Available online at www.sciencedirect.com

ScienceDirect

Journal of Computational Physics 218 (2006) 510–548

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

A generalized finite-difference (GFD) ALE scheme for incompressible flows around moving solid bodies on hybrid meshfree–Cartesian grids

C.S. Chew [a], K.S. Yeo [b,*], C. Shu [b]

[a] *The Institute of High Performance Computing (IHPC), CFD Division, 1 Science Park Road, #01-01, The Capricorn, Singapore Science Park II, Singapore 117528, Singapore*

[b] *Department of Mechanical Engineering, National University of Singapore, 9 Engineering Drive 1, Singapore 117576, Singapore*

Received 9 March 2005; received in revised form 6 January 2006; accepted 22 February 2006
Available online 18 April 2006

Abstract

A scheme using the mesh-free generalized finite differencing (GFD) on flows past moving bodies is proposed. The aim is to devise a method to simulate flow past an immersed moving body that avoids the intensive remeshing of the computational domain and minimizes data interpolation associated with the established computational fluid methodologies; as such procedures are time consuming and are a significant source of error in flow simulation. In the present scheme, the moving body is embedded and enveloped by a cloud of mesh-free nodes, which convects with the motion of the body against a background of Cartesian nodes. The generalized finite-difference (GFD) method with weighted least squares (WLS) approximation is used to discretize the two-dimensional viscous incompressible Navier–Stokes equations at the mesh-free nodes, while standard finite-difference approximations are applied elsewhere. The convecting motion of the mesh-free nodes is treated by the Arbitrary Lagrangian–Eulerian (ALE) formulation of the flow equations, which are solved by a second-order Crank–Nicolson based projection method. The proposed numerical scheme was tested on a number of problems including the decaying-vortex flow, external flows past moving bodies and body-driven flows in enclosures.

Keywords: Moving body; Convecting nodes; Meshless method; Projection method; Generalized finite difference; Arbitrary Lagrangian–Eulerian; Navier–Stokes equations

1. Introduction

Computational fluid dynamics (CFD) is traditionally practised using mesh-based methods like the finite element (FE), finite volume (FV) and finite difference (FD), and they are widely accepted as the mainstream solution tools. While the mesh structure allows the FE and FV methods to handle complex geometries effectively, the same structure becomes a limitation when one has to alter the meshing to accommodate a

[*] Corresponding author. Tel.: +65 6516 2246; fax: +65 6779 1459.
E-mail address: mpeyeoks@nus.edu.sg (K.S. Yeo).

0021-9991/$ - see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2006.02.025

continuously changing computational domain. Problems of this nature are becoming common today, particularly with the growing preoccupation of engineers with biologically motivated problems, where large boundary motions are frequently encountered. Besides appearing cumbersome and inelegant, extensive remeshing and data interpolation in these methods can add to computational costs and solution inaccuracies. The traditional finite-difference method is highly effective when coordinate transformations are available to simplify the computational geometries. Unlike FE and FV methods, however, the transformation approach cannot be extended to complex geometries.

In recent decades, a new class of what is termed mesh-free or meshless numerical methods has flourished. This is due in part to their ability to alleviate some of the shortcomings of the conventional procedures. As opposed to mesh-based methods, they do not require pre-specified connectivity between the computational nodes or grid points. There is no need to identify the element or volume edges and surfaces. These methods are thus better able to represent irregular boundaries and negotiate complex spaces with ease and flexibility. Mesh-free methods are known to originate from the works of astrophysicists. The smooth particle hydrodynamics (SPH) method [1] is used to simulate astrophysical phenomena such as exploding stars and dust clouds. This method belongs to the class of particle methods [2] which adopts kernel approximations to obtain the Lagrangian flow field. Belytschko et al. [3] provided a detailed description of the broad spectrum of mesh-free methods that have been developed, including the diffuse element method (DEM) [4], the element-free Galerkin method (EFG) [5], the reproducing-kernel particle method (RKPM) [6], the partition of unity method (PUM, or PUFEM) [7], hp clouds [8], finite point method (FPM) [9–11] and the generalized finite-difference method (GFD) [12–15]. These methods have been applied to various problems in solid and fluid mechanics.

Particle methods like SPH and the vortex particle method are Lagrangian in nature and avoid grid-based numerical dissipation, which causes the flow to be more viscous. However, these ‘particles’ require good distribution and redistribution, and their Lagrangian nature makes this difficult in flows with extensive boundary motions. DEM and EFG belong to the type of mesh-free methods that are integral by nature. In this sense, they are not strictly ‘free from mesh’, and may suffer from the same deficiencies as the FE methods in moving body problems. The truly mesh-free methods are the non-integral types in hp clouds, FPM and GFD. These methods adopt mesh-free computational nodes that may not necessarily be Lagrangian in nature, and such nodes may be more suited for handling flows with moving boundaries.

As mentioned earlier, there has been growing interest in the modeling of moving boundary problems, which is aided not least by the continuous increase in computing power and corresponding reduction in the cost of computation. In this paper, we shall concern ourselves with the development of a hybrid mesh-free computational scheme for the simulation of flows past and flows driven by moving solid bodies. We believe the advantages of mesh-free methods can be harnessed to solve this class of problem. Before we proceed with the details of the new scheme, we shall first present a broad overview of the various methods that have been devised over the years to deal with the simulation of moving boundary/body problems.

Methods for solving moving boundary or interface problems fall under two main categories: those that employ boundary-fitted grids (also termed body-conformal grids) and those that employ non-boundary-fitted grids. A method is boundary-fitted when the boundaries or interfaces coincide with the computational nodes or grid points. The nodes/grid points then clearly demarcate the interface, and morph and move with the interface during motion. Boundary conditions are applied directly on the interface, and the motion of the interface is explicitly tracked. The two common boundary-fitted methods are composite grids [16,17], which are normally applied with finite differencing, and FV or FE schemes, which are frequently implemented with the arbitrary Lagrangian–Eulerian (ALE) formulation [18].

Composite grids with generalized coordinates and transformed governing equations are commonly used to embed solid objects within flows. A separate set of coordinate frame and grid points (the sub-mesh) are generated for each embedded object. Local sub-meshing is usually performed using an orthogonal or algebraic grid generator. Communication between the main and the local meshes is usually done through extensive interpolation of data between the meshes, typically at the outer boundaries of the mesh systems. Apart from interpolation, the sub-meshes are also restricted in their geometric complexity due to the use of generalized coordinates. The second boundary-fitted method involves the use of unstructured grids with FE or FV methodologies to solve for the flow in arbitrary Lagrangian–Eulerian (ALE) formulation [19,20]. This is an elegant approach in which the solution is derived based on a mixed treatment of grid points in either the Lagrangian or the Eulerian fashion. The use of FE or FV, however, requires constant remeshing near areas of interfacial motion. Computational overheads are incurred in both mesh administration and data interpolation into the new meshes that may considerably slow down the course of simulation. In boundary-fitted methods, there are also potential errors arising from equation transformation, data interpolation and complex discretization along the interface. There are also a number of positive features in this category of methods. The boundary conditions are accurately represented and directly applied onto the interface. One can also have control over the grid resolution near the interface, and, as a result, obtain desirable flow resolution. With the advent of computer technology, the advantage of resolution control may well prove to outweigh the disadvantage of costly overheads.

Non-boundary-fitted (non-body-conformal) methods mainly employ a single 'background' mesh which may be in structured or unstructured form. Mittal and Iaccarino [21] provided a review of these methods. The interface can be traced by a series of 'marker nodes', and methods are distinguished by their different treatments on these nodes, as well as the implementation of the boundary conditions. For interface treatment, the two means of defining the interface are front tracking and front capturing. Front tracking (e.g. [22–27]) employs explicit derivation of the 'marker node' velocity to track the interface motion. The front capturing approach uses a phase function to solve for the evolution of the interface. In volume of fluids (VOF) methods [28,29], the volume fraction is solved in the form of an advection equation. The volume fraction ranges between 0 and 1, of which the two extreme values represent the two different phases within the flow. When the value in a cell is between 0 and 1, the interface lies within that computational cell, and the position of the line of interface will be determined through the contributions of the neighbouring cells. In the case of the level set method [30,31], the solution field provides the interface as the zero level set of a smooth function. Because of this feature, the level set method can handle topographic merging and breaking very easily. Front capturing is considered implicit due to the way it obtains the interface, as compared to the more explicit form of front tracking.

Non-boundary-fitted methods can also be distinguished by their treatment of boundary conditions. They can basically be classified as Cartesian or Immersed Boundary. Cartesian methods are often front tracking methods, whereby the interface is identified by its boundary nodes. The background cells (or stencils) are modified by the intersection between the cell and the interface. By applying the boundary conditions on the interface with interpolation, the cell flow field is obtained.

Immersed boundary methods [32–34] employ force terms as source terms in the equations of motion to account for the effects of the boundary, like mass conservation and no-slip conditions. These modified equations are solved in the background grid, and interfacial motion is then derived through interpolation from the background. The new boundary topology then determines the new interfacial forces to be used in the equations of motion. Traditionally, the force terms are implemented over a small 'band' of cells in the immediate vicinity of the interface via the delta function. This practice enables conservation, but at the expense of a 'diffuse interface' that smears out flow details near the interface, like boundary layers. As a result, finer cells are usually required in the interface region to obtain the desired features.

Recently a class of a hybrid mix of Cartesian and immersed boundary methods has emerged (aptly called hybrid Cartesian/immersed boundary, HCIB, e.g. [27]). Basically, the boundary conditions are applied at the background nodes nearest to the interface. The variables at these nodes are interpolated from the interior nodes with known solved values and from the nodes at the boundary itself. This method retains the boundary as a sharp interface and is relieved of the need for explicit force terms in the governing equations.

From a negative point of view, non-boundary-fitted methods rely solely on the background grid for flow resolution. There is thus limited grid control. Though new schemes have emerged which include the use of adaptive local grid refinement [35] to alleviate this problem, this additional step may still be difficult to implement on moving boundaries. Also, some of the explicit boundary treatment like the cut-cell stencil modification make boundary condition implementation somewhat complex when huge numbers of moving boundary cells are involved. The issue of a diffused interface representation also poses an age-old disadvantage for some of the non-boundary-fitted methods.

Having mentioned the drawbacks, non-boundary-fitted methods are still very attractive and remain as competitive alternatives in recent times. Firstly, the diffuse interface problem has been addressed by many variants, and many of them are now able to maintain the boundary as a sharp interface. Secondly, the established non-boundary-fitted methods are generally conservative and are able to handle phase jumps easily. Compared to boundary-fitted methods, they are better able at managing deforming interface problems, which is often the case in biological applications. These methods can also make use of powerful, low cost numerical tools like line methods and multigrid solvers.

Taking into consideration the advantages and limitations of the various approaches for modeling moving boundary flow problems, there clearly exists an opportunity to develop a fresh approach, in which one hopes to minimize some of the cumbersome and potentially error-prone features of existing schemes, and to strengthen numerical representation in the area of the boundary. In this paper, we formulate a scheme based on a convecting mesh-free grid. The proposed method makes use of generalized finite-difference (GFD) approximation in weighted least squares (WLS) form and the governing flow equations in ALE formulation. The method of WLS-GFD has been discussed by Ding et al. [36,37]. Essentially, the mesh-free scheme is built on the performance of mesh-free computations around a cloud of mesh-free nodes that cluster around the moving body. The nodal cloud convects with the moving body against a background of Cartesian non-staggered grid points, and all computations are carried out within a single Cartesian reference frame. Grid resolution may be easily controlled by regulating the nodal cloud density, and nodes can be added and removed with ease for bodies that deform extensively – we shall, however, be concerned here with solving problems of rigid moving bodies at this stage of the study. There is no need to identify cells, faces, edges and vertices in this scheme; and the least-square based difference approximation permits direct interpolation of computational data to acquire fluid properties on ‘fresh’ or new nodes with ease and accuracy.

The main emphasis of the paper is on the simulation of the incompressible Navier–Stokes flows in the primitive variables of velocity and pressure (u, v, p) in two dimensions. All of the flows simulated involved rigid bodies, both stationary and under prescribed motion. The weighted least squares form of the generalized finite-difference (WLS-GFD) approximation is briefly described in Section 2. The numerical scheme for the flow equations and the treatment of fresh nodes are described in Section 3. The method is first tested on established flow problems with stationary boundary in Section 4, where the overall spatial and temporal accuracy of the scheme is also evaluated. A moving patch/cloud test (with no embedded object) is also carried out to study spatial accuracy and the effects of node convection (with node removal and fresh node creation) on computational error. The full ALE-WLS-GFD scheme is applied to circular cylinders undergoing prescribed acceleration and oscillatory motions in Section 5. The results obtained are compared against published results as well as results that are computed using an equivalent moving frame under the same ALE-WLS-GFD scheme. Finally the method is applied to three flow cases in Section 6. In box enclosures, a flapping ellipse, an asynchronous counter-rotating pair of beams, and an oscillating cylinder demonstrate the capability of the proposed scheme to solve flows driven by multiple, non-collisional moving solid boundaries up to moderately high Reynolds numbers.

2. The weighted least square generalized finite-difference method

The generalized finite-difference (GFD) method is based on the Taylor series expansion. For a two-dimensional function $f(\mathbf{x})$, where $\mathbf{x} = (x, y)$, the value of the function at $\mathbf{x}_1 = \mathbf{x}_0 + \Delta\mathbf{x}_1$ is given in terms of the derivative values of the function at the reference node point $\mathbf{x}_0$ to order m by

$$f(\mathbf{x}_1) = f(\mathbf{x}_0) + \sum_{1\leqslant j_1+j_2\leqslant m-1} \frac{\Delta x_1^{j_1}\Delta y_1^{j_2}}{j_1!j_2!}[\partial_x^{j_1}\partial_y^{j_2}f]_{\mathbf{x}_0} + \mathrm{O}(|\Delta\mathbf{x}_1|^m), \tag{1}$$

where ∂_x and ∂_y denote the partial derivatives with respect to the coordinate variables x and y respectively. If the values $f(\mathbf{x}_i) = f_i$ $(i = 0, 1, \ldots, n)$ of the function are known at a number of points (neighbours; see Fig. 1) $\mathbf{x}_i = \mathbf{x}_0 + \Delta\mathbf{x}_i$, one may truncate the Taylor series Eq. (1) appropriately and approximate the derivatives $(\partial_x^{j_1}\partial_y^{j_2}f)_{\mathbf{x}_0}$ of the function at the reference node $\mathbf{x}_0$ by solving a system of linear equations. For such an approximation, one will typically select the points $\mathbf{x}_i$ $(i = 1, \ldots, n)$ which are in the immediate or near-neighbourhood of $\mathbf{x}_0$. The *neighbours* or *support nodes* $\mathbf{x}_i$ are usually taken to be points within a prescribed distance d_0 from the reference node, $|\mathbf{x}_i - \mathbf{x}_0| = |\Delta\mathbf{x}_i| < d_0$. Alternatively, one may also use only a subset of the neighbouring node points according to certain criteria [9,10] designed to improve the accuracy of derivative approximation. When

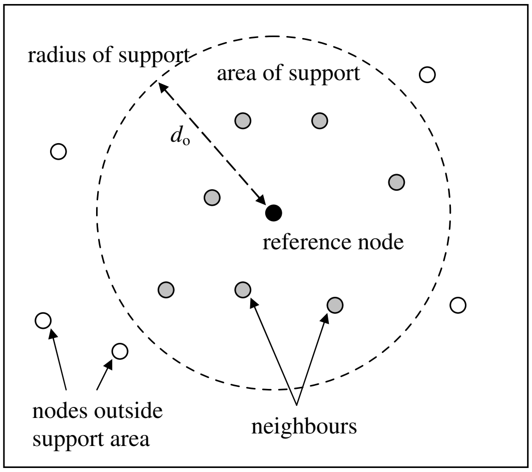


Fig. 1. Schematic of a typical reference node with its 'neighbours' within its radius of support, d_o.

the Taylor series Eq. (1) is truncated after the third-order derivative (i.e. $m = 4$), we obtain the following linear system, which relates the derivatives $\partial\mathbf{f}_{9\times 1} = (\partial_x, \partial_y, \partial_x^2, \partial_x\partial_y, \ldots, \partial_y^3)^{\mathrm{T}} f|_{\mathbf{x}_0}$ of f at $\mathbf{x}_0$ to the functional values of f at $\mathbf{x}_0$ and the n support nodes:

$$\Delta\mathbf{f}_{n\times 1} = [S]_{n\times 9}\partial\mathbf{f}_{9\times 1}, \tag{2}$$

where $\Delta\mathbf{f}_{n\times 1} = (f_1 - f_0, f_2 - f_0, \ldots, f_n - f_0)^{\mathrm{T}}$, and

$$[S]_{n\times 9} = \begin{bmatrix} \Delta x_1 & \Delta y_1 & \Delta x_1^2/2! & \Delta x_1 \Delta y_1 & \cdots & \Delta y_1^3/3! \\ \Delta x_2 & \Delta y_2 & \Delta x_2^2/2! & \Delta x_2 \Delta y_2 & \cdots & \Delta y_2^3/3! \\ \vdots & & & & & \vdots \\ \Delta x_n & \Delta y_n & \Delta x_n^2/2! & \Delta x_n \Delta y_n & \cdots & \Delta y_n^3/3! \end{bmatrix}. \tag{3}$$

For $m = 4$, $n = (m - 1)(m + 2)/2 = 9$ support nodes are needed to exactly close the linear system. If the matrix $[S]$ is non-singular, one may then solve Eq. (2) to obtain the derivative vector $\partial\mathbf{f}_{9\times 1}$; with formal accuracies of $\mathrm{O}(|\Delta\mathbf{x}|^3)$ and $\mathrm{O}(|\Delta\mathbf{x}|^2)$ for the first- and second-order derivative components of $\partial\mathbf{f}_{9\times 1}$ respectively.

While the above approach to approximating the derivatives numerically is sound in theory, it may suffer from ill-conditioning of the matrix $[S]$ due to poor spatial arrangement of the support nodes. Ill-conditioning can arise from a variety of causes such as extremely close separation between some nodes or large number of nodes aligning in a nearly collinear manner.

