

# A robust direct-forcing immersed boundary method with enhanced stability for moving body problems in curvilinear coordinates



L. Nicolaou<sup>a</sup>, S.Y. Jung<sup>a</sup>, T.A. Zaki<sup>a,b,\*</sup>

<sup>a</sup> Department of Mechanical Engineering, Imperial College London, Exhibition Road, London SW7 2AZ, UK

<sup>b</sup> Department of Mechanical Engineering, Johns Hopkins University, Baltimore, MD 21218, USA

## ARTICLE INFO

### Article history:

Received 19 November 2014

Received in revised form 29 May 2015

Accepted 13 June 2015

Available online 2 July 2015

### Keywords:

Immersed boundary method

Numerical stability

Curvilinear coordinates

Moving body

## ABSTRACT

A robust immersed boundary method for semi-implicit discretizations of the Navier–Stokes equations on curvilinear grids is presented. No-slip conditions are enforced via momentum forcing, and mass conservation at the immersed boundary is satisfied via a mass source term developed for moving bodies. The errors associated with an explicit evaluation of the momentum forcing are analysed, and their influence on the stability of the underlying Navier–Stokes solver is examined. An iterative approach to compute the forcing term implicitly is proposed, which reduces the errors at the boundary and retains the stability guarantees of the original semi-implicit discretization of the Navier–Stokes equations. The implementation in generalized curvilinear coordinates and the treatment of moving boundaries are presented, followed by a number of test cases. The tests include stationary and moving boundaries and curvilinear grid problems (decaying vortex problem, stationary cylinder, flow in 90° bend in circular duct and oscillating cylinder in fluid at rest).

© 2015 Elsevier Ltd. All rights reserved.

## 1. Introduction

Immersed boundary (IB) methods have become an established approach for modelling complex and moving geometries. The main advantage and the popularity of these methods are due to their simplicity and efficiency. Unlike body-conforming methods which require a body-fitted grid and remeshing in moving boundary problems, a structured Cartesian grid is adopted for the IB method. The effect of the body surface is included through the addition of boundary forces in the Navier–Stokes equations. Use of structured grids greatly simplifies the task of grid generation, particularly for moving bodies and leads to more efficient computational algorithms with better convergence and stability properties. The present work focuses on the use of IB in the context of semi-implicit Navier–Stokes solvers, which are commonly adopted in simulations of moderate- and high-Reynolds number flows.

IB methods can be grouped into continuous forcing and discrete forcing approaches [1]. The first immersed boundary method was developed by Peskin [2] and was applied to elastic boundaries moved by the fluid. Modifications to this approach for use with rigid boundaries were proposed by Beyer and Leveque [3] and

Goldstein et al. [4] and employed *feedback forcing* to drive the velocity at the boundary to rest. However, these methods produced spurious oscillations and were subject to severe stability constraints. Another drawback of continuous forcing methods is the fact that a sharp representation of the boundary cannot be obtained since smoothing functions are used to transmit the forcing to the fluid, effectively spreading the location of the boundary. This is undesirable, especially when modelling high-Reynolds-number flows in which thin boundary layers need to be resolved accurately.

Mohd-Yusof [5] proposed a discrete derivation of the forcing term, in what is now commonly referred to as *direct forcing*. Other discrete forcing approaches exist, such as immersed interface methods (IIM) and Cartesian grid methods. However, the direct forcing approach remains most popular due to its simplicity and enhanced stability compared to other immersed boundary methods. Many variants of direct forcing methods have therefore appeared in the literature [6–13] and have been implemented in the framework of the fractional step algorithm which is commonly employed for solving the Navier–Stokes equations [7–9,14].

The accuracy and stability of direct forcing approaches used in conjunction with fractional step depend on the formulation of the fractional-step method, the computation of the forcing term and the treatment of mass conservation at the boundary. The two implementations of the fractional step method, referred to as the *p*-form, which neglects the pressure term in the

\* Corresponding author at: Department of Mechanical Engineering, Johns Hopkins University, Baltimore, MD 21218, USA. Tel.: +1 (410)516 6599; fax: +1 (410)516 7254.

E-mail address: [t.zaki@jhu.edu](mailto:t.zaki@jhu.edu) (T.A. Zaki).

intermediate velocity equation, and the  $\Delta p$ -form, which includes the pressure term from the previous time step, are both used extensively. However, only IB conditions applied in conjunction with the  $\Delta p$ -form are second-order accurate in time, whereas the  $p$ -form is only first-order accurate. The computation of the forcing term should be viewed relative to the temporal discretization of the governing equations. Explicit schemes for the solution of the Navier–Stokes equations are straightforward to implement, however they are limited by the viscous stability constraint. Therefore, most fractional step methods apply a semi-implicit approach where the diffusive terms are treated implicitly – these are the focus of the present work. In this context, implicit evaluation of the IB forcing term is not straightforward unless a simplified interpolation is adopted, for example the one used by Fadlun et al. [7]. Kim et al. [8] proposed an alternate approach where they provisionally advance the velocity field explicitly in order to compute the forcing term and then add it to the semi-implicit momentum equations. Their method has a clear advantage in terms of algorithmic efficiency. However, this approach can potentially reduce the stability limit of the numerical scheme due to the mismatch in the temporal discretization of the IB forcing and the governing equations.

In the present study, the errors in the computation of the forcing term are analysed and a stable second-order accurate direct forcing method is proposed. The current method consists of an iterative approach which decreases the errors at the boundary and enhances stability. The proposed method has been developed for use in a generalized curvilinear system allowing a wide range of complex geometries to be modelled efficiently on structured grids. The treatment of moving boundaries is also presented for completeness, and builds on the recent literature.

Sharp-interface IB methods are known to suffer from spurious force oscillations (SFOs) in moving body problems [15–20]. Lee et al. [18] identified two main sources of spurious oscillations: (i) The first source is the temporal discontinuity in the velocity which arises as a point from the fluid becomes solid and its velocity is suddenly changed to satisfy the no-slip condition at the IB. (ii) The second source is the spatial discontinuity in the pressure field which arises due to the momentum forcing and which contaminates the fluid field when a point from the solid becomes fluid. Seo and Mittal [19] found the major source of oscillations to be the violation of mass conservation near the immersed boundary. In order to suppress spurious oscillations, Yang and Balaras [16] proposed a field-extension approach in which the pressure and velocity at solid points becoming fluid were extrapolated from the surrounding fluid. Uhlmann [15] combined the direct forcing approach at Lagrangian points with discrete delta functions [2]. Lee et al. [18] showed that the addition of a mass source/sink inside the solid equally suppressed the SFOs, and Seo and Mittal [19] applied a cut-cell method to improve local mass conservation and reduce spurious oscillations. In the present method, an extension of the mass source term by Kim et al. [8] for use with moving boundaries is applied. Recently, a similar method applied to cells cut by the boundary was presented by Lee and You [20], and the differences will be discussed.

In summary, the stability of explicit and implicit forcing methods in the semi-implicit discretization of the Navier–Stokes equations is examined. An iterative implicit scheme is proposed, which is shown to have favourable stability properties. The method is capable of handling complex geometries on curvilinear grids and moving body problems. The paper is organized as follows: In Section 2, the governing equations and discretization scheme are presented. The accuracy of the IB boundary conditions in the fractional step method is discussed. In Section 3, the stability of explicit and implicit forcing methods is examined. An error analysis of explicit forcing methods is performed and the proposed implicit

forcing approach is then presented. The stability of both methods is studied for flow over a stationary cylinder. The implementation of the immersed boundary conditions, the modifications required for extension onto curvilinear coordinates, and the treatment of moving boundaries are described in Section 4. In Section 5, numerical tests which validate the accuracy of the method are presented. Finally in Section 6, some conclusions are drawn.

## 2. Governing equations and semi-implicit discretization

In order to satisfy both the no-slip and no-penetration conditions at the immersed boundary, a momentum forcing,  $f_i$ , and a mass source term,  $q$ , are applied to the Navier–Stokes equations, similar to the method by Kim et al. [8]. The forcing term sets the velocity at points surrounding the boundary, which are referred to as IB points, to a particular value such that the velocity at the surface of the immersed body satisfies the boundary conditions. Cells containing the immersed boundary do not satisfy mass conservation without appropriate treatment. Therefore a mass source is added in order to ensure that mass is conserved [8].

The governing equations for unsteady incompressible flow are the momentum and continuity equations given below

$$\frac{\partial u_i}{\partial t} + \frac{\partial u_i u_j}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{1}{Re} \frac{\partial^2 u_i}{\partial x_j \partial x_j} + f_i, \quad (1)$$

$$\frac{\partial u_i}{\partial x_i} - q = 0, \quad (2)$$

where  $x_i$  are the Cartesian coordinates,  $u_i$  are the corresponding velocities,  $p$  is the pressure,  $f_i$  are the momentum forcing components and  $q$  is the mass source term. The flow equations are solved on a staggered curvilinear grid using a volume flux formulation [21]. The equations are spatially discretized by a second-order finite-volume scheme and advanced in time with a second-order semi-implicit fractional step method that uses Adams–Bashforth for the convective terms and Crank–Nicolson for the diffusive terms. The flow solver has been extensively validated and adopted in direct numerical simulations of transitional and turbulent flows [22–24].

A number of approaches can be adopted for transformation of the governing equations from Cartesian to curvilinear coordinates, each with different methods of discretization, choice of dependent variables and grid layouts. For example, Cartesian velocities, contravariant velocities or volume fluxes, could be chosen as the dependent variables. While volume fluxes are used in our work, for generality and in keeping with the literature on IB methods, the discretized equations will be shown in Cartesian coordinates. Extension of the direct forcing method to curvilinear grids is independent of the coordinate transformation used and will be discussed in Section 4.1.

The fractional step method decouples the solution of the momentum equations (Eq. (1)) from that of the continuity equation (Eq. (2)) by solving them separately in two steps. An intermediate velocity field which is not divergence-free is computed first and subsequently corrected with a pseudo-pressure,  $\phi$ , such that continuity is satisfied.

The discretized equations are given by

$$\frac{\hat{u}_i - u_i^{n-1}}{\Delta t} = \frac{1}{Re} (\alpha L(\hat{u}_i) + \beta L(u_i^{n-1})) - G p^{n-1} - \gamma N(u_i^{n-1}) - \delta N(u_i^{n-2}) + f_i^n, \quad (3)$$

$$DG\phi^n = \frac{1}{\Delta t} (D\hat{u} - q^n), \quad (4)$$

$$u_i^n = \hat{u}_i - \Delta t G\phi^n, \quad (5)$$

$$p^n = p^{n-1} + \phi^n, \quad (6)$$

where  $N(u_i^n)$  is the discrete form of the convective term,  $L$  is the discrete Laplacian operator,  $D$  is the discrete divergence operator,  $G$  is the discrete gradient operator and the weighting coefficients depend on the numerical scheme adopted ( $\alpha = \beta = 1/2$  for Crank–Nicolson,  $\gamma = 3/2, \delta = -1/2$  for Adams–Bashforth and  $\gamma = 1, \delta = 0$  for Euler).

