INTERNATIONAL JOURNAL FOR NUMERICAL METHODS IN FLUIDS
Int. J. Numer. Meth. Fluids 2008; **56**:1845–1884
Published online 29 August 2007 in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/fld.1565

A divergence-free interpolation scheme for the immersed boundary method

Frank Muldoon[1,*,†] and Sumanta Acharya[2]

[1]*Center for Computation and Technology, Louisiana State University, Baton Rouge, LA 70803, U.S.A.*
[2]*Department of Mechanical Engineering, Louisiana State University, Baton Rouge, LA 70803, U.S.A.*

SUMMARY

The immersed boundary approach for the modeling of complex geometries in incompressible flows is examined critically from the perspective of satisfying boundary conditions and mass conservation. It is shown that the system of discretized equations for mass and momentum can be inconsistent, if the velocity is used in defining the force density to satisfy the boundary conditions. As a result, the velocity is generally not divergence free and the pressure at locations in the vicinity of the immersed boundary is not physical. However, the use of the pseudo-velocities in defining the force density, as frequently done when the governing equations are solved using a fractional step or projection method, combined with the use of the specified velocity on the immersed boundary, is shown to result in a consistent set of equations which allows a divergence-free velocity but, depending on the time step, is shown to have the undesirable effects of inaccurately satisfying the boundary conditions and allowing a significant permeability of the immersed boundary. If the time step is reduced sufficiently, the boundary conditions on the immersed boundary can be satisfied. However, this entails an unacceptable increase in computational expense. Two new methods that satisfy the boundary conditions and allow a divergence-free velocity while avoiding the increased computational expense are presented and shown to be second-order accurate in space. The first new method is based on local time step reduction. This method is suitable for problems where the immersed boundary does not move. For these problems, the first new method is shown to be closely related to the second new method. The second new method uses an optimization scheme to minimize the deviation from the interpolation stencil used to represent the immersed boundary while ensuring a divergence-free velocity. This method performs well for all problems, including those where the immersed boundary moves relative to the grid. Additional results include showing that the force density that is added to satisfy the boundary conditions at the immersed boundary is unbounded as the time step is reduced and that the pressure in the vicinity of the immersed boundary is unphysical, being strongly a function of the time step. A method of computing the total force on an immersed boundary which takes into account

[*]Correspondence to: Frank Muldoon, Center for Computation and Technology, Louisiana State University, Baton Rouge, LA 70803, U.S.A.

[†]E-mail: fmuldoo@me.lsu.edu

Contract/grant sponsor: Office of Naval Research
Contract/grant sponsor: Center for Computation and Technology at Louisiana State University

the specifics of the numerical solver used in the iterative process and correctly computes the total force irrespective of the residual level is also presented. Copyright © 2007 John Wiley & Sons, Ltd.

Received 7 March 2007; Revised 24 May 2007; Accepted 30 May 2007

KEY WORDS: immersed boundary; mass conversion; computational fluid dynamics

1. INTRODUCTION

One of the most difficult tasks in the numerical simulation of fluid flow is the generation of a grid around the object being modeled. This has always involved a large amount of time-consuming user interaction. If the object is moving and/or deforming, then the necessary regridding is an even greater and computationally expensive problem. After regridding, the solution variables must be transferred (i.e. interpolated) from the old to the new grid. This interpolation is a source of error and may also destroy the conservative properties of numerical schemes. The immersed boundary method [1–4] has the potential to simplify these problems associated with the grid. In particular, the computation of the grid around the object being modeled can be completely automated. In the immersed boundary method, the Navier–Stokes equations are generally solved on a Cartesian grid, which removes the effort needed to generate a body-fitted grid and enables the use of efficient numerical methods that can be parallelized in a relatively easy manner. The influence of objects on the flow is simulated by the addition of a force density (which represents the force of the surface of the object on the fluid) to the Navier–Stokes equations. This force density, if chosen properly, should result in a solution to the Navier–Stokes equations which satisfies the boundary conditions on the surface of the object. This is in contrast to other methods such as body-fitted curvilinear or unstructured grids, which require the grid to be built around or inside the objects being modeled.

While there are several methods of defining the force density added to the Navier–Stokes equations, the method of [5] has been widely used for flows in which the motion of the boundary is prescribed [3, 6–16]. The original immersed boundary method of [1] has been generally applied to flows in which the boundary is interacting with and deforming as a result of the flow. The lack of a stability restriction on the time step and lack of user defined parameters in the method of [5] appear to be the major reasons for its use.

The addition of the force density to the momentum equations, as done in the method of [5], to force the velocity to satisfy boundary conditions on the object's surface is equivalent to replacing certain momentum equations with interpolation equations involving only the velocity. As a consequence of these replacements, the resulting system of equations may form an inconsistent set of equations. With the iterative methods used in the present work to solve the discretized system of equations, this inconsistency results in an inability to obtain a divergence-free velocity. As the incompressible Navier–Stokes equations are generally solved using a fractional step or projection method, the force density is frequently defined using the pseudo-velocity instead of the velocity [11, 12, 15]. This is shown to result in a consistent set of equations; however, the present work shows that using the pseudo-velocity to define the force density can result in poor satisfaction of the boundary conditions on the surface of the object, which may result in permeability of an impermeable surface. However, the poor satisfaction of boundary conditions on the surface of the object may not greatly impact gross metrics such as drag coefficients; the authors and others [3, 7, 8, 12, 15] have computed drag coefficients for blunt body flows such as cylinders and spheres that have closely matched those obtained experimentally and from computational work involving body-fitted meshes. The present work demonstrates that satisfaction of the boundary conditions on the surface of the object is strongly a function of the time step, if the pseudo-velocity is used to define the force density. The authors present flows for which permeability, resulting from poor satisfaction of the boundary conditions on the surface of the object, is a significant problem and propose two new methods that correct the problem. This permeability can be a significant problem if the surface of the object contains an opening through which all the flow should pass that has a small area compared with the impermeable surface area of the object; a small leakage through the surface of the object over a large area can drastically reduce the flow through the opening. The new methods presented are based on the concept of satisfying the boundary conditions at the immersed boundary as closely as possible while enforcing a divergence-free velocity. In addition, regardless of whether the velocity or pseudo-velocity is used to define the force density, the pressure at certain points in the vicinity of the immersed boundary is shown to be unphysical. Also, a method of computing the total force on an immersed boundary which correctly computes the total force irrespective of the residual level is presented.

2. GOVERNING EQUATIONS AND DISCRETIZATION

2.1. Spatial discretization

The governing equations of interest in the present work are the non-dimensional non-conservative unsteady Navier–Stokes equations (Equations (1) and (2)). In the present work, the velocity vector is denoted by $\mathbf{u}$, the individual components of which are defined by $\mathbf{u}=\begin{pmatrix} u \\ v \end{pmatrix}$.

$$\frac{\partial \mathbf{u}}{\partial t}+\mathbf{u}\cdot\nabla\mathbf{u}=-\nabla p+\frac{1}{Re}\nabla^2\mathbf{u} \tag{1}$$

$$\nabla\cdot\mathbf{u}=0 \tag{2}$$

Using finite differences, Equations (1) and (2) are discretized on a staggered grid as shown in Figure 1. The staggered grid is used to avoid the appearance of spurious modes in the pressure field. In two dimensions, each pressure grid point is surrounded by four velocity grid points. Pressure is not needed or defined on non-periodic boundaries. For the convective terms, a third-order upwind scheme is used [17]. Due to the staggered grid, none of the velocity components are stored at the same location. They are, however, needed at the same location in order to compute the terms of $u_j(\partial u_i/\partial x_j)$, where $i\neq j$. They are obtained by fitting a two-dimensional Lagrange surface over the point in question. The fitting uses products of six point Lagrange polynomials, resulting in a total of 36 points used to fit the surface, and is sixth-order accurate. The number of points and hence the accuracy involved is retained as the boundary of the flow domain is approached. The diffusive terms are represented by sixth-order accurate central difference schemes [17]. The pressure gradient term and the terms in the continuity equation are represented by two-point second-order accurate centered stencils. In general, except for certain flow problems indicated in the present work, the order of all stencils is retained as a non-periodic boundary is approached. This is done by keeping the number of points in the stencil constant while shifting toward the boundary the point at which the derivative is evaluated. For a centered scheme, this results in

