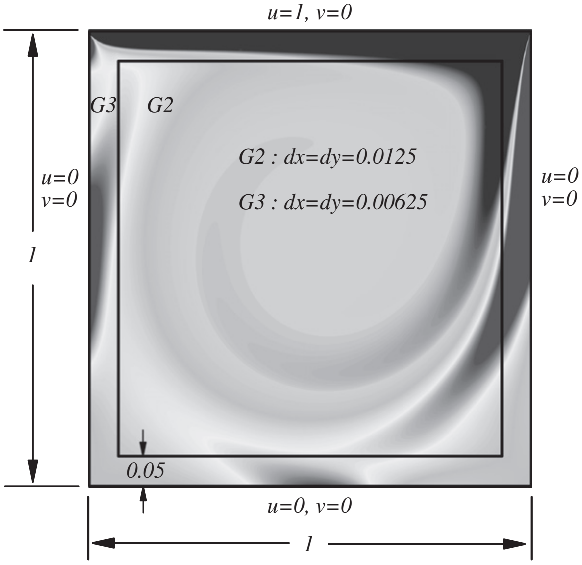

Fig. 3. Details of computational domain, boundary conditions, simulated vorticity contours, and the associated two-block nested grid system B2G3 for the lid-driven cavity flow at $Re = 1000$.

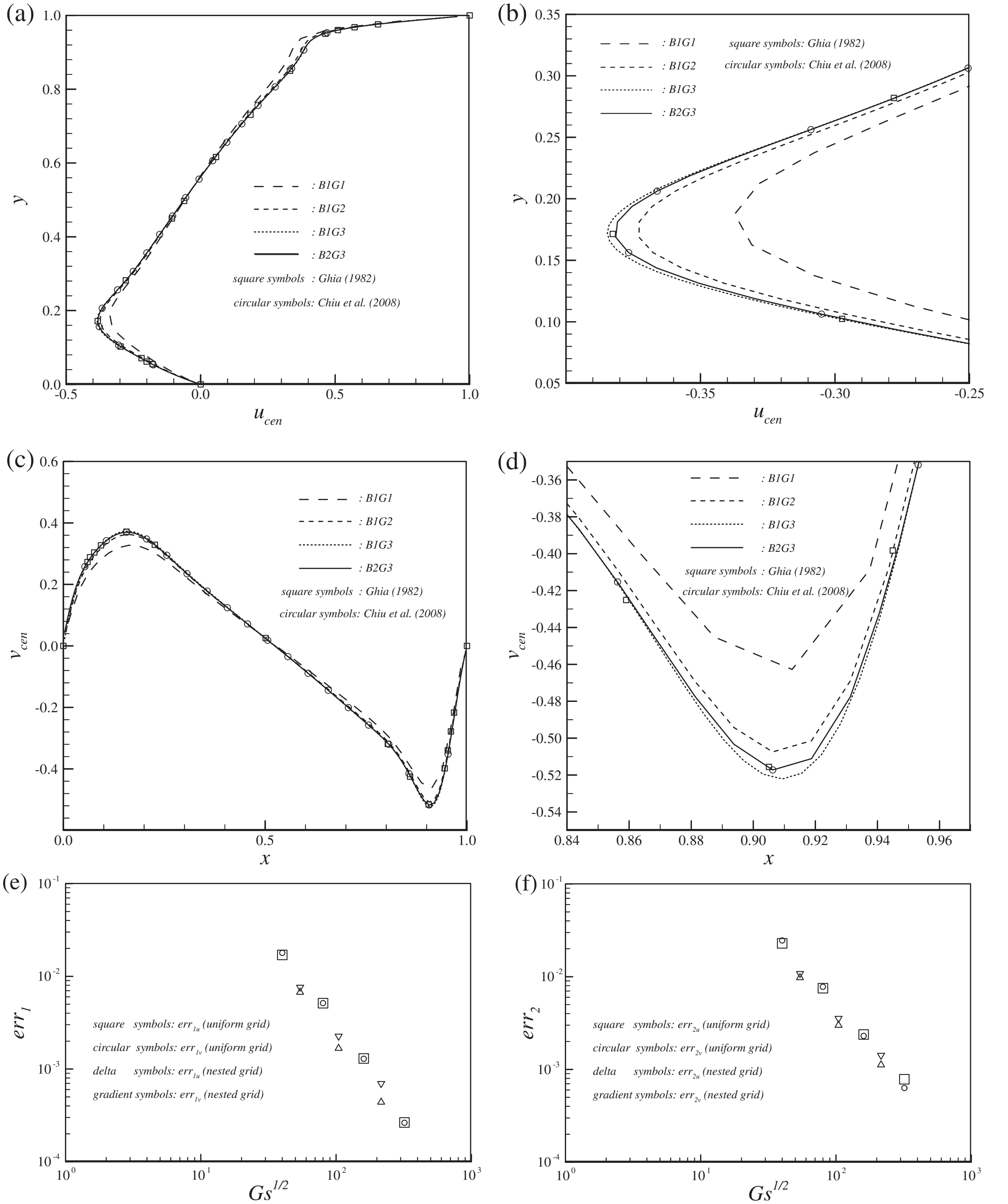

Fig. 4. Simulated velocity profiles and error norms for the square lid-driven cavity flow at *Re* = 1000. (a) Velocity component *u* on the vertical centerline. (b) Enlarged view around largest reversed *u*. (c) Velocity component *v* on the horizontal centerline. (d) Enlarged view around largest negative *v*. (e) Depicted err_1 norm shows the second-order global convergence of *u* and *v* velocities. (f) Depicted second-order global convergence of velocity err_2 norm. (g) Depicted pressure convergences for uniform and nested grids. (h) Convergence tolerance history for various uniform and nested grids.

improvement of resolution has been achieved. Such results clearly show that the present nested-block method works very effectively for the investigated flow.

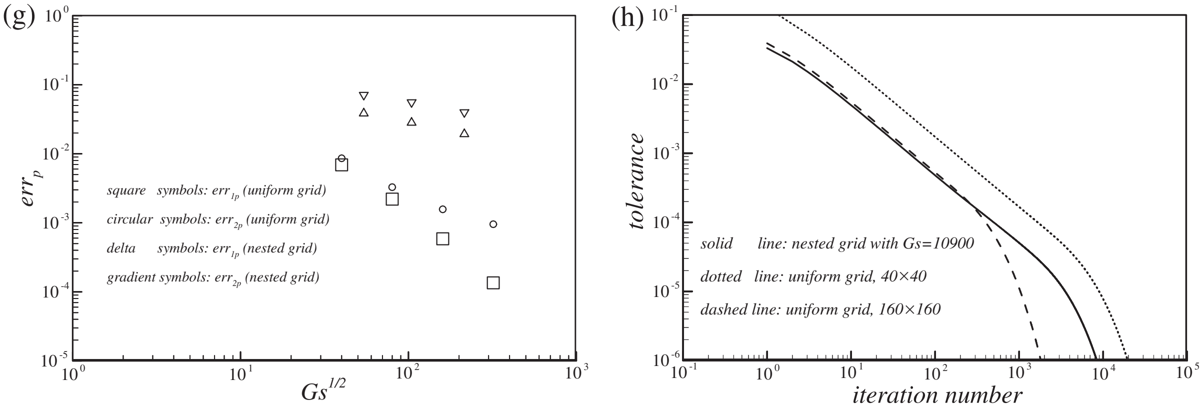

Fig. 4 (*continued*)

In order to further demonstrate spatial accuracy of the underlying method, we now conduct a series of grid refinement study for the steady incompressible lid-driven flow in a square cavity at $Re = 1000$. We simulate the flows using our solver on five different uniform meshes 40 × 40, 80 × 80, 160 × 160, 320 × 320, 640 × 640, and three two-block grid systems with grid size (G_s) of 2932, 10,900, and 43,600. In absence of suitable analytical solution to the problem, the numerical results with the finest grid, 640 × 640, is considered as the benchmark solution. Following this, the *err1* and *err2* norms of the global error (both for the uniform and the nested grids) have been computed as:

$$err_1 = \sum_{i,j=1}^{G_s} \left| \phi_{i,j}^{num} - \phi_{i,j}^{640} \right| \Delta x_i \times \Delta y_j,$$

$$err_2 = \left[\sum_{i,j=1}^{G_s} \left(\phi_{i,j}^{num} - \phi_{i,j}^{640} \right)^2 \times \Delta x_i \times \Delta y_j \right]^{1/2}.$$

Here G_s is the grid size; and while calculating the error norms the area of the grids $\Delta x_i \times \Delta y_j$ is used as weighting factor. This ensures that the computed errors have the same 'scaling' regardless of whether they are computed on a uniform or locally-refined grid. The err_1 and err_2 norms of both the velocity components u, v, and the pressure p are calculated and such results are depicted in Fig. 4(e)–(g) and also listed in Table 1a. Notably, a similar test procedure has been adopted in Ref. [33]. It is noted from Fig. 4(e)–(g) that the velocities u, v appear to be second-order accurate in a global measure like the discrete err_1/err_2 norms for both uniform and nested grids, while the pressure p appears to be second-order accurate for uniform grids and nearly first order accurate for nested grids. The possible reason for the consistency/discrepancy of the order of accuracy may be the following. Because the present scheme uses linear interpolation at coarse–fine interface, second-order like accuracy is maintained for interpolated quantities; however ghost cell values already posses $O(h^2)$ error, gradients will generally have $O(h)$ errors and viscous terms will normally posses $O(1)$ errors at coarse–fine interfaces. $O(h)$ error may thereby be incurred with the approach to computing the discrete divergence at coarse–fine interfaces (e.g., conservatively coursing pressure gradients and velocities from fine levels onto coarse levels while using the discretization Eq. (6)). Therefore, in the proposed nested grid method, while the linear interpolation scheme used to compute ghost cell values maintains second-order global accuracy, at the coarse–fine interface the scheme might behave as first order accurate [16]. This way, it's possible to devise an adaptive scheme that maintains second-order accuracy at places away from the coarse–fine interface and nearly first order accurate at the interface, but overall the scheme yields second-order convergence rates in global error measures (such as discrete err_1/err_2 norms).

We may mention here that, for all the computational cases presented in the paper, the convergence tolerances for viscous and pressure solvers are set as 10^{-13} and 10^{-6}, respectively. It required less than 20 iterations for the viscous solution (in all the cases); however many more iterations (depending on grid size) were needed to solve for pressure. Notably, the overall efficiency of any fractional-step method for solving the incompressible Navier–Stokes equations depends on the approach adopted to solve the pressure correction equation. In the present study the pressure Poisson equation is solved with the line-SOR smoother which leads to a numerical solution that scales almost linearly with the number of grid points. The ability to employ easily such methods is a key advantage of the current nested grid approach over unstructured grid approaches. To be specific, the convergence tolerances of pressure solver with two uniform grids, 40 × 40, 160 × 160, and a nested grid with $G_s = 10{,}900$ for the lid-driven cavity flow at $Re = 1000$ are extracted and plotted in Fig. 4(h). It is seen from Fig. 4(h) that the iterations numbers for solution of the pressure Poisson equation scales almost linearly with the number of grid points. In contrast, for unstructured grids one has to either resort to multigrid methods or Krylov subspace based methods (such as conjugate gradient or GMRES) to provide good performance.

As an extended study concerning the spatial accuracy of the presently developed nested grid method, we now conduct simulations for another test problem which is amenable to the analytic solution. The test case is that of the Pearson vortex