Two slightly different forms of the fractional step method exist. The first, referred to as the  $p$ -form [25,26], neglects the pressure term in Eq. (3) and then solves the Poisson equation for pressure. The method adopted here, the  $\Delta p$ -form, includes the pressure term from the previous time step when computing the intermediate velocity field and then determines the pressure difference from the Poisson equation [27,28]. Although the distinction may seem subtle, the appropriate form of the immersed boundary conditions differs for the two variants of the fractional step method. Note that in direct forcing methods the forcing term,  $f_i$ , is constructed such that  $\hat{u}_i$  satisfies the immersed boundary conditions rather than  $u_i^n$ . Following the analysis by Kim and Moin [26], enforcing  $\hat{u}_r = U_r^n$  on the immersed boundary  $\Gamma$ , where  $U_r^n$  is the boundary velocity, is second-order accurate when the  $\Delta p$ -form of the equations is used. The same condition is only first-order accurate in the  $p$ -form, but has nonetheless been used in the literature (e.g. [29,14]). Following a similar analysis to Kim and Moin [26], a second order accuracy can be achieved in the  $p$ -form by enforcing the immersed boundary condition,  $\hat{u}_r = U_r^n + \Delta t G \phi_r^{n-1}$  (the derivation is deferred to Appendix A). In this work, we adopt the  $\Delta p$ -form of the fractional step method, and ensure second-order accuracy by enforcing  $\hat{u}_r = U_r^n$  on the immersed boundary.

### 3. Stability of explicit and implicit forcing methods

#### 3.1. Explicit forcing

The evaluation of the momentum forcing,  $f_i^n$  in Eq. (3), must be consistent with the temporal evaluation of the intermediate velocity equation. Any disparity can introduce errors near the immersed boundary. In this section, explicit evaluation of  $f_i^n$  is examined when the intermediate velocity equation is also advanced explicitly in time and, after, for the more common case of a semi-implicit discretization of the momentum equation.

Consider a fractional step algorithm where the intermediate velocity equation is discretised with an explicit scheme (instead of the semi-implicit scheme provided in Eq. (3)). In this case the intermediate velocity,  $\hat{u}_{i(\text{ex})}$  is governed by,

$$\frac{\hat{u}_{i(\text{ex})} - u_i^{n-1}}{\Delta t} = \frac{1}{Re} (\gamma L(u_i^{n-1}) + \delta L(u_i^{n-2})) - G p^{n-1} - \gamma N(u_i^{n-1}) - \delta N(u_i^{n-2}) + f_{i(\text{ex})}^n. \quad (7)$$

Both  $f_{i(\text{ex})}^n$  and  $\hat{u}_{i(\text{ex})}$  are unknown, but the velocity field  $\hat{u}_{i(\text{ex})}$  at non-IB points can be computed since, at these points,  $f_{i(\text{ex})}^n = 0$ . Replacing  $\hat{u}_{i(\text{ex})}$  by the target velocity at IB points,  $U_{IB}^n$ , the forcing term is obtained from:

$$f_{i(\text{ex})}^n = \frac{U_{IB}^n - u_i^{n-1}}{\Delta t} - \frac{1}{Re} (\gamma L(u_i^{n-1}) + \delta L(u_i^{n-2})) + G p^{n-1} + \gamma N(u_i^{n-1}) + \delta N(u_i^{n-2}) \quad \text{at IB points,} \\ f_{i(\text{ex})}^n = 0 \quad \text{elsewhere,} \quad (8)$$

and added to Eq. (7) to solve for the intermediate velocity field. Substituting Eq. (8) into Eq. (7) shows that the scheme is consistent, and no error is introduced at the boundary,

$$\hat{u}_{i(\text{ex})} = U_{IB}^n \quad \text{at IB points,} \\ \hat{u}_{i(\text{ex})} = u_i^{n-1} - \left( \frac{1}{Re} (\gamma L(u_i^{n-1}) + \delta L(u_i^{n-2})) + G p^{n-1} + \gamma N(u_i^{n-1}) + \delta N(u_i^{n-2}) \right) \Delta t \quad \text{elsewhere.} \quad (9)$$

Since explicit schemes are limited by the viscous stability constraint, fractional step methods often apply a semi-implicit approach, in which the diffusive terms are treated implicitly (Eq. (3)). Evaluating the forcing term is less straightforward because  $\hat{u}_i$  at non-IB points, which are required to enforce  $\hat{u}_i = U_{IB}^n$  at the IB points by interpolation, now depend on  $\hat{u}_i$  at the IB points. This dependence comes through the second-order derivatives in the implicit diffusive terms. The solution proposed by Kim et al. [8] was to obtain the forcing by provisionally discretizing the intermediate velocity equation (Eq. (3)) explicitly in order to find an approximation,  $\tilde{u}_i$ , of  $\hat{u}_i$ ,

$$\frac{\tilde{u}_i - u_i^{n-1}}{\Delta t} = \frac{1}{Re} (\gamma L(u_i^{n-1}) + \delta L(u_i^{n-2})) - G p^{n-1} - \gamma N(u_i^{n-1}) - \delta N(u_i^{n-2}) + f_{i(\text{ex})}^n. \quad (10)$$

The velocity  $\tilde{u}_i$  at non-IB points is no longer dependent on  $\tilde{u}_i$  at IB points. The latter can be set via interpolation ( $\tilde{u}_i = U_{IB}^n$ ) and used to evaluate the forcing term, as in Eq. (8). The error introduced by the explicit forcing term can be viewed from two perspectives: The error in  $f$  itself which was examined by Kempe and Fröhlich [30]. Alternatively, here we focus on the error in the intermediate velocity which can be derived by substituting Eq. (8) into Eq. (3). This yields the following expression at the immersed boundary,

$$\hat{u}_i = U_{IB}^n + \frac{\Delta t}{Re} (\alpha L(\hat{u}_i) + \beta L(u_i^{n-1}) - \gamma L(u_i^{n-1}) - \delta L(u_i^{n-2})). \quad (11)$$

Due to the implicit treatment of the diffusive term in the intermediate velocity equation, this error also affects neighbouring non-IB points. The error due to the explicit treatment of the diffusive terms in the calculation of  $f_i^n$  can render the scheme unstable in two scenarios: (i) in low-Reynolds-number flows; (ii) in turbulent flows where the resolution near boundaries can render the viscous stability constraint more restrictive [7].

The stability of two explicit schemes for the computation of the forcing term is compared in Section 3.3: (i) Explicit 1, where the diffusive terms in Eq. (8) for  $f_{i(\text{ex})}^n$  are discretised using the Euler method ( $\gamma = 1, \delta = 0$ ) and (ii) Explicit 2, where the diffusive terms are discretised using the Adams–Bashforth scheme ( $\gamma = 3/2, \delta = -1/2$ ).

#### 3.2. Implicit forcing

In order to avoid introducing errors by the IB forcing in the context of the semi-implicit fractional step method, implicit evaluation of  $f_i^n$  is required,

$$f_i^n = \frac{U_{IB}^n - u_i^{n-1}}{\Delta t} - \frac{1}{Re} (\alpha L(\hat{u}_i) + \beta L(u_i^{n-1})) + G p^{n-1} + \gamma N(u_i^{n-1}) + \delta N(u_i^{n-2}). \quad (12)$$

If interpolation at the immersed boundary is performed along a grid line, the forcing term can be recast in such a way that only the intermediate velocity equation needs to be inverted [7]. However, for arbitrary complex geometries, interpolation is generally performed along the normal to the boundary in order to remove any ambiguity over the direction of interpolation. For interpolation along the surface normal, the method by Fadlun et al. [7] leads to a large sparse matrix, which adds substantially to the computational cost [9,16]. An extension to curvilinear coordinates presents a similar challenge to this scheme.

An alternative solution for the implicit calculation of the momentum forcing is to adopt an iterative strategy [20,31]. Here we propose an iterative approach which can be derived by splitting the intermediate velocity equation into two steps which fully recover Eq. (3),

$$\frac{\tilde{u}_i - u_i^{n-1}}{\Delta t} = \frac{1}{Re} (\alpha L(\tilde{u}_i) + \beta L(u_i^{n-1})) - Gp^{n-1} - \gamma N(u_i^{n-1}) - \delta N(u_i^{n-2}) + f_i^{n-1}, \quad (13)$$

$$\frac{\hat{u}_i - \tilde{u}_i}{\Delta t} = \frac{\alpha}{Re} (L(\hat{u}_i) - L(\tilde{u}_i)) + \delta f_i^n. \quad (14)$$

In the first step, an approximation,  $\tilde{u}_i$ , of the intermediate velocity is determined. This velocity field does not satisfy the exact boundary conditions at the immersed boundary since the forcing applied is that from the previous time step. In the second step, the forcing is updated and the no-slip constraint at the immersed boundary is satisfied.

The exact expression for  $\delta f_i^n$  is obtained from Eq. (14) by applying the no-slip constraint at the immersed boundary ( $\hat{u}_i = U_{IB}^n$  at IB points):

$$\delta f_i^n = \frac{U_{IB}^n - \tilde{u}_i}{\Delta t} - \frac{\alpha}{Re} (L(\hat{u}_i) - L(\tilde{u}_i)). \quad (15)$$

Since the diffusive term,  $L(\hat{u}_i)$ , in Eq. (15) is unknown, an iterative method is required to solve for  $\hat{u}_i$  and  $\delta f_i^n$  implicitly. An approximation to  $L(\hat{u}_i)$  is made by setting

$$\begin{aligned} \hat{u}_i &= U_{IB}^n \quad \text{at the forcing points,} \\ \hat{u}_i &= \tilde{u}_i \quad \text{elsewhere.} \end{aligned} \quad (16)$$

The first term in the forcing (Eq. (15)) drives the velocity at the boundary to the target value (i.e. it enforces the no-slip boundary constraint) while the second term ensures that the diffusive term at the boundary and, hence, the velocities  $\hat{u}_i$  at surrounding non-IB points are correct.

The algorithmic implementation for the method is given below and, in the following section, the stability of the proposed scheme (Implicit) is compared to that of the explicit forcing methods (Explicit 1 and Explicit 2) given in Section 3.1.

**Algorithm 1.** Implicit iterative scheme for the evaluation of the forcing term

---

```

 $f_i^{n,0} = 0$  or  $f_i^{n(ex)}$ 
for  $k = 1$  to  $nk$  do
     $\tilde{u}_i^k = u_i^{n-1} + \Delta t \left( \frac{\alpha}{Re} L(\tilde{u}_i^k) + \frac{\beta}{Re} L(u_i^{n-1}) - Gp^{n-1} - \gamma N(u_i^{n-1}) - \delta N(u_i^{n-2}) + f_i^{n,k-1} \right)$ 
     $\hat{u}_i^k = U_{IB}^n$  at IB points and  $\hat{u}_i^k = \tilde{u}_i^k$  elsewhere
     $f_i^{n,k} = f_i^{n,k-1} + \frac{U_{IB}^n - \tilde{u}_i^k}{\Delta t} - \frac{\alpha}{Re} (L(\hat{u}_i^k) - L(\tilde{u}_i^k))$ 
end for
 $f_i^n = f_i^{n,nk}$ 
 $\hat{u}_i = u_i^{n-1} + \Delta t \left( \frac{\alpha}{Re} L(\hat{u}_i) + \frac{\beta}{Re} L(u_i^{n-1}) - Gp^{n-1} - \gamma N(u_i^{n-1}) - \delta N(u_i^{n-2}) + f_i^n \right)$ 

```

---

In order to verify the spatial and temporal accuracy of the current IB method, simulations of a 2D unsteady flow are performed. We consider a decaying vortex problem [32], given by

$$\begin{aligned} u(x, y, t) &= -\cos(\pi x) \sin(\pi y) e^{-2\pi^2 t/Re}, \\ v(x, y, t) &= \sin(\pi x) \cos(\pi y) e^{-2\pi^2 t/Re}, \end{aligned} \quad (17)$$

in a square domain of length  $(-1.5L \leq x \leq 1.5L, -1.5L \leq y \leq 1.5L)$  with the immersed boundary located at  $(x = \pm L, y = \pm L)$ , where  $L$  is the size of the vortex (see Fig. 1). The Reynolds number based on the maximum velocity and the size of the vortex is  $Re = 500$ . The initial field and the velocities at the domain boundaries and at the immersed boundary are prescribed using the analytical solution.