To overcome the above problem associated with solving Eq. (2) directly for the numerically exact functional derivatives $\partial\mathbf{f}_{9\times 1}$, one may instead seek a numerically approximate solution for $\partial\mathbf{f}_{9\times 1}$ that minimizes some measure of the residual error, $\mathbf{E}_{n\times 1} = \Delta\mathbf{f}_{n\times 1} - [S]_{n\times 9}\partial\mathbf{f}_{9\times 1}$ for Eq. (2). It is a common practice in such a case to minimize the squares of the residual error. This is equivalent of course to a minimization of the square of the Euclidean l_2-norm $\|\ \|_2$ of error, i.e. $\min_{\partial\mathbf{f}}\{\|\mathbf{E}_{n\times 1}\|_2^2\}$. We note that

$$\|\mathbf{E}_{n\times 1}\|_2^2 = (\Delta\mathbf{f}_{n\times 1} - [S]_{n\times 9}\mathbf{b})^{\mathrm{T}}(\Delta\mathbf{f}_{n\times 1} - [S]_{n\times 9}\mathbf{b}), \tag{4}$$

where $\mathbf{b}$ is the abbreviated form for $\partial\mathbf{f}_{n\times 1}$, so that the variation of $\|\mathbf{E}_{n\times 1}\|_2^2$ with respect to $\mathbf{b}$ is given by

$$\Delta_b\|\mathbf{E}_{n\times 1}\|_2^2 = (\mathbf{b}^{\mathrm{T}}[S^{\mathrm{T}}][S] - (\Delta\mathbf{f})^{\mathrm{T}}[S])(\Delta\mathbf{b}) + (\Delta\mathbf{b})^{\mathrm{T}}([S^{\mathrm{T}}][S]\mathbf{b} - [S^{\mathrm{T}}]\Delta\mathbf{f}) + \mathrm{O}(|\Delta\mathbf{b}|^2), \tag{5}$$

where some of the subscripts for matrix dimension have been omitted for brevity, and $\Delta\mathbf{b}$ denotes the variation of $\mathbf{b}$. The stationary property at the point of minimum square error then yields the following linear equation:

$$[S^{\mathrm{T}}]_{9\times n}[S]_{n\times 9}\partial\mathbf{f}_{9\times 1} = [S^{\mathrm{T}}]_{9\times n}\Delta\mathbf{f}_{n\times 1}, \tag{6}$$

by which the functional derivatives $\partial\mathbf{f}_{9\times1}$ may be determined. Eq. (6) can be seen to be merely a preconditioned form of the original Eq. (2) for the functional derivatives, where the preconditioner is just the transpose of the coefficient matrix $[S]$. The coefficient matrix $[S^{\mathrm{T}}]_{9\times n}[S]_{n\times9}$ for Eq. (6) is symmetric, and also positive-definite if the columns of $[S]_{n\times9}$ are independent. The advantage of casting the original problem Eq. (2) in the least square form of Eq. (6) is that it allows an over-determined system ($n > 9$) of support nodes. This is a crucial property, for by including more support nodes $\mathbf{x}_i$ ($i = 1, \ldots, n$) than the number of unknown functional derivatives in $\partial\mathbf{f}_{9\times1}$, one can always ensure the columns of $[S]_{n\times9}$ are independent, and hence the existence of solution. It is pertinent to note in passing that there are alternative formulations/algorithms of the least square problem via the orthogonalization or singular value decomposition procedures that may have improved numerically properties, albeit at increased computational cost. The normal equation approach Eq. (6) for approximating the functional derivative $\partial\mathbf{f}_{9\times1}$ is found to be adequate for our purpose here.

The least square approach can be further refined by the incorporation of weight factors w_i ($i = 1, \ldots, n$) for the errors at the support nodes. The weight factors are normally designed to give greater importance to errors at nodes that are located nearer to the reference node. The square of the weighted norm function is

$$\|\mathbf{E}^W_{n\times1}\|_2^2 = (\Delta\mathbf{f}_{n\times1} - [S]_{n\times9}\mathbf{b})^{\mathrm{T}}[W_n](\Delta\mathbf{f}_{n\times1} - [S]_{n\times9}\mathbf{b}), \tag{7}$$

where $[W_n] = \mathrm{Diag}\{w_1, w_2, \ldots, w_n\}$ is the diagonal matrix of weight factors. The least square Eq. (6) is then

$$[S^{\mathrm{T}}]_{9\times n}[W_n][S]_{n\times9}\partial\mathbf{f}_{9\times1} = [S^{\mathrm{T}}]_{9\times n}[W_n]\Delta\mathbf{f}_{n\times1}. \tag{8a}$$

or, in component form,

$$\sum_{k=1}^{9}\sum_{j=1}^{n} w_j S_{ji} S_{jk}\partial f_k = \sum_{j=1}^{n} S_{ji} w_j (f_j - f_0) \quad (i = 1, 2, \ldots, n), \tag{8b}$$

where ∂f_k is the kth component of $\partial\mathbf{f}_{9\times1}$. The following nodal weighting function used by Duarte and Oden [8] is applied in the present study:

$$w_i = \sqrt{4\pi^{-1}}(1 - |\Delta\mathbf{x}_i/d_0|^2)^4. \tag{9}$$

The functional derivatives $\partial\mathbf{f}_{9\times1} = (\partial_x, \partial_y, \partial_x^2, \partial_x\partial_y, \ldots, \partial_y^3)^{\mathrm{T}} f|_{\mathbf{x}_0}$ are obtained by solving Eq. (8), i.e., by inverting the matrix $[S^{\mathrm{T}}][W][S]$ to give

$$\partial\mathbf{f} = ([S^{\mathrm{T}}][W][S])^{-1}[S^{\mathrm{T}}][W]\Delta\mathbf{f} = [C]\Delta\mathbf{f}, \tag{10}$$

where $[C] = ([S^{\mathrm{T}}][W][S])^{-1}[S^{\mathrm{T}}][W]$.

The matrices $[S^{\mathrm{T}}]$, $[S]$ and $[W]$ are all dependent only on the geometrical distances between the reference node and its neighbours. For stationary boundary problems (for which the mesh-free nodes are fixed), one needs only to invert the matrix of $[S^{\mathrm{T}}][W][S]$ in Eq. (8) once to obtain the functional derivatives $\partial\mathbf{f}$ as a linear function of the nodal values $\Delta\mathbf{f}$ for each of the nodes.

3. Numerical procedures

3.1. Classification of computational nodes

We are interested in the class of incompressible viscous flows in domains whose boundaries may subject to prescribed motion in time. In particular, we are interested here in the motion of internal boundaries associated with moving rigid bodies immersed in flow. These boundaries are defined by a set of mesh-free nodes denoted here as *boundary nodes* (see Fig. 2). To facilitate the discretization of the space around possibly complex body shapes or geometries, the immediate neighbourhood of the bodies/boundaries may be covered with a cloud or cluster of or mesh-free nodes that extends to a few boundary-layer thicknesses of the flow past the boundary. We term these the *cloud nodes*. The boundary and cloud nodes are associated with the body that they define and envelop respectively. The body and its enveloping cloud of mesh-free nodes are set against a background of *Cartesian* nodes. The Cartesian nodes are assumed to be fixed in their spatial positions in the present work and the motion of the bodies/boundaries is carried out against this background of fixed Cartesian nodes. The

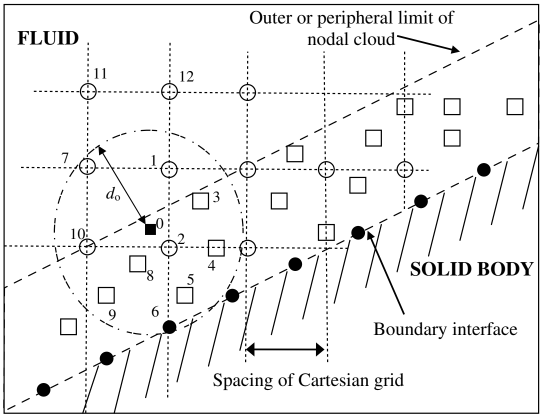

FLUID
Outer or peripheral limit of nodal cloud
11
12
7
1
d_o
3
10
0
2
4
8
5
9
6
SOLID BODY
Boundary interface
Spacing of Cartesian grid

Fig. 2. Support neighbours for reference: Node 0 are nodes 1, 3, 4, 5, 6, 7, 8, 9, while (Cat. 1) Cartesian overlapped nodes 2 and 10 are excluded as support nodes; Nodes 11 and 12 are (Cat. 2) Cartesian nodes using the standard FD treatment; Nodes 1 and 7 are (Cat. 3) Cartesian nodes requiring WLS-GFD treatment due to the presence of cloud nodes 0 and 3 within their $[-\Delta x, \Delta x] \times [-\Delta y, \Delta y]$ neighbourhoods. Boundary nodes ●; Meshfree cloud node □; Cartesian nodes ○.

boundary and cloud nodes will convect with the prescribed motion of their associated body. The boundary nodes can generally be regarded as a part of the cloud nodes. All of these are considered with respect to a single Cartesian reference frame.

The cloud nodes form a contiguous cover around the body. We define the *cloud cover* here (*cloud* for short) to be the contiguous region of space between the body and the outermost line of cloud nodes (Fig. 2). For the evaluation of spatial derivatives, four categories of the background Cartesian nodes may be defined. *Category 1* comprises the set of Cartesian nodes that are overlapped by a body or its cloud cover. These inactive nodes are termed the *overlapped* Cartesian nodes. The *Category 2* Cartesian nodes are nodes that have the exact 5-point Cartesian finite-difference nodal template (see nodes 11 and 12 in Fig. 2). These nodes are usually located at a distance from the body and cloud so that there are none of the cloud nodes within the closed rectangular $[-\Delta x, \Delta x] \times [-\Delta y, \Delta y]$ neighbourhoods of these nodes. The *Category 3* Cartesian nodes are Cartesian nodes around the peripheral limit of the cloud nodes. These nodes have one or more of cloud nodes within their $[-\Delta x, \Delta x] \times [-\Delta y, \Delta y]$ neighbourhoods. Examples are nodes 1 and 7 in Fig. 2, owing to the presence of cloud nodes 0 and 3. The last or *Category 4* Cartesian nodes are *fresh* nodes that are produced during the motion of a body or boundary. They are described in greater detail in Section 3.4. The nodal memberships of the four categories of Cartesian nodes will change with the motion of the body.

The overlapped or Category 1 Cartesian nodes are not directly involved in flow computation. The inexpensive standard 5-point central finite-difference scheme is applied for the evaluation of spatial derivatives at Category 2 Cartesian nodes, which usually form the bulk of the nodes in the computational domain. Spatial derivatives at Category 3 Cartesian nodes and cloud nodes around immersed bodies are evaluated by the mesh-free weight-least-squares generalized-finite-difference (WLS-GFD) scheme of the preceding section. We refer to these two groups of nodes broadly as the mesh-free nodes to reflect the manner spatial derivatives are evaluated. Each mesh-free node has an associated disk of radius d_0, which contains a sufficient number of *support* nodes for the evaluation of the derivatives (Fig. 2). The radius d_0 will in general vary inversely with the spatial density of the computational nodes.

3.2. The fractional step algorithm for fluid flow

The temporal evolution of the velocity and pressure fields $(\mathbf{u}, p)$ of an incompressible viscous fluid is governed by the Navier–Stokes equations,

$$\partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\nabla p + \frac{1}{Re}\nabla^2 \mathbf{u}, \tag{11}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{12}$$

presented in a non-dimensional form here, where $Re = UL/v$ is the Reynolds number based on a characteristic velocity U, characteristic length L and the kinematic viscosity of the fluid v.

The fractional-step or projection method by Chorin [38,39] is applied for the temporal treatment of the flow equations. A recent survey of projection methods for incompressible viscous flows was given by Brown et al. [40], with particular emphasis on the proper implementation of the pressure boundary condition. In the projection method, a modified version of the momentum equation (11) in discretized form is first advanced in time to provide an approximation $\mathbf{u}^*$ for the velocity field $\mathbf{u}^{n+1}$ at the next time level. The intermediate velocity field $\mathbf{u}^*$ will not, in general, satisfy the divergence-free condition for incompressible flow. The velocity correction or projection step involves the solution of a Poisson equation for pressure (or pressure correction) that is derived from the enforcement of the continuity equation (12). The pressure/pressure correction thus obtained is used to correct or project $\mathbf{u}^*$ onto $\mathbf{u}^{n+1}$ in the subspace of divergence-free solution. The pressure equation varies with the different variants of the fractional step method – some solve the pressure directly while others may employ an auxiliary variable related to pressure.

The momentum equation (11) is first discretized into a semi-implicit form here by the application of the second-order Crank–Nicolson scheme:

$$\frac{\mathbf{u}^{n+1}-\mathbf{u}^{n}}{\Delta t}=\frac{1}{2}\left\{\left(-\nabla p-\mathbf{u}\cdot\nabla\mathbf{u}+\frac{1}{Re}\nabla^{2}\mathbf{u}\right)^{n}+\left(-\nabla p-\mathbf{u}\cdot\nabla\mathbf{u}+\frac{1}{Re}\nabla^{2}\mathbf{u}\right)^{n+1}\right\}. \tag{13}$$

The present fractional step procedure involves the following splitting of Eq. (12):

$$\frac{\mathbf{u}^{*}-\mathbf{u}^{n}}{\Delta t}=\frac{1}{2}\left\{\left(-\nabla p-\mathbf{u}\cdot\nabla\mathbf{u}+\frac{1}{Re}\nabla^{2}\mathbf{u}\right)^{n}+\left(-\mathbf{u}\cdot\nabla\mathbf{u}+\frac{1}{Re}\nabla^{2}\mathbf{u}\right)^{n+1}\right\}, \tag{14}$$

$$\frac{\mathbf{u}^{n+1}-\mathbf{u}^{*}}{\Delta t}=-\frac{1}{2}\nabla p^{n+1}. \tag{15}$$

Applying the divergence operator to Eq. (15) and invoking the continuity condition Eq. (12) for the velocity field at the $(n+1)$ time level yields the pressure Poisson equation:

$$\nabla^{2}p^{n+1}=\frac{2}{\Delta t}\nabla\cdot\mathbf{u}^{*}. \tag{16}$$

The computational algorithm involves stepping Eq. (14) forward in time to give the intermediate velocity field $\mathbf{u}^*$ subject to the following condition on the boundary $\partial\Omega$ of the flow domain:

$$\mathbf{u}^{*}_{\partial\Omega}\approx\left(\mathbf{u}^{n+1}+\frac{\Delta t}{2}\nabla\hat{p}\right)_{\partial\Omega}, \tag{17}$$

where $\hat{p}$ represents the latest available approximation to the value of p^{n+1}.

With $\mathbf{u}^*$ thus determined, the pressure Poisson equation (16) is now solved subject to the following Neumann boundary condition:

$$\mathbf{n}\cdot\nabla p^{n+1}|_{\partial\Omega}=\frac{2}{\Delta t}[\mathbf{n}\cdot(\mathbf{u}^{*}-\mathbf{u}^{n+1})]_{\partial\Omega}, \tag{18}$$

which is derived by taking the dot product of Eq. (15) with the outward surface normal $\mathbf{n}$ on the boundary. The solution of the pressure Poisson equation (16) then yields the pressure field p^{n+1}, which is then substituted into Eq. (15) to determine the velocity field $\mathbf{u}^{n+1}$ at all the interior nodes of the flow domain. The typical implementation of the projection method involves the iteration of the first fractional-step equation to determine $\mathbf{u}^*$. This is then followed by the solution of pressure Poisson equation and projection onto $\mathbf{u}^{n+1}$ to complete the time step. The present scheme is similar to that of Kim and Moin [41], who employed p^n in place of $\hat{p}$ in (17). Kim and Moin [41] showed that in their three-level time scheme, the estimate p^n for p^{n+1} is sufficient to give second-order accuracy in the velocities. In the present implementation, the fractional-step equations (14)–(18) are solved together in an iterative manner at each time level. This ensures good overall numerical stability and accuracy under condition of continuously changing boundary. Importantly, the iterative approach allows us to update $\hat{p}$ towards p^{n+1}

at each time level. This enables Eq. (13) to be fully reconciled, thus ensuring the scheme is fully second order, according to the study presented by Brown et al. [40].

3.3. Treatment for convecting nodes

The previous section describes the computational formulation for stationary nodes in a fixed frame of reference. They are not applicable to the boundary nodes and the cloud nodes that convect with the motion of the boundary. The corresponding equations for arbitrary convecting nodes with nodal velocity $\mathbf{u}_c$ can be easily derived by applying Taylor expansion in space and time to account for functional variation due to the motion of the node. The consequence of nodal motion is simply to transform the Navier–Stokes equation into the following slightly modified form:

$$\partial_t \mathbf{u} + (\mathbf{u} - \mathbf{u}_c) \cdot \nabla \mathbf{u} = -\nabla p + \frac{1}{Re} \nabla^2 \mathbf{u}, \tag{19}$$

which had appeared in the context of the arbitrary Lagrangian–Eulerian (ALE) formulation of Hirt et al. [18]. When $\mathbf{u}_c$ is chosen to be equal to the local fluid velocity $\mathbf{u}$, Eq. (19) tracks the variation of $\mathbf{u}$ for a fixed particle of fluid, and $\mathbf{u}$ is then simply the Lagrangian flow field. The fractional-step equations for convecting nodes are essentially the same as those for stationary nodes with the exception of Eq. (14), which must now be replaced by

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = \frac{1}{2}\left\{ \left(-\nabla p - (\mathbf{u} - \mathbf{u}_c) \cdot \nabla \mathbf{u} + \frac{1}{Re} \nabla^2 \mathbf{u} \right)^n + \left(-(\mathbf{u} - \mathbf{u}_c) \cdot \nabla \mathbf{u} + \frac{1}{Re} \nabla^2 \mathbf{u} \right)^{n+1} \right\}. \tag{20}$$

At a solid boundary, the no-slip velocity condition dictates that $\mathbf{u} = \mathbf{u}_c$, where $\mathbf{u}_c$ is the prescribed (Lagrangian) velocity of the solid surface, which is non-zero for moving solid boundaries. The boundary velocity $\mathbf{u}^{n+1}|_{\partial\Omega}$ in the boundary conditions (17) and (18) are then simply the known prescribed velocity of the boundary $\mathbf{u}_c^{n+1}$ at the $n + 1$ time level. For interactive boundary problem $\mathbf{u}_c^{n+1}$ will be determined as part of the numerical iteration subject to fulfillment of other additional dynamical conditions.