DOI: 10.1002/fld

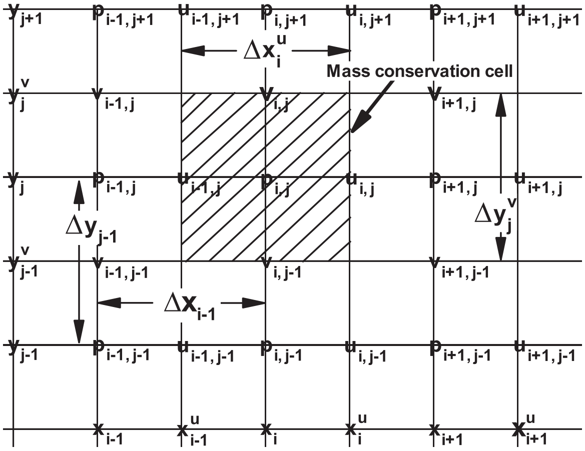

Figure 1. Staggered grid and mass conservation cell (shaded).

moving from a centered stencil to one that is biased away from the boundary. The stencils are not changed in the vicinity of an immersed boundary. The overall spatial accuracy of the scheme is determined by the lowest order finite differences used and is therefore second-order accurate. Further details of the finite difference schemes and the computer code used in the present work are given in [17]. Sixth-order accurate discretizations are used for the convective and diffusive terms to avoid their error dominating the second-order accurate continuity and pressure gradient terms which are the focus of the present work.

2.2. Temporal discretization

The spatially discretized convective and diffusive terms are integrated in time using either the explicit third-order Adams–Bashforth scheme (Equation (3)), the fully implicit first-order scheme or the third-order Adams–Moulton scheme (Equation (4)). The time step for which the solution is sought is denoted by $n+1$.

$$\begin{aligned}\frac{\mathbf{u}^{n+1}-\mathbf{u}^{n}}{\Delta t} &= -\nabla p^{n+1}+\mathbf{\Upsilon}^{n}\\ \mathbf{\Upsilon}^{n} &= \frac{23}{12}\left(\frac{1}{Re}\nabla^2\mathbf{u}^{n}-(\mathbf{u}\cdot\nabla\mathbf{u})^{n}\right)-\frac{16}{12}\left(\frac{1}{Re}\nabla^2\mathbf{u}^{n-1}-(\mathbf{u}\cdot\nabla\mathbf{u})^{n-1}\right)\\ &\quad+\frac{5}{12}\left(\frac{1}{Re}\nabla^2\mathbf{u}^{n-2}-(\mathbf{u}\cdot\nabla\mathbf{u})^{n-2}\right)\end{aligned}\tag{3}$$

The first-order fully implicit scheme ($\beta=1$, $\mathbf{\Phi}^{n}=0$) is used for steady-state problems for which time accuracy is not an issue and the additional stability of the scheme at low Re is desired. The pressure gradient term is integrated in time using the fully implicit scheme. Equation (2), which expresses conservation of mass, does not contain a temporal derivative; all terms in it are evaluated at the $n+1$ time step.

$$\begin{aligned}
&\frac{\mathbf{u}^{n+1}-\mathbf{u}^{n}}{\Delta t} = -\nabla p^{n+1} + \beta\left(\frac{1}{Re}\nabla^2\mathbf{u}^{n+1} - (\mathbf{u}\cdot\nabla\mathbf{u})^{n+1}\right) + \mathbf{\Phi}^n \\
&\text{fully implicit} \rightarrow \beta = 1, \quad \mathbf{\Phi}^n = 0 \\
&\text{Adams–Moulton} \rightarrow \beta = \frac{5}{12}, \quad \mathbf{\Phi}^n = \frac{8}{12}\left(\frac{1}{Re}\nabla^2\mathbf{u}^{n} - (\mathbf{u}\cdot\nabla\mathbf{u})^{n}\right) \\
&\qquad - \frac{1}{12}\left(\frac{1}{Re}\nabla^2\mathbf{u}^{n-1} - (\mathbf{u}\cdot\nabla\mathbf{u})^{n-1}\right)
\end{aligned} \tag{4}$$

2.3. *Discretized system solved at each time step*

If the explicit time integration scheme (Equation (3)) is used for the convective and diffusive terms, the discretized system solved at each time step is linear and relatively simple (as the complicated convective and diffusive terms appear only at previous time steps). Representing the pressure gradient and the continuity equation by finite differences, the system of equations to be solved at each time step is given by the following equations (Δx_i^u, Δy_j^v, Δx_i, Δy_j are defined in Figure 1):

$$\frac{u_{i,j}^{n+1}-u_{i-1,j}^{n+1}}{\Delta x_i^u} + \frac{v_{i,j}^{n+1}-v_{i,j-1}^{n+1}}{\Delta y_j^v} = 0 \tag{5}$$

$$\begin{aligned}
&\frac{u_{i,j}^{n+1}}{\Delta t} + \frac{p_{i+1,j}^{n+1}-p_{i,j}^{n+1}}{\Delta x_i} = X_{i,j}^n \quad \text{where } X_{i,j}^n \text{ represents all terms at previous time steps} \\
&\frac{v_{i,j}^{n+1}}{\Delta t} + \frac{p_{i,j+1}^{n+1}-p_{i,j}^{n+1}}{\Delta y_j} = Y_{i,j}^n \quad \text{where } Y_{i,j}^n \text{ represents all terms at previous time steps}
\end{aligned} \tag{6}$$

Equations (5) and (6) evaluated at all grid points in the computational domain can be written in matrix form as Equation (7), where $\mathbf{q}$ is the vector of all discrete variables (u^{n+1}, v^{n+1}, p^{n+1}) over the entire computational domain and R is on the right-hand side:

$$A\mathbf{q} = R \tag{7}$$

It can be shown that A has a non-zero determinant, if one of the discrete divergence equations is replaced by an equation that sets the pressure to a constant at one grid point, which is necessary as pressure can only be determined up to a constant in an incompressible flow with specified velocity boundary conditions. As a result, there is a guaranteed solution at each time step, which may be found by any number of direct or iterative methods. When the implicit time integration scheme (Equation (4)) is used, the resulting system of equations to be solved at each time step is non-linear and much more complicated with A in Equation (7) depending linearly on the velocity. In the present work, the solution to the systems of equations arising from both explicit and implicit time integration of the convection–diffusion terms is obtained by the use of a colored version of symmetric coupled Gauss Seidel (SCGS) [17, 18].

DOI: 10.1002/fld

3. IMMERSED BOUNDARY METHOD, BI METHOD

The surface of an object placed or immersed in a fluid will exert a stress on the fluid; the concept of the immersed boundary method is that the presence of this object in the fluid can be represented by solving the Navier–Stokes equations (Equations (1) and (2)) while taking into account the stress which the object exerts on the fluid. The stress which the immersed boundary exerts on the fluid has dimensions of *force per area*, however, the momentum equations have dimensions of *force per volume*. Therefore, the stress which the immersed boundary exerts on the fluid must be transformed into a force density which may then be added to the momentum equations.