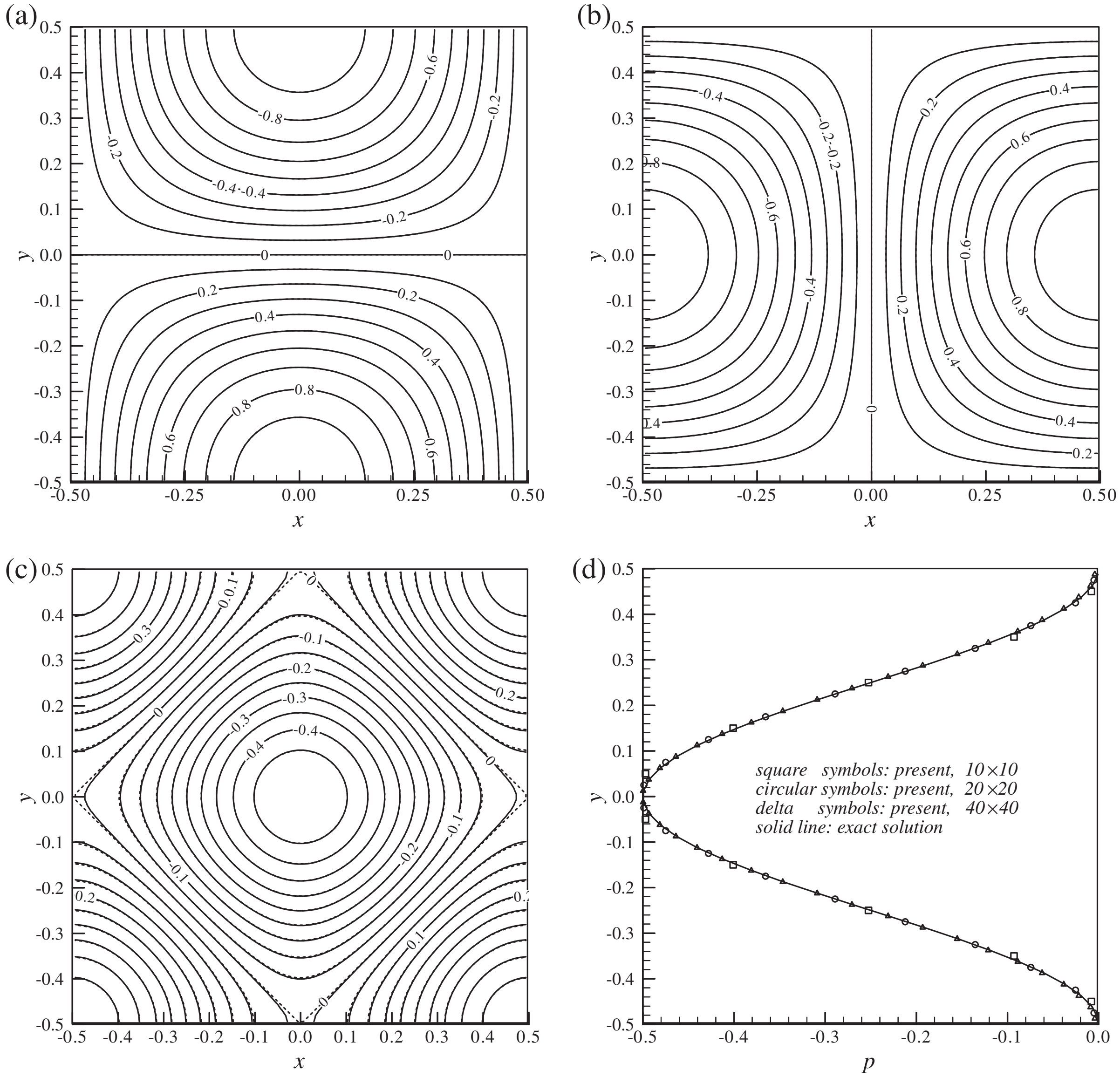

Fig. 5. Code validation for the Pearson vortex problem. (a) $u(x,y)$; (b) $v(x,y)$; (c) $p(x,y)$. The solid lines in (a–c) are exact solutions, and dotted lines denote the corresponding solutions obtained with 40 × 40 grids. (d) grid convergence test on $p(x = 0,y)$.

Table 1a
Various one-block and two-block grid systems and corresponding error norms for square lid-driven cavity flow at $Re = 1000$.

Case No.	Grid		Grid size (Gs)	err_{1u}	err_{1v}	err_{2u}	err_{2v}	err_{1p}	err_{2p}
	Magnitude	Number							
B1G5	$\Delta h = 1/640$	409,600	409,600	0	0	0	0	0	0
B1G4	$\Delta h = 1/320$	102,400	102,400	2.65×10^{-4}	2.63×10^{-4}	7.83×10^{-4}	6.30×10^{-4}	1.35×10^{-4}	9.54×10^{-4}
B1G3	$\Delta h = 1/160$	25,600	25,600	1.30×10^{-3}	1.29×10^{-3}	2.38×10^{-3}	2.30×10^{-3}	5.87×10^{-4}	1.57×10^{-3}
B1G2	$\Delta h = 1/80$	6400	6400	5.13×10^{-3}	5.14×10^{-3}	7.52×10^{-3}	7.79×10^{-3}	2.20×10^{-3}	3.28×10^{-3}
B1G1	$\Delta h = 1/40$	1600	1600	1.70×10^{-2}	1.79×10^{-2}	2.29×10^{-2}	2.46×10^{-2}	6.91×10^{-3}	8.61×10^{-3}
B2G4	$\Delta h = 1/320$ $\Delta h = 1/160$	24,000 19,600	43,600	4.39×10^{-4}	6.97×10^{-4}	1.12×10^{-3}	1.42×10^{-3}	1.93×10^{-2}	3.99×10^{-2}
B2G3	$\Delta h = 1/160$ $\Delta h = 1/80$	6000 4,900	10,900	1.67×10^{-3}	2.26×10^{-3}	2.99×10^{-3}	3.55×10^{-3}	2.82×10^{-2}	5.57×10^{-2}
B2G2	$\Delta h = 1/80$ $\Delta h = 1/40$	1776 1156	2932	6.73×10^{-3}	7.61×10^{-3}	9.76×10^{-3}	1.08×10^{-2}	3.84×10^{-2}	7.12×10^{-2}

Table 1b
Various one-block and two-block grid systems and the corresponding error norms for Pearson vortex flow at *Re* = 10.

Case No.	Grid		Grid size (Gs)	err_{1u}	err_{1v}	err_{2u}	err_{2v}	err_{1p}	err_{2p}
	Magnitude	Number							
B1G5	$\Delta h = 1/160$	25,600	25,600	3.05×10^{-3}	3.03×10^{-3}	3.94×10^{-3}	3.92×10^{-3}	1.28×10^{-1}	1.67×10^{-1}
B1G4	$\Delta h = 1/80$	6400	6400	3.20×10^{-3}	3.19×10^{-3}	4.01×10^{-3}	4.00×10^{-3}	1.31×10^{-1}	1.79×10^{-1}
B1G3	$\Delta h = 1/40$	1600	1600	3.33×10^{-3}	3.31×10^{-3}	4.15×10^{-3}	4.13×10^{-3}	1.38×10^{-1}	2.10×10^{-1}
B1G2	$\Delta h = 1/20$	400	400	4.27×10^{-3}	4.24×10^{-3}	5.65×10^{-3}	5.63×10^{-3}	1.83×10^{-1}	2.76×10^{-1}
B1G1	$\Delta h = 1/10$	100	100	9.20×10^{-3}	9.19×10^{-3}	1.09×10^{-2}	1.09×10^{-2}	2.40×10^{-1}	3.23×10^{-1}
B2G5	$\Delta h = 1/160$	6000	10,900	4.10×10^{-3}	4.12×10^{-3}	4.59×10^{-3}	4.57×10^{-3}	2.30×10^{-1}	2.60×10^{-1}
	$\Delta h = 1/80$	4900							
B2G4	$\Delta h = 1/80$	2800	3700	7.05×10^{-3}	7.03×10^{-3}	6.06×10^{-3}	6.03×10^{-3}	3.94×10^{-1}	3.53×10^{-1}
	$\Delta h = 1/40$	900							
B2G3	$\Delta h = 1/40$	816	1012	9.50×10^{-3}	9.43×10^{-3}	7.89×10^{-3}	7.87×10^{-3}	4.68×10^{-1}	4.33×10^{-1}
	$\Delta h = 1/20$	196							
B2G2	$\Delta h = 1/20$	256	292	1.88×10^{-2}	1.88×10^{-2}	1.42×10^{-2}	1.43×10^{-2}	5.68×10^{-1}	5.13×10^{-1}
	$\Delta h = 1/10$	36							

Table 1c
Various one-block and two-block grid systems and the corresponding error norms for Pearson vortex flow at *Re* = 100.

Case No.	Grid		Grid size (Gs)	err_{1u}	err_{1v}	err_{2u}	err_{2v}	err_{1p}	err_{2p}
	Magnitude	Number							
B1G5	$\Delta h = 1/160$	25,600	25,600	3.92×10^{-4}	3.90×10^{-4}	5.03×10^{-4}	4.96×10^{-4}	1.21×10^{-2}	1.62×10^{-2}
B1G4	$\Delta h = 1/80$	6400	6400	4.28×10^{-4}	4.21×10^{-4}	6.02×10^{-4}	5.89×10^{-4}	1.27×10^{-2}	1.89×10^{-2}
B1G3	$\Delta h = 1/40$	1600	1600	6.18×10^{-4}	6.10×10^{-4}	8.58×10^{-4}	8.52×10^{-4}	1.52×10^{-2}	2.41×10^{-2}
B1G2	$\Delta h = 1/20$	400	400	1.65×10^{-3}	1.64×10^{-3}	2.06×10^{-3}	2.07×10^{-3}	2.16×10^{-2}	2.98×10^{-2}
B1G1	$\Delta h = 1/10$	100	100	5.65×10^{-3}	5.63×10^{-3}	7.11×10^{-3}	7.14×10^{-3}	2.72×10^{-2}	3.45×10^{-2}
B2G5	$\Delta h = 1/160$	6000	10,900	5.84×10^{-4}	5.79×10^{-4}	7.32×10^{-4}	7.16×10^{-4}	2.27×10^{-2}	2.68×10^{-2}
	$\Delta h = 1/80$	4900							
B2G4	$\Delta h = 1/80$	2800	3700	1.39×10^{-3}	1.36×10^{-3}	1.32×10^{-3}	1.30×10^{-3}	3.87×10^{-2}	3.77×10^{-2}
	$\Delta h = 1/40$	900							
B2G3	$\Delta h = 1/40$	816	1012	3.75×10^{-3}	3.80×10^{-3}	3.03×10^{-3}	3.09×10^{-3}	4.15×10^{-2}	4.15×10^{-2}
	$\Delta h = 1/20$	196							
B2G2	$\Delta h = 1/20$	256	292	1.32×10^{-2}	1.34×10^{-2}	9.36×10^{-3}	9.58×10^{-3}	5.87×10^{-2}	5.08×10^{-2}
	$\Delta h = 1/10$	36							

problem [34]. In a square domain of length 1, we solve the flow in primitive variables subject to the boundary values of the velocity and initial condition as given below:

$$u = -\cos \pi x \sin \pi y \; e^{-2t/Re}$$
$$v = \sin \pi x \cos \pi y \; e^{-2t/Re}$$

The analytical expression for the pressure is given by

$$p = -\frac{1}{4}(\cos 2\pi x + \cos 2\pi y)e^{-4t/Re}.$$

Here we performed computations under the conditions $\Delta t = 5 \times 10^{-5}$ and *Re* = 10, 100, and 1000, respectively. The grid systems we consider in this validation test are: uniform grids with resolution 10 × 10, 20 × 20, 40 × 40, 80 × 80, 160 × 160, and two-block nested grids with grid sizes $G_s = 292$, 3700, and 10,900. After 200 time steps the corresponding converged results obtained at $t = 0.01$ showed increasing agreement with the analytical solutions, as depicted in Fig. 5(a)–(d). For this case, we also conducted computations on continuously refined grids and computed corresponding err_1/err_2 norms. Such results are extensively presented in Tables 1a–1d and Fig. 6(a)–(d). Notably, as Fig. 6(a)–(d) reveal, at *Re* = 1000 both for uniform as well as nested grids, the computed rate of convergences of err_1/err_2 appeared about 2 for the velocity components u and v. At *Re* = 100 the convergence rates of u and v became close to 1.6. However, such convergence rates are seen to get reduced to nearly 1.1 for the very low Reynolds number like *Re* = 10. Interestingly, convergence rates for the nested grid cases (Fig. 6(a) and (b)) seem to appear somewhat better than the corresponding uniform grid solutions. On the other hand, the convergence rates for err_1/err_2 of the pressure p (Fig. 6(c) and (d)) at all the simulated Reynolds numbers, *Re* = 1000, 100, 10, remained nearly first order both for uniform and the nested grid test cases. Note carefully that, at lower Reynolds numbers, even with fewer cells, the numerical solutions quickly attained the grid independent results for this test problem. The conducted convergence tests therefore imply that the proposed numerical scheme produces globally second-order accurate velocity field in the moderate to high Reynolds number range, however, at very low *Re* (*Re* ~ 10) the convergence rate becomes nearly first order.

Table 1d
Various one-block and two-block grid systems and the corresponding error norms for Pearson vortex flow at $Re = 1000$.