3.4. Numerical treatment of 'fresh' Cartesian nodes

In the course of its motion, a body and its nodal cloud will overlap new Cartesian nodes in its path. The overlapped nodes then become the inactive Category 1 nodes, which are not directly involved in flow computation. The forward motion of the body and cloud also uncovers Cartesian nodes in its rear. Fig. 3(a) shows node 0 as a Category 1 node under the cover of the nodal cloud at time level n; it is not involved in flow computation at this time level. The forward motion of the nodal cloud uncovers node 0 at time level $(n + 1)$ in Fig. 3(b). Cartesian nodes thus uncovered by the motion of the nodal cloud are termed *fresh* nodes or *Category 4* Cartesian nodes. Fresh node 0 thus uncovered is not able to immediately partake in flow computation at the time level $(n + 1)$ because it lacks the requisite flow field data at the preceding time level n. To prepare fresh node 0 for full participation in flow computation at the next time level $(n + 2)$, its flow field data $(\mathbf{u}, p)$ at the current time level $(n + 1)$ is derived by applying Taylor's series expansion Eq. (1). There are a number of ways this can be done; by forward extrapolation from the nearest one or more cloud nodes using Eq. (2). A somewhat more implicit scheme is used here, which involves backward interpolation from the nearest one or more cloud nodes to the fresh node:

$$\Delta u_1^{n+1} = \left(\Delta x_1^{n+1}, \Delta y_1^{n+1}, \frac{1}{2}(\Delta x_1^{n+1})^2, \frac{1}{2}(\Delta y_1^{n+1})^2, \Delta x_1^{n+1} \Delta y_1^{n+1} \right) \cdot \partial \mathbf{u}_0, \tag{21}$$

where $\Delta u_1^{n+1} = u_1^{n+1} - u_0^{n+1}, \Delta x_1^{n+1} = x_1^{n+1} - x_0^{n+1}, \Delta y_1^{n+1} = y_1^{n+1} - y_0^{n+1}$, and subscript 1 refers to values at the nearest cloud node (see Eq. (2)). In Eq. (21), $\partial \mathbf{u}_0$ consists of (i) the elements of the $[C]$ matrix of the fresh node (which are purely geometrical and already determined), (ii) the variables at its neighbouring nodes $\mathbf{u}_i$, $i = 1, 2, \ldots, n$, neighbours, and (iii) the unknown value of u_0^{n+1} itself. The only unknown in this equation is

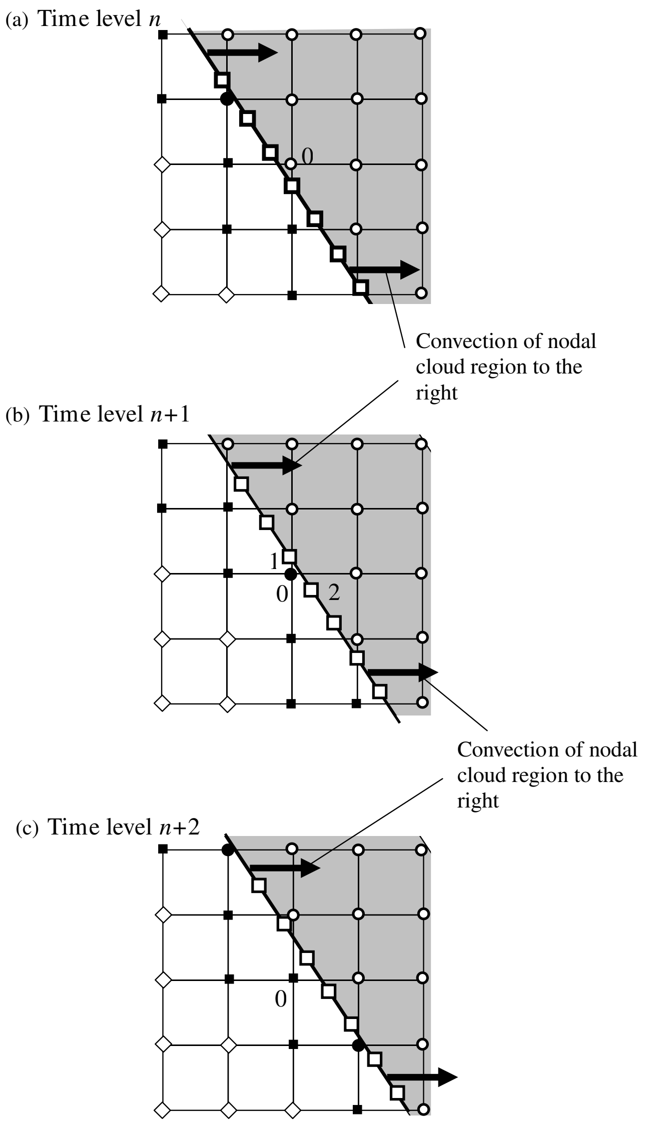

(a) Time level n
0
Convection of nodal cloud region to the right
(b) Time level n+1
1
0
2
Convection of nodal cloud region to the right
(c) Time level n+2
0

Fig. 3. Convection of nodal cloud region (shaded grey) to the right: (a) Node 0 is an inactive overlapped node (Cat. 1) at time level n, (b) Node 0 becomes a fresh node (Cat. 4) at time level $n+1$, with flow values interpolated from cloud nodes 1 and 2, (c) Node 0 becomes a (Cat. 3) node requiring WLS-GFD treatment. Cloud nodes □; overlapped Cartesian nodes (Cat. 1) ○; Standard FD treated Cartesian nodes (Cat. 2) ◆; WLS-GFD treated Cartesian nodes (Cat. 3) ■; Cartesian fresh nodes (Cat. 4) ●.

u_0^{n+1}, which can be solved for. If two nearest cloud nodes 1 and 2 (Fig. 3(b)) are used to give two estimates of u_0^{n+1}, say $(u_0^{n+1})_1$ and $(u_0^{n+1})_2$, then a weighted approximation such as the following may be used:

$$u_0^{n+1} = \frac{r_2(u_0^{n+1})_1 + r_1(u_0^{n+1})_2}{r_1 + r_2}, \quad \text{where } r_k = \sqrt{(\Delta x_k^{n+1})^2 + (\Delta y_k^{n+1})^2}. \tag{22}$$

An alternative approach for interpolation may be through the use of polynomial functions, as described in [26]. With $(\mathbf{u}, p)^{n+1}$ thus determined, fresh node 0 is ready to participate fully in flow computation as a computational Cartesian node at time level $(n + 2)$. Since there are cloud nodes in its close neighbourhood (see Fig. 3(c)), node 0 will be a Category 3 Cartesian node at the time level $(n + 2)$ and will require the mesh-free WLS-GFD spatial derivative treatment. The treatment of the overlapped and fresh nodes reflects the manner in which nodal deletion and addition may be carried out in the present scheme.

In the present hybrid Cartesian–meshfree grid scheme with WLS-GFD for spatial discretization, interpolation of data is confined primarily to acquiring flow field values for fresh nodes. While this is a positive feature of the present method, the WLS-GFD discretization scheme generally requires more supporting neighbours per node than the usual mesh-based methods. Considering the two-dimensional Taylor series expansion (1), to obtain the gradients $\partial_x f$ or $\partial_y f$ with second-order accuracy, the expansion needs to include terms up to the second-order derivatives. With 5 unknown derivative terms, at least 5 neighbour nodes are required. To obtain second-order accuracy for the second-order derivatives, e.g. $\partial_{xx} f$, at least 9 neighbours are required. Usually more nodes than the minimum required are used for improved accuracy. This implies a heavier computational cost for every mesh-free node (i.e., the cloud nodes and Category 3 Cartesian nodes) that requires the WLS-GFD treatment. However, the burden is not too excessive in exchange for the many advantages that it confers because the population of mesh-free nodes in most problems is usually much smaller than the population of the Category 2 Cartesian nodes (a mere few percent) that require only inexpensive Cartesian central FD evaluation.

3.5. Computational sequence

The various components of the present computational scheme for moving bodies in fluid have been explained in the preceding sections. The overall solution sequence is summarized below, where references are made to Sections 3.1 and 3.4 for nodal classification:

(i) Generate an initial set of background Cartesian nodes, boundary nodes (which define the body) and the cloud nodes around the body.
(ii) Identify the inactive Category 1 Cartesian nodes that do not require flow computation.
(iii) Identify Cartesian nodes (Cat. 3) that require the mesh-free WLS-GFD treatment (see Section 3.1).
(iv) Identify the supporting nodes for each of the mesh-free nodes (cloud nodes and Cat. 3. Cartesian nodes) and calculate their coefficient matrices $[C]$ in accordance with Eq. (10).
(iv) Prescribe initial and boundary conditions, and begin simulation.
(v) At each time level n:
  (a) Update and store required nodal variables.
  (b) Obtain new positions and velocities of cloud nodes at time level $n+1$ arising from the motion of the body. For prescribed rigid body motion, which is considered here, the cloud nodes convect with the body in fixed relative positions.
  (c) Identify the overlapped (Cat. 1) and 'fresh' (Cat. 4) Cartesian nodes that are formed at time level $n+1$ due to the motion of the body (see Section 3.4). Also identify the Category 3 Cartesian nodes that require WLS-GFD treatment.
  (d) Identify supporting nodes and compute $[C]$ matrices for all mesh-free nodes.
  (e) Iterate:
  - update latest values of $\mathbf{u}^{n+1}$ into the intermediate momentum Eq. (14) for the Cartesian (stationary) nodes and Eq. (20) for the convecting cloud nodes, and solve for $\mathbf{u}^*$. WLS-GFD discretization is used for cloud nodes and the Category 3 Cartesian nodes, while standard FD discretization is applied to the Category 2 Cartesian nodes.
  - Solve for pressure Poisson equation (16) using boundary conditions Eqs. (17) and (18).
  - Obtain $\mathbf{u}^{n+1}$ using Eq. (15). Iterate for a fixed number of cycles or until satisfactory convergence.
(vi) At the end of each time step, obtain flow field data at the 'fresh' Cartesian nodes (Cat. 4) (see Section 3.4).
(vii) Step forward in time and repeat (v)–(vi) until the desired time level.

The present methodology and its procedures may be naturally extended to three-dimensional (3D) problems without essential changes – beside obvious ones like having to perform all necessary geometric operations and to compute the GFD matrices in three dimensions.

4. Numerical experiments I

In this section, we first verify that the present fractional-step implementation of the weighted-least-square generalized finite-difference (WLS-GFD) scheme for two-dimensional incompressible fluid flows in the primitive variables (u, v, p) is essentially second order in space and time. Then we examine errors that are associated with the ALE implementation of the WLS-GFD scheme for prescribed convecting nodes.

4.1. Spatial and temporal accuracy

To test the spatial accuracy of the numerical formulation described in the preceding section, the problem of *decaying vortex* was first simulated. This has been a fairly popular test problem for accuracy of numerical schemes [41,42] because it has a theoretical solution:

$$\begin{aligned} u^{(e)}(x,y,t) &= -\cos\pi x\sin\pi y \mathrm{e}^{-2\pi^2 t/Re}, \\ v^{(e)}(x,y,t) &= \sin\pi x\cos\pi y \mathrm{e}^{-2\pi^2 t/Re}, \\ p^{(e)}(x,y,t) &= -\frac{1}{4}(\cos 2\pi x+\cos 2\pi y)\mathrm{e}^{-4\pi^2 t/Re}. \end{aligned} \tag{23a–c}$$

Here the Reynolds number is $Re = UL/\upsilon$, where U is the maximum velocity at the initial time $t = 0$, L the size of the vortex and υ, the kinematic viscosity of the fluid. Superscript (e) denotes the exact solution.

The computational domain spans $[-0.5, 0.5]\times[-0.5, 0.5]$ and the Reynolds number is set at $Re = 10$. Simulations are carried with both square Cartesian and ‘uniformly distributed’ random nodes. The latter are generated by applying a random perturbation to the former with two-dimensional nodal displacement of up to 15% of the Cartesian mesh size. Ding et al. [36] had carried out a brief study of different schemes of grid point generation and the performance of the WLS-GFD in solving the Poisson problems. For the present study, the decaying vortex is simulated for the following four configurations:

Nodal distribution	Spatial discretization	Velocity boundary condition	Pressure boundary condition
Cartesian	Central-space FD	Dirichlet	Dirichlet
Cartesian	WLS-GFD	Dirichlet	Dirichlet
Uniformly random	WLS-GFD	Dirichlet	Dirichlet
Uniformly random	WLS-GFD	Dirichlet	Neumann

The first configuration is the reference case, which employs the standard 5-point central-space second-order finite-difference approximation. For the next three configurations, spatial discretization is carried out with the WLS-GFD method. Dirichlet boundary conditions are applied for the velocity $(u, v)^{n+1}$ and pressure p^{n+1} in all cases with the exception of the last, where the Neumann boundary condition Eq. (18) is adopted for the pressure. The initial velocity and pressure fields at $t = 0$ are provided by the exact solutions Eqs. (23), and the solution is marched forward in time numerically until $t = 0.4$, at which time the maximum norm of the error is computed.

The spatial accuracy is investigated by varying the average grid interval h. For the case of randomly distributed nodes, the average grid interval can be taken to be $h = \sqrt{A/N}$, where A is the area of the computational domain and N is the total number of computational node points. The marching time step is set at a small value of $\Delta t = 10^{-5}$ for all the above cases. This is to ensure that the contribution of temporal error to the numerical error is kept very small compared to the spatial error. The numerical error is defined to be the maximum norm of the difference between the numerical and exact solutions, i.e. $\|u - u^{(e)}\|_\infty = \max_\Omega\{| u - u^{(e)}|\}$.

Fig. 4 provides a comparison of the errors in the horizontal velocity u and pressure $p$for the four configurations as a function of the average grid interval h. The gradients of the error curves show immediately that the WLS-GFD scheme indeed has second-order accuracy. Not unexpectedly, the standard second-order

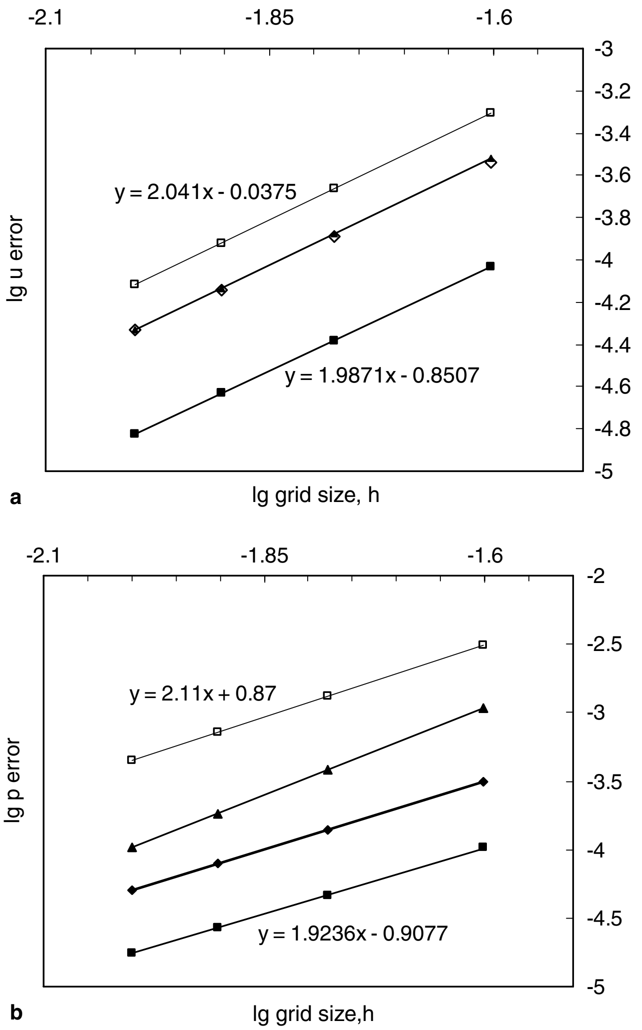


Fig. 4. Log$_{10}$ of maximum errors in u and p for the decaying vortex Eq. 23 at $t = 0.4$ and time step $\Delta t = 10^{-5}$. Best-fit straight lines displaying a gradient of 2. (■) error under standard FD scheme, (◆) error under WLS-GFD in Cartesian grid, (▲) error under WLS-GFD in random grid, (□) error under WLS-GFD in random grid with Neumann boundary conditions. (a) velocity u (b) pressure p.

finite-difference scheme on square Cartesian grid has the smallest numerical error, while the WLS-GFD scheme with randomly distributed nodes and Neumann boundary condition for pressure has the largest error.

What this comparison indicates is that the WLS-GFD is second-order in all cases including the worst case of randomly scattered nodes with Neumann boundary conditions. However, due to the latter's comparatively large absolute errors, such random arrangement of nodes is clearly to be avoided. Indeed the mesh-free literature is filled with frequent discussions concerning the desirability of maintaining star-quality in node distributions, while mesh regularity is a key issue in preserving low discretization errors in FE methods. Consequently in the applications of our hybrid Cartesian–Meshfree discretization scheme, we have selected to use 'layered' mesh-free grids that offer partial 'grid orthogonality' at boundaries whenever possible. Alternatively, triangulation routines, such as those available in FE packages, can also be used to generate near star-quality nodal arrangements at boundaries.

The temporal accuracy of the fractional-step scheme was investigated for flow in a square box [0, 1] × [0, 1] with the following initial field state:

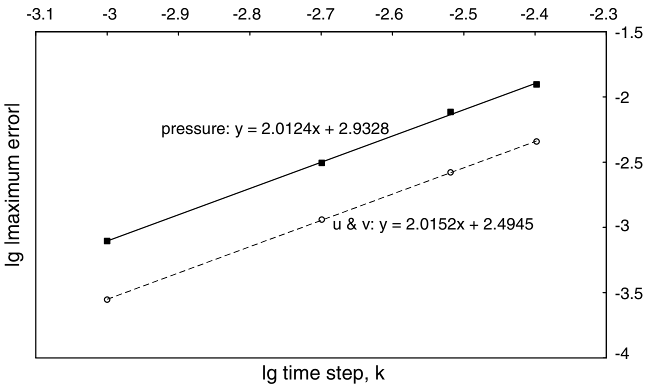


Fig. 5. Log_{10} of maximum errors in u and p for the decaying function Eq. (23) at $t = 0.1$ referenced with computational solution derived at time step $k = 0.00001$. Best-fit straight lines displaying a gradient of 2. — pressure, --- u and v.

$$\begin{aligned} u(x,y,0) &= (1-\cos 2\pi x)\sin 2\pi y, \\ v(x,y,0) &= (\cos 2\pi y - 1)\sin 2\pi x, \\ p(x,y,0) &= 0, \end{aligned} \tag{24a–c}$$

and a Reynolds number of $Re = 1000$. The no-slip boundary condition is applicable at the four walls. This problem has been used by a number of researchers, including Perot [43], to evaluate the temporal accuracy of their numerical schemes.

An orthogonal Cartesian mesh of 81×81 grid points was used for this study, in which the reference numerical solution $(u, v, p)^{(r)}$ at time $t = 0.1$ was obtained with a small time step size of $\Delta t = 10^{-5}$. The problem was repeated for larger time steps, and the maximum errors for velocity $\|u - u^{(r)}\|_\infty$ and pressure $\|p - p^{(r)}\|_\infty$ at $t = 0.1$ were recorded. Fig. 5 shows the variation of the errors with the size of time step Δt. The gradients of the error curves show that the method is second order in time.

4.2. Decaying vortex with convecting nodal patch

The present work concerns the use of mesh-free convecting nodes to model the prescribed motion of solid bodies or boundaries against a background grid of Cartesian nodes. In the course of motion of the body, some Cartesian background nodes are eliminated from computation while new ones are reinstated into active simulation as they emerge from under the cover of the cloud nodes. In order to understand whether a convecting cloud or patch of mesh-free nodes will cause any stability problems or create any additional computational errors, a test problem is deliberated. The aim is to check the numerics of the moving cloud methodology in isolation, by quantifying the errors associated purely with the motion of a patch of active mesh-free nodes during the simulation of the flow, without the assumption of a physically moving body.