3.1. Defining the force density

This force density (denoted by $\mathbf{F}^{n+1}$) represents the force of the object on the fluid, and is chosen so that the velocity will satisfy the appropriate boundary conditions on the object's surface (which is also referred to as the immersed boundary). A number of different forms have been used for $\mathbf{F}^{n+1}$ [1, 2, 4, 5]. The form of the force density used in [1, 4] is suitable for objects that are interacting (i.e. deforming) with the flow. The present work is concerned with rigid non-deforming objects for which the method of [5] is intended and used in [3, 7–16]. This method avoids the severe time step restrictions and the difficulty in choosing two free parameters of the method of [2] and is the basis for the methods used in the present work. In the method of [5], after applying the spatial and temporal discretization schemes (without loss of generality the method is described using the explicit scheme, Equation (3)), the velocity at the $n+1$ time step can be written as:

$$\frac{\mathbf{u}^{n+1}-\mathbf{u}^{n}}{\Delta t}=-\nabla p^{n+1}+\boldsymbol{\Upsilon}^{n}+\mathbf{F}^{n+1} \tag{8}$$

Denoting the desired velocity on the immersed boundary at the $n+1$ time step by $\mathbf{u}_{\mathrm{IB}}^{n+1}$ (where IB indicates points on the immersed boundary), the value of $\mathbf{F}^{n+1}$ that results in $\mathbf{u}_{\mathrm{IB}}^{n+1}$ at the boundary is simply:

$$\mathbf{F}^{n+1}=\frac{\mathbf{u}_{\mathrm{IB}}^{n+1}-\mathbf{u}^{n}}{\Delta t}-(-\nabla p^{n+1}+\boldsymbol{\Upsilon}^{n}) \tag{9}$$

This is equivalent to setting $\mathbf{u}^{n+1}=\mathbf{u}_{\mathrm{IB}}^{n+1}$ at every time step (i.e. replace $\mathbf{F}^{n+1}$ in Equation (8) with $\mathbf{F}^{n+1}$ from Equation (9)), if the immersed boundary coincides with the velocity grid points; if not, then an interpolation operator will be necessary to find $\mathbf{u}^{n+1}$ at the grid points (near the immersed boundary), where the force density $\mathbf{F}^{n+1}$ is added. This operator is written as $L(\mathbf{u}_{\mathrm{NB}}^{n+1},\mathbf{u}_{\mathrm{IB}}^{n+1})=L_{\mathrm{NB}}(\mathbf{u}_{\mathrm{NB}}^{n+1})+L_{\mathrm{IB}}(\mathbf{u}_{\mathrm{IB}}^{n+1})$, where the subscript NB indicates neighboring grid points on the finite difference grid (not necessarily on the immersed boundary) and $L_{\mathrm{NB}}, L_{\mathrm{IB}}$ are a set of coefficients depending on the interpolation scheme, the grid and the position of the immersed boundary. If Equation (9) is substituted into Equation (8) (using the interpolation stencil), the following equation results:

$$\frac{\mathbf{u}^{n+1}-\mathbf{u}^{n}}{\Delta t}=-\nabla p^{n+1}+\boldsymbol{\Upsilon}^{n}+\overbrace{\left(\frac{L_{\mathrm{NB}}(\mathbf{u}_{\mathrm{NB}}^{n+1})+L_{\mathrm{IB}}(\mathbf{u}_{\mathrm{IB}}^{n+1})-\mathbf{u}^{n}}{\Delta t}-(-\nabla p^{n+1}+\boldsymbol{\Upsilon}^{n})\right)}^{\mathbf{F}^{n+1}} \tag{10}$$

DOI: 10.1002/fld

The above equation simplifies to Equation (11), which replaces the discretized momentum equation at grid points adjacent to the immersed boundary at which the force density $\mathbf{F}^{n+1}$ is added (hereafter called immersed boundary points). This method is referred to in the present work by the acronym 'BI' for 'Basic Interpolation'.

$$\mathbf{u}^{n+1} = L_{\mathrm{NB}}(\mathbf{u}_{\mathrm{NB}}^{n+1}) + L_{\mathrm{IB}}(\mathbf{u}_{\mathrm{IB}}^{n+1}) \tag{11}$$

3.2. Selection of immersed boundary points

As the Cartesian grid points will generally not coincide with the immersed boundary, it is necessary to decide at which grid points $\mathbf{F}^{n+1}$ will be added; these grid points being termed immersed boundary points. It is conceivable to add the force density at all grid points. However, as the influence of an immersed boundary on a distant grid point would be expected to be small, it makes physical and computational sense to limit the addition of $\mathbf{F}^{n+1}$ to a subset of grid points near the immersed boundary. The selection scheme to determine this subset of grid points is as follows. All interior velocity points are examined to see if a boundary lies between them and one or more of its four neighbors $(i-1, j)$, $(i+1, j)$, $(i, j-1)$, $(i, j+1)$. These grid points are marked as possible immersed boundary points. Each mass conservation cell (Figure 1) is then examined to see if the four velocity grid points that surround it are possible immersed boundary points. If they are, then all four of the points are defined as immersed boundary points. The result of this selection process is that all immersed boundary points belong to either one or two mass conservation cells that have all four velocity grid points determined by the immersed boundary interpolation procedure. Some investigators [15], when representing flow over an object, have chosen to add the force density only at grid points on or inside the object. The rational for doing this is that only the flow outside the object is of interest. However, this approach prevents modeling of immersed boundaries with a thickness less than the spacing between velocity grid points, where grid points on either side of the immersed boundary lie in the fluid region and must have the force density added to satisfy the boundary conditions on the immersed boundary. This one-sided approach is not used in the present work as no distinction is made between the inside or outside of an immersed boundary.

3.3. Interpolation operator

Once a choice has been made of the grid points at which the force density will be added, a value for the force density must be determined. The determination of $\mathbf{F}^{n+1}$ at a velocity grid point is equivalent to setting the velocity at that grid point to a desired value. As the grid points will generally not coincide with the immersed boundary, the determination of what value to set the velocity to will require an interpolation operator $L(\mathbf{u}_{\mathrm{NB}}^{n+1}, \mathbf{u}_{\mathrm{IB}}^{n+1})$. This interpolation operator will interpolate between the velocity of the immersed boundary $(\mathbf{u}_{\mathrm{IB}}^{n+1})$ and velocities $(\mathbf{u}_{\mathrm{NB}}^{n+1})$ at neighboring grid points to find a value for the velocity at a point at which the force density is added, i.e. an immersed boundary point. The simplest choice is one-dimensional interpolation along coordinate lines. For instance, the immersed boundary point $u_{i,j-1}$ in Figure 2 can be determined by second-order accurate one-dimensional interpolation in the $+j$ direction using the points u_{IB}^{+j}, $u_{i,j}$, $u_{i,j+1}$ (where u_{IB} is the x component of velocity on the immersed boundary). However, $u_{i,j-1}$ could also be determined by the second-order accurate one-dimensional interpolation in the $+i$ direction using the points u_{IB}^{+i}, $u_{i+1,j-1}$, $u_{i+2,j-1}$. In general, in two dimensions, there can be up to four possible directions to interpolate. The present work resolves this ambiguity by interpolating in the direction in which the immersed boundary is closest to the grid point (the stencil chosen is shown in

DOI: 10.1002/fld

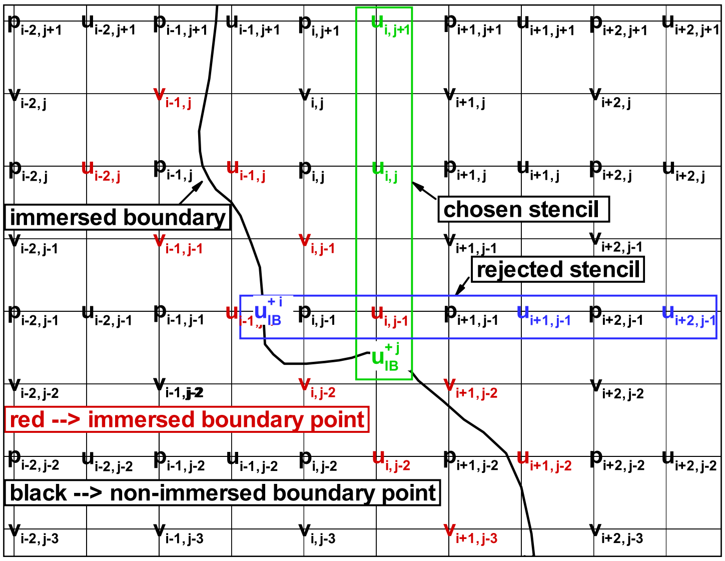

Figure 2. A stencil for interpolation of immersed boundary points.