For the spatial accuracy test, a grid refinement study is performed using five uniform Cartesian grids  $\{13^2, 25^2, 49^2, 97^2\}$ . In order to test the temporal accuracy of the scheme, various computational time steps are examined  $\{10, 5, 2.5, 1.25, 0.625\} \times 10^{-3}$  on the finest grid. The number of iterations in the implicit forcing scheme is set to  $nk = 5$ . The  $L_2$  and  $L_\infty$  norms of the  $u$  and  $v$  velocities, measured relative to the exact solution, are computed at time  $t = 0.1$  everywhere in the fluid enclosed by the immersed boundary as well as on the immersed boundary itself. The error norms are given by

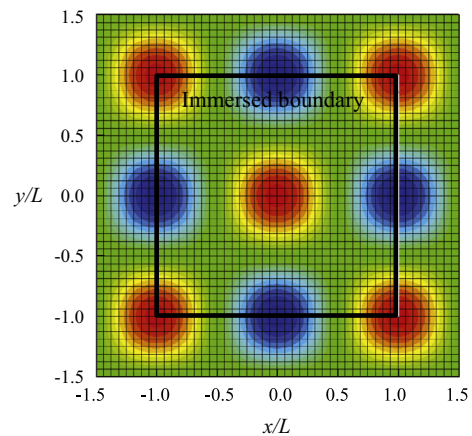
$$L_2 = \left[ \frac{1}{N^2} \sum_{i=1}^N \sum_{j=1}^N (u_{ij}^{num} - u_{ij}^{exact})^2 \right]^{\frac{1}{2}}, \quad L_\infty = \max |u_{ij}^{num} - u_{ij}^{exact}|. \quad (18)$$

Fig. 2a and b show the spatial and temporal convergence of the two norms, alongside reference lines displaying first and second-order rates of convergence. The results demonstrate that the IB scheme is second-order accurate in both space and time, and hence preserves the order of accuracy of the numerical method used for the solution of the Navier–Stokes equation.

In order to verify the convergence of the iterative scheme, the error in the intermediate velocity at the immersed boundary is evaluated. The  $L_2$  and  $L_\infty$  error norms are plotted versus number of iterations in Fig. 3, where zero iterations corresponds to the conventional explicit forcing. The error at the boundary diminishes rapidly with iterations. After four iterations, the error is approximately five orders of magnitude smaller than that due to the explicit IB method.

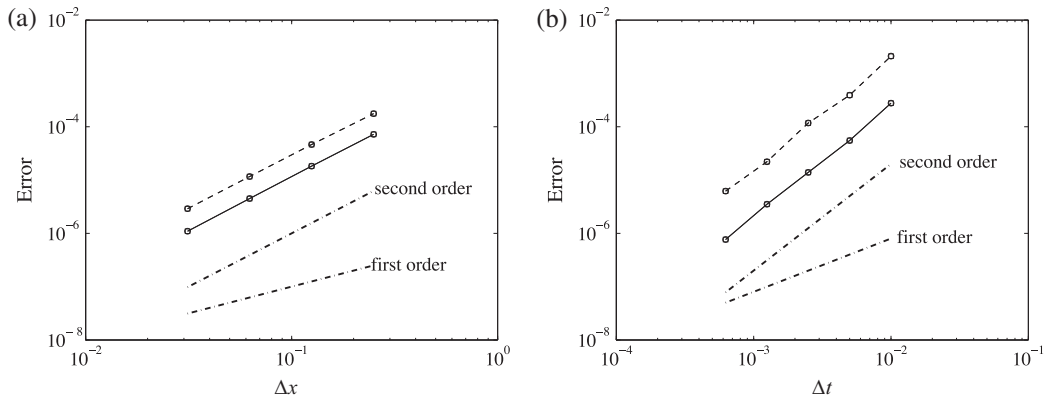
### 3.3. Stability

The stability of the explicit and the proposed implicit forcing schemes is examined for flow around a stationary circular cylinder. In all cases the intermediate velocity equation is discretized using Adams–Bashforth for the convective terms and Crank–Nicolson for the diffusive terms. The three schemes described in Sections 3.1 and 3.2 for the evaluation of the forcing term are compared: (i) Explicit 1, (ii) Explicit 2 and (iii) Implicit. A uniform Cartesian grid with  $769^2$  points is employed on a  $4D \times 4D$  domain, where  $D$  is the

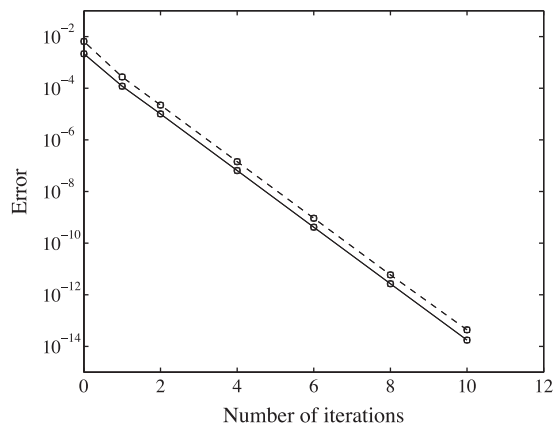


**Fig. 1.** Computational domain, grid (every fourth line in  $x$  and  $y$  of the finest grid are shown) and immersed boundary for decaying vortex problem. Contours of spanwise vorticity from  $-6.0$  to  $6.0$  with intervals of  $0.6$  at  $t = 0$ .





**Fig. 2.**  $L_2$  (—) and  $L_\infty$  (---) norms for the  $u$  ( $\square$ ) and  $v$  ( $\circ$ ) velocities versus (a) grid spacing, and (b) time step.



**Fig. 3.**  $L_2$  (—) and  $L_\infty$  (---) norms for the  $\hat{u}$  ( $\square$ ) and  $\hat{v}$  ( $\circ$ ) velocities at the immersed boundary versus number of iterations.

cylinder diameter, with the cylinder located at the centre. Free-slip is applied at the top and bottom boundaries and the simulations are performed at Reynolds numbers  $Re_D = \{20, 40\}$ , based on the diameter of the cylinder,  $D$ , and the far-field velocity,  $U_\infty$ . Different time steps are examined in order to determine the maximum allowable  $\Delta t$  for stability for each scheme.

**Table 1**  
Stability of the IB methods at different time steps for flow over a stationary cylinder at  $Re_D = \{20, 40\}$ .

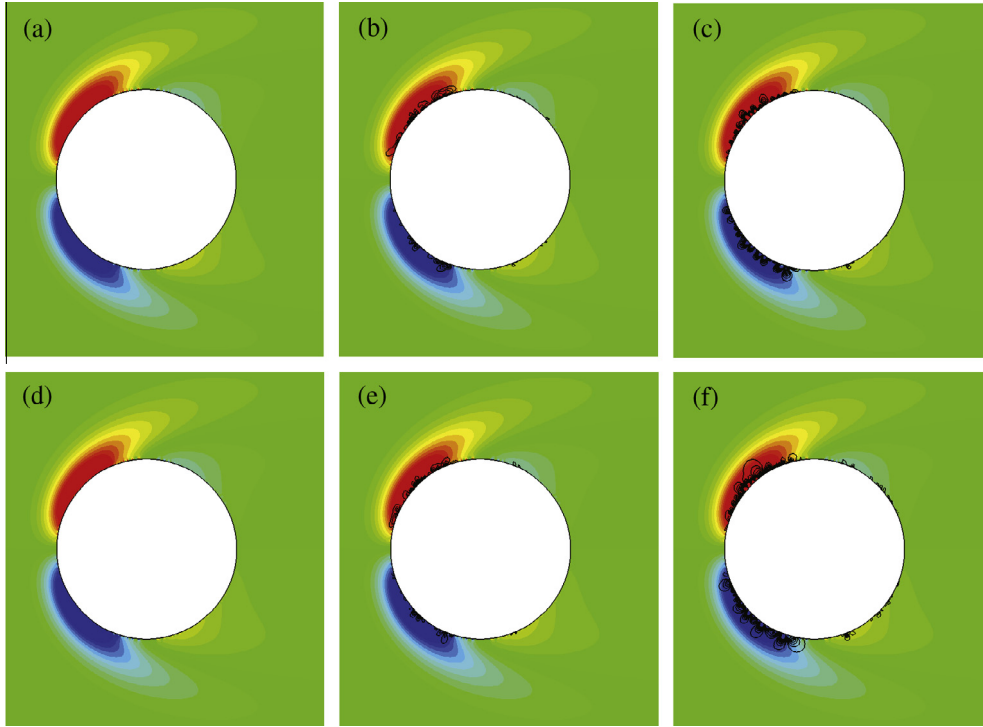
$Re_D$	$\Delta t$	Implicit	Explicit 1	Explicit 2	CFL	$\Delta t/Re\Delta x^2$
20	0.00110	unstable	unstable	unstable	0.402	2.028
	0.00105	stable	unstable	unstable	0.384	1.935
	0.00045	stable	unstable	unstable	0.164	0.829
	0.00040	stable	stable	unstable	0.146	0.737
	0.00030	stable	stable	unstable	0.110	0.553
	0.00025	stable	stable	stable	0.091	0.461
40	0.0022	unstable	unstable	unstable	0.831	2.028
	0.0021	stable	unstable	unstable	0.793	1.935
	0.0009	stable	unstable	unstable	0.340	0.829
	0.0008	stable	stable	unstable	0.302	0.737
	0.0006	stable	stable	unstable	0.226	0.553
	0.0005	stable	stable	stable	0.189	0.461

The test cases reported in Table 1 are specifically designed to test the viscous stability limit of the forcing methods, without approaching the CFL limit for convective instability. The motivation is to determine whether the inconsistency in the discretization of the viscous term in the intermediate velocity equation and in the expression for the forcing term can lead to instability. Below is a summary of the results:

1. Stability depends on the discretization scheme used for the diffusive terms in the equation for  $f_i^n$ . The implicit iterative method remains stable for larger  $\Delta t$  than the explicit methods. Compared to the explicit method of Kim et al. [8], the implicit approach remains stable up to approximately 2.5 times larger  $\Delta t$ , and up to four times the maximum allowable  $\Delta t$  of the Adams–Bashforth explicit method.
2. Stability is limited by the viscous stability constraint rather than the convective CFL, since errors are introduced in the diffusive terms (see Eq. (11)). The implicit method remains stable for  $\Delta t < 2.0Re\Delta x^2$ . The explicit forcing methods lower the stability of the overall algorithm even though forcing is only applied in a small region of the flow. Explicit method 1 remains stable for  $\Delta t < 0.75Re\Delta x^2$  and explicit method 2 is stable for  $\Delta t < 0.5Re\Delta x^2$ .

In this example, only one iteration was performed for the evaluation of the implicit forcing term, which reduced the errors at the immersed boundary by two to three orders of magnitude and only required an additional 50% of the computational cost required for the evaluation of an This added cost is insignificant for two reasons: First, it becomes marginal when normalised by the overall cost of the time step, where the most expensive element is the solution of the pressure equation. Second, the added cost is easily offset by the enhanced stability of the scheme. For example, the results in Table 1 show that the implicit force evaluation can increase the simulation  $\Delta t$  by 2.5 times relative to the explicit forcing scheme.

Next, errors in  $u$  and  $v$  velocities around the cylinder are examined at  $Re_D = 40$  in order to verify the source of the instability. Since there is no exact solution for this flow, the solution obtained with the implicit iterative method using the smallest time step is used as the benchmark. Fig. 4 shows line contour plots of the  $u$  and  $v$  errors for the three methods with  $\Delta t = 0.001$ . Flood contours of the wall-normal diffusive terms from the benchmark case have also been plotted and demonstrate that the errors leading to instability occur in regions where the diffusive terms are largest, which reaffirms the role of the viscous term in the instability of the explicit forcing scheme.