The test problem is the decaying-vortex problem Eqs. (23), in which the discretized computational domain comprises a square Cartesian grid overlaid with a small patch or cloud of mesh-free nodes initially residing near the bottom left-hand corner of the domain (see Fig. 6). The convecting mesh-free cloud is a small 5 by 5 square nodal array with a nodal interval of $0.8h$, where h is the grid spacing of the background Cartesian grid. The convecting nodes are treated by the WLS-GFD method in the ALE formulation, while the standard FD scheme is applied at the stationary Cartesian background nodes whenever possible. The decaying vortex is set for $Re = 100$ and the nodal patch is convected with a velocity of $u = v = U\sin(2\pi t/T)$, where $U = 0.25$ and $T = 6.0$. The decaying vortex is simulated for a series of Cartesian grid spacings h, at a constant time step of $\Delta t = 0.002$, and the solutions at $t = 1.0$ are obtained for comparison with the exact solution Eqs. (23). The maximum errors, the L2 error norms (defined by $\sqrt{\sum_{i=1}^{N}\left(f_i - f_i^{(e)}\right)^2/N}$) and root-mean-square (RMS) error norms (defined by $\sqrt{\sum(f - f^{(e)})^2/\sum(f^{(e)})^2}$) of velocity and pressure as functions of grid spacing h are

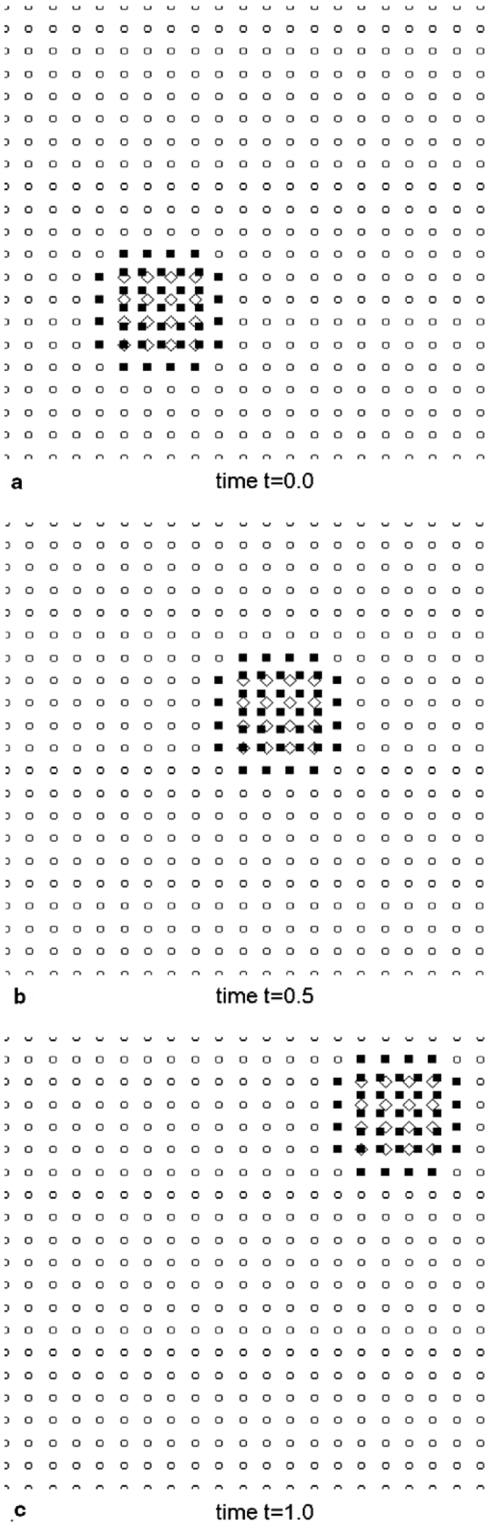


Fig. 6. Motion of moving patch across the decaying vortex. (○) computational Cartesian node, (■) meshfree node, (◇) exterior node.

depicted in Fig. 7. The gradients of the error curves show that the solutions with the convecting nodal patch retain second-order spatial accuracy.

To gain a better understanding of the effects that Cartesian node removal and addition (fresh nodes) have on the solution during the convection of the nodal patch, we decided to record the variation of numerical errors with time and to compare these against the numerical errors of similar computations carried out on the same Cartesian grid without the nodal patch. The central-space FD scheme was used in the latter case. The ratio of the root-mean-square (RMS) errors in the ALE-WLS-GFD moving-patch solution to those in the stationary-node Cartesian-grid FD solution are computed as a function of time and shown in Fig. 8. Two neighbours are used for fresh node interpolation using Eq. (22).

It is observed that the errors in the convecting-patch solution get amplified at regular intervals. Analysis reveals that the intervals correspond precisely with the rate at which the convecting nodal patch is overlapping new Cartesian background nodes in front and uncovering 'fresh' Cartesian nodes in its rear. The removal and addition of the Cartesian nodes during simulation clearly alter the composition and spatial configuration of the nodes involved in the WLS-GFD computation around the edge region of the convecting patch. This is the direct cause of the temporal fluctuations in the numerical errors of the moving-patch solution. The constantly changing nodal distribution does not, however, destabilise the solution process. The errors in the pressure solution grow during the initial phase of patch motion but stay well bounded to less than 1.2 times the errors of the standard FD solution (no convecting patch) over the solution interval. The errors in the velocity solutions, on the other hand, suffer a minor initial rise, but quickly decay to levels similar to those of the standard FD solution with time. It is worth noting that, from Fig. 7, the enhanced levels of error of the moving patch solution above are no worse than the relative errors between the WLS-GFD solutions and the standard Cartesian FD solutions for stationary-node cases with the same average grid interval, presented earlier in

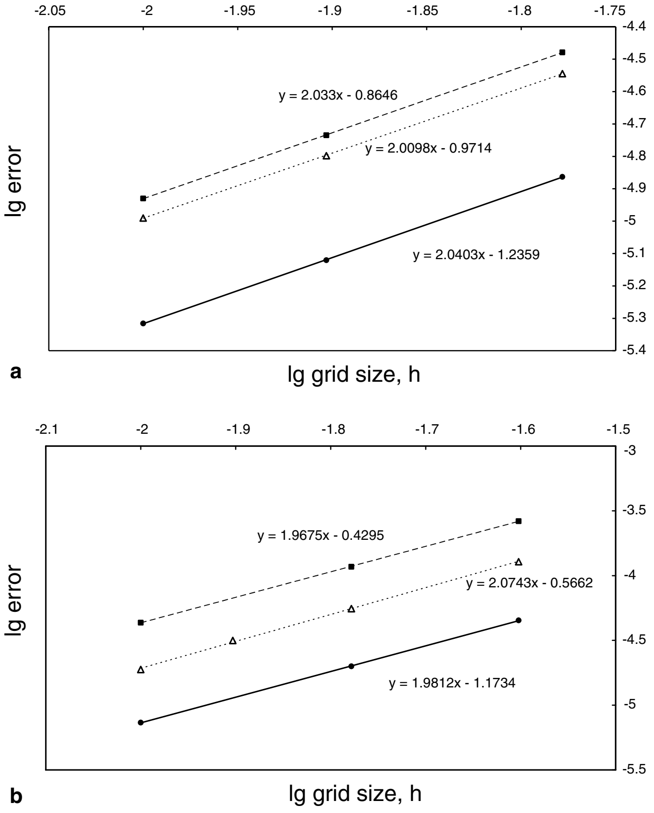


Fig. 7. Spatial accuracy level of 2 for the WLS-GFD method under ALE formulation: (a) ■ lg(RMS error u), ● lg(L2 error u), △ lg(maximum error u); (b) ■ lg(RMS error p), ● lg(L2 error p) △ lg(maximum error p).

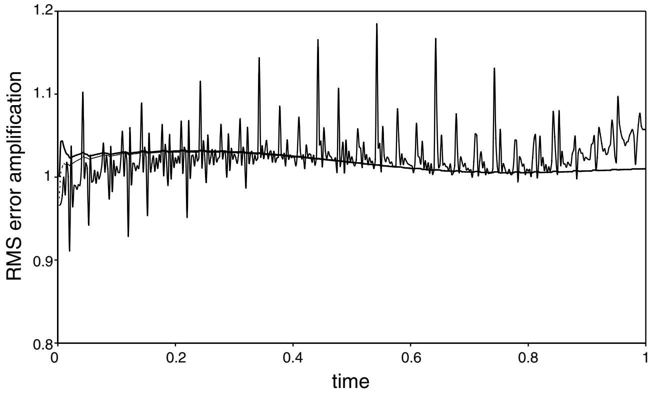


Fig. 8. Amplification of RMS errors of u, v, p vs time, comparing the errors in a stationary domain with the moving patch against the errors in a domain without the moving patch. Based on two-neighbour interpolation; -- u error, — v error, ▬ p error.

Fig. 4. This suggests that the increased errors of the moving patch solution, which arose from the removal and addition of nodal points and changing nodal configurations, is no larger than the inherent discretization error of the WLS-GFD approximation.

The next series of results involves the application of the method to some well-known flow problems with stationary boundaries.

4.3. Flow past stationary circular cylinders

The first series of test was carried out to verify that the numerical scheme is able to simulate the simpler case of flow past stationary two-dimensional circular cylinders with a good degree of accuracy. Cylinders subject to external open flow and channel flow were simulated. It is well known that external flow past a stationary circular cylinder is steady and symmetric at low Reynolds numbers of up to 40, where *Re* is based on the diameter D of the cylinder and the free stream speed U_∞ of the oncoming flow. The flow in the wake takes the form of a symmetric pair of steady counter-rotating vortices, whose streamwise extent increases with *Re*. Above the *Re* of 40, the wake becomes unstable and breaks into unsteady flow comprising periodic build-up and alternate shedding of vortices from the two sides of the cylinder.

For this series of studies, the cylinder, which is defined by its boundary nodes, is enveloped by a few layers of cloud nodes and set against a background of uniform square grid (Fig. 9). This in turn is embedded within a larger computational domain of non-uniform stretched grid to cater for free stream simulation, where the boundaries of the computational domain have to be placed at a sufficient distance from the solid body (e.g. Fig. 10). The non-dimensional external boundary conditions are:

at the inflow boundary,

$$\mathbf{u} = (1,0), \tag{25a}$$

at the outflow boundary,

$$(\partial_t + \partial_x)\mathbf{u} = (0,0), \tag{25b}$$

and along the top and bottom fluid boundaries,

$$\partial_y \mathbf{u} = (0,0). \tag{25c}$$

On the surface of the cylinder,

$$\mathbf{u} = \mathbf{u}_c, \tag{25d}$$

where $\mathbf{u}_c$ is the prescribed velocity of the cylinder surface, which is zero if the cylinder is stationary. The boundary conditions for the intermediate velocity $\mathbf{u}^*$ and pressure p are implemented as described in Section 3.2.

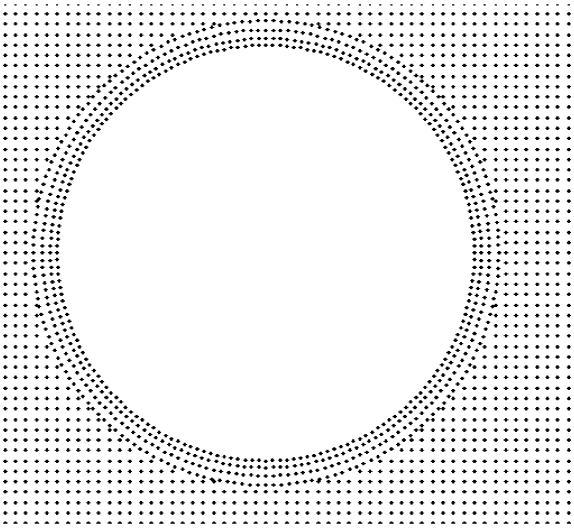


Fig. 9. Nodal arrangement for meshless cloud around circular cylinder.

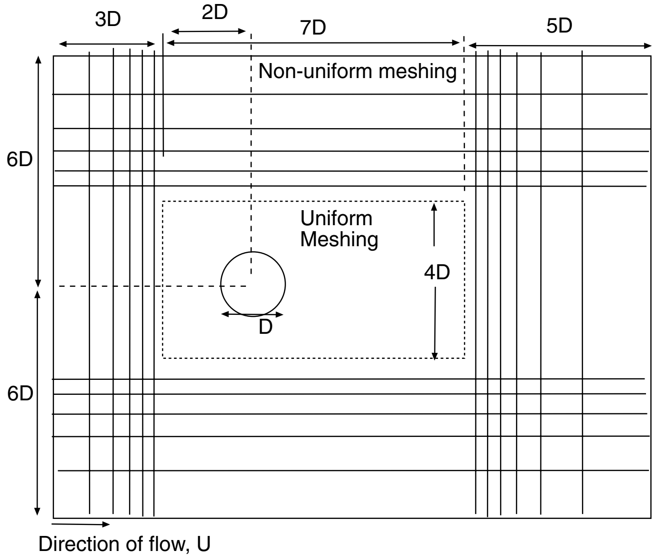


Fig. 10. Flow domain of computation for the free stream, open field flow.

Free stream, low Reynolds number steady state flows were first solved. Referring to Fig. 10, the computational domain is of size $15 \times 12D^2$. A total of 1260 cloud nodes and boundary nodes are used, which make up just 1.5% of the total number of 85,869 nodes employed in this external flow simulation. The cylinder centre is located at $5D$ downstream of the inflow boundary. Flows at $Re = 10$, 20 and 40 were simulated, and the results are shown in Table 1. The recirculation bubble length, separation angle and drag coefficient for each of the above flows compare very well with results from the reference publications.

The pressure distributions over the cylinder surface at $Re = 20$ are computed for two sizes of computational domain $15 \times 12D^2$ and $19 \times 19D^2$. The results are compared against similar results from He and Doolen [45] in Fig. 11. Closer agreement is observed for results obtained in the larger domain. The difference between the two sets of computed results shows clearly the significant effect of domain size on the results at this low Re. This may also partially explain the spread in the published results in Table 1.

Table 1
Comparison of recirculation lengths, separation angles and drag coefficients of flows around circular cylinder at $Re = 10$, 20 and 40 for open field flow

Re	References	Recirculation length, $2L/D$	Separation angle (deg)	Drag coefficient, C_D
10	Present	0.56	28.6	3.09
	a. Nieuwstadt (1973)	0.434	28.0	2.828
	b. Coutanceau (1977)	0.68	32.5	
	c. He and Doolen [45]	0.474	26.9	3.17
	d. Dennis and Chang (1970)	0.53	29.6	2.846
	e. Guo et al. [44]	0.533	31.6	
20	Present	1.90	44.2	2.19
	a	1.79	43.4	2.053
	b	1.86	44.8	
	c	1.84	42.9	2.152
	d	1.88	43.7	2.045
	e	1.87	42.3	
40	Present	4.6	53.47	1.59
	a	4.357	53.34	1.55
	b	4.26	53.5	
	c	4.49	52.84	1.499
	d	4.69	53.8	1.522
	e	4.4	53.13	

References a–d taken from e.

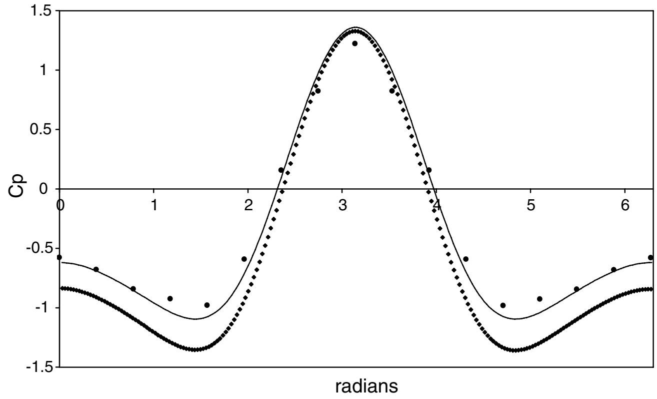


Fig. 11. Plots of surface pressure coefficient, C_p at $Re = 20$; ··· results in domain space of $15 \times 12D^2$, — results in domain space of $19 \times 19D^2$, ● results from [45].

Next, using the same spatial setting, unsteady flow at $Re = 100$ was simulated. A timed sequence of vorticity plots are presented in Fig. 12. The time histories of the drag and lift coefficients are illustrated in Figs. 13 and 14 respectively, where a Strouhal number $St = Df/U_\infty$ of 0.17 may be deduced, where f is the stabilized frequency of the lift fluctuation and vortex shedding. This is close to the value of 0.167 cited in [46–48].

A number of runs were also made for flow past the cylinder in a channel of width $4D$. The computational domain extends $2D$ ahead of the cylinder and ends $5D$ after the cylinder in the streamwise direction and is coverd by a square Cartesian grid with grid interval of 0.025. The cylinder surface is discretized by 188 boundary nodes and enclosed by 3 layers of cloud nodes that extend up to a distance $0.06D$ from the surface. The out-flow boundary conditions remain as before, while the non-slip flow condition $\mathbf{u} = (0, 0)$ is applicable at the channel walls. The parabolic in-flow velocity profile is given by $\mathbf{u} = (1.5 - 6(y/H)^2, 0)$ (x-axis located along the centre line $y = 0$ of the channel), which give a mean u velocity of 1.0 across the inflow.

The channel flow cases actually offer a better test of accuracy from a numerical viewpoint because the flow boundary conditions can be more accurately defined here than for open field, free-stream cases, where the

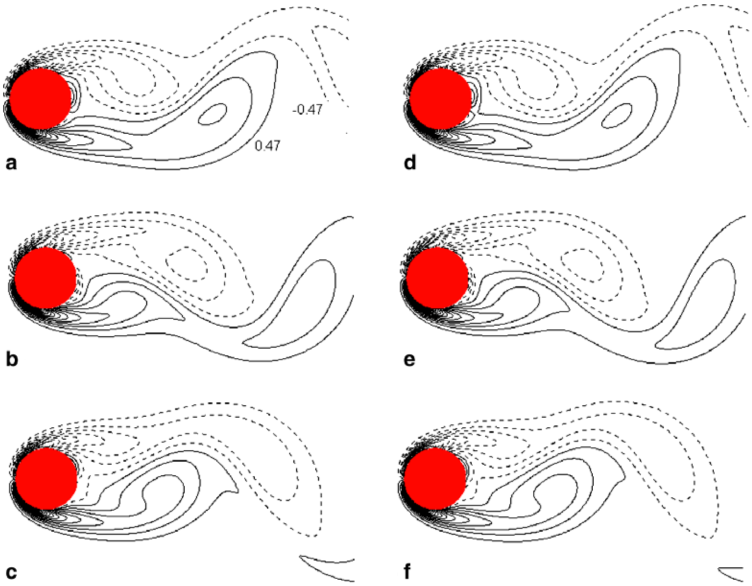


Fig. 12. Vorticity plots of stationary cylinder in free stream at $Re = 100$, from time $t = 65$ to $t = 75$. Sequence starts from top left down to top right down. Negative vorticity is denoted by dotted lines. Vorticity lines of values ±0.47 indicated in (a), with subsequent intervals of 0.94.