Figure 2). While a wide variety of interpolation schemes have been developed and used [3, 7–16], the key point here is that the method of [5] is equivalent to replacing the momentum equation at each immersed boundary point with an equation for the velocity that is a function of velocities (not pressure) at other grid points. The present work uses second-order accurate one-dimensional interpolation with the choice of stencils described above for the operator $L_{\mathrm{NB}}(\mathbf{u}_{\mathrm{NB}}^{n+1})+L_{\mathrm{IB}}(\mathbf{u}_{\mathrm{IB}}^{n+1})$ in Equation (11).

3.4. Rank deficient matrix

An example of the result of replacing the momentum equation at immersed boundary points with an interpolation stencil is shown for the simple geometry and immersed boundary in Figure 3. For clarity, one-sided linear interpolation of only the x component of velocity is used at the immersed boundary. The grid spacing is a constant in both dimensions, while u_{IB} is the x component of velocity on the immersed boundary; no slip boundary conditions are applied on all four walls. Using the method of [5] force density is added at the circled points, i.e. the momentum equation at the circled points in Figure 3 is replaced with the interpolation stencil arising from the immersed boundary. The resulting matrix equation (i.e. Equation (7), where A has been modified by the immersed boundary method) for all interior velocities and pressures is found (using the symbolic algebra package Maple©) to be rank deficient of degree 1 less than the dimension of the matrix. Note that as pressure, in an incompressible fluid with specified velocity boundary conditions, can be determined only up to a constant, the pressure at one point is set to a constant. This rank deficiency means that, depending on the right-hand side of Equation (7), the system of equations may be inconsistent and have no solution or may have an infinite number of solutions. More complicated immersed boundaries may result in matrices with higher degrees of rank deficiency.

DOI: 10.1002/fld

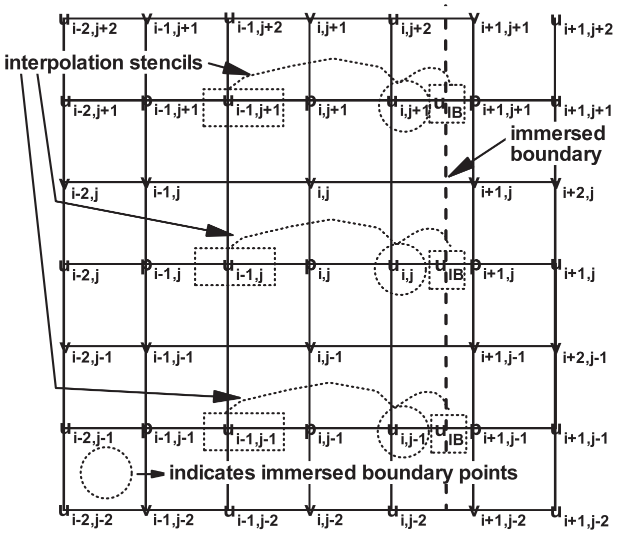

Figure 3. An immersed boundary resulting in a rank deficient matrix.

3.5. Conservation of mass

A problem with the interpolation procedure of Section 3.3 is that the continuity equation is not taken into account. Thus, it is likely that the velocity resulting from applying Equation (11) will not be divergence free. For an example of how the interpolation procedure can violate conservation of mass, consider the four velocities, $u_{i-2,j}$, $u_{i-1,j}$, $v_{i-1,j-1}$, $v_{i-1,j}$, defining the mass conservation cell at the $p_{i-1,j}$ location in Figure 2. These velocities are not determined from the solution of the discretized Navier–Stokes equations. They are determined by the interpolation scheme, which makes them a linear combination of velocities at other grid points. In general, as the interpolation scheme is defined without regard to the continuity equation, the velocities obtained from the interpolation scheme will not satisfy conservation of mass for the cell shown. This will cause the magnitude of the pressure in these mass conservation cells to slowly increase without bound, if a solution method such as artificial compressibility, SCGS, colored SCGS, SCSG-PPV [17] or SCGS-PP [17] is used, in which the change in pressure from one iteration to the next is proportional to the divergence. However, as this pressure does not appear in any discrete momentum equation (as a result of the two-point pressure gradient stencil and the selection criteria for immersed boundary points), the solution does not diverge. This pressure is, therefore, completely decoupled from all other discrete variables. With the iterative solver used in the present work, the result of attempting to solve the inconsistent set of equations is a ‘solution’ that satisfies the boundary conditions resulting from the immersed boundary, but does not conserve mass. In the present work, this ‘solution’ is obtained from the BI method. Another way of viewing this failure to conserve mass is that the replacement of the momentum equation at each immersed boundary point with an equation for the velocity that is a linear combination of velocities at non-immersed boundary grid points can result in changing A in Equation (7) to a rank deficient matrix. This rank deficient matrix equation is an inconsistent set of equations as a result of attempting to satisfy both the divergence-free constraint and the boundary conditions resulting from the immersed boundary (as expressed by the interpolation operator, Equation (11)).

DOI: 10.1002/fld

There is little work in the literature regarding this issue of inconsistency and mass conservation; a method by which a mass source term is added to the continuity equation (Equation (2)) is presented in [15]. This mass source term is equal to the negative of the mass source (resulting only from the contribution of immersed boundary points in that cell) at each mass conservation cell. The velocities at these immersed boundary grid points are obtained by interpolating the pseudo-velocities, which are obtained by advancing the momentum equations in time without the pressure gradient at the current time step. By definition, the sum of this added mass source term over the entire domain is zero and, therefore, global mass conservation is not affected. However, because adding the force density term (which using the method of [5] is equivalent to interpolating the velocity onto the immersed boundary point) should depend on the velocity at the time step being solved for, and not on the pseudo-velocity, the resulting velocity will not satisfy the desired velocity at the boundary. Another issue is that due to the added mass source term, the velocity obtained by using the pressure determined from the pressure-Poisson equation will not be divergence free. However, in [15], results for various cases are presented showing improvements due to the addition of the mass source term. A method of ensuring mass conservation was discussed in [19], where the scheme was found to be spatially first-order accurate. Specifics of the scheme in [19] were not presented and we are unable to make comparisons with it.

While only the Navier–Stokes equations are solved in the present work, the ability to obtain a mass conserving velocity field is critical if this velocity is used in the solution of equations governing the transport of energy and/or chemical species. In mass conservation cells in which mass is not conserved, there will be a source or sink of energy and/or chemical species which can result in unbounded solutions.