**Fig. 4.** Flood contours of the wall-normal diffusive terms from  $-2.0$  to  $2.0$  with intervals of  $0.2$  around the cylinder. (Top row) Line contours of  $u$ -errors and (bottom row)  $v$ -errors, from  $-0.25$  to  $0.25$  with intervals of  $0.025$ .  $Re_D = 40$ ,  $\Delta t = 0.0010$ . (a, d) Implicit; (b, e) Explicit 1; (c, f) Explicit 2.

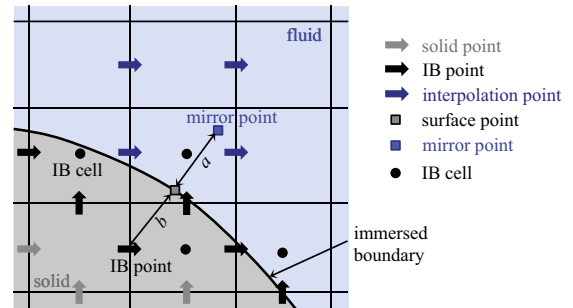
In order to validate our findings, we examined the stability of a one-dimensional unsteady heat diffusion problem. Starting with zero initial condition, the temperature was suddenly increased to unity at one boundary. In addition, we applied a forcing term to set the temperature at a given point in the domain to a prescribed value. The spatial discretization was a second-order central differencing scheme and time advancement was performed using Crank–Nicolson. Without the forcing term, a Crank–Nicolson discretization is unconditionally stable. Explicit forcing by Euler method was benign in this simple problem. However, explicit forcing by Adams–Bashforth reduced the stability boundary. The case of the Navier–Stokes equations is clearly more complex and the explicit forcing was demonstrated to reduce the finite stability region of the Crank–Nicolson discretization of the viscous terms.

#### 4. Implementation in curvilinear coordinates and for moving boundaries

The complete description of the iterative IB method presented in Section 3 requires two further considerations: implementation in curvilinear coordinates and treatment of moving boundaries. Both elements are presented in this section, and are followed by test cases for validation.

The immersed boundary is represented as a three-dimensional triangulated surface. Velocity points are classified into fluid or solid via a ray tracing algorithm which establishes whether a point lies inside or outside the geometry based on the number of times a ray starting from the point and passing by an arbitrary point in the domain intersects the surface. Solid points with at least one neighbour in the fluid are classified as IB points where the forcing,  $f_i$ , is to be applied. The cells cut by the immersed boundary are classified as IB cells.

In order to calculate the forcing term at an IB point, the velocity  $U_{IB}^n$  such that the immersed boundary conditions are satisfied must be determined. The evaluation of  $U_{IB}^n$  uses linear extrapolation from



**Fig. 5.** 2D schematic showing the definition of IB points and IB cells and the interpolation scheme applied in the present method, including the mirror point, surface point and interpolation points used for calculation of the mirror point velocity.

the velocity at the mirror point,  $u_M$ , and the velocity at the surface point,  $U_r^n$ :

$$U_{IB}^n = \frac{(a+b)}{a} U_r^n - \frac{b}{a} u_M. \quad (19)$$

Here,  $a$  and  $b$  are the distances of the mirror and the IB points to the surface respectively (see Fig. 5). The location of the mirror point is determined following a similar approach to that by Mittal et al. [33], which ensures the method works with highly complex geometries, and the mirror point velocity is computed by interpolation of the eight surrounding velocity points.

##### 4.1. Implementation in curvilinear coordinates

Certain geometries are better-suited to curvilinear rather than Cartesian grids. Curvilinear IB methods are particularly useful in bioflow problems which often involve complex geometries, e.g.

flow in a mechanical heart valve [34] or flow in the respiratory airways [35]. A curvilinear grid can improve efficiency by minimising the number of grid points outside the fluid domain, can have a more natural alignment with the streamlines which is desirable for the accuracy of the solution, and might provide a better wall-normal resolution than a Cartesian grid. This last advantage is particularly beneficial in simulations of turbulent flows.

The main issues when applying IB methods on a curvilinear grid are (i) the identification of the host cell for a particular point of interest and (ii) the interpolation method. On a Cartesian grid identification of host cells is a trivial procedure. On a curvilinear grid, however, there is no explicit relation between a physical position and the grid cell that contains it. Therefore a search algorithm is required in order to determine the cell location of the mirror points. The algorithm is based on ‘tetrahedrization’, whereby each cell is split into six tetrahedrons (or two triangles in 2D) [36]. The cell search is confined to ten surrounding cells in each dimension in order to speed up the algorithm. This approach is particularly advantageous in moving boundary problems since it avoids global search for host cells as the IB and mirror points move through the flow.

The second required modification is in the interpolation scheme for the evaluation of the velocity at the mirror points. Trilinear interpolation can be applied on curvilinear grids by mapping the grid points from physical space to a rectilinear computational space. Alternatively, an inverse distance weighting scheme can be adopted in physical space directly, where the weights of the eight surrounding velocity points are calculated as a function of the Euclidean distances,  $d_i$ , to the mirror point. In this case, the velocity at the mirror point  $u_M$  is given by,

$$u_M = \begin{cases} \frac{\sum_{i=1}^N \frac{u_i/d_i^2}{\sum_{i=1}^N 1/d_i^2}} & \text{if } d_i > \epsilon \text{ for all interpolation points,} \\ u_i & \text{if } d_i < \epsilon \text{ for one interpolation point,} \end{cases} \quad (20)$$

where  $u_i$  is the velocity at an interpolation point and  $\epsilon$  is the minimum tolerance for  $d_i$ . When the distance between the mirror point and an interpolation point is very small, the computational error becomes significant as  $d_i \approx 0$ . Therefore if a distance is less than  $\epsilon$ , the velocity is set to that of the interpolation point and all other interpolation points are given zero weighting [37].

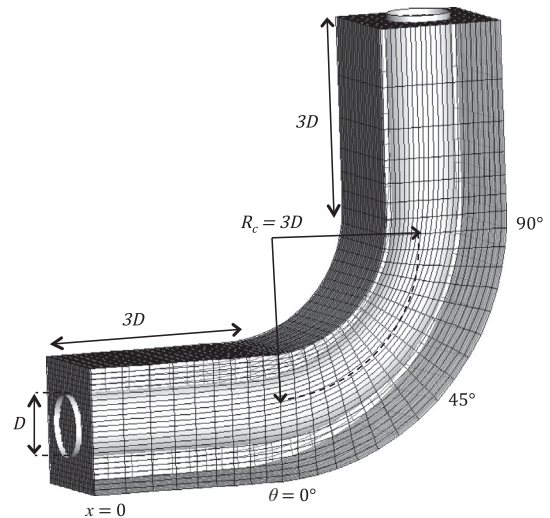
When all the surrounding points are in the fluid, the interpolation to the mirror point is straightforward. If any of the points is in the solid, it can be ignored, but the interpolation accuracy

deteriorates. Instead, the interpolation point within the solid is replaced by the nearest surface point. This approach would not be feasible with a standard trilinear interpolation scheme.

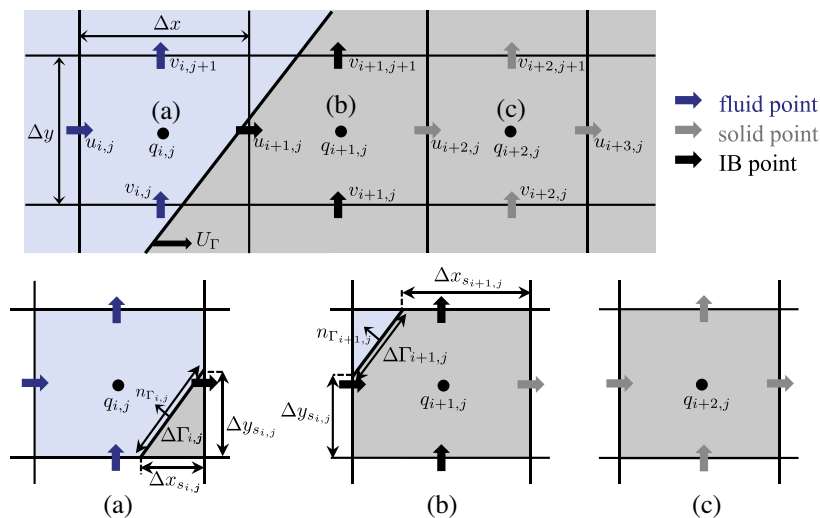
The method presented is capable of handling complex geometries on curvilinear grids. For example, we have applied the IB scheme in simulations of the flow in realistic extrathoracic airways [35]. However, for validation purposes, canonical test cases which can be reproduced by the readers are presented in Section 5.

## 4.2. Treatment of moving boundaries

The two main challenges encountered with moving boundaries are the treatment of freshly-cleared cells and the spurious force oscillations. Freshly-cleared cells arise when the moving boundary withdraws from the fluid and cells which were inside the solid at the previous time step are relocated in the fluid. The flow variables at these points do not have a valid time history, and therefore special treatment is required for the momentum equation in these cells. In order to ensure the correct flow field around a moving boundary, the intermediate velocity in the freshly-cleared cells is obtained by interpolation between the immersed boundary and a point in the fluid.



**Fig. 7.** The geometry and the grid for pulsatile flow in a circular 90° pipe bend. Every fourth gridline of the coarse grid has been plotted.



**Fig. 6.** 2D schematic of the mass source term for moving immersed boundaries.

In order to reduce spurious force oscillations at the boundary, a mass source/sink is applied inside the solid [8]. The mass source term proposed by Kim et al. [8] can be formulated for moving body problems [18]. This expression can also be extended in order to take into account the solid fractions of the cell faces,

$$q^n = \frac{1}{\Delta V} \sum_{i=1}^6 [\omega \beta_s (\dot{\mathbf{u}} - \mathbf{U}_r^n) + \psi \beta_s (\dot{\mathbf{u}} - \mathbf{U}_r^n)] \cdot \mathbf{n} \Delta S_i, \quad (21)$$

where  $\Delta V$  is the cell volume,  $\Delta S_i$  is the area of each cell face,  $\mathbf{n}$  is the unit normal vector outward at each cell face and  $\beta_s$  is the fraction of the cell face inside the solid. The coefficient  $\omega = 1$  for IB points and 0 otherwise, and  $\psi = 1$  for fluid points and 0 otherwise.