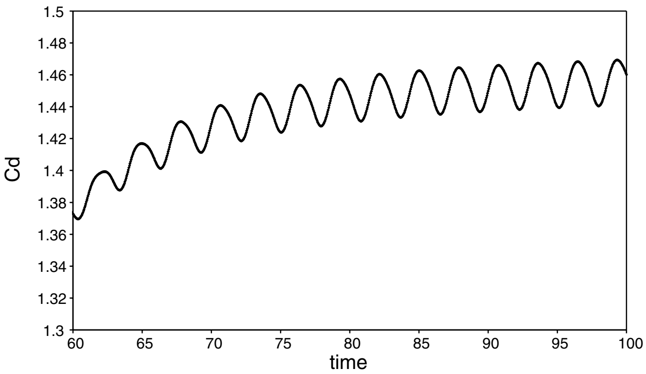


Fig. 13. Drag coefficient, C_D, against time for open field, free stream flow around cylinder at $Re = 100$.

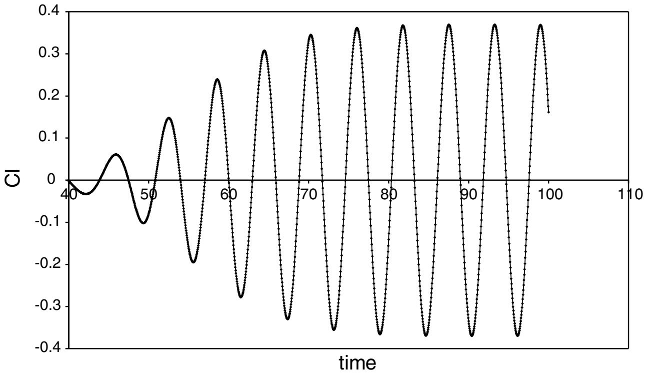


Fig. 14. Lift coefficient, C_L, against time for open field, free stream flow around cylinder at $Re = 100$.

Table 2
Comparison of Strouhal numbers, drag and lift coefficients of flows across circular cylinder at $Re = 100$ for channel flow and open field flow with blockage

$Re = 100$ Channel flow	St	$\overline{C}_D$	$C_{D,max}$	$C_{D,min}$	$C_{L,max}$	$C_{L,min}$
Lilek et al. [47]	0.2944	2.908	2.931	2.884	0.7418	−0.7672
Present	0.2985	3.016	3.048	2.975	0.7599	−0.7602
$Re = 100$ Open field; blockage = 0.25	St	$\overline{C}_D$				
Stansby and Slaouti [46]	0.222	1.910				
Present	0.224	2.011				

computational domain has to be sufficiently large. The computed quantities of the flow at $Re = 100$ are compared against the published results of Lilek et al. [47] in Table 2. The agreement is very good for all the computed flow properties. The squeezing effect of the channel walls, compared to the open flow, results in a higher Strouhal number.

Free stream flow with blockage of 25% was also computed for comparison with Stansby and Slaouti [46]. This is an artificial flow situation in which the non-slip impermeable boundary condition $\mathbf{u} = (0, 0)$ at the walls of the above channel flow is now replaced by the free stream velocity condition of Eq. (25c). The present results for Strouhal number and mean drag again compare very well with the published results (see Table 2).

To illustrate simulation of higher Reynolds numbers, channel flow at Re of 1000 is simulated, where a finer mesh-free grid interval of 0.0125 on the cylinder surface and 0.0167 for the Cartesian background grid were used to capture the finer features of the flow. The channel width is now $5D$ indicating a blockage ratio of 0.2. Fig. 15 shows the resulting vorticity evolution. A mean drag coefficient and Strouhal number of 3.08 and 0.32 were obtained which compares fairly well with the values of 3.0 and 0.3 from Zovatto and Pedrizzetti [49] respectively. The Strouhal number is less accurate, partly due to the difference in the channel lengths used for our simulation ($10D$) and the referred results ($55D$). It is also a well-known fact that flows of higher Reynolds numbers are generally more difficult to solve. Apart from requiring higher grid resolution, artificial dissipation or upwind treatment may be required to maintain numerical stability. The present scheme does not incorporate any numerical dissipation treatment at this point of time, although this may be needed eventually for applications at higher Reynolds numbers.

4.4. Flow past stationary odd-shaped cylinder

An odd-shaped cylinder of cross-sectional width $D = 1.0$ is subjected to a free stream flow as in Section 4.3. Fig. 16 shows the geometry of the cylinder and the mesh-free nodal cloud generated around the cylinder by a triangulation routine. Periodic vortex shedding is observed for the cylinder at $Re = 200$. Fig. 17(a) and (b) show the instantaneous streamlines and vorticity contours around the cylinder. The streamlines and vorticity contours can be seen to extend through a narrow gap into the cavity on the underside of the cylinder, suggesting the presence of weak flow re-circulations within the cavity. This example demonstrates the facility of the present hybrid Cartesian–Meshfree to handle objects with complex geometry that may not be easily amenable to treatment by conventional boundary-fitting O- or C-grid discretization.

5. Numerical experiments II

The full ALE-WLS-GFD scheme is applied to a series of problems involving moving circular cylinders: impulsively and sinusoidally-started cylinders, symmetric and asymmetric flows past longitudinally oscillating cylinders and flows past transversely oscillating cylinders with lock-on and lock-off vortex shedding. The results are compared against solutions obtained from equivalent moving-frame simulations and published results in literature. These cases provide a test of the ability of the scheme to handle moving body problems.

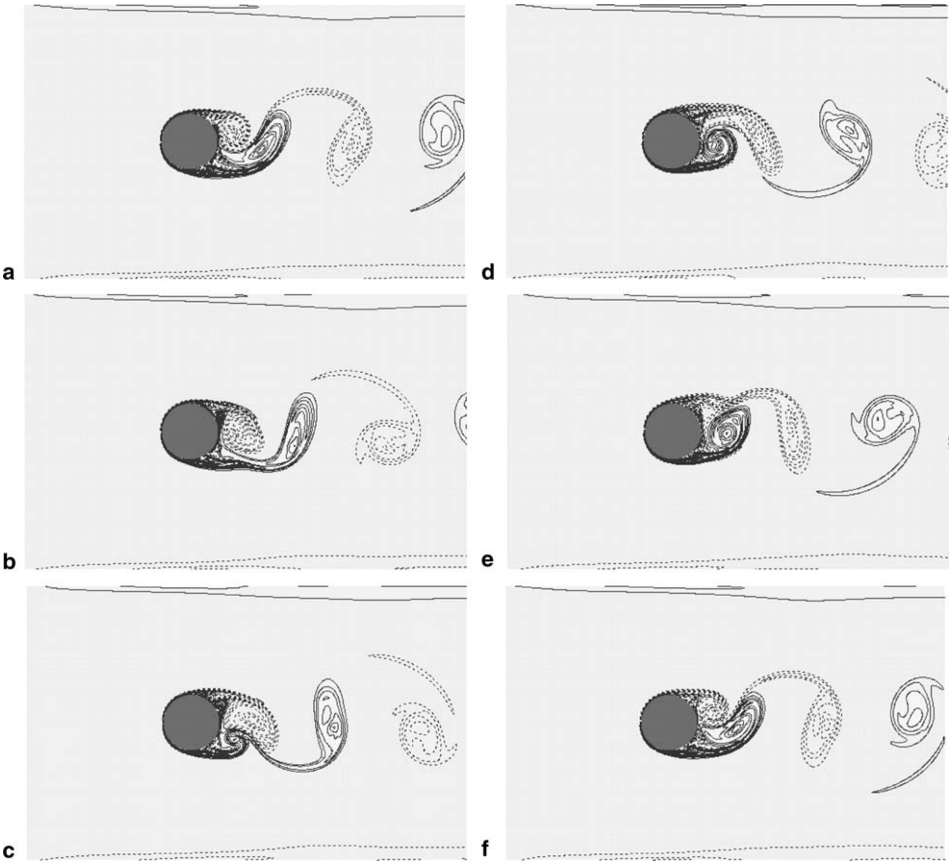


Fig. 15. Vorticity plots of channel flow past stationary cylinder at $Re = 1000$. Sequence starts from top left down to top right down, from time $t = 77$ to time $t = 79.5$. Negative vorticity is denoted by dotted lines.

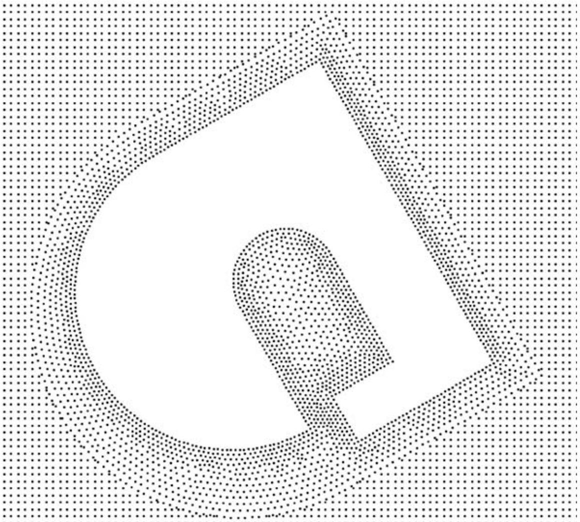


Fig. 16. Nodal arrangement of an odd-shaped cylinder in a free stream flow.

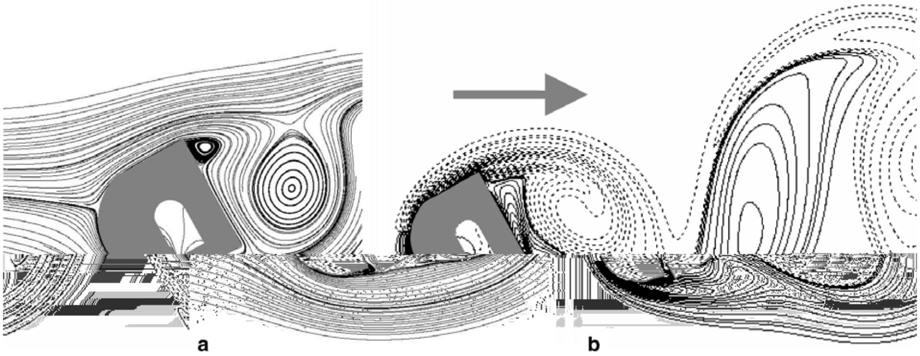


Fig. 17. (a) Streamline plot of free stream flow past stationary odd-shaped cylinder at $Re = 200$. (b) Vorticity plot of similar cylinder. Direction of flow indicated by arrow, from left to right. Negative vorticity is denoted by dotted lines.

5.1. Accuracy test on ALE implementation

The first task is to verify the accuracy of our numerical method when handling problems with moving boundary. A common test case to perform is the oscillating circular cylinder (or sphere for 3D) in a box enclosure, as performed by both Udaykumar et al. [26] and Gilmanov and Sotiropoulos [27].

A circular 2D cylinder of size $D = 0.5$ is placed in a square box of length $2D$ and is made to oscillate at a frequency of $f = 1/T = 1/0.4\pi$, at $Re = 40$. Since there is no known exact solution for this problem, a reference solution is first computed for time $t = 0.3$, with the cylinder surface grid spacing of $h^{(\text{reference})} = 0.00455$ $(2D/220)$, and background grid size of 0.005. Subsequently, three runs were conducted with varying surface (and background) spacings of $h = 0.01$, 0.0126 and 0.0167. The obtained flow fields at $t = 0.3$ are compared with the reference solution, and the maximum errors in velocity and pressure are noted. Fig. 18 shows the log–log curves of the errors against the surface grid spacing. The gradients are found to be close to the value of 2, signifying that the order of accuracy is still preserved at 2 after implementing the ALE formulation.

5.2. Flow past impulsively and sinusoidally started circular cylinders

As further tests of the present ALE implementation, flows past sinusoidally and impulsively started cylinders in an otherwise stationary fluid medium are computed. The ALE results for moving cylinders against a fixed stationary Cartesian grid (the present scheme) are compared against similar results computed in a moving/

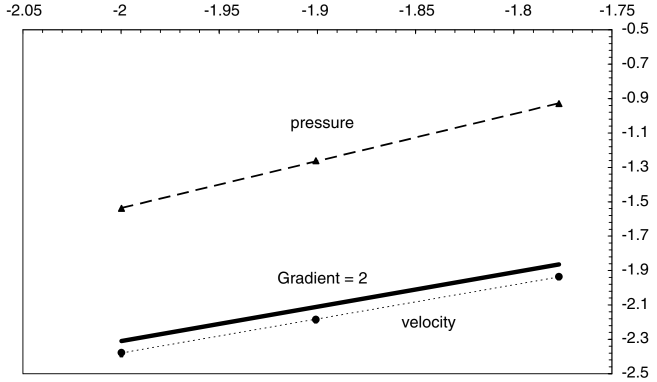


Fig. 18. Log_{10} of maximum errors in u, v, p for the oscillating circle in a box enclosure at $t = 0.3$ referenced with computational solution derived at cylinder surface grid size, $h = 0.00455$. Best-fit straight lines display a gradient of 2. –▲– pressure, --●-- u and v.

accelerating reference frame attached to the moving cylinders. In the latter mode, the cylinders have fixed coordinates, and the governing NS equations incorporate a frame acceleration term:

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla)\mathbf{u} = -\nabla p + \frac{1}{Re}\nabla^2 \mathbf{u} + \frac{\partial \mathbf{u}_{\mathrm{frame}}}{\partial t}. \tag{26}$$

Furthermore, the boundary conditions of the computational domain would be modified to reflect their values in the moving reference frame of the cylinder. No ALE/moving-node algorithm or 'fresh' node creation are involved here at all. The moving or accelerating frame method has been an established method because of its great simplicity. However, it is greatly limited in that it cannot be applied to flow problems involving two or more bodies undergoing relative motions, and even to single body moving within a finite enclosure. It is through the use of this established method that we can derive comparison in results to ensure that our scheme is stable and accurate.

Simulations using the stationary reference frame were conducted similar to the previous external flow cases, with the incorporation of the ALE moving body algorithm for the starting cylinders. Eq. (25c) is applicable at the top and bottom sides of the domain and the convective boundary condition is now applied at the left and right boundaries of the domain:

$$(\partial_t + u_{\mathrm{mean}}\partial_x)\mathbf{u} = (0, 0), \tag{27}$$

where u_{mean} denotes the average inlet/outlet velocity along the boundaries. One can also replace it with the local flow velocity at the previous time step, u^n. For the impulsively started cylinder,

$$\mathbf{u}_c(t) = \mathbf{u}_c(0) = (1, 0), \tag{28a}$$

and for the sinusoidally started cylinder,

$$\mathbf{u}_c(t) = (\sin 2\pi t/T, 0), \tag{28b}$$

where T is the period of cylinder oscillation.

For the moving or accelerating frame simulations, the boundary conditions are

$$u_{\mathrm{frame},\partial\Omega}(t) = -u_c(t) = -1, \tag{29a}$$

and

$$u_{\mathrm{frame},\partial\Omega}(t) = -u_c(t) = -\sin 2\pi t/T, \tag{29b}$$

for the impulsively and sinusoidally started cylinders respectively along the outer boundary $\partial\Omega$ of the computational domain. For the impulsively started cylinder, the sharp initial acceleration of the cylinder is carried out numerically by increasing the cylinder velocity linearly from 0.0 to 1.0 in a short time interval of $t = 0.02$.

Tang and Aubry [50] employed the impulsively started cylinder at $Re = 100$ to validate their numerical scheme, and their resulting drag coefficient history agreed well with other published results [51,52]. The drag coefficient histories of the present simulation are plotted and compared with that of [50] in Fig. 19. From the figure, it can be seen that the results from both the present ALE-WLS-GFD moving body formulation and the accelerating or moving frame formulation are generally consistent with each other and match well the results of Tang and Aubry. The moving frame results are obtained with WLS-GFD discretization on the modified NS Eqs. (26) with $\mathbf{u}_c \equiv 0$ for the cylinder. Similar to the impulsively started case, both the ALE moving body and the moving frame simulations are performed for a sinusoidally started cylinder. The resulting drag coefficient histories at $Re = 20$ can be seen in Fig. 20. The runs from $t = 0$ to $t = 1.5$ show that the two curves follow each other closely with generally small deviations.

Dynamically, the ALE formulation and the accelerating/moving frame formulations should be equivalent for the present cylinder cases. However, in numerical implementation, some minor differences in results may still arise due to the different configurations of the finite computational domains and the non-exact equivalence of the boundary conditions under the two approaches. Nevertheless, the results from the moving body simulations are seen to match very well corresponding results from the moving frame simulations, indicating that the moving-body ALE-WLS-GFD formulation is working very well indeed.

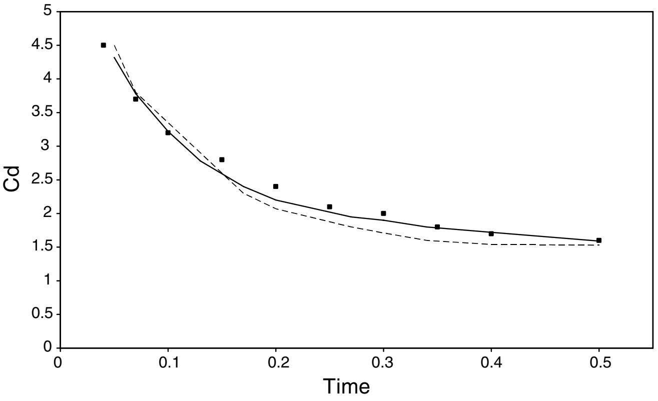


Fig. 19. History of drag coefficient, C_D, for the impulsively started cylinder at $Re = 100$: moving frame of reference vs moving cylinder with ALE formulation: (■) Ref. [50], (—) moving body, (---) moving frame.

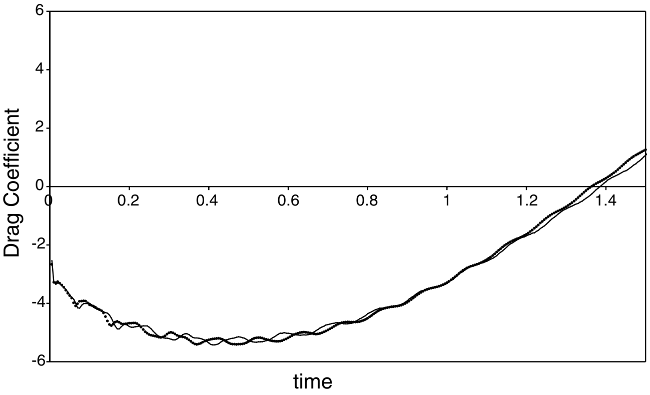


Fig. 20. Comparison of drag coefficient, C_D, of sinusoidally started cylinder at $Re = 20$: moving frame of reference vs moving cylinder with ALE formulation. (···) moving frame, (—) moving cylinder.