4. DEFINITION OF FORCE DENSITY USING PSEUDO-VELOCITY, PV METHOD

4.1. Fractional step and definition of pseudo-velocities

A fractional step (or projection) method [20] to solve Equation (3) or (4) and Equation (2) is defined by splitting Equation (3) (explicit time integration) or Equation (4) (implicit time integration) into two parts (Equations (12) and (14) or Equations (13) and (14), respectively). The splitting is done to define a pseudo-velocity ($\widetilde{\mathbf{u}}$) that does not depend on the pressure at the $n+1$ time step (the time step at which the solution is being sought).

$$\frac{\widetilde{\mathbf{u}}-\mathbf{u}^n}{\Delta t}=\boldsymbol{\Upsilon}^n \tag{12}$$

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

$$\frac{\mathbf{u}^{n+1}-\widetilde{\mathbf{u}}}{\Delta t}=-\nabla p^{n+1} \tag{14}$$

Equation (12) or (13) is advanced in time to obtain a pseudo-velocity $\widetilde{\mathbf{u}}$ (note that Equation (13) is a coupled non-linear system of the components of the velocity vector). This pseudo-velocity generally will not satisfy the divergence-free condition. An equation for pressure, which can be used to obtain a divergence-free velocity, can be obtained by taking the divergence of

Equation (14) and setting the divergence of the velocity at the $n+1$ time step to zero, resulting in Equation (15).

$$\nabla \cdot \tilde{\mathbf{u}} = \Delta t \nabla^2 p \tag{15}$$

Once the pressure field is obtained from the solution of Equation (15), the divergence-free velocity at the $n+1$ time step can be obtained from Equation (14). While the fractional step approach and numerous related variants (such as PISO [21], SIMPLE [22] and SIMPLER [22]) which involve an intermediate velocity that does not depend on the pressure gradient at the current time step are the most common methods for solving the incompressible Navier–Stokes equations, other methods such as SCGS [18], SCGS-PP [17], SCGS-PPV [17], artificial compressibility [23] and the direct solution of Equation (3) or (4) and Equation (2) can be used to obtain a solution.

4.2. Calculation of force density

4.2.1. Explicit time integration scheme. In the commonly used fractional step or projection methods, the pseudo-velocity is readily available, while the velocity is unknown at the step in which the force density is needed. For this reason, the force density has been defined using the pseudo-velocity [8, 9, 11, 12, 15]. In this case, at immersed boundary points, instead of choosing $\mathbf{F}^{n+1}$ such that the velocity ($\mathbf{u}^{n+1}$) satisfies a boundary condition on the immersed boundary, $\mathbf{F}^{n+1}$ is chosen so that the pseudo-velocity ($\tilde{\mathbf{u}}$) satisfies a boundary condition on the immersed boundary. The result is Equation (16), where the force density depends on the pseudo-velocity and therefore depends only on the previous time step and not on the pressure gradient or velocity on the finite difference grid at the $n+1$ time level.

$$\mathbf{F}^{n+1} = \frac{L_{\mathrm{NB}}(\tilde{\mathbf{u}}_{\mathrm{NB}}) + L_{\mathrm{IB}}(\mathbf{u}_{\mathrm{IB}}^{n+1}) - \mathbf{u}^n}{\Delta t} - \boldsymbol{\Upsilon}^n \tag{16}$$

As the immersed boundary generally does not coincide with the grid points, the interpolation operator $L_{\mathrm{NB}}() + L_{\mathrm{IB}}()$ is used to define the pseudo-velocity at immersed boundary points in Equation (16). Points in the interpolation operator that lie on the immersed boundary can be taken as the velocity ($\mathbf{u}_{\mathrm{IB}}^{n+1}$, since this is known), as opposed to the pseudo-velocity.

Substituting for $\mathbf{F}^{n+1}$ in Equation (8) using $\mathbf{F}^{n+1}$ from Equation (16):

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^n}{\Delta t} = -\nabla p^{n+1} + \boldsymbol{\Upsilon}^n + \left(\overbrace{\frac{L_{\mathrm{NB}}(\tilde{\mathbf{u}}_{\mathrm{NB}}) + L_{\mathrm{IB}}(\mathbf{u}_{\mathrm{IB}}^{n+1}) - \mathbf{u}^n}{\Delta t} - \boldsymbol{\Upsilon}^n}^{\mathbf{F}^{n+1}} \right)$$

Simplifying:

$$\frac{\mathbf{u}^{n+1} - (L_{\mathrm{NB}}(\tilde{\mathbf{u}}_{\mathrm{NB}}) + L_{\mathrm{IB}}(\mathbf{u}_{\mathrm{IB}}^{n+1}))}{\Delta t} = -\nabla p^{n+1} \tag{17}$$

Using the definition of $\tilde{\mathbf{u}}$ from Equation (12) and the linearity of L_{NB}, this can be written as Equation (18), which replaces the momentum equation (Equation (3)) at immersed boundary points.

$$\frac{\mathbf{u}^{n+1} - (L_{\mathrm{NB}}(\mathbf{u}_{\mathrm{NB}}^n) + L_{\mathrm{IB}}(\mathbf{u}_{\mathrm{IB}}^{n+1}))}{\Delta t} = -\nabla p^{n+1} + L_{\mathrm{NB}}(\boldsymbol{\Upsilon}_{\mathrm{NB}}^n) \tag{18}$$

DOI: 10.1002/fld

4.2.2. Implicit time integration scheme. Note that the splitting of Equation (4) as used in the fractional step method results in the sum of the two split parts (Equations (13) and (14)) equaling Equation (19) instead of Equation (4). It may be thought that the problem of satisfying the boundary conditions on the immersed boundary is related to the error in splitting the implicit scheme. This possibility is eliminated by using colored SCGS (which has no splitting error) as the solver, instead of the fractional step method.

$$\frac{\mathbf{u}^{n+1}-\mathbf{u}^{n}}{\Delta t}=-\nabla p^{n+1}+\beta\left(\frac{1}{Re}\nabla^{2}\widetilde{\mathbf{u}}-(\widetilde{\mathbf{u}}\cdot\nabla\widetilde{\mathbf{u}})\right)+\mathbf{\Phi}^{n} \tag{19}$$

For the implicit time integration scheme, the force density is given by the following equation:

$$\mathbf{F}^{n+1}=\frac{L_{\mathrm{NB}}(\widetilde{\mathbf{u}}_{\mathrm{NB}})+L_{\mathrm{IB}}(\mathbf{u}_{\mathrm{IB}}^{n+1})-\mathbf{u}^{n}}{\Delta t}-\left(\beta\left(\frac{1}{Re}\nabla^{2}\widetilde{\mathbf{u}}-(\widetilde{\mathbf{u}}\cdot\nabla\widetilde{\mathbf{u}})\right)+\mathbf{\Phi}^{n}\right) \tag{20}$$

However, unlike the explicit time integration scheme, there is no simple expression for the pseudo-velocity as it is obtained from solving a coupled system of equations (Equation (13)); the equation which replaces the momentum equation (Equation (4)) at immersed boundary points is given by the following equation:

$$\begin{aligned}\frac{\mathbf{u}^{n+1}-(L_{\mathrm{NB}}(\widetilde{\mathbf{u}}_{\mathrm{NB}})+L_{\mathrm{IB}}(\mathbf{u}_{\mathrm{IB}}^{n+1}))}{\Delta t}&=-\nabla p^{n+1}+\beta\left(\frac{1}{Re}\nabla^{2}\mathbf{u}^{n+1}-(\mathbf{u}\cdot\nabla\mathbf{u})^{n+1}\right)\\&\quad-\beta\left(\frac{1}{Re}\nabla^{2}\widetilde{\mathbf{u}}-(\widetilde{\mathbf{u}}\cdot\nabla\widetilde{\mathbf{u}})\right)\end{aligned} \tag{21}$$