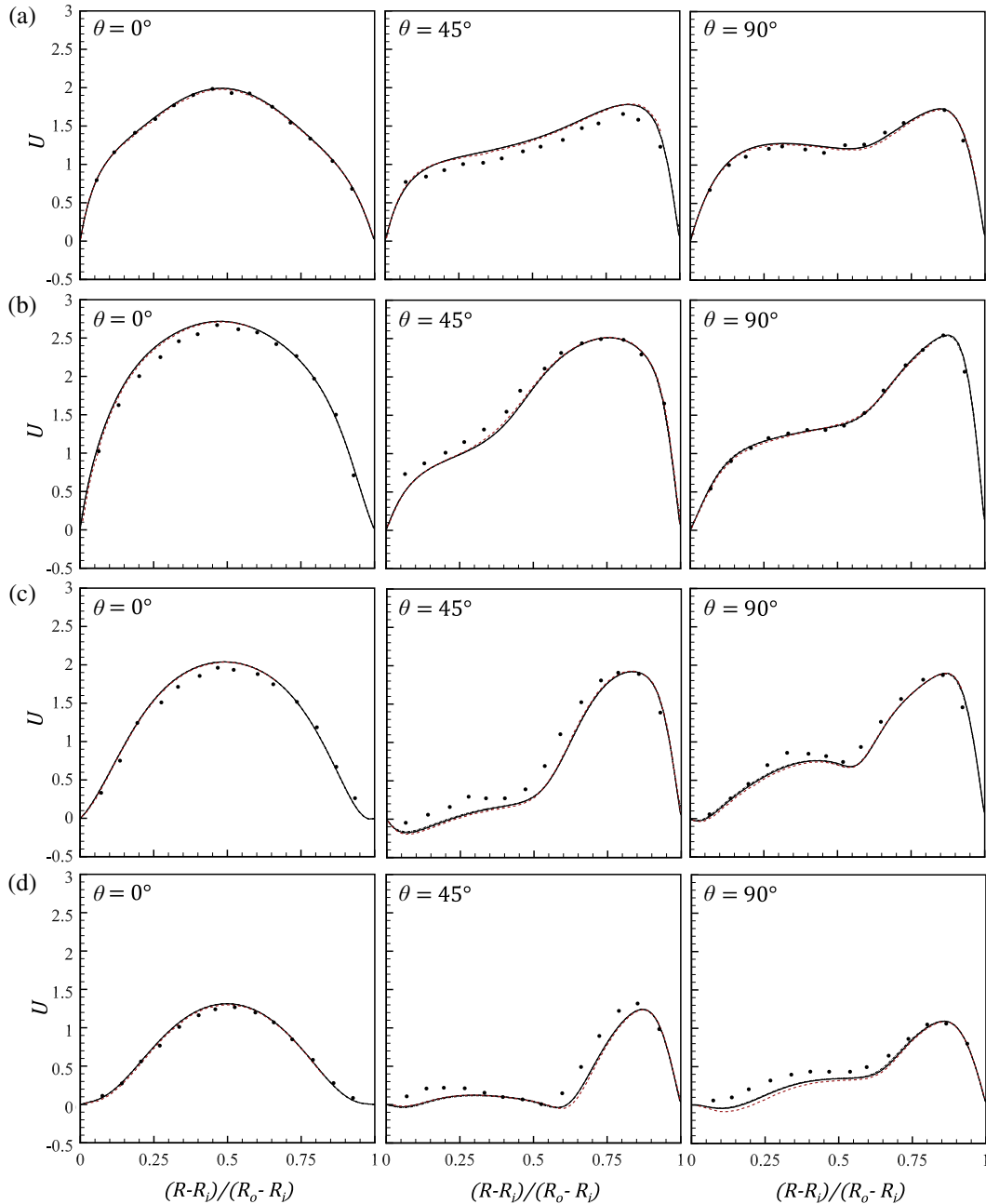
In the present work, a variant of the mass source approach is adopted for use with moving boundaries. The mass source term is given by

$$q^n = \frac{1}{\Delta V} \left( \sum_{i=1}^6 \beta_s \dot{\mathbf{u}} \cdot \mathbf{n} \Delta S_i + \mathbf{U}_r^n \cdot \mathbf{n}_r \Delta_r \right), \quad (22)$$

which is equivalent to

$$q^n = \frac{1}{\Delta V} \sum_{i=1}^6 \beta_s (\dot{\mathbf{u}} - \mathbf{U}_r^n) \cdot \mathbf{n} \Delta S_i, \quad (23)$$

where  $\mathbf{n}_r$  and  $\Delta_r$  are the surface outward unit normal vector (pointing towards the fluid), and the area of the boundary within the cell,



**Fig. 8.** Pulsatile flow in a circular 90° pipe bend. Streamwise velocity profiles normalised by the median bulk velocity. --- 129 × 97 × 97 grid; --- 129 × 129 × 129 grid; - 257 × 129 × 129 grid; - - - body-fitted data [34]; • experimental measurements [40]. (a)  $t = 0$ ; (b)  $t = T/4$ ; (c)  $t = T/2$ ; (d)  $t = 3T/4$ .  $R$  is the radial distance from the centre of bend curvature, and  $R_i$  and  $R_o$  are the inner and outer radius of the bend, respectively.



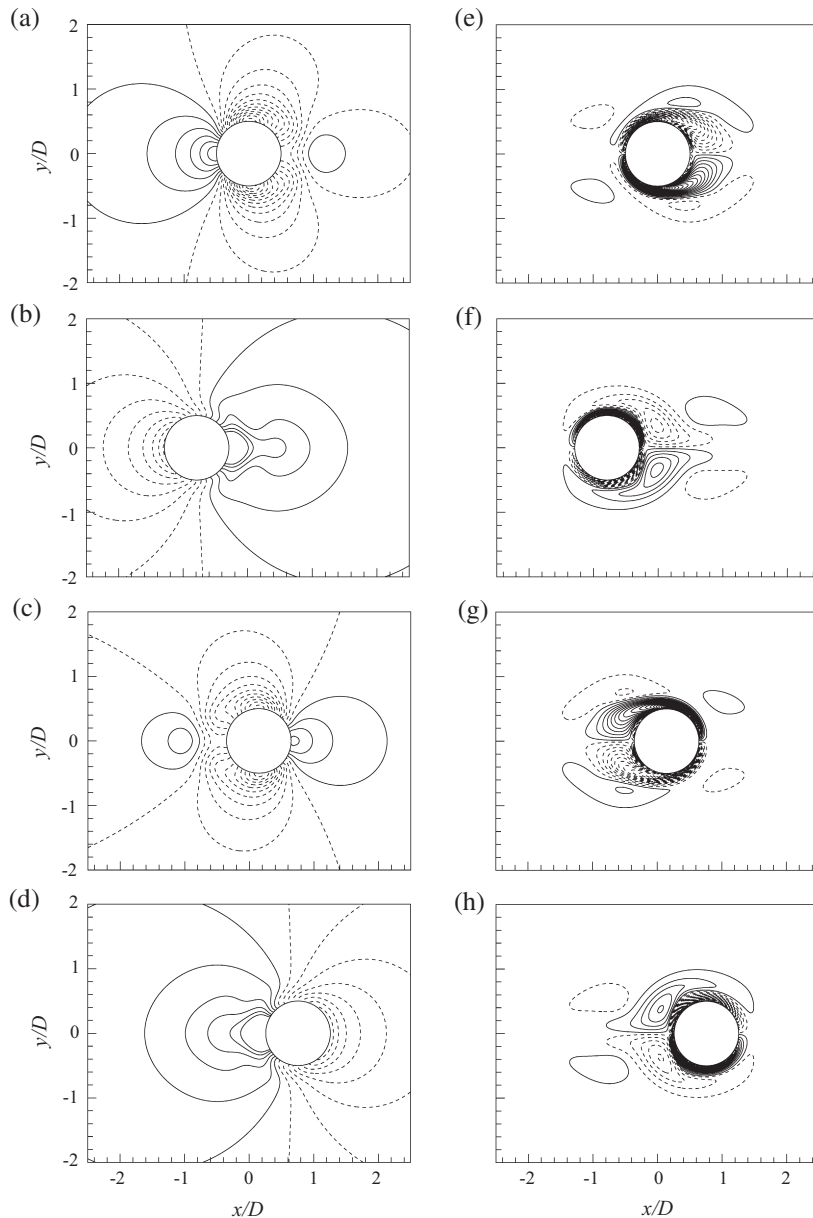
respectively. For the two-dimensional example shown in Fig. 6, the mass source terms in the three cells are

$$\begin{aligned}
 q_{ij} &= \frac{1}{\Delta x \Delta y} \left( u_{i+1,j} \frac{\Delta y_{s_{ij}}}{\Delta y} - v_{ij} \frac{\Delta x_{s_{ij}}}{\Delta x} + U_r \cdot n_{\Gamma_{ij}} \Delta \Gamma_{ij} \right), \\
 q_{i+1,j} &= \frac{1}{\Delta x \Delta y} \left( -u_{i+1,j} \frac{\Delta y_{s_{ij}}}{\Delta y} + u_{i+2,j} \Delta y - v_{i+1,j} \Delta x + v_{i+1,j+1} \frac{\Delta x_{s_{i+1,j}}}{\Delta x} \right. \\
 &\quad \left. + U_r \cdot n_{\Gamma_{i+1,j}} \Delta \Gamma_{i+1,j} \right), \\
 q_{i+2,j} &= \frac{1}{\Delta x \Delta y} (-u_{i+2,j} \Delta y + u_{i+3,j} \Delta y - v_{i+2,j} \Delta x + v_{i+2,j+1} \Delta x). \quad (24)
 \end{aligned}$$

The first term on the right-hand-side of Eq. (22) mimics the cut-cell/virtual cell-merging technique [19] without requiring construction of new polyhedral cells. It ensures that the continuity equation is satisfied for the fraction of the cell inside the fluid by excluding the contribution from the solid region. Since two

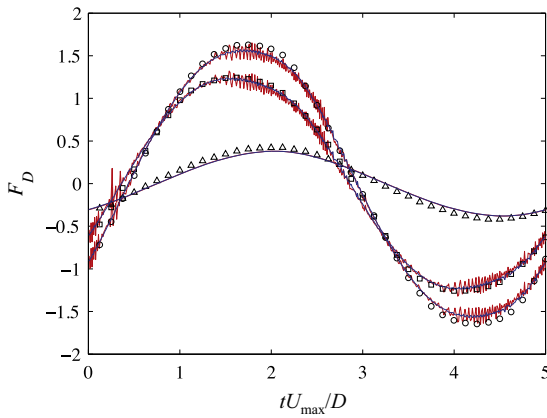
neighbouring cells which share a common face have the same value of the surface flux, but with opposite signs, the global contribution of this term vanishes [8]. The second term in Eq. (22) ensures that the continuity equation is satisfied for moving body problems by including the flux at the boundary. Eq. (22) can be interpreted as an expression of the source term from a grid-based reference frame. Cast in the form (23), it can be interpreted as the source term relative to the moving body.

While the mass source term (Eq. (22)/(23)) appears similar to that by Huang and Sung [38] for stationary boundaries or by Lee and You [20] for moving bodies, two important differences should be noted, and they relate to global mass conservation and the suppression of spurious force oscillations on staggered grids. The first notable distinction is that the method by Lee and You [20] was developed for collocated grids, on which the mass source term was applied in IB cells only. This is not the case in our method for staggered grids, where the mass source term in Eq. (22) (or (23)) is applied to IB cells (cut by the boundary) as well as any



**Fig. 9.** In-line oscillating cylinder in a fluid at rest at four different phase angles: (a) 0°; (b) 96°; (c) 192°; (d) 288°. At left, pressure contours from −1.1 to 0.6 with intervals of 0.09. At right, spanwise vorticity contours from −26.0 to 26.0 with intervals of 0.95. —, positive; ---, negative.

adjacent cells that share the IB fluxes at their common cell face, in order to satisfy global mass conservation. Second, unlike Huang and Sung [38] and others who compute the source term at IB points only, the present approach applies the forcing and mass source terms throughout the solid region. Specifically, the first term in Eq. (22) is evaluated at all solid and IB cell faces, as shown in Eq. (24). This requirement avoids a temporal discontinuity in the velocity of solid points suddenly becoming IB points, which we demonstrate in Appendix B. The benefit of the present approach has been remarked in some previous studies: For example, Kim and Choi [39] reported that this choice improved the stability of their simulations at high-Reynolds-number; Liao et al. [17] showed that applying the forcing term throughout the solid reduces the otherwise large spurious oscillations that arise when the forcing is applied on the immersed boundary only.