5.3. Transient flow past longitudinal oscillating circular cylinders

Computation is carried out for flow past circular cylinders undergoing sinusoidal longitudinal oscillations. This is an extension of the case for sinusoidally-started cylinder in the preceding section. The flow configuration remains the same. The flow is, however, simulated for a much longer computational time of about 20 oscillation cycles to allow the flow to reach a nearly stationary periodic state.

Depending on the Reynolds number $Re = U_{\max}D/\upsilon$ and the Keulegan–Carpenter number $KC = U_{\max}T/D$ (where $U_{\max}$ and T are the velocity amplitude and period respectively), the flow around the cylinder may develop either symmetric or asymmetric vortical structures. The boundary in the *Re–KC* parameter plane separating symmetric flows and asymmetric flows has been determined in visualization experiments by Tatsuno and Bearman [53] and also numerically by Iliadis and Anagnostopoulos [54]. The latter employed the finite element method and the streamfunction-vorticity (ψ, ζ) flow formulation with moving frames and oscillating boundary conditions.

Two cases with *Re–KC* parameters in the vicinity of the boundary delineating the two flow regimes are specially selected to test the capability of the present numerical scheme to develop the correct flow patterns. The first case with $Re = 200$ and $KC = 4$ is shown in Fig. 21. It displays a symmetric pattern of vortical flow formation through one half of an oscillation cycle. The pattern of vortex formation through the cycle compares very well with similar symmetric flow results presented by [54].

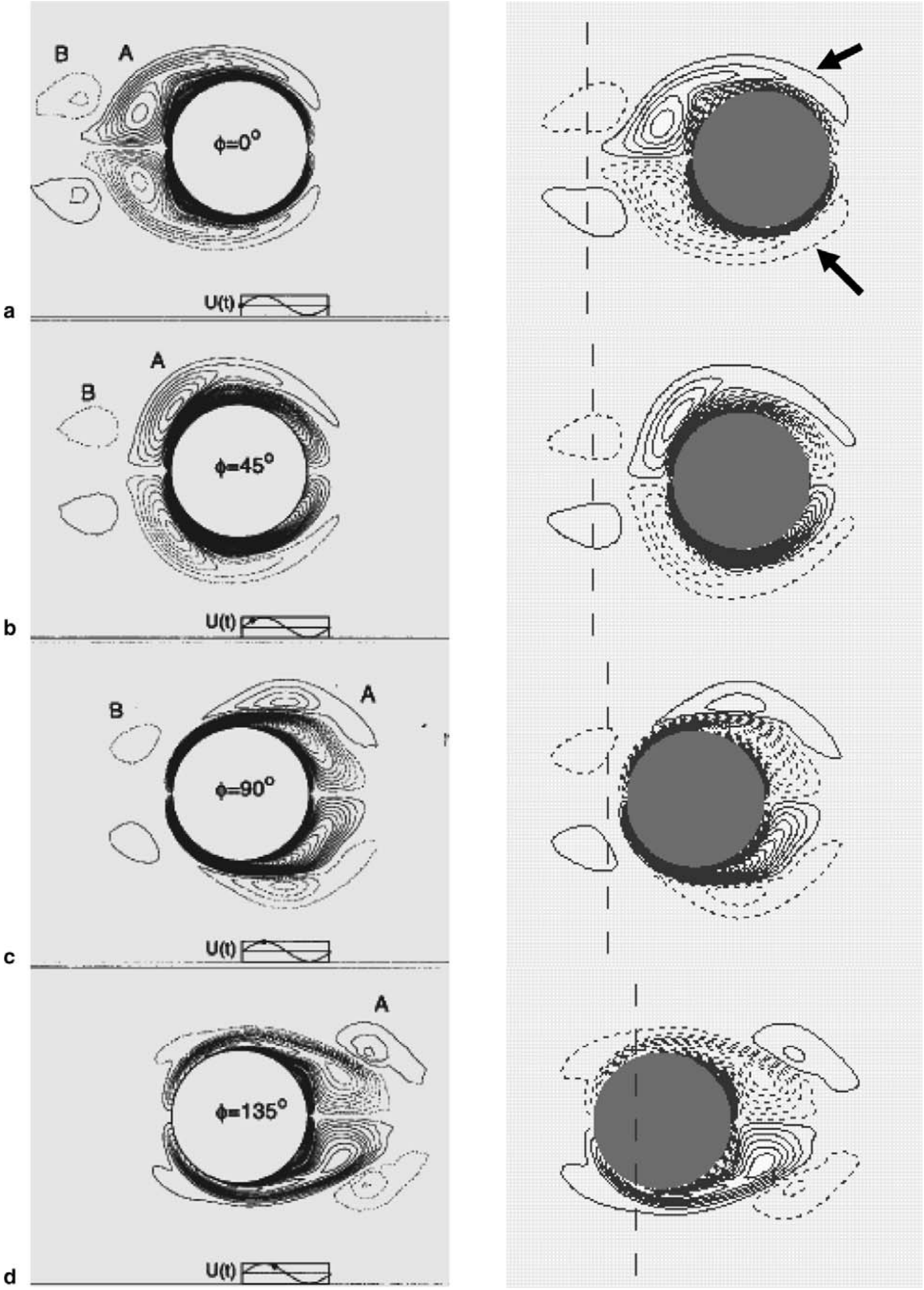


Fig. 21. Comparison of vortex shedding in half a cycle, $KC = 4$, $Re = 200$. Left-hand side: Ref. [54], right-hand side: present study. Negative vorticity denoted by dotted lines. Vorticity lines of values ±0.646 indicated in (a) by arrows, with subsequent intervals of 1.29. Dashed vertical line denotes the extreme left position of the cylinder centroid.

The second case was computed with $Re = 200$, $KC = 10$, which falls within the regime of asymmetric flows. Fig. 22 shows the comparison of the vorticity distributions with [54] in half an oscillation cycle, and Figs. 23 and 24 show the drag and lift histories compared to the similar results in [54]. The present simulation captures all the major vortical features of the flow around the cylinders, with some smaller differences arising in the weak vortical features further away from the cylinder. The drag and lift histories show good agreement with [54] with regards to the overall trend and phasing of the coefficient peaks. The peak values of C_D are also in

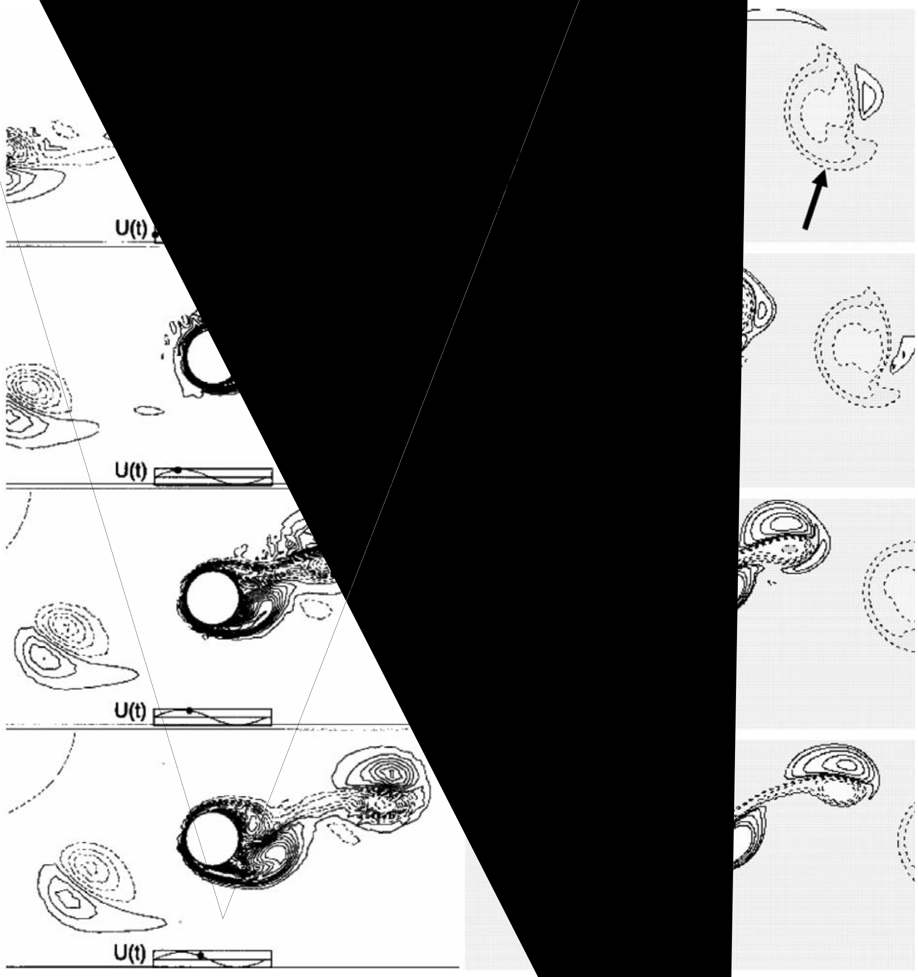
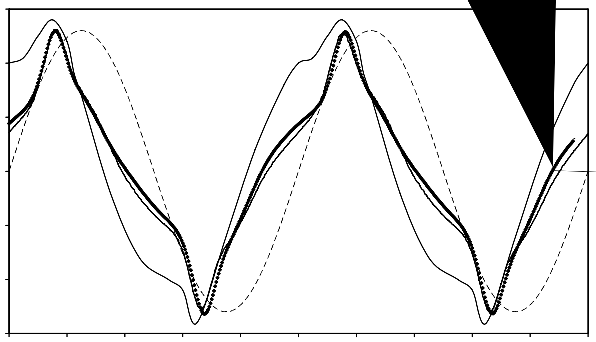


Fig. 22. Comparison of vortex shedding in half a cycle, $KC = 10$, $Re = 200$. Cyl[redacted]ghtmost position to leftmost position. Left-hand side: Ref. [54], right-hand side: present study. Negative vorticity denot[redacted]. Vorticity lines of values −0.308 and 0.463 indicated by arrows in (a), with subsequent intervals of 0.772.



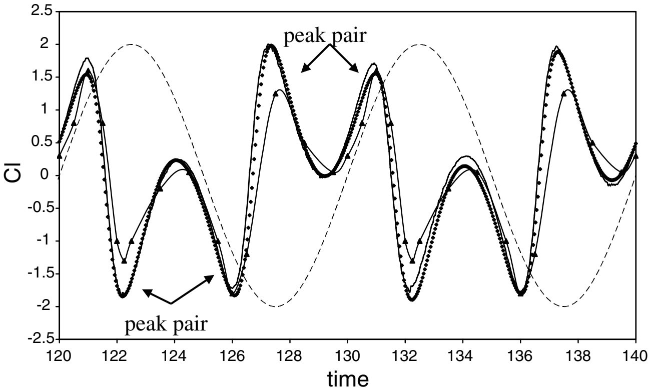


Fig. 24. History of lift coefficient, C_L, for the longitudinally oscillating cylinder at $Re = 200$, $KC = 10$: (▲) Ref. [54], (––) cylinder sinusoidal motion, (■) present moving frame, (—) present moving body (ALE scheme).

very good agreement, while there is some disagreement with regards to the inflexion regions in the C_D curves (Fig. 23) and the C_L peaks (Fig. 24).

The differences in the details with [54] may be attributable to the relatively limited size of our computational domain of $15 \times 12D^2$ compared to the much larger $40 \times 20D^2$ employed in [54]. The smaller domain may not be able to fully accommodate and account for the spatially extended systems of interacting vortices that develop in the asymmetric cases. It is worth noting, however, that the mentioned results from Iliadis and Anagnostopoulos [54] have not been verified against other experimental or numerical results, and the comparison above should hence be regarded purely on a qualitative basis. Probably of more significance in the present instance is the very good agreement and mutual consistency between the present moving body (ALE) results (see Figs. 23 and 24) and the physically equivalent moving frame results. The very good match with the moving frame solution demonstrates the stability of the present moving body ALE method for extended time computation.

5.4. Flow around transversely oscillating cylinder

Koopmann [55] carried out one of the first detailed experimental studies of wake formation behind transversely oscillating circular cylinders in external flow. He observed a phenomenon, which is now popularly known as *lock-on*, whereby the vortex shedding frequency f_v in the wake is identical to the driving frequency f_c of the oscillating cylinder. The lock-on of f_v onto f_c occurs in the vicinity of the natural shedding frequency f_n of the cylinder, which is the vortex shedding frequency of the corresponding stationary or non-oscillating cylinder. At $Re = 100$, the non-dimensional natural frequency $f_n = St = 0.17$ in Section 4.3. The lock-on region is the range of frequency about f_n for which the phenomenon of frequency lock-on occurs. The upper and lower limits of the frequency lock-on region may be determined by directly comparing the shedding frequency f_v in the wake against the driving frequency f_c, when the latter is progressively varied. The frequency lock-on region is dependent on the Re of the flow (which governs f_n) as well as the amplitude A of cylinder oscillation.

For the present ALE-WLS-GFD simulations, the transverse trajectory and velocity of the centre of the cylinder are given by $y_{cen} = A\sin(2\pi f_c t)$ and $v_{cen} = A2\pi f_c\cos(2\pi f_c t)$, with corresponding expressions for boundary nodes on the cylinder and the cloud nodes. Flows with $Re = 100$ and oscillation amplitude $A = 0.1$ and 0.2 are simulated in the present work. The ranges of frequencies studied are 0.86–1.19f_n and 0.775–1.28f_n respectively at the two amplitudes. Lock-on is reckoned to have occurred if $|f_v - f_c| < 0.04f_c$. Lock-on can be observed from the phasing of the lift curve relative to cylinder oscillation. The results are summarized in Fig. 25, where the full circular symbol ● denotes that lock-on had been observed while the empty square symbol □ denotes no lock-on. The occurrences of both are in very good accordance with approximate upper and lower frequency limits of the lock-on region established by [55].

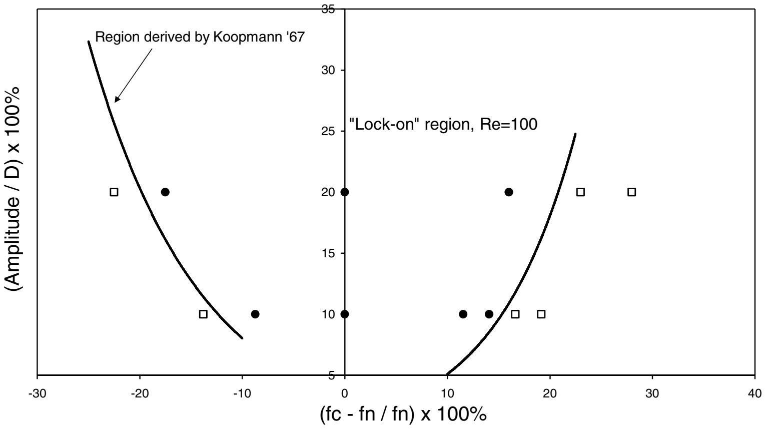


Fig. 25. Region of locking in of vortex street. Amplitude vs percentage change in cylinder driving frequency, f_c. (▬) upper and lower limits from [55], (●) 'locked on', (□) 'locked off'.

The characteristic features of the flow in the wake are also consistent with descriptions given by [55,56]. Corresponding wake formations for the same cylinder driven with amplitude $A = 0.2$ at frequencies $f_c = f_n$, $f_c > f_n$ and $f_c < f_n$ are depicted in Figs. 26–28 respectively. For a cylinder driven at its natural shedding frequency, i.e. $f_c = f_n$, they observed a decrease in the average lateral spacing between the shedded vortices, while the longitudinal spacing remains largely unchanged. Furthermore, the synchronous shedding of the vortices

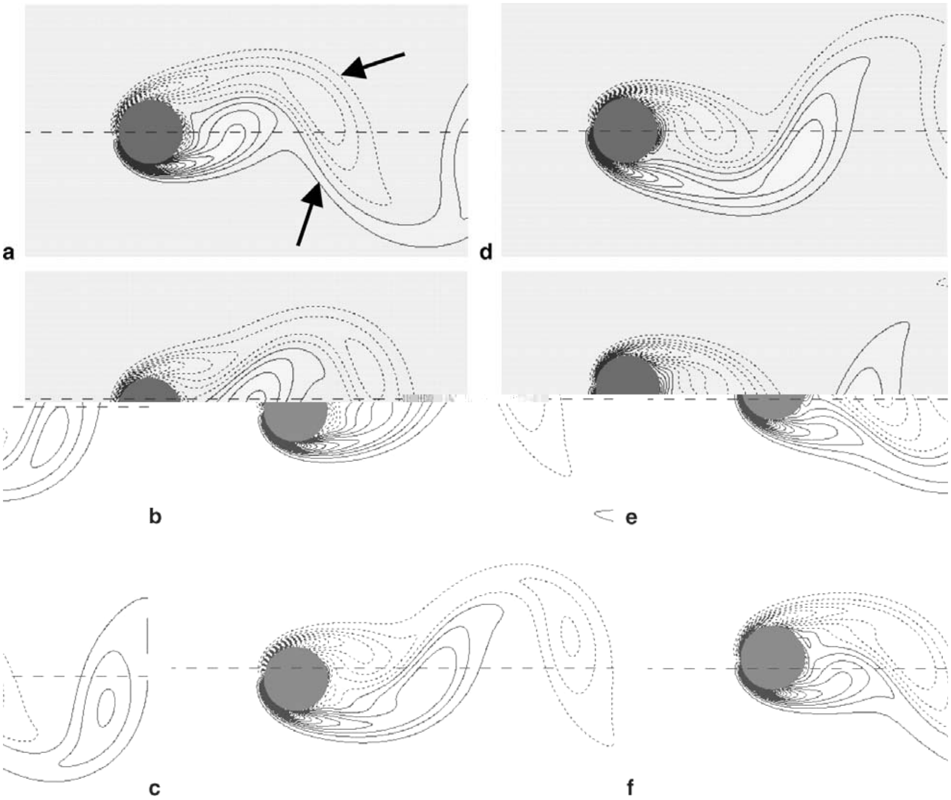


Fig. 26. Vorticity plots at $Re = 100, f_c = f_n$, $A = 0.2$. Negative vorticity denoted by dotted lines. Vorticity lines of values ±0.377 indicated in (a) by arrows, with subsequent intervals of 0.754. Sequence starts from top left down to top right down, from time $t = 73.5$ to 78.5. Centre dashed horizontal line denotes mean y position.