4.2.3. Structure of equations. Note that Equation (18) (explicit time integration) and Equation (21) (implicit time integration) have exactly the same terms evaluated at the $n+1$ time step, respectively, as Equation (3) or (4). Therefore, for the explicit scheme, the matrix for the resulting system of discretized equations is the same as in Equation (7) with only the right-hand side being different. As a result, the solution to Equation (18) along with the continuity equation will guarantee a divergence-free velocity. For both the explicit and implicit schemes, the addition of the force density has resulted only in a change in the source term (i.e. the term not at the $n+1$ time step) at each immersed boundary point. The problem with this approach is that the force density is determined such that the pseudo-velocity at the immersed boundary points satisfies the desired velocity at the immersed boundary. This approach is incorrect because the force density should be chosen such that the velocity (not the pseudo-velocity) at the immersed boundary points satisfies the desired velocity at the immersed boundary. As the difference between the pseudo-velocity and the velocity is $\Delta t\nabla p^{n+1}$ the use of the pseudo-velocity means that the pressure gradient at the $n+1$ time step does not influence the force density. Also, since the difference between the pseudo-velocity and the velocity is proportional to Δt, the issue arises as to how the time step affects the solution, including the possible undesirable dependence of a steady-state solution on the time step. It is important to note that the ability to define the force density using the pseudo-velocity is a feature not only of fractional step methods, but can also be applied to any method used to solve Equation (3) or (4) and Equation (2). In the present work, this method of defining the force density is referred to by the acronym 'PV' for 'Pseudo Velocity'.

5. TWO-DIMENSIONAL OBSTACLE

5.1. Grid dependence

The BI and PV methods are demonstrated on a two-dimensional obstacle with a single tube through which the flow should pass. A schematic of the problem along with inflow and outflow boundary conditions, the origin of the coordinate system and dimensions is given in Figure 4. All quantities are non-dimensionalized by the tube diameter d and the average velocity through the tube u_0, with a Reynolds number of 10^6 based on d and u_0. This high Reynolds number is chosen to avoid the instability of the diffusive terms for the explicit time integration scheme. The initial conditions are $u=\frac{1}{15}$, $v=0$ on the boundaries and interior, and constant pressure in the interior. For all time steps, a uniform profile of $u=\frac{1}{15}$ at $x=-10d$ is maintained, so that the non-dimensional mass flow is 1. The tube extends from $-d \leqslant x \leqslant d$ and $-0.5d \leqslant y \leqslant 0.5d$. The spatially averaged mass flow within the tube is determined by finding the mass flow in the x direction within $-0.5d \leqslant y \leqslant 0.5d$ at each x_i^u (x_i^u is defined in Figure 1) for which $-d \leqslant x_i^u \leqslant d$ and averaging. The immersed boundary method is used to model the no-slip surfaces of the obstacle indicated in Figure 4 by dashed lines, while no slip wall boundaries (without the use of immersed boundaries) are applied at $y=-7.5d$ and $y=7.5d$. A convective outflow boundary condition is applied for both the velocities at $x=30d$. Results are obtained on four different grids with dimensions of 41×21, 81×41, 161×81 and 321×161, denoted on plots as grids $(1,2,3,4)$. The grids are stretched in both dimensions, to concentrate points near the origin (center of the tube). The time step on the 41×21 grid is 0.05 for the explicit time integration scheme and 0.25 for the implicit Adams–Moulton time integration scheme, resulting in maximum CFL numbers of ~ 0.2, ~ 0.1, respectively. The time step is halved when the next finer grid in the sequence is used. The flow is advanced a total of 0.05 (explicit time integration scheme) or 0.25 (implicit time integration scheme) non-dimensional units in time; this results in twice as many discrete time steps on the next finest grid in the sequence. Therefore one, two, four and eight time steps are taken on the 41×21, 81×41, 161×81 and 321×161 grids, respectively. Due to the boundary conditions, all the mass flow should go through the opening in the immersed boundary, if the boundary conditions on the immersed boundary are enforced.

The L_∞ norms of the residuals for both the explicit and implicit schemes can be decreased at each time step to machine zero for the PV method, while for the BI method they remain ~ 1. As reducing the residuals to machine zero at each time step is computationally expensive, the effect of reducing the residuals to different levels at each time step for this particular problem is studied; it is found that the results are independent of the residual if the L_∞ norms of the residuals $\leqslant 10^{-4}$.

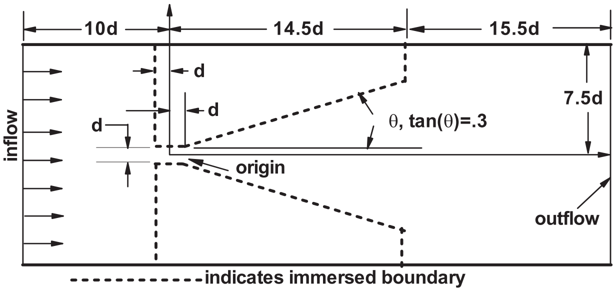

Figure 4. Schematic of a two-dimensional obstacle.

DOI: 10.1002/fld

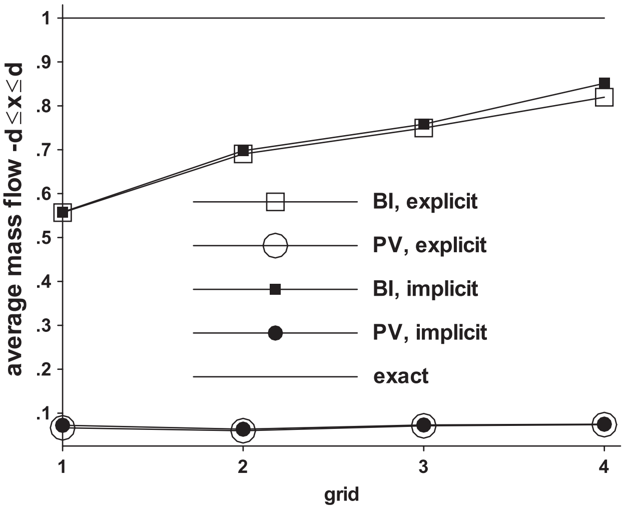

Figure 5. Spatial ($-d\leqslant x\leqslant d$) average of mass flow through a tube as a function of grid size, two-dimensional obstacle, $Re=10^6$.

For the PV method, the L_∞ norms are reduced to this level at each time step. However, for the BI method, since the addition of the force density has resulted in an inconsistent set of equations, the L_∞ norm of the residuals cannot be reduced beyond ~ 1. Figure 5 shows that the PV method results in very low values of mass flow through the tube, which is not improved by reducing the grid spacing. As global and local mass is conserved with the PV method, the mass that does not pass through the tube goes through the immersed boundary (which should be impermeable); therefore, the velocity boundary condition at the immersed boundary is poorly satisfied. The BI method gives much better results which improve as the grid spacing is reduced, however, local (and global) mass is not conserved.