**Fig. 10.** Temporal evolution of the drag force on an oscillating cylinder with grid-size  $\Delta x = \Delta y = 0.02D$ : —, momentum forcing without a mass source; —, momentum forcing with a mass source. Symbols are boundary-fitted data [42]:  $\circ$ , total drag force;  $\square$ , pressure component;  $\triangle$ , shear component.

## 5. Numerical examples

### 5.1. Pulsatile flow in 90° circular bend

Pulsatile flow in a circular duct with a 90° bend is simulated in order to validate the immersed boundary method on curvilinear coordinates. The non-dimensional parameters that characterize this flow are the Reynolds number,  $Re_d = \frac{UD}{\nu}$ , the Womersley parameter,  $\alpha = \frac{D}{2} \left( \frac{\Omega}{\nu} \right)^{\frac{1}{2}}$ , and the curvature ratio,  $\delta = \frac{D}{2R_c}$ , where  $D$  is the pipe diameter,  $U$  is the instantaneous bulk velocity,  $\nu$  is the kinematic viscosity,  $\Omega$  is the angular frequency and  $R_c$  is the radius of curvature of the pipe axis.

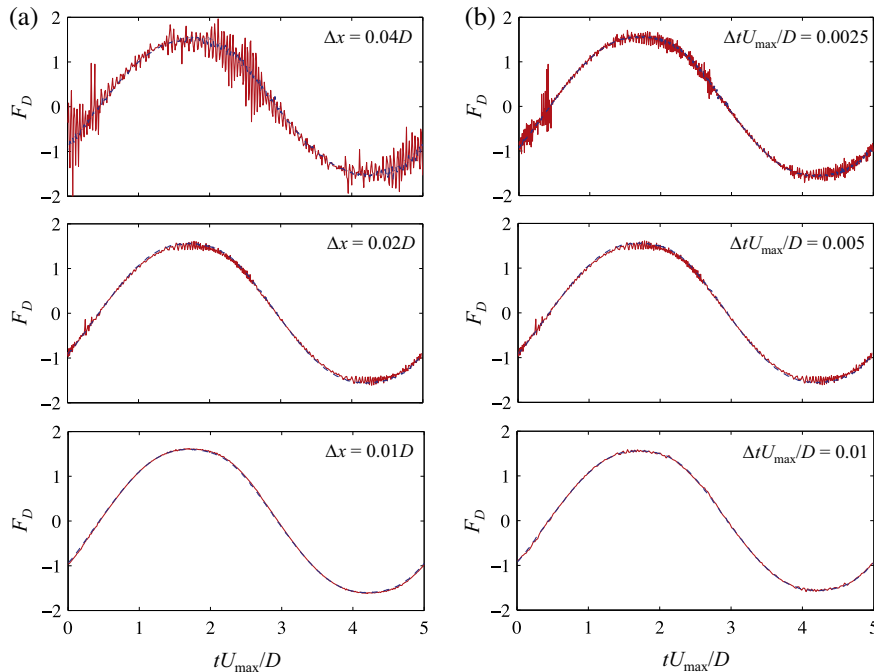
A sinusoidally varying flow rate ( $200 < Re_d < 800, \alpha = 7.8$ ) is applied at the inlet of the pipe with curvature ratio,  $\delta = \frac{1}{6}$ , for comparison with available body-fitted and experimental data from the literature [34,40]. The velocity profile at the inlet,  $x = 0$ , is computed from the Womersley solution for fully-developed pulsatile flow in a circular pipe given by

$$u_{inlet}(r, t) = 2[1 - r^2] + \frac{A_0}{\rho} \frac{1}{i\Omega} \left[ 1 - \frac{J_0(\alpha r i^{3/2})}{J_0(\alpha i^{3/2})} \right] e^{i\Omega t}, \quad (25)$$

$$A_0 = Q_{max} \frac{4\rho}{\pi D^2} i\Omega \left[ 1 - \frac{2}{i^{3/2} \alpha} \frac{J_1(\alpha i^{3/2})}{J_0(\alpha i^{3/2})} \right]^{-1} e^{-i\Omega t_{max}}, \quad (26)$$

where  $A_0$  is the amplitude of pulsation;  $J_0$  denotes the Bessel function of the first kind and order zero;  $r$  is the normalised radial distance from the pipe centre;  $\rho$  is the fluid density;  $Q_{max}$  is the flow rate at the maximum bulk velocity or, equivalently, the maximum Reynolds number [41].

Simulations were performed on three successively finer curvilinear grids:  $129 \times 97 \times 97$ ,  $129 \times 129 \times 129$  and  $257 \times 129 \times 129$  in the streamwise,  $\xi$ , and transverse,  $\eta$  and  $z$ , directions. The grids are stretched in the streamwise direction in order to provide a higher resolution inside the bend. The geometry and the grid are



**Fig. 11.** Temporal evolution of the drag force on an oscillating cylinder: (a) on different grid resolutions with time step  $\Delta t = 0.005$ ; (b) for different time steps with grid size  $\Delta x = \Delta y = 0.02D$ . —, momentum forcing without a mass source; —, momentum forcing with a mass source.

shown in Fig. 7. The pulsatile flow cycle is divided into 2000 time steps, corresponding to  $CFL_{\max} = \{0.51, 0.63, 0.82\}$  on the coarse, medium and fine grid, respectively.

Profiles of the streamwise velocity normalised by the median bulk velocity are shown in Fig. 8, at three different locations along the bend,  $\theta = \{0^\circ, 45^\circ, 90^\circ\}$ . Four different time instants are shown,  $t = \{0, T/4, T/2, 3T/4\}$ . The results are plotted alongside body-fitted data [34] and experimental measurements [40]. The figures demonstrate excellent agreement between our IB results and the body-fitted data even on the coarsest mesh, and establish the accuracy of our IB method in complex, 3D curvilinear flow simulations.

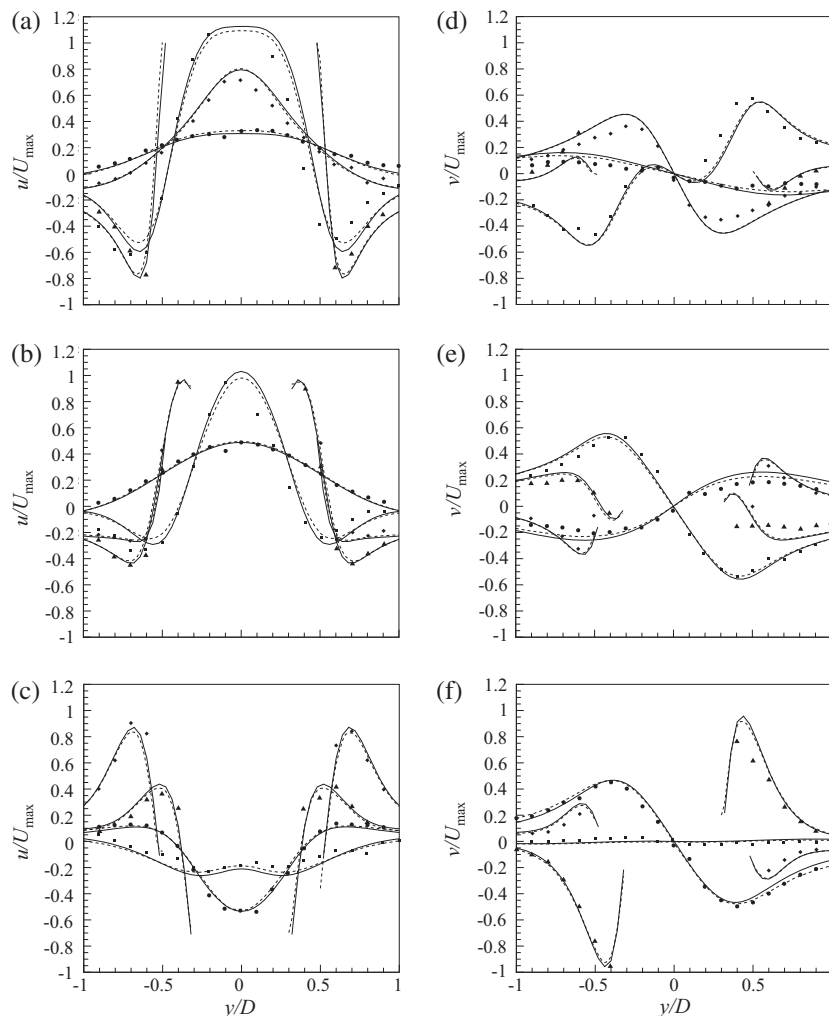
## 5.2. Oscillating cylinder in fluid at rest

In order to demonstrate the ability of the proposed immersed boundary method to handle moving boundaries, simulations of a cylinder oscillating in a fluid at rest are presented and compared to both numerical and experimental results from the literature. In this configuration, the flow parameters are the Reynolds number,  $Re_D = U_{\max} D / \nu = 100$ , and Keulegan–Carpenter number,  $KC = U_{\max} / fD = 5$ , where  $U_{\max}$  is the maximum velocity of the cylinder,  $D$  is the diameter of the cylinder,  $\nu$  is the kinematic viscosity of the fluid, and  $f$  is the frequency of oscillation. The streamwise location of the cylinder is given by  $x(t) = -A \sin(2\pi ft)$ , where

$A$  is the amplitude of the oscillation. The cylinder is located at the centre of a  $50D \times 50D$  domain. A  $512 \times 512$  non-uniform Cartesian grid is employed, with a minimum local spacing  $\Delta x_{\min} = \Delta y_{\min} = 0.02D$  in the vicinity of the cylinder.

Pressure and spanwise vorticity contours around the oscillating cylinder at different phase angles are shown in Fig. 9. The contours are in good qualitative agreement with the results reported by Dütch et al. [42] and thus demonstrate good prediction of the general flow characteristics. As the cylinder starts to move to the left, two boundary layers are formed at the top and bottom of the cylinder (see Fig. 9e). These boundary layers later shed, forming two counter-rotating vortices behind the cylinder (Fig. 9f). The cylinder then begins to move in the opposite direction, breaking the existing vortices apart (Fig. 9g) while at the same time developing new boundary layers which shed on the other side of the cylinder (Fig. 9h).

In order to assess the accuracy of the IB method near the moving boundary and the ability of the proposed mass source term to suppress spurious force oscillations, the drag force acting on the cylinder is examined. The pressure force and skin friction are computed using the pressure and velocity gradients at the centres of the surface triangles, which are obtained by linear extrapolation from the outer fluid along the surface normal. Fig. 10 shows the time evolution of the total drag force along with its constituent pressure and shear contributions predicted by the present IB



**Fig. 12.** In-line oscillating cylinder in a fluid at rest at three different phase angles: (a)  $180^\circ$ ; (b)  $210^\circ$ ; (c)  $330^\circ$ . At left, streamwise velocity. At right, vertical velocity. —, current IB method; ---, body-fitted results [42]; symbols are experimental data [42]:  $\blacksquare$ ,  $x = -0.6D$ ;  $\blacktriangle$ ,  $x = 0.0D$ ;  $\blacklozenge$ ,  $x = 0.6D$ ;  $\bullet$ ,  $x = 1.2D$ .

method with and without mass source term. Without a mass source, larger spurious oscillations in the drag force can be observed. The SFOs are due to the pressure component of the drag, while the shear force remains smooth. With the mass source term, a smoother solution of the pressure drag is obtained. Very good agreement with boundary-fitted data from the literature [42] is observed, which demonstrates the ability of the current IB method to accurately predict the body forces on a moving object.