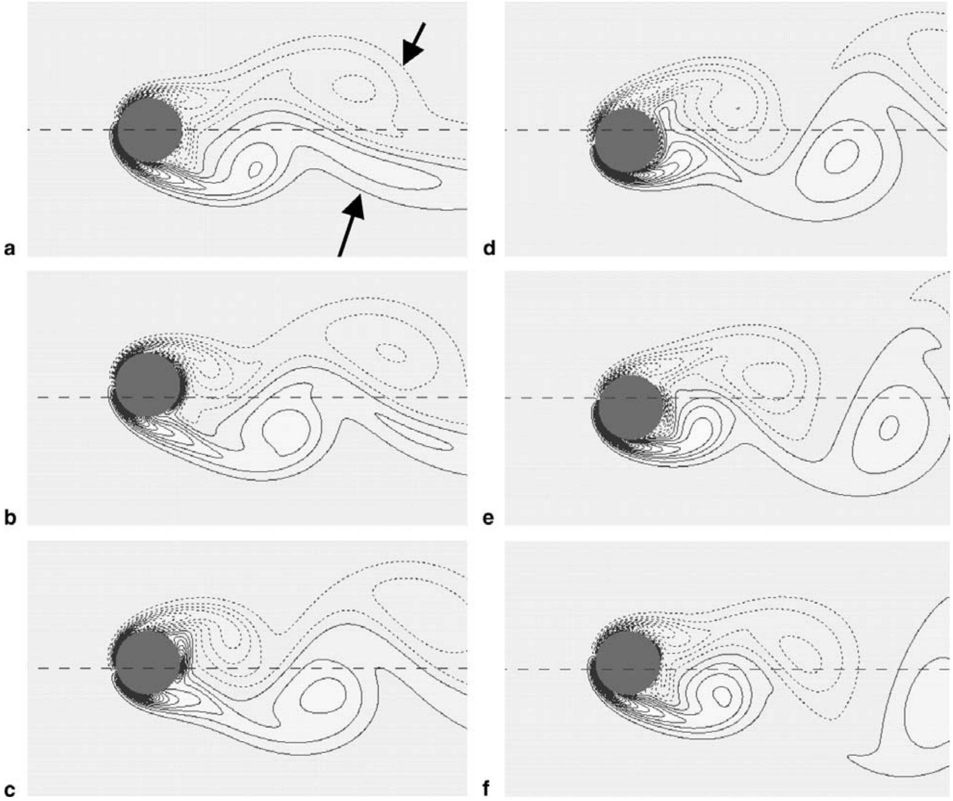


Fig. 27. Vorticity plots of oscillating cylinder at $A = 0.2, f_c = 128\% f_n$. Negative vorticity denoted by dotted lines. Vorticity lines of values ±0.391 indicated in (a) by arrows, with subsequent intervals of 0.782. Sequence starts from top left down to top right down, from time $t = 73.5$ to 78.5. Centre dashed horizontal line denotes mean y position.

occurs on the same side as displacement of the cylinder. These features are also reproduced in the present simulation.

When $f_c > f_n$, the vortices are more promptly shed (Fig. 27), which is consistent with the lock-on of the shedding frequency f_v to the higher values of f_c. The vortices are now shedded on the opposite side to the cylinder displacement in agreement with the observations of [55,56]. When $f_c < f_n$, referring to Fig. 28, vortex shedding appears more sluggish so much so that it approaches 'collinear' pattern, instead of well-defined counter-rotating vortices along the two sides of the vortex street. The longitudinal vortex interval is about $2.5D$ compared with $2.2D$ for $f_c = f_n$, and vortex shedding now occurs on the same side as cylinder displacement.

The above test cases for flows past starting cylinders, longitudinally oscillating and transversely oscillating cylinders show that present the WLS-GFD method under ALE formulation is an effective numerical scheme for simulating flows with moving solid bodies. The good agreement between the moving body and moving frame simulations also provides strong and ample evidence that the hybrid ALE FD-GFD methodology moving-body computation is working very well.

6. Numerical experiments III

Extensions to complex geometries and motions, including uncorrelated motions, of multiple bodies can be carried out with relative ease in the present moving-boundary formulation. We demonstrate this with three body-driven cases: a flapping elliptical cylinder, a pair of asynchronously counter-rotating elongated cylinders and an oscillating circular cylinder. All three flows have been carried out for confined flow within a rectangular box. The moving frame method, mentioned in the preceding section, cannot be used to treat this class of problems.

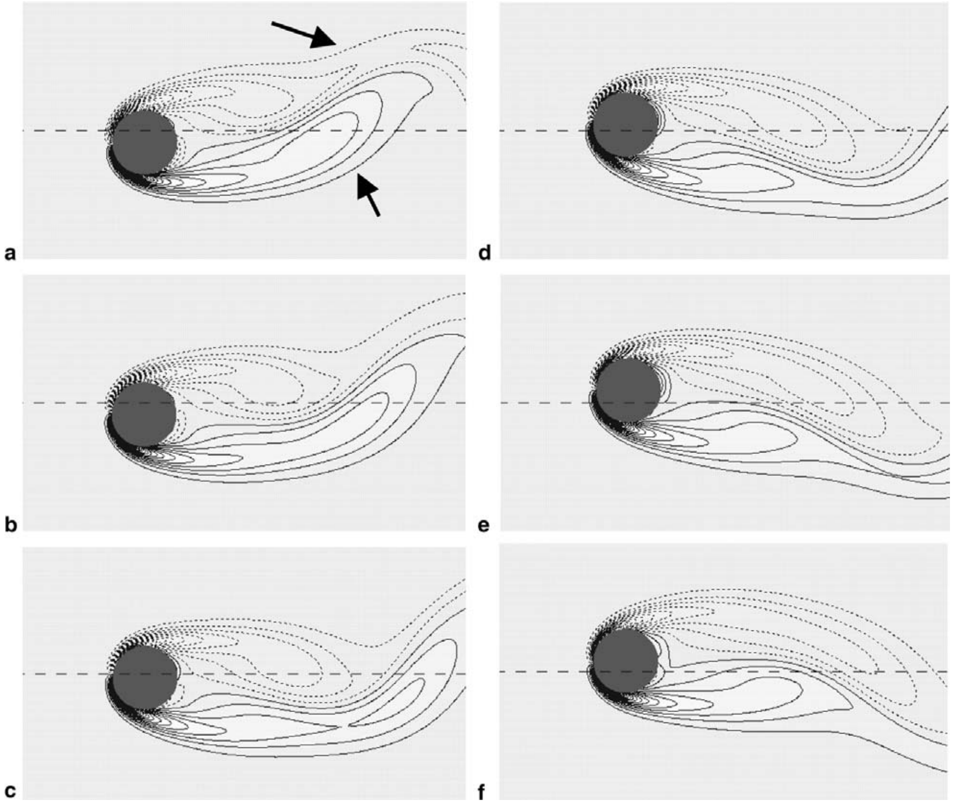


Fig. 28. Vorticity plots of oscillating cylinder at $A = 0.2, f_c = 77.5\% f_n$. Negative vorticity denoted by dotted lines. Vorticity lines of values ±0.333 indicated in (a) by arrows, with subsequent intervals of 0.667. Sequence starts from top left down to top right down, from time $t = 73.5$ to 78.5. Centre dashed horizontal line denotes mean y position.

6.1. Flapping elliptical cylinder in a box

Fig. 29 shows the arrangement of nodes around the flapping elliptical cylinder. Three elliptically-arranged layers of cloud nodes are employed. The ellipse, which has major and minor diameters of 0.5 and 0.15 respectively, is positioned in a square box spanning $[-1, 1] \times [-1, 1]$. The flapping motion involves the simultaneous translation and rotation of the ellipse. It is generated by an imaginary oscillating arm of length 2.5 and having its centre of oscillation at $(-1.5, 0)$. The arm is oscillated about the x-axis in accordance with the angular position and velocity functions $\theta = -\dot{\theta}_o T \cos(2\pi t/T)/2\pi$ and $\dot{\theta} = \dot{\theta}_o \sin(2\pi t/T)$ respectively. The trajectory schematic of the elliptic cylinder is shown in Fig. 30. The Reynolds number based on the major diameter and maximum tip velocity is $Re = 7.5$ in Figs. 31 and 32 ($T = 20$, $\dot{\theta} = 0.1$) and $Re = 22.5$ in Fig. 33 ($T = 10$, $\dot{\theta} = 0.3$).

The flow is simulated for a period of 5–6 cycles of flapping motion, and the resulting streamline and vorticity plots are shown in Figs. 31 and 32 respectively. Fig. 31 shows an approximate quarter cycle, with the ellipse in an upward motion (parts (a) to (b)). This is followed by a turnaround (part (c)) and the ‘flap’ back downwards (parts (d) to (f)).

The motion of the elliptic cylinder generates two tip vortices. They are shown in part (a) as vortex pair **A**, which is highlighted by dotted ovals. The tip vortices are shed as the cylinder approaches the top end of its trajectory (parts (a) to (c)). The more dominant tip vortex is the outer tip vortex $\mathbf{A_r}$ since the outer tip of the ellipse has the higher velocity. As the cylinder reverses its direction downwards, a new pair of counter rotating tip vortices $\mathbf{C_l}$ and $\mathbf{C_r}$ is formed (part (d)). The right vortex then merges with the nearby corner vortex **B** of the same sign to form a larger tip vortex $\mathbf{C_r}$ (parts (d) to (e)). Meanwhile the shed tip vortex $\mathbf{A_l}$ from the preceding up-flap motion disappears (parts (c)–(e)). Finally, the remaining outer tip vortex $\mathbf{A_r}$ (formed during

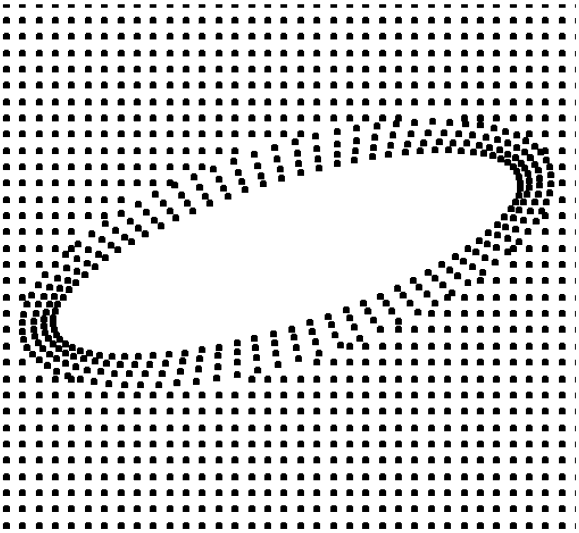


Fig. 29. Nodal arrangement of the flapping ellipse. The meshless cloud represented by three layers of nodes along ellipses of varied 'radii'.

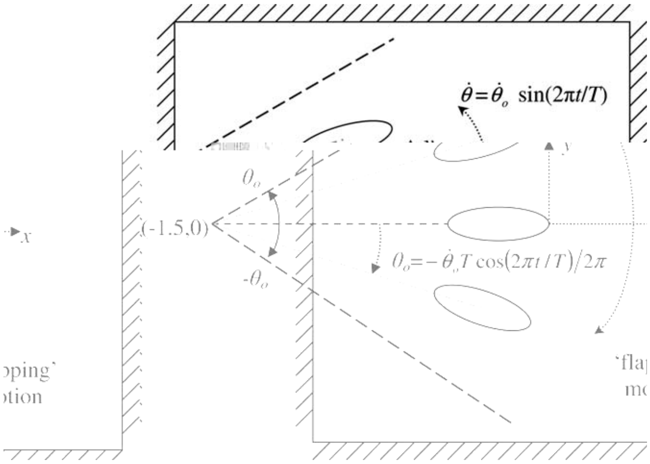

$\dot{\theta}=\dot{\theta}_o \sin(2\pi t/T)$
y
θ_o
$(-1.5,0)$
x
$\theta_o=-\dot{\theta}_o T\cos(2\pi t/T)/2\pi$
$-\theta_o$
pping'
tion
'fla
mo

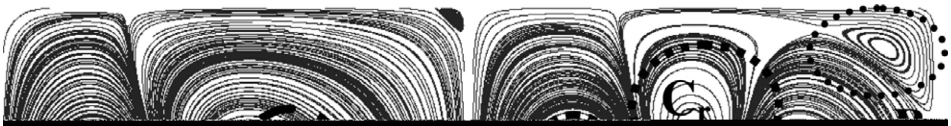
Fig. 30. Flapping motion of the elliptic cylinder. Cylinder starts oscillation at $\theta_o = -\dot{\theta}_o T/2\pi$ at $t = 0$.

the up-flap) can be seen to merge with the new inner tip vortex $\mathbf{C_l}$ in parts (e) to (f), both having the same sign. In part (f), we may also observe the right outer tip vortex $\mathbf{C_r}$ reaching out to merge with the lower right corner vortex of the same sign. This simulation shows the rather interesting phenomenon in which the vortical flow generated by the outer tip in one quarter of the cycle is passed on to reinforce the inner tip vortex in the next quarter of the cycle.

The flow is next solved for an oscillation period of $T = 10$ and $\dot{\theta}_o = 0.3$, with all other parameters remaining unchanged. The Reynolds number is $Re = 22.5$. Fig. 33 shows a Lissajous plot of the vertical force coefficient F_y against the horizontal force coefficient F_x on the ellipse. It can be seen that there are two horizontal force cycles for every vertical force cycle.

6.2. Non-synchronously counter-rotating cylinders in a rectangular box

The second test case demonstrates the application of the method to two bodies undergoing independently prescribed non-colliding motion in a rectangular box. The two moving bodies comprise two elongated cylinders rotating about their centres in opposite directions. The left cylinder rotates in the anti-clockwise direction



[redacted]

Fig. 31. Streamline plots of flapping ellipse within an enclosed box, from time $t = 48$ to 53, at time intervals of 1.0. The ellipse approaches the highest point at (c) in the upstroke, and turns back to make a downstroke ((d)–(f)). Annotated dashed ovals trace the development of the key vortical structures in the flow.

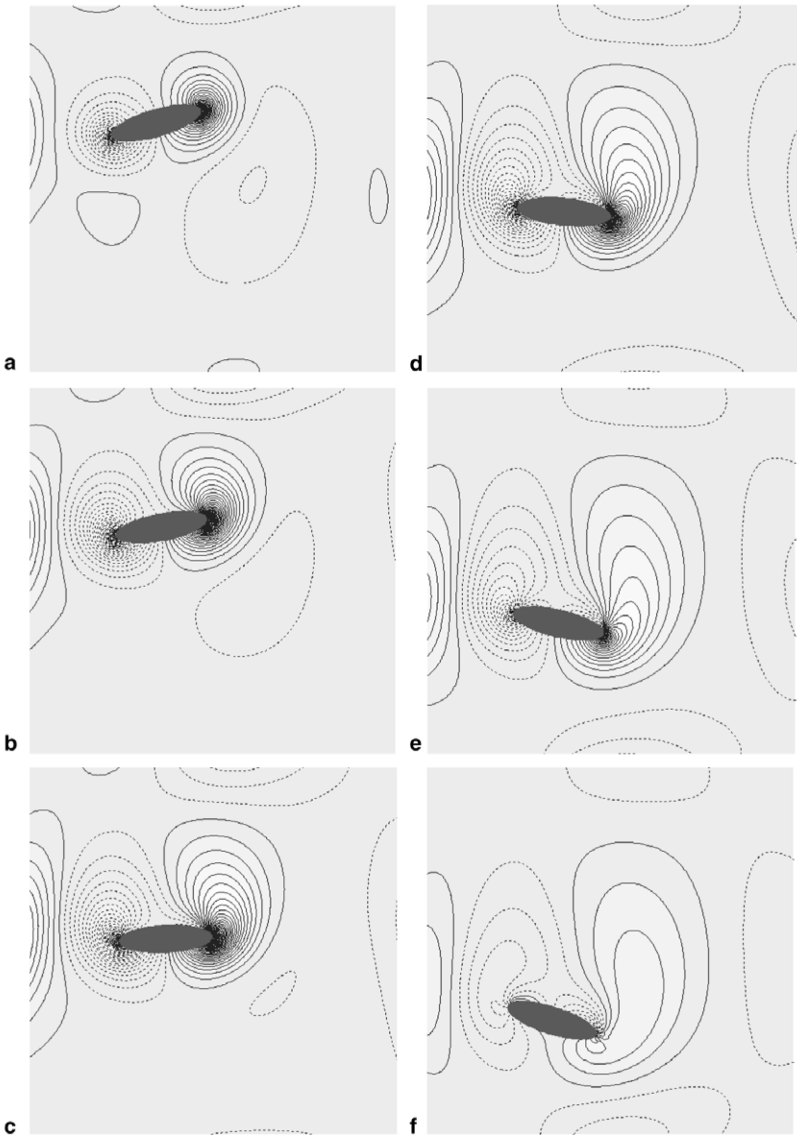


Fig. 32. Vorticity plots of flapping ellipse within an enclosed box. Sequence starts from top left down to top right down in a half cycle, from time $t = 50$ to $t = 60$. Negative vorticity is denoted by dotted lines.

with a non-dimensional period $T = 2\pi$, while the right cylinder rotates in the clockwise direction with $T = 4\pi$. A Reynolds number of $Re = 50$ based on the tip-to-tip length of the cylinder and the tip velocity of the left cylinder is used in this simulation. The ensuing time-dependent flow in the box is relatively complex, not unexpectedly – some snapshots of vorticity are presented in Fig. 34.

The overall dynamics of the flow may be briefly described as follows. Focusing attention first on the left cylinder, the vorticity distribution around the cylinder shows that the fluid elements adjacent to the flat sides of the cylinder are driven into anti-clockwise rotation (positive vorticity) by the anti-clockwise motion of the cylinder. Regions of strong negative vorticity (clockwise rotating flow) are generated near the two ends of the cylinder as the fluid is forced to go around them. The exact opposite distribution of positive and negative vorticities occur around the right clockwise rotating cylinder. The vorticity magnitudes are smaller because of the lower rotational speed of the right cylinder. The figures show that the bulk of the negative vorticity is produced by the tips of the left cylinder. The negative vorticity thus generated by the left cylinder convects

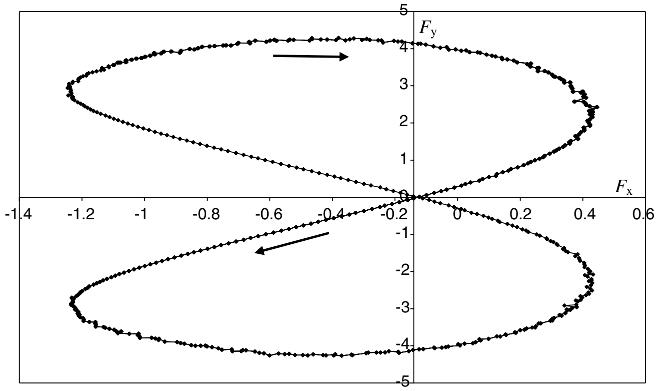


Fig. 33. Lissajou plot of F_y against F_x of the flapping ellipse within an enclosed box. Arrows point in the direction of progress. $Re = 22.5$, $T = 10$, $\dot{\theta} = 0.3$.