Case No.	Grid		Grid size (Gs)	err_{1u}	err_{1v}	err_{2u}	err_{2v}	err_{1p}	err_{2p}
	Magnitude	Number							
B1G5	$\Delta h = 1/160$	25,600	25,600	5.20×10^{-5}	4.92×10^{-5}	9.87×10^{-5}	9.24×10^{-5}	1.22×10^{-3}	1.74×10^{-3}
B1G4	$\Delta h = 1/80$	6400	6400	1.08×10^{-4}	1.06×10^{-4}	1.71×10^{-4}	1.70×10^{-4}	1.38×10^{-3}	2.28×10^{-3}
B1G3	$\Delta h = 1/40$	1600	1600	3.32×10^{-4}	3.36×10^{-4}	4.80×10^{-4}	4.84×10^{-4}	3.57×10^{-3}	4.42×10^{-3}
B1G2	$\Delta h = 1/20$	400	400	1.38×10^{-3}	1.36×10^{-3}	1.80×10^{-3}	1.80×10^{-3}	4.46×10^{-3}	5.66×10^{-3}
B1G1	$\Delta h = 1/10$	100	100	5.44×10^{-3}	5.38×10^{-3}	6.85×10^{-3}	6.88×10^{-3}	1.09×10^{-2}	1.42×10^{-2}
B2G5	$\Delta h = 1/160$	6000	10,900	2.04×10^{-4}	1.98×10^{-4}	2.76×10^{-4}	2.72×10^{-4}	2.29×10^{-3}	3.02×10^{-3}
	$\Delta h = 1/80$	4900							
B2G4	$\Delta h = 1/80$	2800	3700	8.54×10^{-4}	8.66×10^{-4}	7.68×10^{-4}	7.93×10^{-4}	3.72×10^{-3}	3.97×10^{-3}
	$\Delta h = 1/40$	900							
B2G3	$\Delta h = 1/40$	816	1012	3.11×10^{-3}	3.20×10^{-3}	2.54×10^{-3}	2.64×10^{-3}	5.10×10^{-3}	4.91×10^{-3}
	$\Delta h = 1/20$	196							
B2G2	$\Delta h = 1/20$	256	292	1.24×10^{-2}	1.28×10^{-2}	8.94×10^{-3}	9.18×10^{-3}	1.22×10^{-2}	1.04×10^{-2}
	$\Delta h = 1/10$	36							

3.2. Flow past a circular cylinder

While the benchmark solutions of lid-driven cavity flow and Pearson vortex problem allow us to confirm the global accuracy of the present nested-block method, here we validate it further by simulating different unsteady viscous flows past a circular cylinder (of diameter d) placed symmetrically in a planar channel of height h. Fig. 7(a) shows a schematic of the flow configuration that has been used for simulations with implemented parabolic inlet profile and the immersed boundary method. Notably, the two main parameters in this flow are the blockage ratio $\beta = d/h$, and the Reynolds number $Re = Q/\nu$, where Q is the inlet volume flux. A systematic numerical investigation of the flow over a parameter range has been presented by Chen et al. [35], and their results (and the definition of Re) have been used to further validate our simulations. Importantly, the presently investigated flow shares some common feature with the case of a cylinder immersed in a free stream; in particular, the vortex shedding behavior observed beyond a critical Reynolds number. The value of the critical Reynolds number, however, remains strongly dependent on the blockage ratio. Furthermore, the development of vortices is also significantly affected by the boundary layers that form on the channel walls. Thus, overall, this is a more complicated flow and a good test case for our nested grid simulation methodology.

We now first conduct a resolution study by solving the vortex shedding flow at $Re = 500$ with blockage ratio 0.2. A series of grid systems including one-block, two-block, and three-block grids are used for simulation of this flow. The resolution study included two parts, namely, the locally-refined-domain inference study and the grid independence study, which are separately conducted and results are collected/listed systematically in Tables 2 and 3. Computed results concerning the dimensionless dynamic forces acting on the cylinder (e.g., the mean drag force coefficient $\overline{C}_d$, the lift force amplitude C_l, and the vortex shedding frequency, i.e., Strouhal number St) for different uniform/non-uniform (single/nested) grid systems are listed in Table 2; where B1G1 denotes a one-block coarse uniform grid system with $\Delta x = \Delta y = 0.02h$, and B1G2 denotes a one block finer uniform grid system with $\Delta x = \Delta y = 0.01h$. Notably, the simulated data (Table 2) clearly suggest that, in the case of single block uniform grids the use of higher grid resolution is very much essential for an effective simulation of this class of flows, as solutions appear to depend strongly on the grid size (e.g., the difference of C_l, as obtained with the coarse B1G1 grid and the fine B1G2 grid (Table 2) is about 34.4%). However, in order to make sure that the presented results remain quite close to the grid independent solution, another single block simulation with 151,200 uniform grids (B1G2′) was subsequently conducted, and notably, now the difference of C_l with respect to B1G2 grid became 3.2%. On the other hand, differences of $\overline{C}_d$ between the coarse (B1G1) and the finer (B1G2) grids, and between B1G2 and B1G2' became about 6.8% and 0.3%, respectively. This clearly shows the G2 grids are fine enough to produce significantly accurate results. On the other hand, note carefully that, although the improvement of numerical results from B1G1 to B1G2 is significant, the B1G2 grid numbers are four times that of B1G1, and the required CPU time for B1G2 is more than six times that of B1G1; which undeniably suggests there is a need for local grid refinement. Now, in order to investigate the effect of local refinement domain inference on the simulated results, we conducted a series of computations using the nested two-block method by suitably placing the refinement area around the cylinder. For this, simulations in eight two-block grid systems, as denoted by B2D1 to B2D8 (see Table 2 for detail), are performed. The corresponding refined area for each grid system is shown in Fig. 7(b), and the grid numbers, including coarse (G1) and finer (G2) grids, are listed in Table 2. From the tabulated data (Table 2) it may be carefully noted that the employed grid refinement around the cylinder improves the numerical results significantly. For instance just with extra 6000 refined (G2) grids (e.g., B2D5 with 29,250 grids, Table 2), which is about 23% of the G1 grid number, the present nested two-block method can predict very closely resembled results to that of B1G2 (a single block grid with 105,000 nodes!). Notably, the differences of each of the computed $\overline{C}_d$ and C_l values, as obtained with B1G2 and B2D5 grids, now appear within 1.0%. This locally-refined-domain inference study, therefore, not only clearly demonstrates efficiency of the present nested-block method, but also strongly suggests that the local-refinement around the cylinder plays a very important role in an effective simulation.

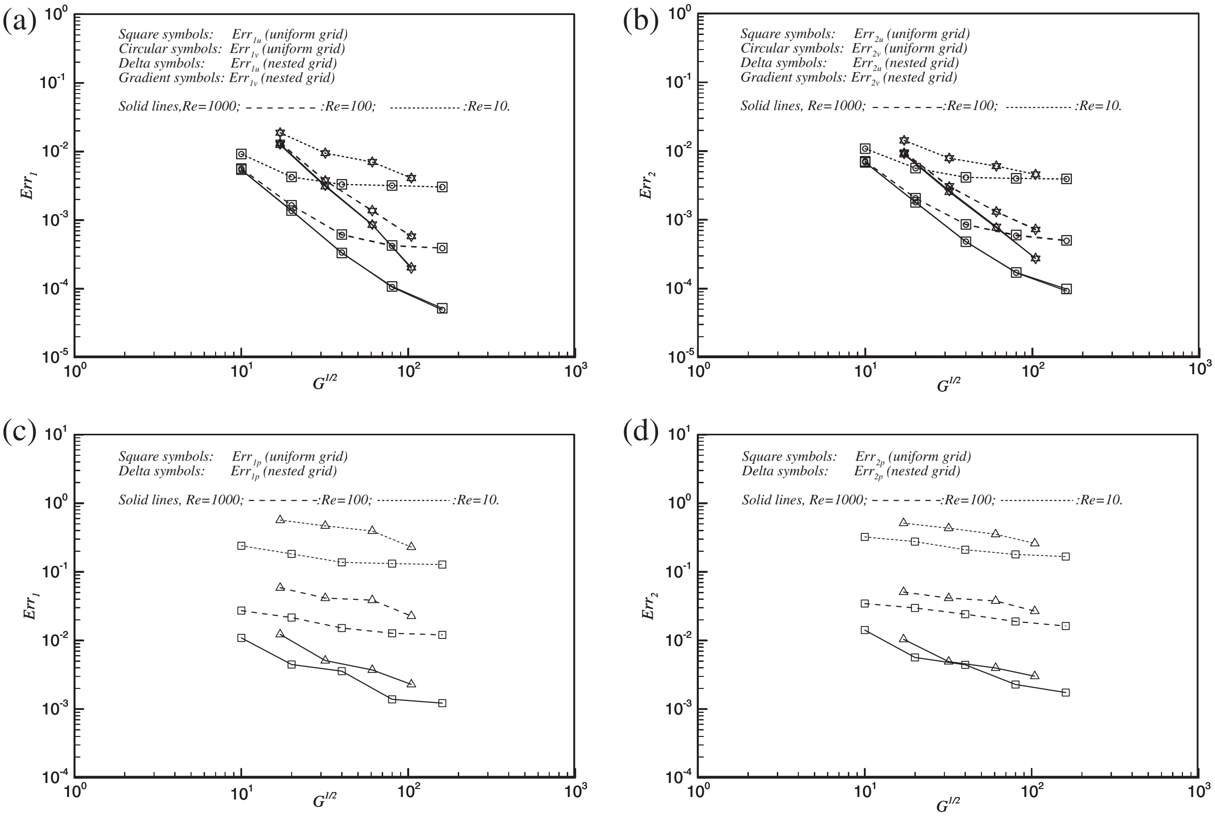

Fig. 6. Error norms for the simulated Pearson vortex problem. (a) err_1 norms for u and v; (b) err_2 norms for u and v; (c) err_1norms of p; (d) err_2 norms of p.

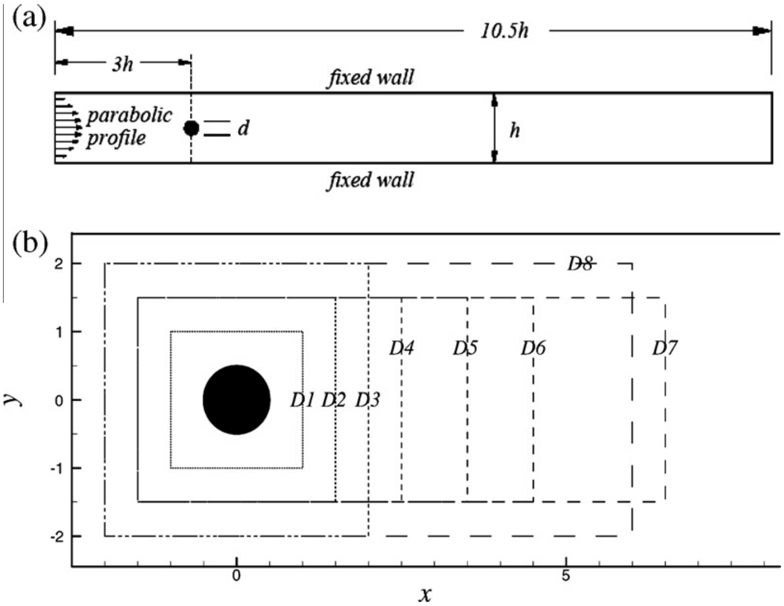

Fig. 7. (a) Physical domain for simulation of flow past a cylinder placed symmetrically in a planar channel. (b) Refinement domains of B2D1 to B2D8 for the flow past a cylinder placed in a channel.

Table 2
Computed results of mean drag coefficient $\overline{C}_d$, and the lift force amplitude C_l for a circular cylinder at $Re = 500$. Details of various one-block and two-block grid systems are also listed.

Grid system	$\overline{C}_d$	Difference to B1G2	C_l	Difference to B1G2	Gl ($\Delta h = 0.02h$), G2($\Delta h = 0.01h$)	Grids G1 + G2
B1G1	2.608	6.8%	0.510	34.4%	26,250 0	26,250
B1G2	2.797	0.0%	0.777	0.0%	0 105,000	105,000
B1G2′	2.806	0.3%	0.802	3.2%	0 151,200	151,200
B2D1	2.759	1.4%	0.687	11.6%	25,450 1600	27,050
B2D2	2.785	0.4%	0.760	2.2%	24,450 3600	28,050
B2D3	2.796	0.0%	0.779	0.3%	23,050 6400	29,450
B2D4	2.791	0.2%	0.768	1.2%	23,850 4800	28,650
B2D5	2.791	0.2%	0.771	0.8%	6000 6000	29,250
B2D6	2.791	0.2%	0.770	0.9%	22,650 7200	29,850
B2D7	2.791	0.2%	0.770	0.9%	21,450 9600	31,050
B2D8	2.792	0.2%	0.772	0.6%	19,850 12,800	32,650

Table 3
Computed results of mean drag coefficient $\overline{C}_d$, lift force amplitude C_l, and Strouhal number (St) for a circular cylinder at $Re = 500$. Details of various one-block, two-block, and three-block grid systems and the corresponding CPU times are also listed.