The effect of grid spacing on the magnitude of the SFOs is examined on three different Cartesian grids  $\{257^2, 513^2, 1025^2\}$  with local spacing in the vicinity of the cylinder equal to  $0.04D, 0.02D$  and  $0.01D$  respectively. The time step is fixed to  $\Delta t U_{\max}/D = 0.005$ , which corresponds to  $CFL_{\max} = \{0.29, 0.44, 0.76\}$ . As shown in Fig. 11a, the magnitude of oscillations rapidly decreases as the grid is refined. In addition, smaller oscillations are observed with the mass source term. The effect of time step on the SFOs is also examined using three different time steps  $\{0.0025, 0.005, 0.01\}$  on the  $513^2$  grid, where  $CFL_{\max} = \{0.21, 0.44, 0.85\}$ . The results are shown in Fig. 11b. A decrease in spurious oscillations is observed with increasing time-step, which is consistent with the behaviour reported in the literature [18–20]. For all time-step sizes, the solution obtained with the mass source term is much smoother.

Finally, the velocity profiles are plotted across the cylinder at various streamwise locations and three different phase-angles, alongside the body-fitted results and experimental data of Dutsch et al. [42] (see Fig. 12). Very good agreement is observed between the profiles obtained with the immersed boundary method and the results from the literature.

## 6. Summary

In the present study, the accuracy and stability of direct forcing methods used in conjunction with the fractional step algorithm are shown to depend on the variant of the fractional step method and the computation of the forcing term. An iterative scheme is presented and shown to have favourable stability properties compared to explicit forcing in the semi-implicit discretization of the Navier–Stokes equations. The method is also effective for stationary and moving boundary problems, and is applicable on curvilinear grids.

A mismatch in the temporal discretization of the momentum equation and the forcing term is shown to introduce errors near the immersed boundary which can destabilise the scheme. In the semi-implicit fractional step method, the implicit discretization of the viscous terms in the Navier–Stokes equations requires an implicit forcing term. Otherwise, the stability of the scheme can be compromised. An implicit iterative scheme is proposed that decreases the errors at the boundary and, as a result, enhances stability.

A variant of the mass source approach [8] is adopted for use with moving boundaries, which improves local mass conservation near the IB and suppresses spurious force oscillations in moving body problems. The momentum forcing and mass source terms are applied throughout the solid region in order to satisfy mass conservation in cells cut by the boundary and remove the incompressibility constraint inside the solid. This requirement avoids temporal discontinuities in the velocity which can otherwise arise when solid points suddenly become IB points as a result of the boundary movement.

The IB method presented can be applied in generalized curvilinear systems, which facilitates efficient simulations of a wide range of complex geometries. The convergence, second-order accuracy and enhanced stability of the method are demonstrated through a number of test cases.

## Acknowledgements

The authors would like to acknowledge the financial support from the UK Engineering and Physical Sciences Research Council (EPSRC).

## Appendix A. Immersed boundary method in fractional step algorithm: $p$ versus $\Delta p$ -form

Using the same analysis as that performed by Kim and Moin [26], and introducing a forcing term on the right-hand-side of the intermediate velocity equation,  $\hat{u}_i$  is regarded as an approximation of the continuous function,  $u_i^*$ , that satisfies:

$$\frac{\partial u_i^*}{\partial t} = -\frac{\partial u_i^* u_j^*}{\partial x_j} - \alpha \frac{\partial p^*}{\partial x_i} + \frac{1}{Re} \frac{\partial^2 u_i^*}{\partial x_j \partial x_j} + f_i^*,$$

$$u_i^*(x, t_{n-1}) = u_i(x, t_{n-1}), \quad (A.1)$$

where  $\alpha = 0$  for the  $p$ -form of the fractional step method, and  $\alpha = 1$  for the  $\Delta p$ -form. Therefore,

$$\begin{aligned} \hat{u}_i &\approx u_i^*(x, t_{n-1} + \Delta t) \\ &= u_i^*(x, t_{n-1}) + \Delta t \frac{\partial u_i^*}{\partial t} + \frac{1}{2} \Delta t^2 \frac{\partial^2 u_i^*}{\partial t^2} + \mathcal{O}(\Delta t^3) \\ &= u_i^*(x, t_{n-1}) + \Delta t \left( -\frac{\partial u_i^* u_j^*}{\partial x_j} - \alpha \frac{\partial p^*}{\partial x_i} + \frac{1}{Re} \frac{\partial^2 u_i^*}{\partial x_j \partial x_j} + f_i^* \right) \\ &\quad + \frac{1}{2} \Delta t^2 \frac{\partial}{\partial t} \left( -\frac{\partial u_i^* u_j^*}{\partial x_j} - \alpha \frac{\partial p^*}{\partial x_i} + \frac{1}{Re} \frac{\partial^2 u_i^*}{\partial x_j \partial x_j} + f_i^* \right) + \mathcal{O}(\Delta t^3). \end{aligned} \quad (A.2)$$

Since  $u_i^*(x, t_{n-1}) = u_i(x, t_{n-1})$ , the above equation becomes,

$$\begin{aligned} \hat{u}_i &= u_i(x, t_{n-1}) + \Delta t \left( -\frac{\partial u_i u_j}{\partial x_j} - \alpha \frac{\partial p}{\partial x_i} + \frac{1}{Re} \frac{\partial^2 u_i}{\partial x_j \partial x_j} + f_i \right) + \mathcal{O}(\Delta t^2) \\ &= u_i(x, t_{n-1}) + \Delta t \left( \frac{\partial u_i}{\partial t} + (1 - \alpha) \frac{\partial p}{\partial x_i} \right) + \mathcal{O}(\Delta t^2) \\ &= u_i(x, t_n) + (1 - \alpha) \Delta t \frac{\partial p}{\partial x_i} + \mathcal{O}(\Delta t^2). \end{aligned} \quad (A.3)$$

Eq. (A.3) demonstrates that the immersed boundary conditions, which are applied on  $\hat{u}_i$  rather than  $u_i(x, t_n)$ , are second-order accurate in the  $\Delta p$ -form,  $\hat{u}_i = u_i(x, t_n) + \mathcal{O}(\Delta t^2)$ , but only first-order accurate in time in the  $p$ -form,  $\hat{u}_i = u_i(x, t_n) + \Delta t \frac{\partial p}{\partial x_i} + \mathcal{O}(\Delta t^2)$ .

Kim and Moin [26] demonstrated that if boundary conditions are inconsistent with the governing equations the solution can incur considerable numerical errors. For global boundaries, they proposed the appropriate treatment of the boundary conditions for the intermediate velocity field,  $\hat{u}_i = u_i^n + \Delta t G \phi^{n-1}$ , in order to maintain second-order accuracy. Similarly, a second-order accurate IB method in the  $p$ -form of the fractional step method would therefore require the intermediate velocity at the immersed boundary to be computed according to,

$$\hat{u}_r = U_r^n + \Delta t G \phi_r^{n-1}, \quad (A.4)$$

where  $U_r$  and  $\phi_r$  are the velocity and pseudo-pressure at the immersed boundary, respectively.

## Appendix B. Forcing and mass source terms for moving boundaries

We demonstrate, using a simple example, the need to apply the forcing and mass source terms throughout the solid region in order to avoid a temporal discontinuity in the velocity of solid points suddenly becoming IB points. First, we consider the case where the mass source terms are computed from IB fluxes only

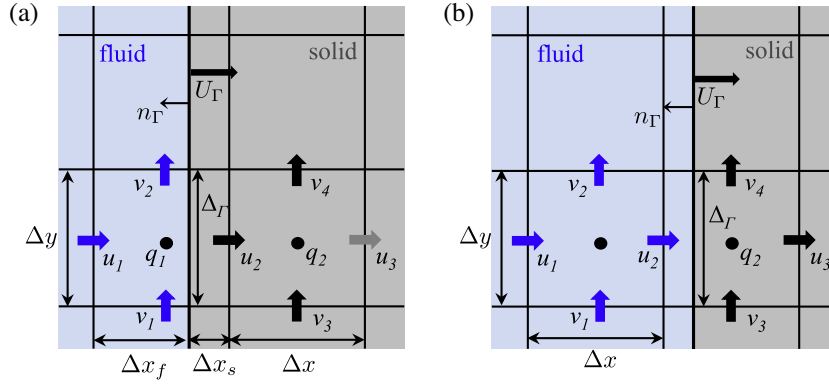


Fig. B.13. Mass conservation for cells near moving boundary: (a) at time  $t^n$ ; (b) at time  $t^{n+1}$ .

(Fig. B.13). This approach can lead to a temporal discontinuity in the velocity of solid points that suddenly become IB points due to the boundary motion. The temporal velocity discontinuity subsequently leads to spatial oscillations in the pressure near the boundary and SFOs.<sup>1</sup> In Fig. B.13a, the continuity equations at time  $t^n$  are given by,

$$u_2^n \Delta y - u_1^n \Delta y + v_2^n \Delta x - v_1^n \Delta x - q_1^n \Delta x \Delta y = 0 \quad \text{in cell 1,} \quad (\text{B.1})$$

$$u_3^n \Delta y - u_2^n \Delta y + v_4^n \Delta x - v_3^n \Delta x - q_2^n \Delta x \Delta y = 0 \quad \text{in cell 2,} \quad (\text{B.2})$$

where  $q_1^n = \frac{1}{\Delta x \Delta y} (u_2^n \Delta y + \beta_s v_2^n \Delta x - \beta_s v_1^n \Delta x - U_r^n \Delta y)$  and  $q_2^n = \frac{1}{\Delta x \Delta y} (-u_2^n \Delta y - v_3^n \Delta x + v_4^n \Delta x)$  are the source terms in those cells,<sup>2</sup> and  $\beta_s = \Delta x_s / \Delta x$ . The continuity equations reduce to,

$$U_r^n \Delta y - u_1^n \Delta y + v_2^n \Delta x_f - v_1^n \Delta x_f = 0 \quad \text{in cell 1,} \quad (\text{B.3})$$

$$u_3^n \Delta y = 0 \quad \text{in cell 2.} \quad (\text{B.4})$$

At the next time step (Fig. B.13b), solid point  $u_3$  becomes an IB point and its velocity is set to satisfy the no-slip condition at the boundary via a second-order interpolation:

$$u_3^{n+1} = U_r^{n+1} + \mathcal{O}(\Delta x^2). \quad (\text{B.5})$$

The velocity  $u_3$  therefore jumps from  $u_3(t^n) = 0$  to  $u_3(t^{n+1}) \approx U_r^{n+1} + \mathcal{O}(\Delta x^2)$ . This simple example demonstrates the temporal discontinuity in velocity that takes place when the mass source term is computed from IB fluxes only.

The temporal discontinuity in  $u_3$  is eliminated by including all IB and also all solid fluxes in the mass source term in Eq. (22). In the above example, at time  $t^n$  (Fig. B.13a), the source term and the continuity equation for the IB cell 1 are unchanged from above:

$$q_1^n = \frac{1}{\Delta x \Delta y} (u_2^n \Delta y + \beta_s v_2^n \Delta x - \beta_s v_1^n \Delta x - U_r^n \Delta y),$$

$$u_2^n \Delta y - u_1^n \Delta y + v_2^n \Delta x - v_1^n \Delta x - q_1^n \Delta x \Delta y = 0 \quad \text{in cell 1.}$$

In the solid cell 2, however, the source term should be  $q_2^n = \frac{1}{\Delta x \Delta y} (u_3^n \Delta y - u_2^n \Delta y - v_3^n \Delta x + v_4^n \Delta x)$ . In terms of the fractional step method, this source term explicitly balances the divergence in that cell,

$$D\hat{u} - q_2^n = 0, \quad (\text{B.6})$$

and the pressure Poisson equation reduces to,

$$DG\phi_2^n = 0. \quad (\text{B.7})$$

By virtue of the staggered grid arrangement, the same treatment carries over to all adjacent solid cells. The above expression for the solid cells can equivalently be described as eliminating the incompressibility constraint from those cells (but not from IB cell 1). Finally, note that the momentum forcing is also applied to all solid points to set their velocities to the body velocity. If the forcing term is evaluated explicitly, this step is only approximate. However, in the implicit forcing method described above, the intermediate velocity inside the solid is set exactly to the body velocity. As a result, the divergence-free constraint is automatically satisfied in solid cells away from the boundary and the mass source term in those cells vanishes. The mass source term is only non-zero in (a) cells cut by the boundary (e.g. cell 1 in Fig. B.13) in order to satisfy the continuity equation in the fluid fraction of the cell; and (b) adjacent solid cells (e.g. cell 2 in Fig. B.13) in order to explicitly balance the divergence in that cell and remove the incompressibility constraint as in the rest of the solid.