5.2. *Effect of time step*

The satisfaction of the velocity boundary condition on the immersed boundary is greatly improved by reducing the time step. This is shown by a simulation of the two-dimensional obstacle on the 41×21 grid at a Reynolds number of 20, chosen so that a steady flow results from the combination of Reynolds number, grid and finite differencing scheme. Since the solution is steady, the results presented are of the steady-state solution. The time step is decreased progressively by a factor of 2 starting from $\Delta t_{\max}=0.08$ (explicit time integration scheme) or $\Delta t_{\max}=0.5$ (implicit Adams–Moulton time integration scheme), while all other parameters remain the same. At these $\Delta t_{\max}$, the maximum CFL numbers are ~ 0.21, ~ 1.35, respectively. Using the PV method, the mass flow converges (Figure 6) to the correct value as the time step is decreased. However, for the explicit time integration scheme, the mass flow does not asymptote until $\log_2(\Delta t_{\max}/\Delta t)\geqslant 8$, while for the implicit time integration scheme it does not asymptote until $\log_2(\Delta t_{\max}/\Delta t)\geqslant 10$, meaning the computational expense to obtain this solution is 2^8 or 2^{10} times that required to obtain the solution using $\Delta t=\Delta t_{\max}$. This dependence of the steady-state solution on the time step is a consequence

DOI: 10.1002/fld

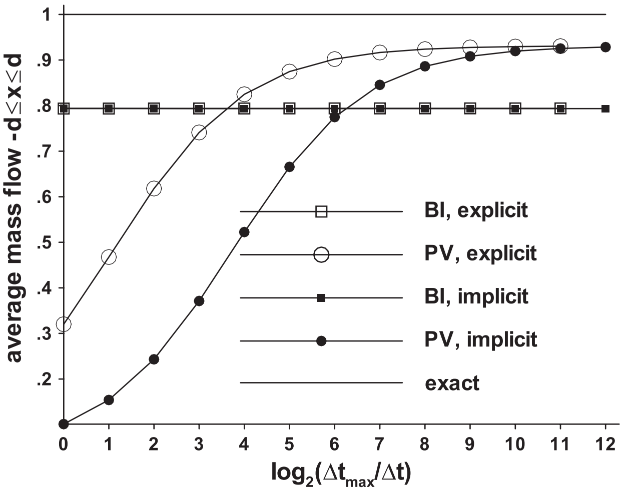

Figure 6. Spatial ($-d\leqslant x\leqslant d$) average of mass flow through a tube as a function of Δt, two-dimensional obstacle, 41×21 grid, $Re=20$.

of the inability of the time step to drop out at steady state in Equations (18) and (21). Note that the solution using the BI method is not affected by the time step.

6. SUGGESTED IMPROVED APPROACHES

6.1. Reducing Δt only at immersed boundary points, PVR method

The numerical example in the preceding section shows that for the PV method, the boundary conditions at the immersed boundary are better satisfied as the time step goes to zero. However, two to three orders of magnitude reduction in the time step (and hence two to three orders of magnitude increase in computational expense) is still required for acceptable results. What is desired is a method that exhibits the same correct behavior as reducing the time step, but does not incur the severe computational expense of reducing the time step. Examination of Equation (18) shows that $\mathbf{u}^n$ does not appear but instead is approximated by $L_{\mathrm{NB}}(\mathbf{u}^n_{\mathrm{NB}})+L_{\mathrm{IB}}(\mathbf{u}^{n+1}_{\mathrm{IB}})$. Therefore, the time accuracy of Equation (18) is suspect and a modification of the terms arising from the time derivative may be appropriate. A simple modification of Equation (18) or (21), which are the replacements for the momentum equations at immersed boundary points, can be made by reducing the time step in these equations (and in the equations that define the pseudo-velocity i.e. Equation (12) or (13)) by a factor (φ), without changing the time step in the momentum equations at non-immersed boundary points. This modification will result in Equation (22) (explicit time integration) and Equation (23) (implicit time integration). In the present work, this method of reducing the time step at immersed boundary points is referred to by the acronym 'PVR' for 'Pseudo-Velocity Reduced time step'.

$$\frac{\mathbf{u}^{n+1}-(L_{\mathrm{NB}}(\mathbf{u}^n_{\mathrm{NB}})+L_{\mathrm{IB}}(\mathbf{u}^{n+1}_{\mathrm{IB}}))}{\Delta t/\varphi}=L_{\mathrm{NB}}(\boldsymbol{\Upsilon}^n_{\mathrm{NB}})-\nabla p^{n+1} \tag{22}$$

DOI: 10.1002/fld

$$\frac{\mathbf{u}^{n+1}-(L_{\mathrm{NB}}(\widetilde{\mathbf{u}}_{\mathrm{NB}})+L_{\mathrm{IB}}(\mathbf{u}_{\mathrm{IB}}^{n+1}))}{\Delta t/\varphi}=-\nabla p^{n+1}+\beta\left(\frac{1}{Re}\nabla^2\mathbf{u}^{n+1}-(\mathbf{u}\cdot\nabla\mathbf{u})^{n+1}\right)-\beta\left(\frac{1}{Re}\nabla^2\widetilde{\mathbf{u}}-(\widetilde{\mathbf{u}}\cdot\nabla\widetilde{\mathbf{u}})\right) \tag{23}$$

Note that the steady-state solution is not affected by changing the time step only in Equation (18) or (21) *vs* changing the time step globally, since at steady state the left-hand side of Equation (3) or (4) is zero. For unsteady problems, however, there will be a difference in the solution since the immersed boundary points are advanced using smaller time steps than the non-immersed boundary points. It may seem that time accuracy has been adversely affected by the use of two different time steps. However, the proper viewpoint is that the equations at immersed boundary points are intended to apply a boundary condition whose behavior in time is known *a priori* and not to solve a time accurate evolution equation. Consider two solutions; one of which is obtained by reducing the global time step by a factor φ, the other by reducing by φ the time step only in Equation (18), which represents the momentum equation at immersed boundary points. After evolving for the same temporal period, the solutions will differ, but they will do so only as a result of the difference in the error involved in the temporal integration of Equation (3) or (4) using Δt *vs* $\Delta t/\varphi$.

6.1.1. Force density. For the PV method, the force density added to the momentum equations can be defined by either Equation (16) (explicit time integration) or Equation (20) (implicit time integration). This is not the case for the PVR, however, the added force density can be determined from the concept of the immersed boundary method. This force density $\mathbf{F}^{n+1}$ can be found by writing Equation (22) (explicit time integration) or Equation (23) (implicit time integration) with the left hand the same as Equation (3) (explicit time integration) or Equation (4) (implicit time integration), subtracting the equations and solving for $\mathbf{F}^{n+1}$. For the PVR method, this results in Equation (24) (explicit time integration) and Equation (25) (implicit time integration), which is the force density that must be added to Equation (3) or (4) to change them to Equation (22) or (23).

$$\mathbf{F}^{n+1}=-\mathbf{\Upsilon}^n+L_{\mathrm{NB}}(\mathbf{\Upsilon}_{\mathrm{NB}}^n)+\frac{L_{\mathrm{NB}}(\mathbf{u}_{\mathrm{NB}}^n)+L_{\mathrm{IB}}(\mathbf{u}_{\mathrm{IB}}^{n+1})-\mathbf{u}^{n+1}}{\Delta t/\varphi}+\frac{\mathbf{u}^{n+1}-\mathbf{u}^n}{\Delta t} \tag{24}$$

$$\mathbf{F}^{n+1}=-\left(\beta\left(\frac{\nabla^2\widetilde{\mathbf{u}}}{Re}-(\widetilde{\mathbf{u}}\cdot\nabla\widetilde{\mathbf{u}})\right)+\mathbf{\Phi}^n\right)+\frac{L_{\mathrm{NB}}(\widetilde{\mathbf{u}}_{\mathrm{NB}})+L_{\mathrm{IB}}(\mathbf{u}_{\mathrm{IB}}^{n+1})-\mathbf{u}^{n+1}}{\Delta t/\varphi}+\frac{\mathbf{u}^{n+1}-\mathbf{u}^n}{\Delta t} \tag{25}$$

6.2. Constrained velocity method, CV method

The principle of the immersed boundary method is to add a force density to the momentum equations to cause the velocity to reach specified values at the immersed boundary. Application of this principle using the method of [5] results in Equation (11), which has the defect of causing the discretized system of equations to become inconsistent. If the force density is defined using the pseudo-velocity a consistent set of discretized equations results, but this approach is not correct as it results in a force density that does not depend on the velocity at the discrete grid points, but instead depends on the pseudo-velocity at the discrete grid points (the force density does, however, depend on the velocity on the immersed boundary). What is desired is an algorithm that combines the positive and drops the negative attributes of the above approaches. The ideal solution

DOI: 10.1002/fld

is to be as close as possible to the modified momentum equation obtained by defining the force density using the velocity (Equation (11)), while retaining the divergence-free constraint. This can be expressed by adding a small correction term $\Delta\mathbf{u}$ to Equation (11) resulting in Equation (26) and then minimizing $\Delta\mathbf{u}$ in some norm, subject to the divergence-free constraint.

$$\mathbf{u}^{n+1} = L_{\mathrm{NB}}(\mathbf{u}^{n+1}_{\mathrm{NB}}) + L_{\mathrm{IB}}(\mathbf{u}^{n+1}_{\mathrm{IB}}) + \Delta\mathbf{u} \tag{26}$$

As $\Delta\mathbf{u}$ depends (because of the term $L_{\mathrm{NB}}(\mathbf{u}^{n+1}_{\mathrm{NB}})$) on the velocity at non-immersed boundary points, the minimization of this norm is coupled to the solution of the entire velocity field.