Grid system	$\overline{C}_d$ Difference to B3G3	C_l Difference to B3G3	St Difference to B3G3	G1: $\Delta h = 0.02h$, G2: $\Delta h = 0.01h$, G3: $\Delta h = 0.005h$	Grids G1 + G2 + G3	CPU (min)
B1G1	2.608 5.2%	0.510 39.1%	0.277 0.0%	26,250 0 0	26,250	174
B1G2	2.797 1.1%	0.777 7.2%	0.276 0.3%	0 105,000 0	105,000	800
B2D5 (B2G2)	2.791 1.3%	0.771 7.9%	0.277 0.0%	23,250 6000 0	29,250	227
B3G3	2.828 0.0%	0.837 0.0%	0.277 0.0%	22,650 4000 6400	33,050	300

Based on experience gained from the local-refined-domain inference study, we now construct a three-block nested grid system, B3G3, for simulating this vortex shedding flow with high resolution. In B3G3 grid system the computational domain is divided into three blocks, with suitably arranged uniform distribution of coarse (G1), finer (G2), and finest (G3) Cartesian grids, respectively; where G1 denotes a uniform grid with$\Delta x = \Delta y = 0.02h$, G2 denotes a uniform grid with $\Delta x = \Delta y = 0.01h$, and G3 denotes a uniform grid with $\Delta x = \Delta y = 0.005h$. Notably, the computed vorticity contours with this three-block grid system B3G3, together with the marked three blocks, are presented in Fig. 8. First, it is clearly seen from Fig. 8 that the vorticity contours remain smoothly distributed across those block-interfaces, which seem to ensure that the present nested-block method works perfectly well even for complex (highly reversed) flow situations. Second, from the extracted corresponding u and v velocity contours, and the pressure contours for this vortex shedding flow, as extensively depicted in Fig. 9(a)–(c), respectively, it becomes quite evident that such flow characteristics in the high resolution region can be predicted more smoothly than that in the low resolution region. For further improved clarity, the computed values of $\overline{C}_d$, C_l, St, together with (G1, G2, G3) grid numbers and CPU time for B3G3 are collected/listed in Table 3, and the corresponding values are compared to those obtained from B1G1, B1G2, and B2D5. All the above mentioned computations are carried out on a

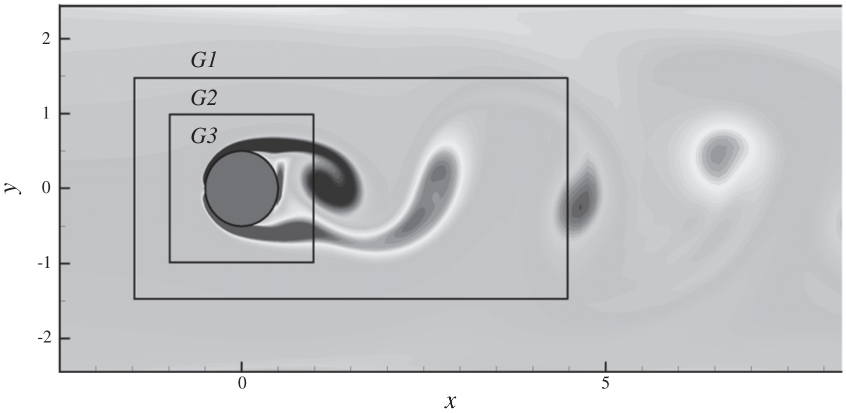

Fig. 8. Computed vorticity contour plots in the near wake of a circular cylinder at $Re = 500$ obtained with the three-block nested grid system B3G3.

personal computer with Pentium IV (3.4G) processor and Fortran 90 compiler. First, as evident from Table 3, with increased blocks the computational results are seen to greatly improve over those of B1G1, B1G2, and B2D5. Second, note carefully here that, while the computed results (Table 3) presented in the first and the second rows (with single block B1G1 and B1G2 grids of sizes 26,250 and 105,000, respectively) quite expectedly suggest the need for higher grid resolution in correctly simulating such a complex flow, interestingly, the two (B2D5, which is same as B2G2) and the three (B3G3) block results, as presented in the last two rows of Table 3, clearly reveal the effectiveness of the nested blocks, particularly with respect to B1G2 (both in terms of total grid numbers and the CPU time used). Therefore, while the present nested-block method undeniably improves the coarse single-block results greatly (as evident from the values of C_l and St in Table 3), the grid size and the CPU time in the nested-block method are still preserved at the coarse-grid (B1G1) level (see columns five and seven in Table 3). Thus, so far it has clearly been observed from the above mentioned numerical experiments that our nested-block method can indeed achieve an equivalent accuracy as obtainable on the finest uniform mesh (e.g., B1G2), and at the same time it greatly saves the CPU time. In addition, note that our simulated vortex shedding behavior (Fig. 10) and the corresponding velocity and pressure contours (Fig. 11(a)–(c)) at a further higher Reynolds number ($Re = 750$), as obtained by using the nested three-block (B3G3) grid system, seem to appear very much encouraging in this context.

In Fig. 11(d) and (e) we now extract the time average normalized pressure distribution around the cylinder at $Re = 40$ and 750, respectively. Simulation at $Re = 40$ (Fig. 11(d)) is particularly conducted to compare our computed results with the existing data. Important it may be to note that, Fig. 11(d) clearly demonstrates our simulated C_p matches quite encouragingly with those of Berthelsen and Faltinsen [33], Tseng and Ferziger [36], Dennis and Chang [37], and Grove et al. [38]. Moreover, values of C_p obtained with different refined grid sizes appear very much consistent (in fact they coincide exactly); which seem to further enhance the reliability of the present numerical model.

We now focus on numerically predicting the critical Reynolds number of the flow for a blockage ratio of 0.2. Notably, the bifurcation analysis of Chen et al. [35] indicates that the value of the critical Reynolds number associated with the particular blockage ratio is 231. This has also been confirmed by Ye et al. [21]. Herein, we focus on computing the critical value with each of B1G1, B1G2, and B3G3 grids. For this, with each of B1G1, B1G2, and B3G3 grid system we performed a series of numerical simulations within the range $225 \leqslant Re \leqslant 270$ to pinpoint the corresponding critical Reynolds numbers. From the flow stability analysis, it is expected that the first bifurcation of the steady flow should occur for a Re lying between 225 and 270. Note that, the transitional bifurcation processes of quasi-steady flows past circular as well as square cylinders have been clarified as supercritical Hopf bifurcations [39]. In an effort to extract an identifiable characteristic signifying the occurrence of such a Hopf bifurcation, variation of C_l^2 as a function of Re is now plotted in Fig. 12. Notably, since a linear relation, $C_l^2 \propto Re$, holds here, the occurrence of the supercritical Hopf bifurcation during transition from the steady flow to the oscillatory vortex shedding flow is clearly confirmed [4,39]. Herein we now concentrate on more accurate exploration of the bifurcation physics (in terms of Reynolds number) for the presently investigated vortex shedding flow by further examining the natures of variations of C_l^2 with respect to Re for all the three, B1G1, B1G2, B3G3, grid systems. Such variations of C_l^2 as functions of Re are also plotted in Fig. 12. Interestingly, the values of Re_{cr} for B3G3, B1G2, and B1G1, as evaluated, became 230.7, 234.2, and 262.6, respectively. Notably, when compared with the numerical results ($Re_{cr} = 231$) of Chen et al. [35] and Ye et al. [21], it becomes quite evident that our predicted $Re_{cr} = 230.7$, obtained by using the nested three-block grid system B3G3, shows excellent agreement with the previous studies, while the errors for the simulated results with B1G2 and B1G1 grids became about 1.4% and 13.7%, respectively.

3.3. Flow past an elliptical cylinder with angle of attack

In this section, the developed numerical method has been further tested by conducting a series of simulations for the two-dimensional unsteady viscous flow over an inclined elliptic cylinder placed in a uniform stream. The purpose of this test is to

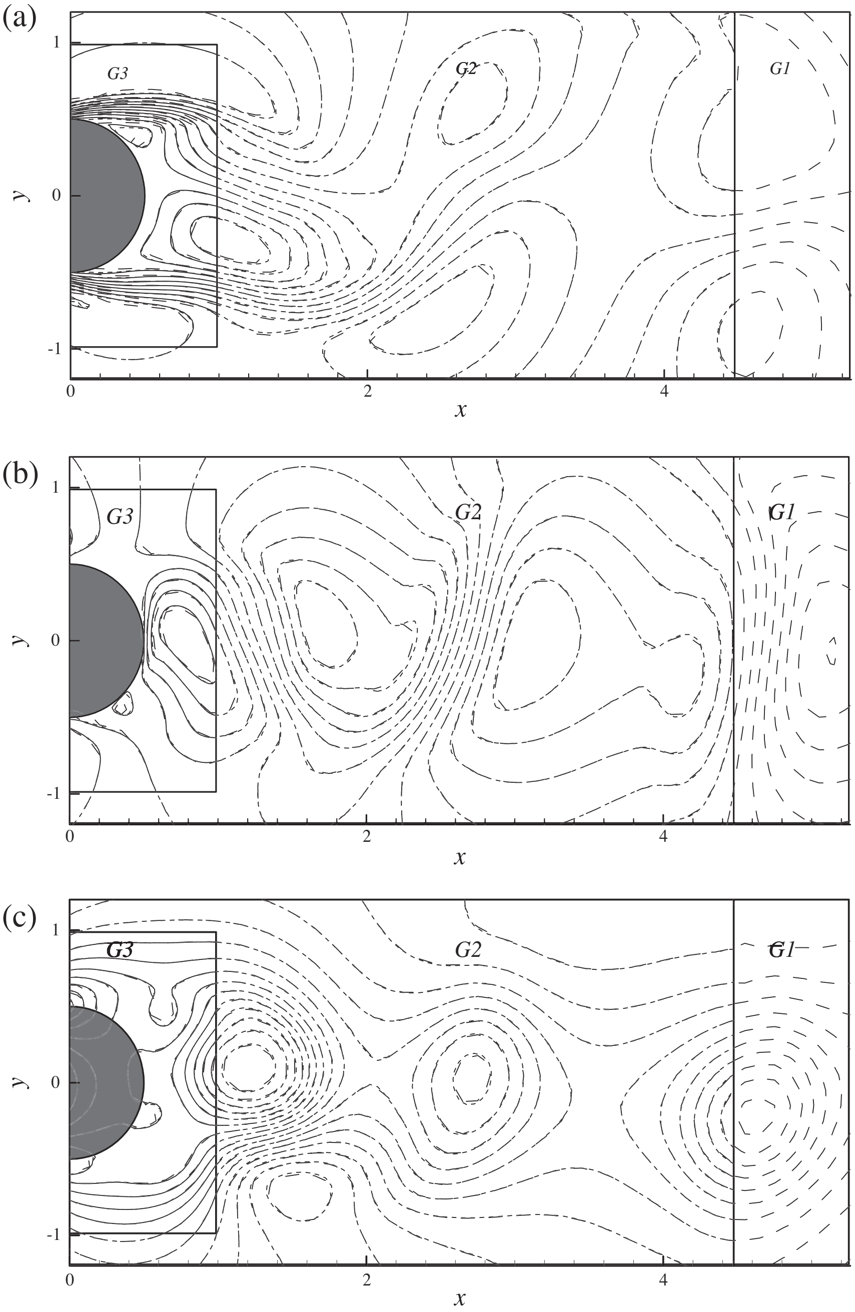

Fig. 9. Simulated near-wake velocity and pressure contours for the flow past a circular cylinder at $Re = 500$ obtained with the grid system B3G3. (a) Computed u velocity contours. (b) Computed v velocity contour plots. (c) Computed pressure (p) contour plots.

investigate the performance of the present nested grid method in a situation when more than two fine grid domains are present and they overlap. Uniform flows past an elliptic cylinder with the angle of attack 45° and Reynolds numbers (Re^*) 163 and 525 are considered here. Based on the projected width, L^*, of the ellipse, the Strouhal number (St^*) and the Reynolds number (Re^*) of the flow are defined as $St^* = fL^*/U_\infty$ and $Re^* = L^*U_\infty/\upsilon$, respectively, where f is the oscillation frequency. Notably, the calculation of St^* is useful here in the sense that, it can be used for comparison between flows over elliptic cylinders of various thickness ratios and angles of attack. In particular, the simulated St^* can be compared with the corresponding Strouhal number for circular cylinders, in order to validate our results.

The computational domain for the flow is taken to be rectangular of size $25L_x \times 10L_x$, where L_x is the length of the semi-major axis of the ellipse. Four different grids, as used to compute the flows are listed in Table 4. Note that, B1G1 and B1G2

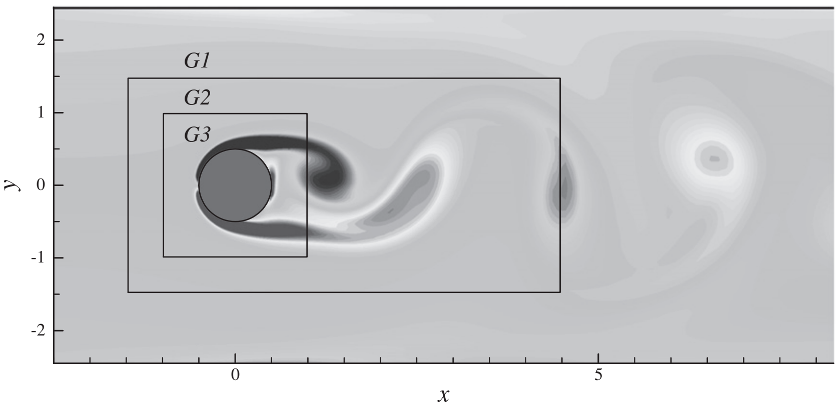

Fig. 10. Simulated vortex shedding behavior in the near wake of a circular cylinder at $Re = 750$, obtained with the three-block nested grid system B3G3.

(Table 4) use uniform grids with $\Delta x = \Delta y = 0.1L_x$ and $\Delta x = \Delta y = 0.05L_x$, respectively, and for B2G2 the computational domain was divided into two blocks. In the outer block, G1 grids with $\Delta x = \Delta y = 0.1L_x$ are used, and in the inner block of $4.1L_x \times 2.7L_x$, G2 grids with $\Delta x = \Delta y = 0.05L_x$ were used to compute the flow around the elliptical cylinder. For B3G2, the computational domain was divided into three blocks. In the outer block B1, as shown in Fig. 13, the G1 grids with $\Delta x = \Delta y = 0.1L_x$ were used to compute the outer flow. In the inner blocks B2 and B3 (Fig. 13) the G2 grids with $\Delta x = \Delta y = 0.05L_x$ were used to compute the flow around the elliptic cylinder. For clarity, the total number of cells used in each grid system is listed in Table 4, and the associated vortex shedding behavior of the flow at $Re^* = 163$ is presented in Fig. 13.