Note that approaches (21) and (22)/(23) yield different mass source terms in the solid cells adjacent to the boundary, e.g. cell 2 in Fig. B.13. In this cell, the source term (21), is equivalent to enforcing the incompressibility constraint,  $u_3^n \Delta y - U_r^n \Delta y = 0$ . With the herein proposed method (Eq. (22)/(23)), the source term annuls the continuity constraint entirely since  $q_2^n \equiv D\hat{u}$ . This cell is therefore consistent with the treatment of the remaining solid region.

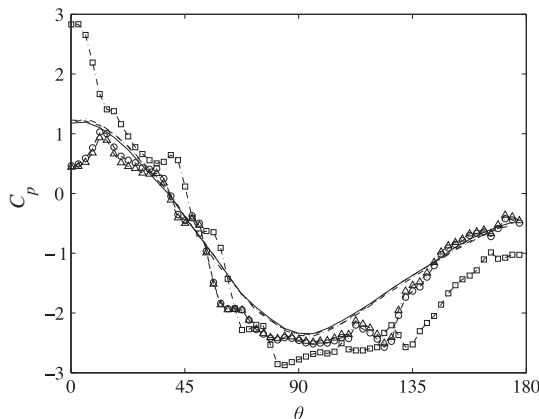
The above discussion is further supported by a numerical example of flow around an oscillating cylinder. Two approaches are considered: In the first, the momentum forcing and the mass source term take into account IB points only, which is expected to lead to pressure oscillations. In the second, the forcing and the mass source given by Eq. (22)/(23) are implemented at IB cells and throughout the solid, which is expected to eliminate SFOs. The Reynolds number is  $Re_D = U_{\max} D / \nu = 100$ , and the Keulegan-Carpenter number is  $KC = U_{\max} / fD = 5$ , where  $U_{\max}$  is the maximum velocity of the cylinder,  $D$  is the diameter of the cylinder,  $\nu$  is the kinematic viscosity of the fluid, and  $f$  is the frequency of oscillation. The streamwise location of the cylinder is given by  $x(t) = -A \sin(2\pi ft)$ , where  $A$  is the amplitude of the oscillation. The cylinder is located at the centre of a  $50D \times 50D$  domain. A  $513^2$  Cartesian grid is employed, with local spacing of  $0.02D$  in the vicinity of the cylinder, and the time step is set to  $\Delta t U_{\max} / D = 0.005$ .

Fig. B.14 shows the pressure coefficient along the surface of the cylinder at three consecutive time steps for the two cases. When the mass source term only accounts for IB fluxes, a temporal discontinuity in the velocity arises resulting in spatial oscillations in

<sup>1</sup> Note that the source of SFOs here is different from that reported by Lee et al. [18]. In that work, the temporal discontinuity arises from enforcing the no-slip condition on fluid points that become IB points.

<sup>2</sup> Note that the velocities  $u_i^n$  are unknown until Eqs. (4) and (5) are solved, therefore the intermediate velocities  $\tilde{u}_i$  are used to compute the mass source instead, without affecting the second-order accuracy of the scheme [8].





**Fig. B.14.** Pressure coefficient along the surface of an oscillating cylinder. Symbols correspond to mass source applied only near the immersed boundary; lines correspond to mass source applied everywhere in solid domain.  $tU_{\max}/D = 4.975$  ( $\circ$ , —);  $tU_{\max}/D = 4.985$  ( $\square$ , ---);  $tU_{\max}/D = 4.995$  ( $\Delta$ , -.-).

the pressure. For this case, significant oscillations in the pressure coefficient along the cylinder surface can be observed in Fig. B.14. The temporal discontinuity in velocity is removed and a smooth pressure coefficient is observed for the case where the mass source and forcing terms are applied at IB cells and throughout the solid region.

## References

- [1] Mittal R, Iaccarino G. Immersed boundary methods. *Annu Rev Fluid Mech* 2005;37:239–61.
- [2] Peskin CS. Flow patterns around heart valves: a numerical method. *J Comput Phys* 1972;10:252–71.
- [3] Beyer RP, Leveque RJ. Analysis of a one-dimensional model for the immersed boundary method. *SIAM J Numer Anal* 1992;29:332–64.
- [4] Goldstein D, Handler R, Sirovich L. Modeling a no-slip flow boundary with an external force field. *J Comput Phys* 1993;105:354–66.
- [5] Mohd-Yusof J. Combined immersed-boundary/B-spline methods for simulations of flow in complex geometries. *Center Turbul Res Annu Res Briefs* 1997;317–27.
- [6] Verzicco R, Mohd-Yusof J, Orlandi P, Haworth D. LES in complex geometries using boundary body forces. In: *Center for turbulence research (proceedings of the summer program)*; 1998. p. 171–86.
- [7] Fadlun EA, Verzicco R, Orlandi P, Mohd-Yusof J. Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations. *J Comput Phys* 2000;161:35–60.
- [8] Kim J, Kim D, Choi H. An immersed-boundary finite-volume method for simulations of flow in complex geometries. *J Comput Phys* 2001;171:132–50.
- [9] Balaras E. Modeling complex boundaries using an external force field on fixed Cartesian grids in large-eddy simulations. *Comput Fluids* 2004;33:375–404.
- [10] Taira K, Colonius T. The immersed boundary method: a projection approach. *J Comput Phys* 2007;225:2118–37.
- [11] Ikeno T, Kajishima T. Finite-difference immersed boundary method consistent with wall conditions for incompressible turbulent flow simulations. *J Comput Phys* 2007;226:1485–508.
- [12] Domenichini F. On the consistency of the direct forcing method in the fractional step solution of the Navier–Stokes equations. *J Comput Phys* 2008;227:6372–84.
- [13] Guy R, Hartenstein D. On the accuracy of direct forcing immersed boundary methods with projection methods. *J Comput Phys* 2010;229:2479–96.
- [14] Roman F, Napoli E, Milici B, Armenio V. An improved immersed boundary method for curvilinear grids. *Comput Fluids* 2009;38:1510–27.
- [15] Uhlmann M. An immersed boundary method with direct forcing for the simulation of particulate flows. *J Comput Phys* 2005;209:448–76.
- [16] Yang J, Balaras E. An embedded-boundary formulation for large-eddy simulation of turbulent flows interacting with moving boundaries. *J Comput Phys* 2006;215:12–40.
- [17] Liao CC, Chang YW, Lin CA, McDonough JM. Simulating flows with moving rigid boundary using immersed-boundary method. *Comput Fluids* 2010;39:152–67.
- [18] Lee J, Kim J, Choi H, Yang K-S. Sources of spurious force oscillations from an immersed boundary method for moving-body problems. *J Comput Phys* 2011;230:2677–95.
- [19] Seo JH, Mittal R. A sharp-interface immersed boundary method with improved mass conservation and reduced spurious pressure oscillations. *J Comput Phys* 2011;230:7347–63.
- [20] Lee J, You D. An implicit ghost-cell immersed boundary method for simulations of moving body problems with control of spurious oscillations. *J Comput Phys* 2013;233:295–314.
- [21] Rosenfeld M, Kwak D, Vinokur M. A fractional step solution method for the unsteady incompressible Navier–Stokes equation in generalized coordinate systems. *J Comput Phys* 1991;94:102–37.
- [22] Zaki TA, Wissink JG, Rodi W, Durbin PA. Direct numerical simulations of transition in a compressor cascade: the influence of free-stream turbulence. *J Fluid Mech* 2010;665:57–98.
- [23] Zaki TA. From streaks to spots and on to turbulence: exploring the dynamics of boundary layer transition. *Flow Turbul Combust* 2013;91:451–73.
- [24] Lee J, Yoon Jung S, Jin Sung H, Zaki TA. Effect of wall heating on turbulent boundary layers with temperature-dependent viscosity. *J Fluid Mech* 2013;726:196–225.
- [25] Chorin AJ. Numerical solution of the Navier–Stokes equations. *Math Comput* 1968;22:745–62.
- [26] Kim J, Moin P. Application of a fractional-step method to incompressible Navier–Stokes equations. *J Comput Phys* 1985;59:308–23.
- [27] Van Kan J. A second-order accurate pressure correction scheme for viscous incompressible flow. *SIAM J Sci Stat Comput* 1986;7:870–91.
- [28] Bell J, Colella P. A second-order projection method for the incompressible Navier–Stokes equations. *J Comput Phys* 1989;85:257–83.
- [29] Tyagi M, Acharya S. Large eddy simulation of turbulent flows in complex and moving rigid geometries using the immersed boundary method. *Int J Numer Methods Fluids* 2005;48:691–722.
- [30] Kempe T, Fröhlich J. An improved immersed boundary method with direct forcing for the simulation of particle laden flows. *J Comput Phys* 2012;231:3663–84.
- [31] Wang S, Zhang X. An immersed boundary method based on discrete stream function formulation for two- and three-dimensional incompressible flows. *J Comput Phys* 2011;230:3479–99.
- [32] Taylor G, Green AE. Mechanism of the production of small eddies from large ones. *Proc Roy Soc London: Ser A, Math Phys Sci* 1937;158:499–521.
- [33] Mittal R, Dong H, Bozkurtas M, Najjar FM, Vargas A, von Loebbecke A. A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries. *J Comput Phys* 2008;227:4825–52.
- [34] Ge L, Sotiropoulos F. A numerical method for solving the 3D unsteady incompressible Navier–Stokes equations in curvilinear domains with complex immersed boundaries. *J Comput Phys* 2007;225:1782–809.
- [35] Nicolaou L, Zaki TA. Direct numerical simulations of flow in realistic mouth-throat geometries. *J Aerosol Sci* 2013;57:71–87.
- [36] Shirayama S. Processing of computed vector fields for visualization. *J Comput Phys* 1993;30–41.
- [37] Shepard D. A two-dimensional interpolation function for irregularly-spaced data. In: *Proceedings 1968 ACM national conference*; 1968. p. 517–24.
- [38] Huang W-X, Sung HJ. Improvement of mass source/sink for an immersed boundary method. *Int J Numer Methods Fluids* 2007;53:1659–71.
- [39] Kim D, Choi H. Immersed boundary method for flow around an arbitrarily moving body. *J Comput Phys* 2006;212:662–80.
- [40] Rindt CCM, Van Steenhoven AA, Janssen JD, Vossler G. Unsteady entrance flow in a 90° curved tube. *J Fluid Mech* 1991;38:1510–27.
- [41] Womersley JR. Method for the calculation of velocity, rate of flow and viscous drag in arteries when the pressure gradient is known. *J Physiol* 1955;127:553–63.
- [42] Dütsch H, Durst F, Becker S, Lienhart H. Low-Reynolds-number flow around an oscillating circular cylinder at low Keulegan–Carpenter numbers. *J Fluid Mech* 1998;360:249–71.