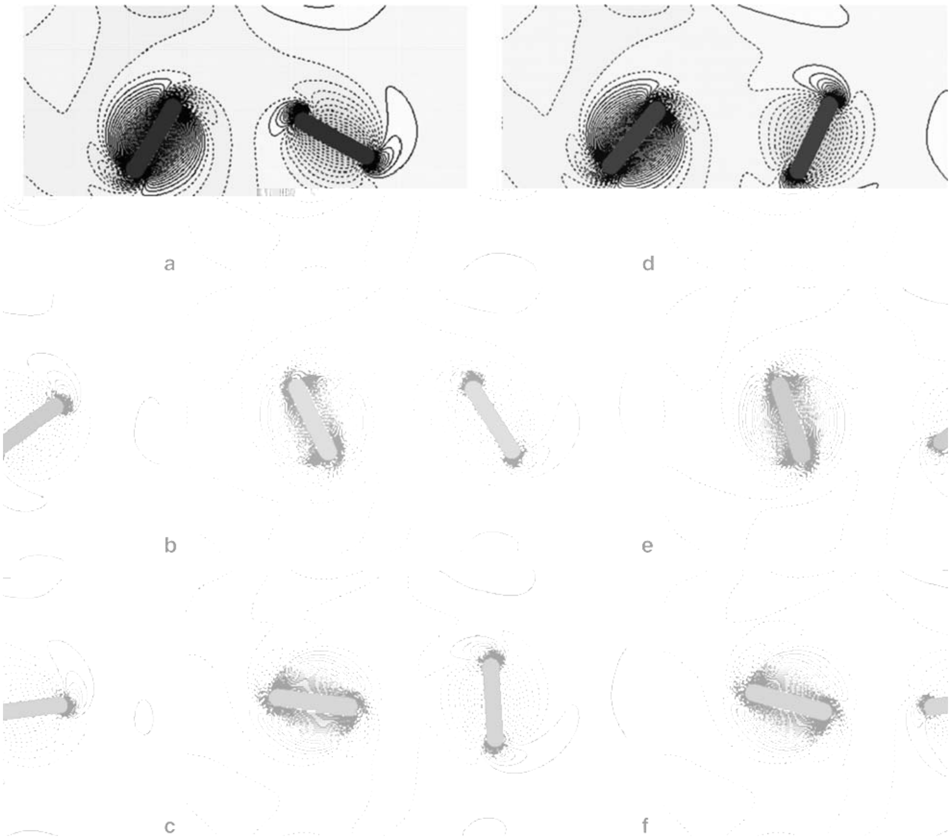


Fig. 34. Vorticity plots of the non-synchronously counter rotating cylinders through time $t = 45$ to $t = 50$ at time intervals of 1.0. Negative vorticity is denoted by dotted lines.

and diffuses to the right, where it may be observed to reinforce the regions of negative vorticity along the sides of the right cylinder – note the continuous region of negative vorticity that extends from the tips of the left cylinder to one side or even two sides of the right cylinder. The negative vorticity from the right cylinder also acts to help neutralize the positively-signed vorticity emanating from the tips of the right cylinder, which may be deduced from careful examinations of the results.

6.3. Driven flow by an oscillating circular cylinder within an enclosed box

In this simulation, a circular cylinder is made to oscillate sinusoidally within a square enclosure of dimensions $4 \times 4D^2$. The period of oscillation is set such that the Reynolds number based on the cylinder diameter and the cylinder's maximum velocity is 1000. Fig. 35 displays four consecutive instances of the vorticity and streamfunction fields. The top half of each plot displays the vorticity contours, while the bottom half displays the streamlines. Starting from (a), the cylinder begins its travel towards the right side of the enclosure. A pair of symmetric trailing vortices can be seen at its rear in part (b). While an 'older' pair of vortices are seen residing in front of the cylinder, they immediately get compressed by the approaching cylinder and make their way around it in part (c) as the cylinder reaches the end of its travel. This is then followed by the cylinder making its way back to the left in part (d).

This result shows that the present scheme can solve body-driven flows for up to moderately large Reynolds numbers without the explicit inclusion of numerical dissipation. Numerical dissipation will be needed, however, to maintain computational stability at higher Reynolds numbers. A QUICK scheme was used by Gilmanov and Sotiropoulos [27] for their Cartesian/immersed boundary method. This will be a subject for our future study.

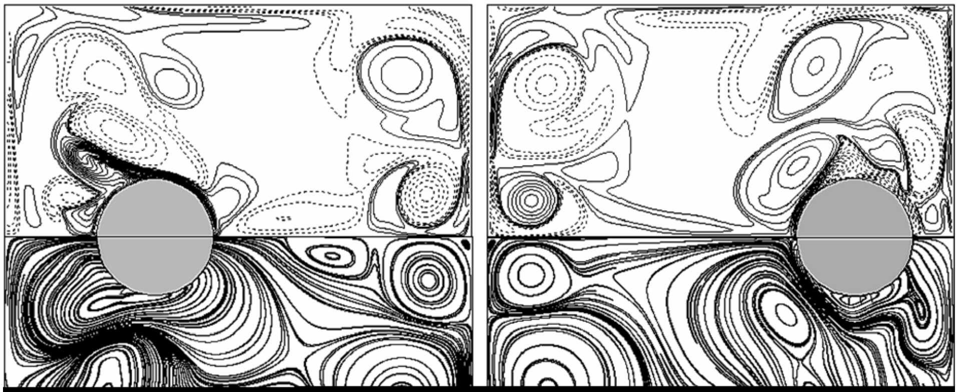


Fig. 35. Vorticity-streamline plots of flow generated by circular cylinder in oscillation at $Re = 1000$ in an enclosed box at instances of time (a) $t = 0.149T$, (b) $t = 0.308T$, (c) $t = 0.546T$ and (d) $t = 0.652T$ respectively, where T is the period of oscillation. Upper half of each plot displays vorticity contours, while lower half displays streamlines. Cylinder travels to the right in (a) and (b), and to the left in (c) and (d). Negative vorticity is denoted by dotted lines.

7. Conclusions

A numerical scheme adapted for simulation of moving boundary and moving body is proposed in this work. The method is based on a weighted least squares (WLS) generalized finite-difference (GFD) scheme for mesh-free spatial discretization and the Arbitrary Lagrangian–Eulerian (ALE) formulation of the governing incompressible flow equations for convecting nodal points. Solution of the flow equations is accomplished with a fractional-step projection method based on the second-order Crank–Nicolson time scheme. Moving bodies in flow are defined and enveloped by mesh-free clouds of nodes and allow to convect against a background of stationary Cartesian nodes. Away from boundaries and bodies, the standard 5-point FD scheme is applicable on the Cartesian grids. Typically the mesh-free nodes form a small fraction of the total nodal population. The present hybrid-grid method combines the advantages of mesh-free discretization for precise definition of bodies and good resolution of boundary regions with the efficiency of standard Cartesian FD scheme.

The ALE-WLS-GFD scheme was applied to a series test problems involving stationary and moving bodies in flow, including flow past circular cylinders in open and channel domains, impulsively and sinusoidally started cylinders, and longitudinally and transversely oscillating cylinders in flows. The results obtained showed very good consistency with similar results derived using a moving-frame method. The results also agreed well with published literature and the well-documented flow phenomena for oscillating cylinders are correctly reproduced. The method was further applied to body-driven flows in enclosures to demonstrate the flexibility of the method and the handling of multiple bodies undergoing independent non-collisional motion: a flapping elliptic cylinder in a square box and a pair of asynchronous counter-rotating elongated cylinders in a rectangular box. Examples for an odd-shaped cylinder illustrates the scheme's ability to handle complex body geometry. The method has been successfully applied on flows with Reynolds numbers up to 1000 without explicit use of artificial dissipation. For higher-*Re* flows and complex bodies with highly angular features, some degree of artificial dissipation will be needed.

The method has several advantages over the conventional numerical schemes as it can model complex bodies and motions without invoking domain transformation and remeshing. Interpolation is kept at a very minimal level associated with fresh nodes creation. Although discretization at mesh-free nodes tends to be more computationally expensive, they usually form only a small fraction of the total nodal population. Furthermore, the density of mesh-free nodes around bodies may be freely varied to give good resolution to boundaries and boundary events, which may be more difficult to achieve with purely Cartesian-based grid or volume methods. The treatment of the overlapped and fresh nodes exemplifies the ease with which nodal removal and addition can be accomplished during the simulation process. The work is done in two dimensions, but the primitive-variables formulation of the flow equations will permit a natural extension of the scheme to three-dimensional moving body/boundary flows. The method also has in-built flexibilities that will facilitate its extension to treat deforming bodies or boundaries, which are frequently encountered in biological flows.

Acknowledgement

The authors acknowledge the financial support provided by the National University of Singapore (NUS) for the present research.

References

[1] L.B. Lucy, A numerical approach to testing of the fission hypothesis, Astro. J. 82 (1977) 1013.

[2] J.U. Brackbill, Particle methods, Int. J. Numer. Methods Fluids 47 (2005) 693.

[3] T. Belytschko, Y. Krongauz, D. Organ, M. Fleming, P. Krysl, Meshless method: an overview and recent developments, Comput. Methods Appl. Mech. Engrg. 139 (1996) 3.

[4] B. Nayroles, G. Touzot, P. Villon, Generalizing the finite element method: diffuse approximation and diffuse elements, Comput. Mech. 10 (1992) 307.

[5] T. Belytschko, L. Gu, Y.Y. Lu, Fracture and crack growth by element-free Galerkin methods, Model. Simulat. Mater. Sci. Engrg. 2 (1994) 519.

[6] W.K. Liu, S. Jun, Y.F. Zhang, Reproducing kernel particle methods, Int. J. Numer. Methods Engrg. 20 (1995) 1081.

[7] I. Babuska, J.M. Melenk, The partition of unity finite element method, Int. J. Numer. Methods Engrg. 40 (1997) 727.
[8] C.A. Duarte, J.T. Oden, Hp clouds – a meshless method to solve boundary-value problems, Technical Report 95-05, Texas Institute for Computational and Applied Mathematics, University of Texas at Austin, 1995..
[9] E. Onate, S. Idelsohn, O.C. Zienkiewicz, R.L. Taylor, C. Sacco, A stabilized finite point method for analysis of fluid mechanics problems, Comput. Methods Appl. Mech. Engrg. 139 (1996) 315.
[10] E. Onate, C. Sacco, S. Idelsohn, A finite point method for incompressible flow problems, Comput. Visual Sci. 3 (2000) 67.
[11] B. Mendez, A. Velazquez, Finite point solver for the simulation of 2-D laminar incompressible unsteady flows, Comput. Methods Appl. Mech. Engrg. 193 (2004) 825.
[12] N. Perrone, R. Kao, A general finite difference method for arbitrary meshes, Comput. Struct. 5 (1975) 45.
[13] T. Liszka, J. Orkisz, The finite difference method at arbitrary irregular grids and its application in applied mechanics, Comput. Struct. 11 (1980) 83.
[14] J.J. Benito, F. Urena, L. Gavete, R. Alvarez, An *h*-adaptive method in the generalized finite differences, Comput. Methods Appl. Mech. Engrg. 192 (2003) 735.
[15] L. Gavete, M.L. Gavete, J.J. Benito, Improvements of generalized finite difference method and comparison with other meshless method, Appl. Math. Modell. 27 (2003) 831.
[16] G. Chesshire, W.D. Henshaw, Composite overlapping meshes for the solution of partial differential equations, J. Comput. Phys. 90 (1990) 1.
[17] H.S. Tang, S.C. Jones, F. Sotiropoulos, An overset-grid method for 3D unsteady incompressible flows, J. Comput. Phys. 191 (2003) 567.
[18] C.W. Hirt, A.A. Amsden, J.L. Cook, An arbitrary Lagrangian–Eulerian computing method for all flow speeds, J. Comput. Phys. 14 (1974) 227, Reprint J. Comput. Phys. 135 (1997) 203.
[19] H.H. Hu, N.A. Patankar, M.Y. Zhu, Direct numerical simulations of fluid–solid systems using arbitrary Lagrangian–Eulerian technique, J. Comput. Phys. 169 (2001) 427.
[20] J. Sarrate, A. Huerta, J. Donea, Arbitrary Lagrangian–Eulerian formulation for fluid-rigid body interaction, Comput. Methods Appl. Mech. Engrg. 190 (2001) 3171.
[21] R. Mittal, G. Iaccarino, Immersed boundary methods, Annu. Rev. Fluid Mech. 37 (2005) 239.
[22] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, Y.J. Jan, A front-tracking method for the computations of multiphase flows, J. Comput. Phys. 169 (2001) 708.
[23] G. Yang, M. Causon, D.M. Ingram, R. Saunders, P. Batten, A Cartesian cut cell method for compressible flows, Aero. J. 101 (1997) 47.
[24] 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.
[25] H.S. Udaykumar, R. Mittal, W. Shyy, Computation of solid–liquid phase fronts in the sharp interface limit on fixed grids, J. Comput. Phys. 153 (1999) 535.
[26] H.S. Udaykumar, R. Mittal, P. Rampunggoon, A. Khanna, A sharp interface Cartesian grid method for simulating flows with complex moving boundaries, J. Comput. Phys. 174 (2001) 345.
[27] A. Gilmanov, F. Sotiropoulos, A hybrid Cartesian/immersed boundary method for simulating flows with 3D, geometrically complex, moving bodies, J. Comput. Phys. 207 (2005) 457.
[28] C.W. Hirt, B.D. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, J. Comput. Phys. 39 (1981) 201.
[29] R. Scardovelli, S. Zaleski, Direct numerical simulation of free-surface and interfacial flow, Annu. Rev. Fluid Mech. 31 (1999) 567.
[30] S. Osher, J.A. Sethian, Fronts propagating with curvature dependent speed: algorithms based on Hamilton–Jacobi formulations, J. Comput. Phys. 79 (1988) 12.
[31] S. Osher, R.P. Fedkiw, Level set methods: an overview and some recent results, J. Comput. Phys. 169 (2001) 463.
[32] C.S. Peskin, Numerical analysis of blood flow in the heart, J. Comput. Phys. 25 (1977) 220.
[33] R. Glowinski, T.W. Pan, J. Periaux, A fictitious domain method for Dirichlet problems and applications, Comput. Methods Appl. Mech. Engrg. 111 (1994) 283.
[34] R. Glowinski, T.W. Pan, J. Periaux, Fictitious domain method for external incompressible viscous flow modeled by Navier–Stokes equations, Comput. Methods Appl. Mech. Engrg. 112 (1994) 133.
[35] P.A. Durbin, G. Iaccarino, An approach to local refinement of structured grids, J. Comput. Phys. 181 (2002) 639.
[36] H. Ding, C. Shu, K.S. Yeo, D. Xu, Development of least-square-based two-dimensional finite-difference schemes and their application to simulate natural convection in a cavity, Comput. Fluids 33 (2004) 137.
[37] H. Ding, C. Shu, K.S. Yeo, D. Xu, Simulation of incompressible viscous flows past a circular cylinder by hybrid FD scheme and meshless least square-based finite difference method, Comput. Methods Appl. Mech. Engrg. 193 (2004) 727.
[38] A.J. Chorin, Numerical solution of the Navier–Stokes equations, Math. Comput. 22 (1968) 745.
[39] A.J. Chorin, On the convergence of the discrete approximations to the Navier–Stokes equations, Math. Comput. 23 (1969) 341.
[40] D.L. Brown, R. Cortez, M.L. Minion, Accurate projection methods for the incompressible Navier–Stokes equations, J. Comput. Phys. 168 (2001) 464.
[41] J. Kim, P. Moin, Application of a fractional step method to incompressible Navier–Stokes equations, J. Comput. Phys. 59 (1985) 308.
[42] D. Kim, H. Choi, A second-order time-accurate finite volume method for unsteady incompressible flow on hybrid unstructured grids, J. Comput. Phys. 162 (2000) 411.
[43] J.B. Perot, An analysis of the fractional step method, J. Comput. Phys. 108 (1993) 51.
[44] Z.L. Guo, B.C. Shi, N.C. Wang, Lattice BGK model for incompressible Navier–Stokes equation, J. Comput. Phys. 165 (2000) 288.

[45] X.Y. He, G. Doolen, Lattice Boltzmann method on curvilinear coordinates system: flow around a circular cylinder, J. Comput. Phys. 134 (1997) 306.

[46] P.K. Stansby, A. Slaouti, Simulation of vortex shedding including blockage by the random-vortex and other methods, Int. J. Numer. Methods Fluids 17 (1993) 1003.

[47] Z. Lilek, S. Muzaferija, M. Peric, V. Seidl, Computation of unsteady flows using nonmatching blocks of structured grids, Numer. Heat Transfer, Part B 32 (1997) 403.

[48] P. Anagnostopoulos, G. Iliadis, S. Richardson, Numerical study of the blockage effects of viscous flow past a circular cylinder, Int. J. Numer. Methods Fluids 22 (1996) 1061.

[49] L. Zovatto, G. Pedrizzetti, Flow about a circular cylinder between parallel walls, J. Fluid Mech. 440 (2001) 1.

[50] S. Tang, N. Aubry, Suppression of vortex shedding inspired by a low-dimensional model, J. Fluids Struct. 14 (2000) 443.

[51] W.M. Collins, S.C.R. Dennis, The initial flow past an impulsively started circular cylinder, Q. J. Mech. Appl. Math. 26 (1973) 53.

[52] M. Bar-Lev, H.T. Yang, Initial flow field over an impulsively started circular cylinder, J. Fluid Mech. 72 (1975) 625.

[53] M. Tatsuno, P.W. Bearman, A visual study of the flow around an oscillating cylinder at low Keulegan–Carpenter number and low Stokes numbers, J. Fluid Mech. 211 (1990) 157.

[54] G. Iliadis, P. Anagnostopoulos, Viscous oscillatory flow around a circular cylinder at low Keulegan–Carpenter numbers and frequency parameters, Int. J. Numer. Methods Fluids 26 (1998) 403.

[55] G.H. Koopmann, The vortex wakes of vibrating cylinders at low Reynolds numbers, J. Fluid Mech. 28 (3) (1967) 501.

[56] P. Anagnostopoulos, Numerical study of the flow past a cylinder excited transversely to the incident stream. Part 1: Lock-in zone, hydrodynamic forces and wake geometry. Part 2: Timing of vortex shedding, aperiodic phenomena and wake parameters, J. Fluids Struct. 14 (2000) 819.