Before we discuss our computational results, it may be worth mentioning that, unlike in the symmetrical case ($\alpha = 0$), for the case of flow past an inclined elliptic cylinder ($\alpha > 0$) the lift remains a dominant force. Accordingly, the simulated time histories of the drag (C_d) and the lift (C_l) forces for the flow with $\Gamma = 0.2$, $\alpha = 45°$, $Re^* = 163$ are presented in Fig. 14(a) and (b), respectively, where Γ is the thickness ratio of the semi-minor axis to the semi-major axis of the cylinder (i.e., $\Gamma = L_y/L_x$). Notably, the depicted variation of C_l, as presented in Fig. 14(b), clearly suggests that there acts a positive (upward) lift force on the elliptical cylinder, and its periodic fluctuation seems to remain closely associated with the local vortex shedding behavior.

At this point, in order to investigate the efficiency of the presently implemented nested-block method, a series of numerical simulations for $Re = 163$ and 525 using four different grid systems (B1G1, B1G2, B2G2, B3G2) have been performed, and the corresponding results are presented in Tables 4 and 5. For all these simulations, a time increment Δt equals to 0.001 had been used and a dimensionless time $t = 300$ was chosen as the end of iterations. Notably, in Tables 4 and 5 the numerical results concerning the mean drag coefficient $\overline{C}_d$, mean lift coefficient $\overline{C}_l$, lift force amplitude $C_{l,Amp}$, and the dimensionless vortex shedding frequency (Strouhal number St^*) are presented. Findings of Mittal and Balachandar [40], whenever appropriate, are also listed there for comparison. We may mention here that, the results of Mittal and Balchandar [40] were obtained with a body-fitted grid system with merely 8100 cells. Moreover, we would also like to state that, the observed difference between the two sets of single block data (with B1G1 and B1G2), as presented in the first two rows of Tables 4 and 5, only reveal the (expected) important role of finer grid resolution (even when they are uniform) in such a flow complex simulation, and they seem to no way disregard the usefulness of the nested blocks. On the other hand, the results (Tables 4 and 5) presented in the second, third, and the fourth rows (with B1G2, B2G2, and B3G2) clearly reveal the efficiency of the nested-block method (both in terms of grid number and the CPU time required). To be explicit, the differences of 8.7% and 20.8%, between the computed two single block values (with coarse B1G1 and the fine B1G2 grid) of $C_{l,Amp}$(Table 7) and C_l (Table 5), respectively, suggest the need for higher grid resolution in the effective simulation of the flow. On the other hand, as evident from Table 4, at $Re^* = 163$ the observed differences of $\overline{C}_d$ and St^* between the coarse (B1G1) and the finer (B1G2) grids stood about 6.7% and 3.2%, respectively; and such differences of $\overline{C_d}$ and St^* (Table 5) at $Re^* = 525$ between B1G1 and B1G2 became about 7.2%, and 4.4%. Note that, although the improvement of (single block) numerical results from B1G1 to B1G2 is significant, the B1G2 grid number is four times the B1G1 grid number, and the corresponding CPU time is about 6 times that of B1G1. Moreover, from the numerical results listed in Tables 4 and 5, it may be observed that the refinement of grids around the ellipse improves the numerical results significantly. On the other hand, it is very interesting to note that, with just extra 3321 (and 3888) refined grids in B2G2 (and B3G2), consisting of about 13% (and 16%) of the B1G1 grid number, the present two block (three block) method can produce a closely resembled results to that of B1G2 (which consists of 100,000 grids!). Note also that, the differences of computed $\overline{C_d}$, Cl, and St^* values between B1G2 and B2G2 (or B3G2) are, remarkably, all now within 1.0%. Therefore, this local-refined-domain inference study not only demonstrates the efficiency of the present nested block immersed boundary method, but also clearly justifies the validity/usefulness of the presence of local-refinement overlap blocks in the flow simulation. In order to further establish the efficiency of such a nested-block grid system, in Figs. 15 and 16 we present the simulated vortex shedding behavior and the corresponding variations of the drag and the lift forces, respectively, for a higher aspect-ratio (thickness ratio) cylinder with $\Gamma = 0.5$, $\alpha = 45°$, $Re^* = 525$, obtained

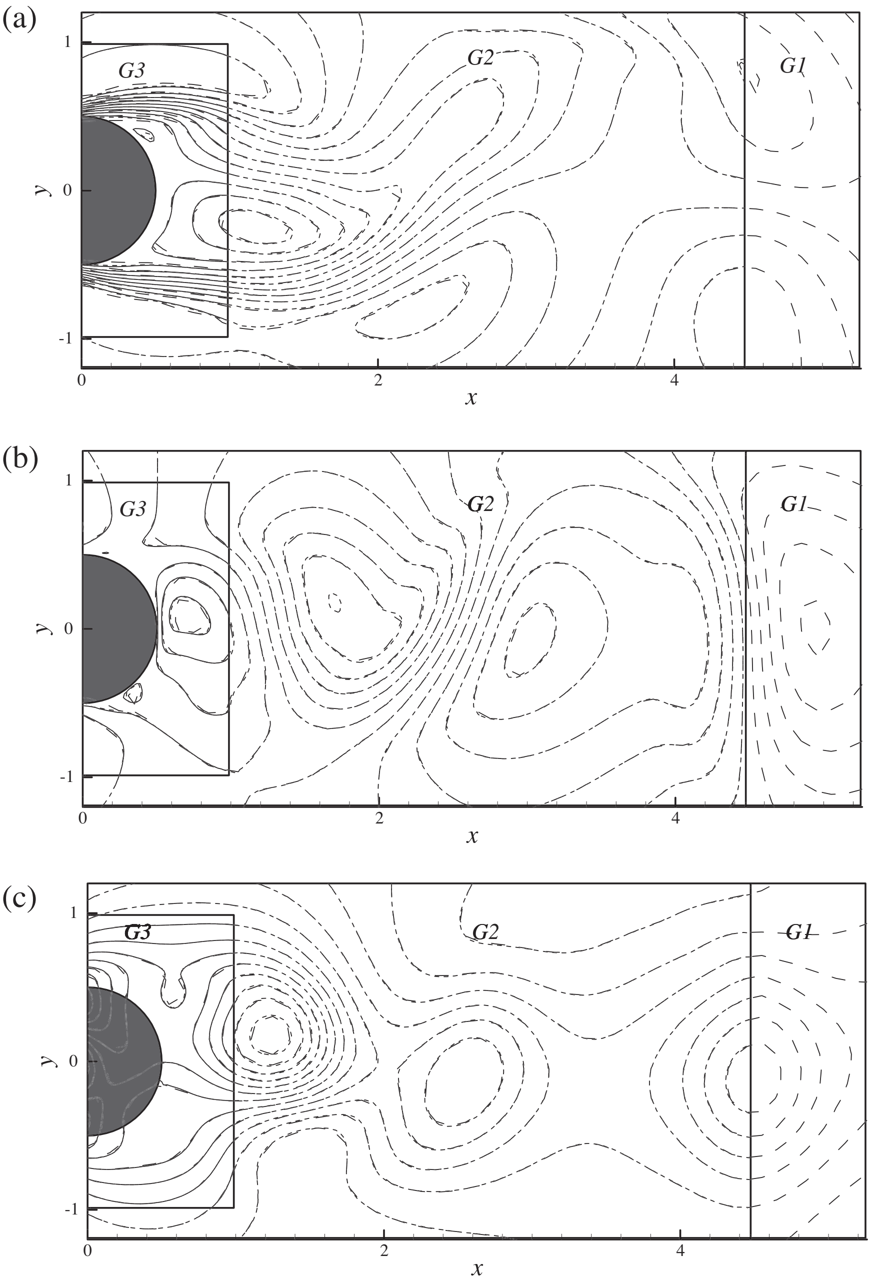

Fig. 11. Simulated near-wake velocity and pressure contours for the flow past a circular cylinder obtained with the grid system B3G3. (a) Computed u velocity contours at $Re = 750$. (b) Computed v velocity contour plots at $Re = 750$. (c) Computed pressure (p) contours at $Re = 750$. (d) Simulated wall pressure distribution at $Re = 40$, and comparison with existing results. (e) Computed wall pressure distribution at $Re = 750$.

by using the same grid distribution as in Fig. 13. Interestingly, the vortex shedding pattern at $Re^* = 525$, as presented in Fig. 15, clearly reveals encouragingly smooth variations of local flows across different block interfaces.

3.4. Flow past two circular cylinders with different diameters

Flow past two cylinders in tandem or in side-by-side arrangement exhibits an interesting but remarkably complex class of wake structure. Depending on the relative arrangement of the two cylinders, formations of a variety of interesting unstable/stable bifurcation patterns/modes in the near-wake have been widely documented in the literature. However, most of the reported studies concerning the flow past two cylinders of different diameters appear to be primarily devoted to the flow control. It is well known that the vortex shedding can cause structural damage through the oscillations of the flow, and in-

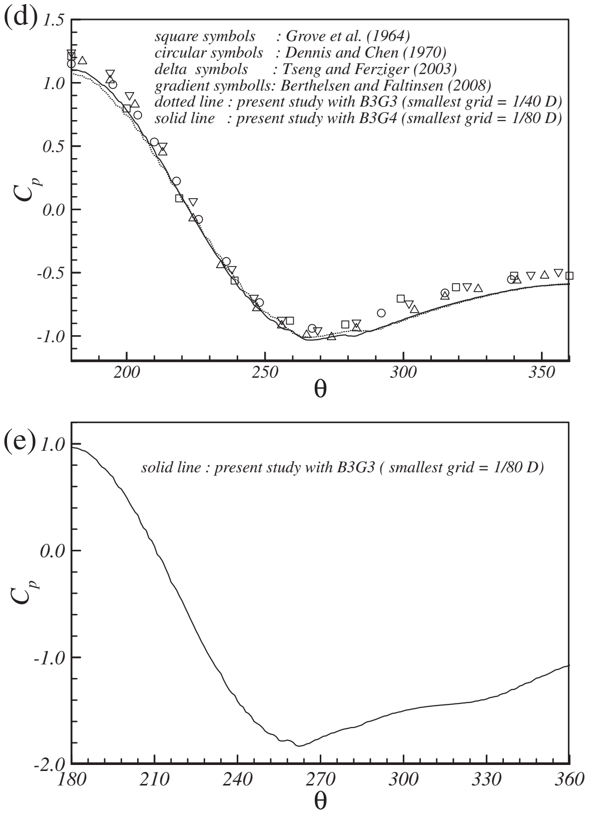

Fig. 11 (*continued*)

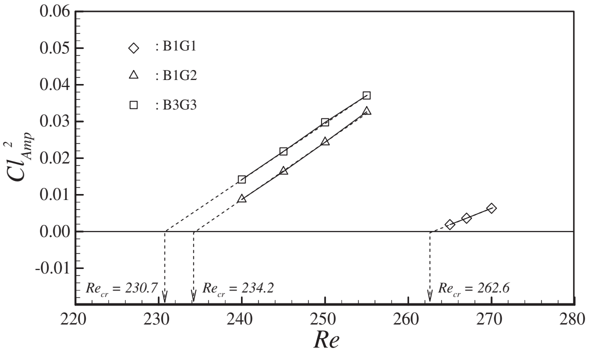

Fig. 12. Variations of $C^2_{l,Amp}$ as a function of *Re*, for the single block coarse (B1G1), single block fine (B1G2), the nested three-block (B3G3) grids, and the extracted corresponding critical Reynolds numbers Re_{cr}.

crease drag/noise. Importantly, Strykowski and Sreenivasan [41] investigated the suppression of vortex shedding from a circular cylinder at relatively low Reynolds numbers by means of introducing another small neighboring parallel cylinder. Following the experiment of Strykowski and Sreenivasan [41], many subsequent experimental and computational studies have been performed in order to control wake flows at various Reynolds numbers. Later Sakamoto et al. [42], and Sakamoto and Haniu [43] studied the phenomenon of suppression of the fluid forces on square and circular cylinders, respectively, in the sub-critical regime. Important it may be to mention here that, in a similar investigation Dalton et al. [44] also report the suppression of lift force on a circular cylinder in the presence of another neighboring small cylinder.

On the other hand, over the recent years there appeared a number of computational studies aimed at providing improved understanding of the flow behavior past two cylinders of different diameter (e.g., Zhao et al. [45], Delaunay and Kaiktsis [46], Young et al. [47], Dalton et al. [44]). These studies were either based on the finite-element or finite-volume method with

Table 4
Computed results of mean drag coefficient $\overline{C}_d$, mean lift coefficient $\overline{C}_l$, lift force amplitude $C_{l,Amp}$ and the Strouhal number St^* for flow past an elliptical cylinder with $\Gamma = 0.2$, $\alpha = 45°$, and $Re^* = 163$.

Grid system	Grids	$\overline{C}_d$ Difference to B1G2	$\overline{C}_l$ Difference to B1G2	$C_{l,Amp}$ Difference to B1G2	St^* Difference to B1G2	CPU
B1G1	25,000	3.603 6.7%	2.700 0.2%	1.386 8.7%	0.1981 3.2%	208.5
B1G2	100,000	3.378 0.0%	2.705 0.0%	1.275 0.0%	0.2045 0.0%	1257.0
B2G2	28,321	3.374 0.1%	2.698 0.3%	1.270 0.4%	0.2056 0.5%	383.9
B3G2	28,888	3.377 0.0%	2.701 0.1%	1.272 0.2%	0.2038 0.4%	390.8
Mittal (1996)	8100	3.71	Na	Na	0.19	Na

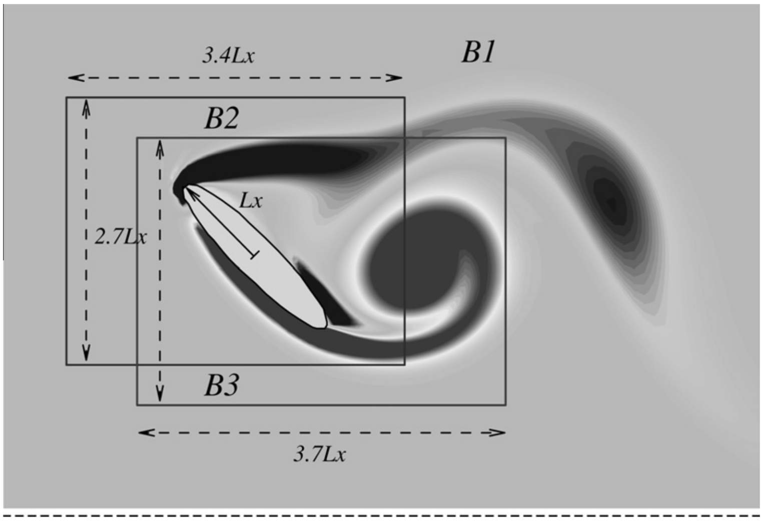

Fig. 13. Physical configuration and the vortex shedding flow behavior past a thin elliptical cylinder with an angle of attack. $\Gamma = 0.2$, $\alpha = 45°$, $Re^* = 163$.

body fitted grids. In our study, we primarily focus on wake evolution behind two circular cylinders of different diameters, in which the governing Navier–Stokes equations are solved using the present nested grid formulation and the immersed boundary method. The aim here is to examine the ability/efficiency of the nested grid solver in reproducing the widely reported phenomenon of suppression of the vortex shedding behind the two-cylinder system, which was kind of difficult to reach by a Cartesian grid method. The two cylinders considered in this study are shown in Fig. 17, and the diameter ratio of the small cylinder to the large cylinder is taken as 0.2. The values of Reynolds numbers, based on the diameter of the large cylinder, are taken as 100 and 80, and hence the corresponding values of *Re* with respect to the small cylinder become 20 and 16, respectively. The gap between the small cylinder and the large cylinder is taken as 2.0 times the diameter of the large cylinder. The position/angle of the small cylinder relative to the flow direction ranges from 20° to 40°. In all of such computations, a rectangular computational domain has been used. The large cylinder is placed at $5D$ downstream of the inflow boundary, and the distance between the large cylinder and the outflow boundary is fixed to $20D$. The two lateral boundaries, however, remained located at $7D$ away from the large cylinder (Fig. 17). As far as the reported results are concerned, the effects of the position/angle (of the small cylinder) on the drag and the lift forces over the large cylinder, the pressure distributions around it, corresponding vortex shedding frequencies, and the combined governing streamwise flow characteristics are critically examined here.

In order to demonstrate the efficiency of our numerical model, we first consider simulating the uniform flow past a single circular cylinder with Reynolds number varying from 10 to 200. Subsequently, the model is applied to simulate the flow over the two-cylinder system. For the single cylinder case, the computational domain of 25D × 14D was divided into two blocks. In the outer block, G1 grids with $\Delta x = \Delta y = 0.1D$ were used to compute the outer flow. In the inner block of $6D \times 4D$ [see Fig. 20(a)] the G2 grids with $\Delta x = \Delta y = 0.05D$ were used to compute the flow around the cylinder. In Fig. 18 our simulated time average drag coefficients, for Reynolds number varying between 10 and 100, are compared with the finite difference results of Lei et al. [48] and the finite element solutions of Zhao et al. [44]. Notably, an excellent agreement between the existing results [45,48] and the present solutions has been achieved. Fig. 19 provides a comparison of the simulated Strhoual

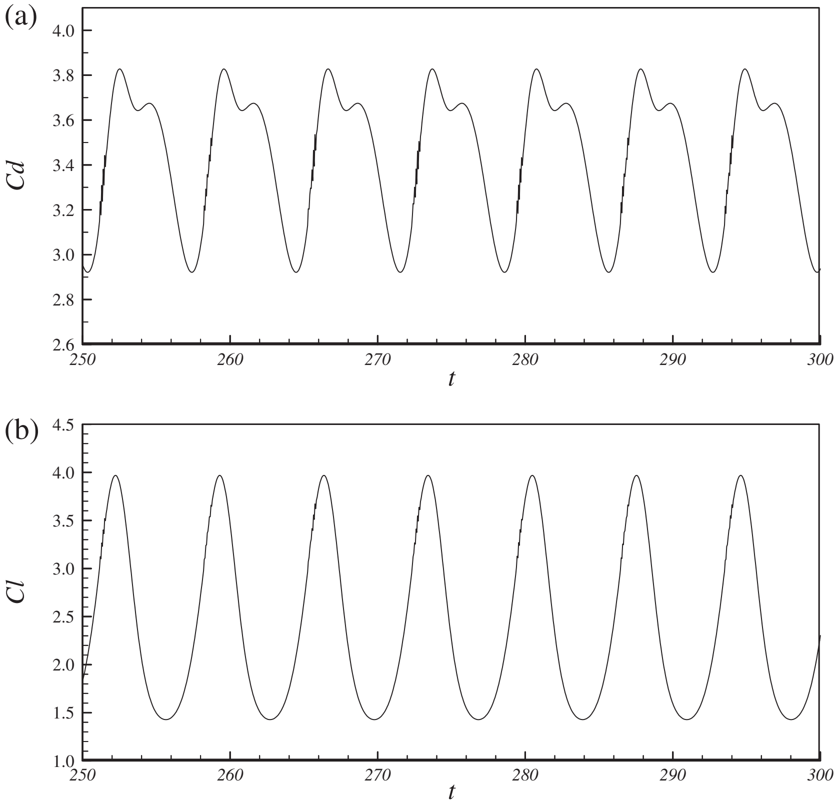

Fig. 14. Temporal variations of drag and the lift forces for the flow past a thin inclined elliptical cylinder with $\Gamma = 0.2$, $\alpha = 45°$, $Re^* = 163$, (a) distribution of the drag force C_d, (b) distribution of the lift C_l.

Table 5
Computed results of mean drag coefficient $\overline{C}_d$, mean lift coefficient $\overline{C}_l$, lift force C_l and the Strouhal number St^* for flow past an elliptical cylinder with $\Gamma = 0.5$, $\alpha = 45°$ and $Re^* = 525$.

Case No.	Grid size	$\overline{C}_d$ Difference to B1G2	$\overline{C}_l$ Difference to B1G2	C_l Difference to B1G2	St^* Difference to B1G2	CPU
B1G1	25,000	3.039 7.2%	1.859 0.3%	1.753 20.8%	0.2423 4.4%	285.5
B1G2	100,000	3.275 0.0%	1.864 0.0%	2.213 0.0%	0.2321 0.0%	1612.0
B2G2	28,321	3.278 0.1%	1.878 0.8%	2.215 0.1%	0.2318 0.1%	503.2
B3G2	28,888	3.276 0.0%	1.874 0.5%	2.216 0.1%	0.2315 0.3%	523.3
Mittal (1996)	8100	2.52	Na	Na	0.24	Na

number variations for $Re \leqslant 200$. Again, the present computational results agree very well with the previous studies. The conducted numerical tests therefore clearly show that the present nested grid immersed boundary method can in fact predict the flow over a single circular cylinder quite reliably.

The numerical model is now applied to study the flow past two circular cylinders, as depicted in Fig. 20(b). Before we discuss our detailed simulated results concerning the suppression of vortex shedding, here we briefly describe the implemented three-block domain decomposition method. A typical computational mesh together with the associated vortex shedding behavior for the flow at $Re = 100$ is presented in Fig. 20(b). Herein the computational domain of $25D \times 14D$ is divided into three blocks. In the outer block G1 grids with $\Delta x = \Delta y = 0.1D$ were used. In the middle block of $6D \times 4D$ the G2 grids with $\Delta x = \Delta y = 0.05D$ were used to compute the flow around the main cylinder. Whereas, in the inner block of $2.5D \times 1D$ the G3 grids with $\Delta x = \Delta y = 0.025D$ were used to compute the depicted flow (Fig. 20(b)) around the control cylinder. In all, the total number of cells employed in the computation was 45,200.

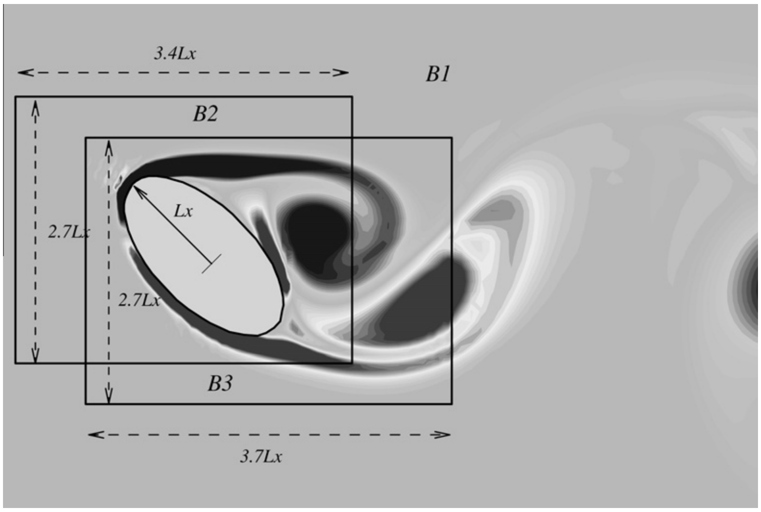

Fig. 15. Physical configuration and the associated vortex shedding flow behavior past a thick elliptical cylinder with $\Gamma = 0.5$, $\alpha = 45°$, $Re^* = 525$.

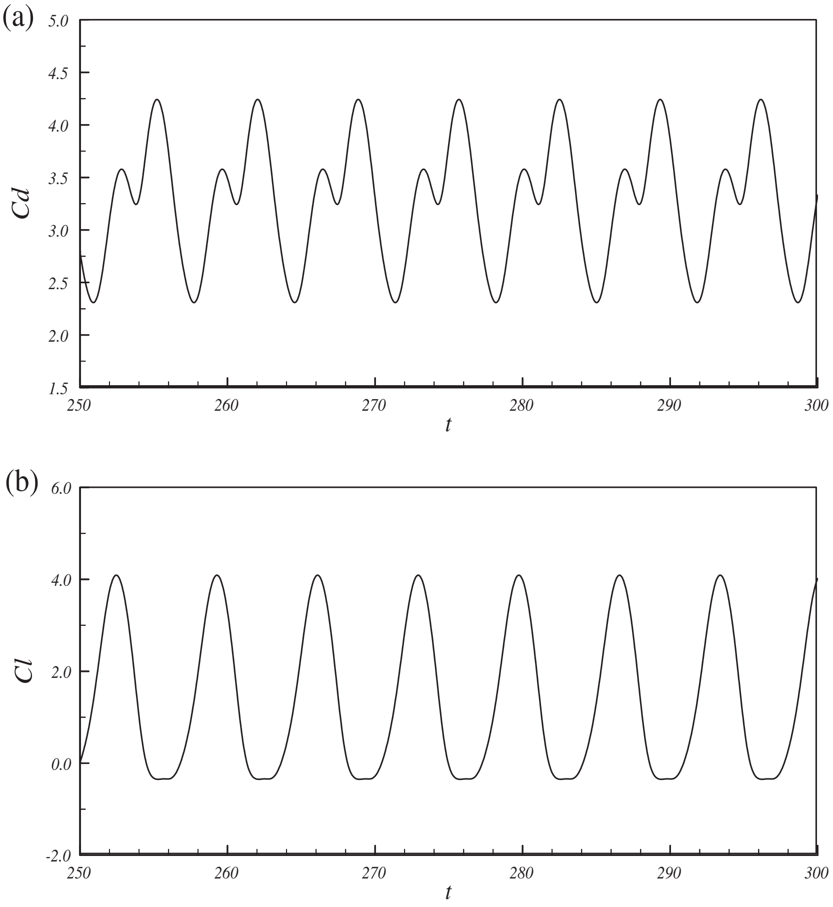

Fig. 16. Temporal variations of drag and the lift forces for the flow past a thick inclined elliptical cylinder with $\Gamma = 0.5$, $\alpha = 45°$, $Re^* = 525$, (a) distribution of the drag force C_d, (b) distribution of the lift force C_l.

Fig. 21 shows the computed time history of lift coefficient of the main cylinder at $Re = 100$. For the sake of comparison, the lift coefficient with an isolated single cylinder is also plotted here. It may be noted from the Fig. 21 that the effect of the small cylinder on the lift coefficients is quite significant. Specifically, the suppression of the mean lift on the large cylinder is very

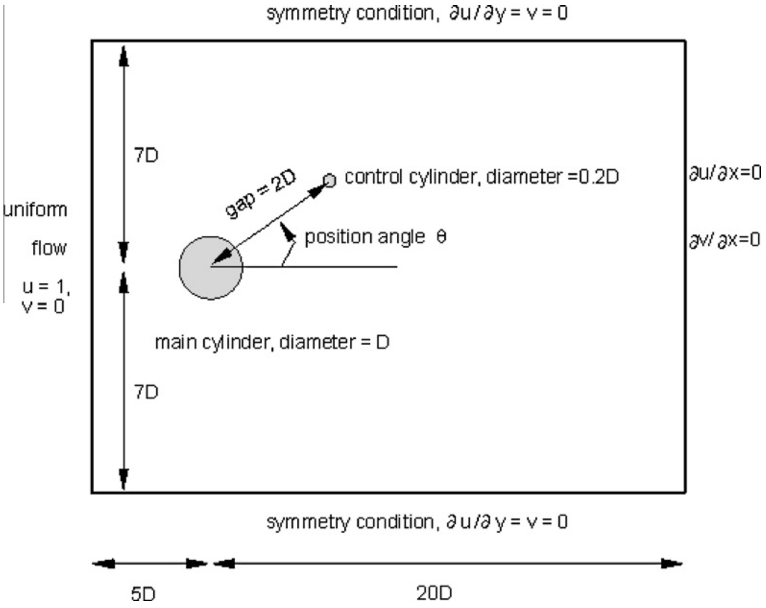
symmetry condition, ∂u/∂y = v = 0
7D
gap = 2D
control cylinder, diameter =0.2D
∂u/∂x=0
uniform
flow
u = 1,
v = 0
position angle θ
∂v/∂x=0
main cylinder, diameter = D
7D
symmetry condition, ∂u/∂y = v = 0
5D
20D

Fig. 17. Flow configuration and the boundary conditions for the simulation of flow past two circular cylinders of different diameters.

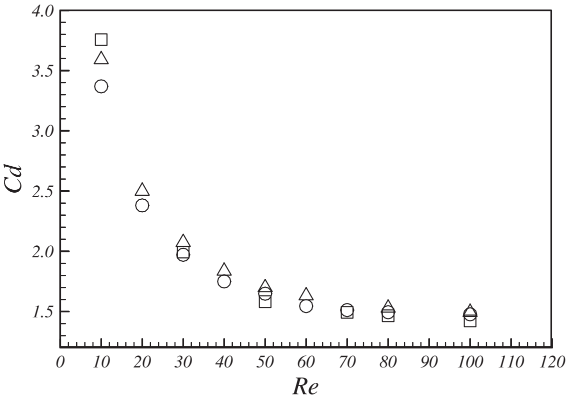

Fig. 18. Comparisons of simulated variations of drag coefficient C_d with Reynolds number *Re* for the flow past an isolated circular cylinder. Square symbols: present results; circular symbols: results from Lie et al. [48]; triangles: results from Zhao [45].

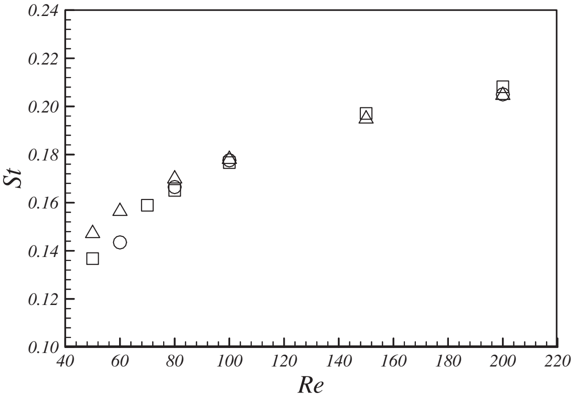

Fig. 19. Comparisons of simulated variations of *St* with Reynolds number *Re* for the flow past an isolated circular cylinder. Square symbols, present results; circular symbols, results from Lie et al. [48]; triangular symbols, results from Zhao [45].

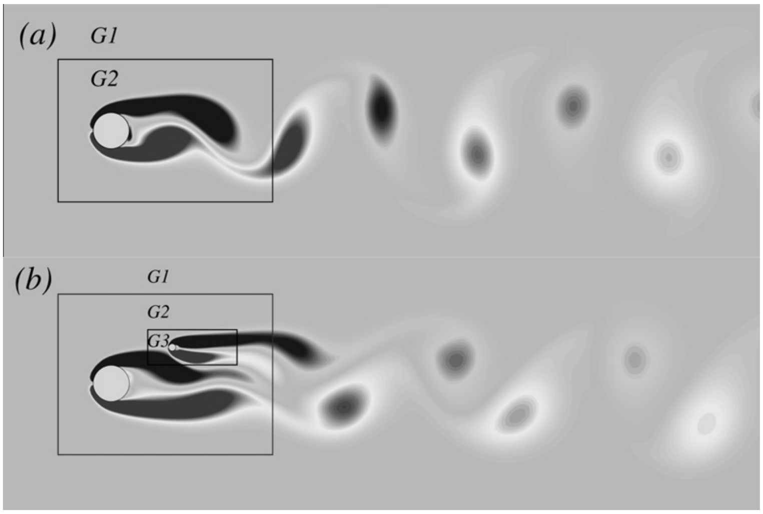

Fig. 20. Computed iso-vorticity plots and the associated nested grid domains: (a) flow past a single circular cylinder at $Re = 100$, (b) flow past two circular cylinders of different diameters at $Re = 100$ and $\theta = 30°$.

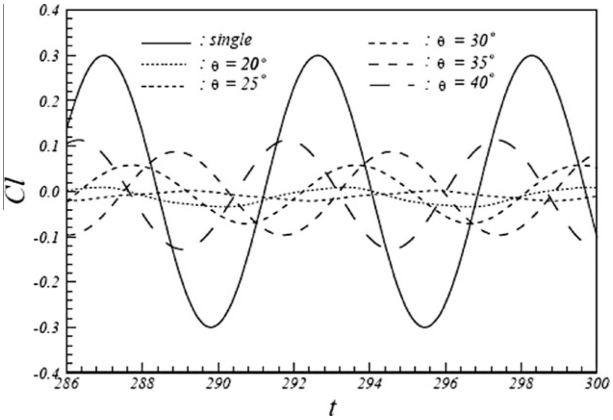

Fig. 21. Simulated temporal fluctuations of C_l for the flow past two cylinders of different diameters at $Re = 100$ with various position angles. The solid line represents the corresponding single cylinder case.

much noteworthy for all the computed cases (Fig. 21) with $20° \leqslant \theta \leqslant 40°$. The maximum suppression of the lift on the main cylinder is clearly noted (Fig. 21) to occur at $\theta = 25°$. For further clarity, the variations of mean lift and drag coefficients, amplitudes of the lift coefficients, and the Strouhal number St for the main cylinder as functions of the position angle of the small cylinder were collected from our simulated data and listed in Table 6. It is noted that (Table 6) the maximum suppression ratio of C_l on the main cylinder is about 3.7%, and it occurs at $\theta = 25°$. For the sake of enhanced clarity, corresponding simulated results for the two-cylinder system with $Re = 80$ are plotted/presented in Fig. 22 and Table 7. Notably, from Tables 6 and 7 it becomes absolutely clear that the suppression of the vortex shedding is most dominant at $Re = 80$ and $\theta = 25°$,

Table 6
Computed results of mean drag coefficient $\overline{C}_d$, mean lift coefficient $\overline{C}_l$, lift coefficient amplitude C_l, and the Strouhal number for the flow past two cylinders of different diameters at $Re = 100$ with various position angles.

θ	$\overline{C}_d$	$\overline{C}_l$	C_l	$C_l/C_{l,\text{single}}$ (%)	St
20	1.270	−0.0125	0.0215	7.2	0.155
25	1.263	−0.0098	0.0111	3.7	0.156
30	1.300	−0.0065	0.0645	21.5	0.169
35	1.316	−0.0045	0.0921	30.8	0.176
40	1.339	−0.0075	0.1205	40.2	0.182
Single	1.422	0.0	0.2994	100	0.177

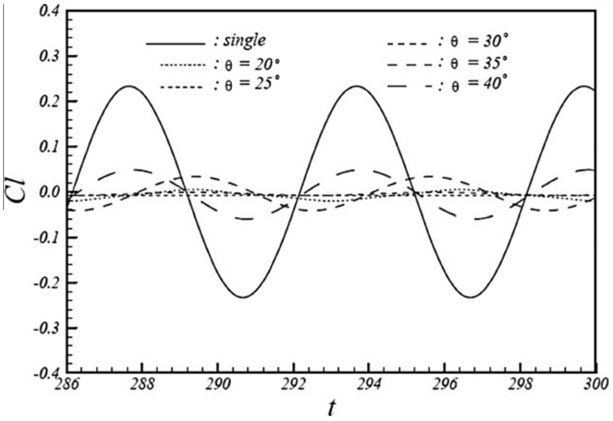

Fig. 22. Simulated temporal fluctuations of C_l for the flow past two cylinders of different diameters at $Re = 80$ with various position angles. The solid line represents the corresponding single cylinder case.

Table 7
Computed results of mean drag coefficient $\overline{C}_d$, mean lift coefficient $\overline{C}_l$, lift coefficient amplitude C_l, and the Strouhal number for the flow past two cylinders of different diameters at $Re = 80$ with various position angles.

θ	$\overline{C}_d$	$\overline{C}_l$	C_l	$C_l/C_{l,single}$ (%)	St
20	1.335	−0.008	0.0127	5.4	0.142
25	1.337	−0.008	0.0	0.0	0.0
30	1.349	−0.0053	0.0035	1.5	0.152
35	1.371	−0.0043	0.0377	16.2	0.160
40	1.389	−0.0058	0.0540	23.1	0.166
Single	1.465	0.0	0.2333	100	0.166

where the suppression ratio of C_l on the main cylinder is equal to 0.0%. Which means a total suppression can indeed be established in the coupled system. For completeness, the computed C_l history at $Re = 80$ and $\theta = 25°$ is extracted in Fig. 23. It clearly shows, for $t \geqslant 80$, the value of C_l has almost vanished (Fig. 23).

In order to demonstrate the effectiveness of the control cylinder on the suppression of vortex shedding more convincingly, in Fig. 24 we depict the vorticity distributions at $Re = 80$ and $\theta = 25°$. It now appears quite clear from Fig. 24 that, in the presence of a control cylinder the vortex shedding pattern/behavior becomes totally suppressed. The suppression of the vortex shedding is even more clear when the Reynolds number is further decreased. Note that, at $Re = 80$ and $\theta = 25°$, the suppression ratio of C_l on the main cylinder (Table 7) is found to be 0.0%, and importantly all such results are obtained with our three-block nested grid immersed boundary method. Therefore, as extensively demonstrated through a number of important but relatively complex simulations, our nested grid solver indeed produces very encouraging results, and they not only compare quite well with the existing experimental and theoretical predictions (Figs. 4–7, 11, 18, 19)(d), but the method

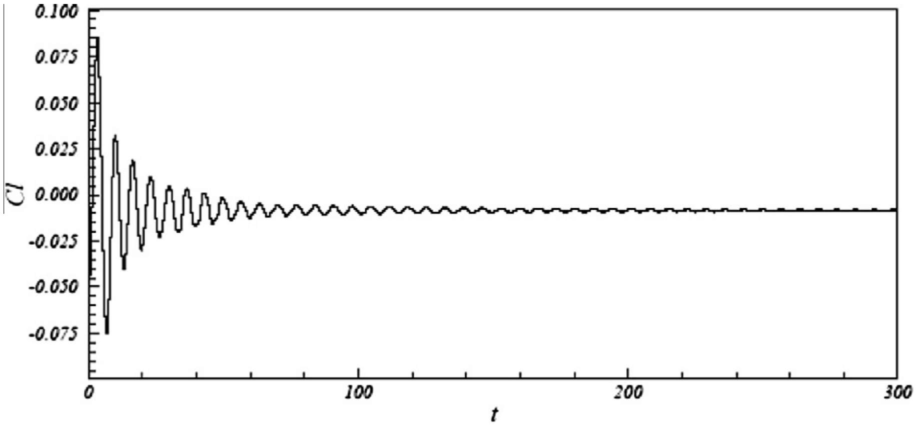

Fig. 23. Computed variations of C_l with time for the flow past two cylinders of different diameters at $Re = 80$ and $\theta = 25°$.

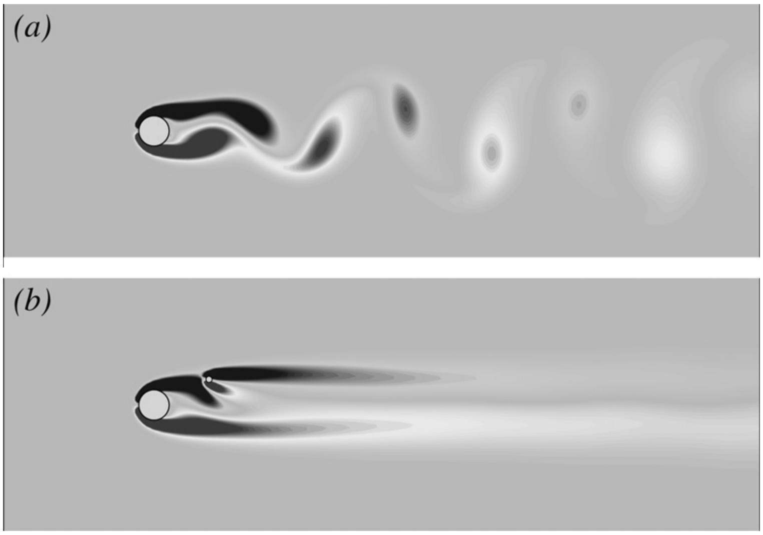

Fig. 24. Computed iso-vorticity contour plots and the associated nested grid domains: (a) flow past a single cylinder at $Re = 80$, (b) flow past two cylinders with different diameters at $Re = 80$ and $\theta\ = 25°$.

is far more economical (Tables 3–5) with respect to a single block method. Here we would like to add that the vortex shedding behind a circular cylinder (in absence of the control cylinder) starts approximately at $Re = 47$ through a single mode Hopf bifurcation, as the spanwise pressure oscillation along the corelins of the shedded vortices reaches a critical threshold value. This in turn facilitates the development of a local pressure maximum at the point of bifurcation (e.g., see [49]). Moreover, at $Re = 80$ owing to enhanced spanwise pressure oscillation (and growing local convective instability), there appears nearly symmetrical growth of two such isolated local pressure maxima [49–51] along the corelines of the shedded vortices (both in presence/absence of the control cylinder). The vortex shedding behind the (isolated) cylinder at this stage ($Re = 80$) occurs through the growth of a second mode of supercritical Hopf bifurcation. Therefore, suppression of vortex shedding till $Re = 80$, in the presence of a control cylinder, is indeed an important phenomenon/goal to achieve, which has been quite successfully obtained through our present nested grid formulation.

4. Conclusions

In this work, a nested-block finite-volume based Cartesian grid method has been developed, which allows us to effectively simulate the unsteady viscous incompressible flows with complex immersed boundaries. The underlying method is based on a colocated arrangement of variables, in which a globally second-order accurate central-difference scheme is used for spatial differencing. The solution is advanced in time using a two-step fractional-step scheme. The key aspects that needed careful consideration in developing such a nested-block solver are imposition of interface conditions on the nested-block boundaries and accurate discretization of the governing equation in cells that are with block-interface of a control-surface. The interpolation procedure adopted here allows systematic development of a discretization scheme that preserves the second-order global spatial accuracy of the underlying solver. As a result, high efficiency and accuracy nested-block method is developed.

Herein, five different complex numerical experiments have been conducted in order to extensively examine/demonstrate the performance of the present nested-block solver. For the lid-driven cavity flow our numerical results demonstrate that even with marginally refined grids the developed method helps to effectively improve the global flow features adequately. Such an observation clearly shows that the present nested-block method works very well for the investigated flow. In another numerical experiment, simulations of flows past a circular cylinder when placed symmetrically in a channel have been carried out in the Reynolds number range 225–750. Notably, the important key quantities, such as, the mean drag coefficient, the length of recirculation zone, and the Strouhal number, as obtained from our simulations, agree quite well with the available well established numerical results. In addition, the solver could predict the critical Reynolds number, associated with the vortex shedding, with a high degree of precision. For the flow past an inclined elliptic cylinder placed in a uniform stream the validity/effectiveness of the present numerical method in a situation when more than two fine nested grid domains are present/overlap is confirmed.

For the flow past two circular cylinders of different diameters (in tandem) the ability of the present numerical method in effectively predicting the suppression of the vortex shedding phenomenon, which has been reported to be difficult to obtain by a Cartesian grid method, is widely demonstrated. In addition, our numerical experiments clearly indicate that even with considerably reduced grid size the present nested-block method can achieve a solution of comparable accuracy to that obtained by the traditional finite difference methods on a finest uniform grid, and the present method also helps to significantly lower the CPU time requirement. In combination with the implemented immersed boundary method and the fractional-step procedure the current nested-block solver is able to simulate the reported flows with ease, and thereby, demonstrate the suitability/efficiency of the present approach in solving various physically important complex flow phenomena of recent interest.

References

[1] L. Chacon, G. Lapenta, A fully implicit, nonlinear adaptive grid strategy, J. Comput. Phys. 212 (2006) 703–717.

[2] H. Ding, C. Shu, A stencil adaptive algorithm for finite difference solution of incompressible viscous flows, J. Comput. Phys. 214 (2006) 397–420.

[3] Y.H. Shiau, Y.F. Peng, R.R. Hwang, C.K. Hu, Multistability and symmetry breaking in the 2-D flow around a square cylinder, Phys. Rev. E 60 (1999) 6188–6191.

[4] Y.F. Peng, Y.H. Shiau, R.R. Hwang, Transition in a 2-D lid-driven cavity flow, Comp. Fluids 32 (2003) 337–352.

[5] A. Sau, T.W. Sheu, S.T. Tsai, R. Hwang, T.P. Chiang, Structural development of vertical flows around a square jet in cross-flow, Proc. R. Soc. Lond. Ser. A 460 (2004) 3339–3368.

[6] M.J. Berger, J. Oliger, Adaptive mesh refinement for hyperbolic partial differential equations, J. Comput. Phys. 53 (1984) 484–512.

[7] A.M. Khokhlov, Fully threaded tree algorithms for adaptive refinement fluid dynamics simulations, J. Comput. Phys. 143 (1998) 519–543.

[8] J.Z. Zhu, O.C. Zienkiewicz, Adaptive techniques in the finite element method, Commun. Appl. Numer. Meth. 4 (1988) 197–204.

[9] P.A. Durbin, G. Iaccarino, An approach to local refinement of structured grids, J. Comput. Phys. 181 (2002) 639–653.

[10] R.B. Pember, J.B. Bell, P. Colella, W.Y. Curtchfield, M.L. Welcome, An adaptive Cartesian grid method for unsteady compressible flow in irregular regions, J. Comput. Phys. 120 (1995) 278–304.

[11] A.S. Almgren, J.B. Bell, P. Colella, L.H. Howell, M.L. Welcome, A conservative adaptive projection method for the variable density incompressible Navier–Stokes equations, J. Comput. Phys. 142 (1998) 1–46.

[12] M. Gerritsen, P. Olsson, Design and efficient solution strategy for fluid flows II. Stable high-order central finite difference schemes on composite adaptive grids with sharp shock resolution, J. Comput. Phys. 147 (1998) 293–317.

[13] W.M. Cao, W.Z. Huang, R.D. Russell, An r-adaptive finite element method based upon moving mesh PDEs, J. Comput. Phys. 149 (1999) 221–244.

[14] R. Li, T. Tang, P. Zhang, Moving mesh methods in multiple dimensions based on harmonic maps, J. Comput. Phys. 170 (2001) 562–588.

[15] A.M. Roma, C.S. Peskin, M.J. Berger, An adaptive version of the immersed boundary method, J. Comput. Phys. 153 (1999) 509–534.

[16] M.L. Minion, A projection method for locally refined grids, J. Comput. Phys. 127 (1996) 158–178.

[17] D.F. Martin, P. Colella, A cell-centered adoptive projection method for the incompressible Euler equation, J. Comput. Phys. 163 (2000) 271–312.

[18] D.F. Martin, P. Colella, D. Graves, A cell-centered adaptive projection method for the incompressible Navier–Stokes equations in three dimensions, J. Comput. Phys. 227 (2008) 1863–1886.

[19] B.E. Griffith, R.D. Hornung, D.M. McQueen, C.S. Peskin, An adaptive formally second order accurate version of the immersed boundary method, J. Comput. Phys. 123 (2007) 10–49.

[20] P. McCorquodale, P. Colella, H. Johansen, A Cartesian grid embedded boundary method for the heat equation on irregular domains, J. Comput. Phys. 173 (2001) 620–635.

[21] T. Ye, R. Mittal, H.S. Udaykumar, W. Shyy, An accurate Cartesian grid method for viscous incompressible flows with complex immersed boundaries, J. Comput. Phys. 156 (1999) 209–240.

[22] C.S. Peskin, Flow Patterns around Heart Valves: A Digital Computer Method for Solving the Equations of Motion. Ph.D. thesis. Physiol., Albert Einstein Coll. Med., Univ. Microfilms. vol. 378, 1972, pp. 72–30.

[23] C.S. Peskin, The fluid dynamics of heart valves: experimental, theoretical and computational methods, Annu. Rev. Fluid Mech. 14 (1981) 235–259.

[24] M.C. Lai, C.S. Peskin, An immersed boundary method with formal second-order accuracy and reduced numerical viscosity, J. Comput. Phys. 160 (2000) 705–719.

[25] D. Clarke, M. Salas, H. Hassan, Euler calculations for multi-element airfoils using Cartesian grids, AIAA J. 24 (1986) 1128–1135.

[26] H.S. Udaykumar, W. Shyy, M.M. Rao, A mixed Eulerian–Lagrangian method for fluid flows with complex and moving boundaries, Int. J. Numer. Meth. Fluids 22 (1996) 691–705.

[27] P. Angot, C.H. Bruneau, P. Frabrie, Apenalization method to take into account obstacles in viscous flows, Numer. Math. 81 (1999) 497–520.

[28] K. Khadra, P. Angot, S. Parneix, J.P. Caltagirone, Fictitious domain approach for numerical modeling of Navier–Stokes equations, Int. J. Numer. Meth. Fluids 34 (2000) 651–684.

[29] J.F. Ravoux, A. Nadim, H. Haj-Hariri, An embedding method for bluff body flows: interactions of two side-by-side cylinder wakes, Theor. Comput. Fluid Dyn. 16 (2003) 433–466.

[30] C.W. Hirt, B.W. Nichols, Volume of fluid (VOF) method for dynamics of free boundaries, J. Comput. Phys. 39 (1981) 201–225.

[31] U. Ghia, K.N. Ghia, C.T. Shin, High-Re solutions for incompressible flow using the Navier–Stokes equations and a multigrid method, J. Comput. Phys. 48 (1982) 387–411.

[32] P.H. Chiu, T.W.H. Sheu, R.K. Lin, An effective explicit pressure gradient scheme implemented in the two-level non-staggered grids for incompressible Navier–Stokes equations, J. Comput. Phys. 227 (2008) 4018–4037.

[33] P.A. Berthelsen, O.M. Faltinsen, A local directional ghost cell approach for incompressible viscous problems with irregular boundaries, J. Comput. Phys. 227 (2008) 4354–4397.

[34] A.J. Chorin, Numerical solution of the Navier–Stokes equations, Math. Comput. 22 (1968) 745–762.

[35] J.H. Chen, W.G. Pritchard, S.J. Tavener, Bifurcation for flow past a cylinder between parallel planes, J. Fluid Mech. 284 (1995) 23–52.

[36] Y.H. Tseng, J.H. Ferziger, A ghost cell immersed boundary method for flow in complex geometry, J. Comput. Phys. 192 (2003) 593–623.

[37] S.C.R. Dennis, G.Z. Chang, Numerical solutions for steady flow past a circular cylinder at Reynolds number up to 100, J. Fluid Mech. 42 (1970) 471–489.

[38] A.S. Grove, F.H. Shair, E.E. Petersen, A. Acrivos, An experimental investigation of the steady separated flow past a circular cylinder, J. Fluid Mech. 19 (1964) 60–80.

[39] M. Provansal, C. Mathis, L. Boyer, Benard von Karman instability: transient and forced regimes, J. Fluid Mech. 182 (1987) 1–22.

[40] R. Mittal, S. Balachandar, Direct numerical simulation of flow past elliptic cylinders, J. Comput. Phys. 124 (1996) 351–367.

[41] B.J. Strykowski, K.R. Sreenivasan, On the formation and suppression of vortex 'shedding' at low Reynolds numbers, J. Fluid Mech. 218 (1990) 71–107.

[42] H. Sakamoto, K. Tan, H. Haniu, An optimum suppression of fluid forces by controlling a shear layer separated from a square prism, J. Fluid Eng. 113 (1991) 183–189.

[43] H. Sakamoto, H. Haniu, Optimum suppression of fluid forces acting on a circular cylinder, J. Fluid Eng. 116 (1994) 221–237.

[44] C. Dalton, Y. Xu, J.C. Owen, The Suppression of lift on a circular cylinder due to vortex shedding at moderate Reynolds numbers, J. Fluid Struct. 15 (2001) 61–68.

[45] M. Zhao, L. Cheng, B. Teng, D. Liang, Numerical simulation of viscous flow past two circular cylinders of different diameters, Appl. Ocean Res. 27 (2005) 39–55.

[46] Y. Delaunay, L. Kaiktsis, Control of circular cylinder wakes using base mass transpiration, Phys. Fluids 13 (2001) 3285–3302.

[47] D.L. Young, J.L. Huang, T.I. Eldho, Simulation of laminar vortex shedding flow past cylinders using a coupled BEM and FEM model, Comput. Method Appl. Mech. Eng. 190 (2001) 5975–5998.

[48] C. Lei, L. Cheng, K. Kavanagh, A finite difference solution of the shear flow over a circular cylinder, Ocean Eng. 27 (2000) 271–290.

[49] A. Sau, Hopf bifurcations in the wake of a square cylinder, Phys. Fluids 21 (2009) 0341051–03410519.

[50] A. Sau, T.W. Hsu, S.H. Ou, Three-dimensional evolution of vortical structures and associated flow bifurcations in the wake of two side-by-side square cylinders, Phys. Fluids 19 (2007) 0841051–08410517.

[51] A. Sau, Growth of unsteady Hopf bifurcation modes and their swapping in a transitional coupled wake, Phys. Fluids 20 (2008) 1041081–10410818.