6.2.1. Relationship between pressure and Lagrange multipliers. Consider the pseudo-velocity $\widetilde{\mathbf{u}}$ defined by Equation (12) or (13) which is not divergence free, having been obtained by advancing the momentum equations in time without invoking the divergence-free constraint and without using the pressure at the current time step. To transform $\widetilde{\mathbf{u}}$ into a divergence-free velocity $\mathbf{u}^{n+1}$, a quantity $\Delta\mathbf{u}$ is added.

$$\mathbf{u}^{n+1} = \widetilde{\mathbf{u}} + \Delta\mathbf{u} \tag{27}$$

A problem can now be defined where it is desired that $\mathbf{u}^{n+1}$ be as close as possible to $\widetilde{\mathbf{u}}$ subject to the constraint that $\mathbf{u}^{n+1}$ be divergence free. This can be accomplished by minimizing a norm of $\Delta\mathbf{u}$ subject to the constraint that $\widetilde{\mathbf{u}} + \Delta\mathbf{u}$ is divergence free. Choosing the norm (i.e. the objective function) as the sum of the squares of $\Delta\mathbf{u}$, by using Lagrange multipliers (the λ's) this constrained problem can be converted to an unconstrained problem with a modified objective function defined by the following equation:

$$\begin{aligned} G = {} & \sum^{N_u} \Delta u_{i,j}^2 + \sum^{N_v} \Delta v_{i,j}^2 \\ & + \sum^{N_p} \lambda_{i,j} \left(\frac{\overbrace{\widetilde{u}_{i,j} + \Delta u_{i,j}}^{u^{n+1}_{i,j}} - \left(\overbrace{\widetilde{u}_{i-1,j} + \Delta u_{i-1,j}}^{u^{n+1}_{i-1,j}} \right)}{\Delta x^u_i} + \frac{\overbrace{\widetilde{v}_{i,j} + \Delta v_{i,j}}^{v^{n+1}_{i,j}} - \left(\overbrace{\widetilde{v}_{i,j-1} + \Delta v_{i,j-1}}^{v^{n+1}_{i,j-1}} \right)}{\Delta y^v_j} \right) \end{aligned} \tag{28}$$

Δx^u_i, Δy^v_j are defined in Figure 1, and N_u, N_v, N_p are the number of interior u, v, p points, respectively.

Differentiating Equation (28) with respect to the Δu's, Δv's and λ's and setting the partial derivatives to zero results in Equation (29). Note that the two-point stencil for the divergence is the reason why each equation obtained by differentiating the modified objective function with respect to the Δu's and Δv's contains two and only two λ's. Another way of viewing this is to note that each $\Delta u_{i,j}$ or $\Delta v_{i,j}$ appears in two discrete divergence (i.e. constraint) equations, therefore, differentiating G with respect to $\Delta u_{i,j}$ will result in an equation containing $\lambda_{i,j}$ and

$\lambda_{i+1,j}$ (with respect to $\Delta v_{i,j}$ will result in an equation containing $\lambda_{i,j}$ and $\lambda_{i,j+1}$). Also, note that differentiating the modified objective function with respect to $\lambda_{i,j}$ returns the (i, j)th discrete divergence-free constraint.

$$\begin{aligned}\frac{\partial G}{\partial(\Delta u_{i,j})} &= 2\Delta u_{i,j} - \frac{\lambda_{i+1,j}}{\Delta x^u_{i+1}} + \frac{\lambda_{i,j}}{\Delta x^u_i} = 0\\ \frac{\partial G}{\partial(\Delta v_{i,j})} &= 2\Delta v_{i,j} - \frac{\lambda_{i,j+1}}{\Delta y^v_{j+1}} + \frac{\lambda_{i,j}}{\Delta y^v_j} = 0\\ \frac{\partial G}{\partial(\lambda_{i,j})} &= \frac{\Delta u_{i,j} - \Delta u_{i-1,j}}{\Delta x^u_i} + \frac{\Delta v_{i,j} - \Delta v_{i,j-1}}{\Delta y^v_j} + \frac{\widetilde{u}_{i,j} - \widetilde{u}_{i-1,j}}{\Delta x^u_i} + \frac{\widetilde{v}_{i,j} - \widetilde{v}_{i,j-1}}{\Delta y^v_j} = 0\end{aligned} \tag{29}$$

Obtaining the solution of Equation (29) for $\Delta u_{i,j}$ results in:

$$\Delta u_{i,j} = \frac{\lambda_{i+1,j}}{2\Delta x^u_{i+1}} - \frac{\lambda_{i,j}}{2\Delta x^u_i}$$

Adding to the x component of the momentum equation results in:

$$u^{n+1}_{i,j} = \widetilde{u}_{i,j} + \overbrace{\frac{\lambda_{i+1,j}}{2\Delta x^u_{i+1}} - \frac{\lambda_{i,j}}{2\Delta x^u_i}}^{\Delta u_{i,j}}$$

If pressure is defined as $p^{n+1} = -\lambda/2\Delta t$, and $\Delta x^u_i = \Delta x^u_{i+1} = \Delta x_i$ (which will occur if the grid spacing is constant), then the result is Equation (30), which, if the pressure gradient is discretized using a two-point central difference stencil, is one component of the vector equation (Equation (14)).

$$u^{n+1}_{i,j} = \widetilde{u}_{i,j} - \overbrace{\Delta t\left(\frac{p^{n+1}_{i+1,j}}{\Delta x_i} - \frac{p^{n+1}_{i,j}}{\Delta x_i}\right)}^{\Delta u_{i,j}} \tag{30}$$

This shows that on a grid with constant spacing, there is a linear relationship between the discrete pressure and the Lagrange multipliers.

6.2.2. Constrained velocity and immersed boundaries. The relationship between pressure and the Lagrange multipliers suggests that the minimization of $\Delta\mathbf{u}$ in Equation (26) at immersed boundary points can be combined with that of $\Delta\mathbf{u}$ in Equation (27) at non-immersed boundary points. This may be done by defining the velocity at each non-immersed boundary point by Equation (27) and at each immersed boundary point by Equation (26), and minimizing the sum of the squares of $\Delta\mathbf{u}$, subject to the constraint that $\nabla\cdot\mathbf{u}^{n+1} = 0$. As before, Lagrange multipliers can be used to turn the problem into an unconstrained optimization problem. However, the modified objective function (Equation (31)) has a different structure than the modified objective function without immersed boundaries (Equation (28)), as a component of $\Delta\mathbf{u}$ may (because of L_{NB}) appear in more than two constraint equations. To make the problem more tractable, the operator $L_{\mathrm{NB}}() + L_{\mathrm{IB}}(u^{n+1}_{\mathrm{IB}})$ is assumed to contain only non-immersed boundary points; therefore, points with the subscript

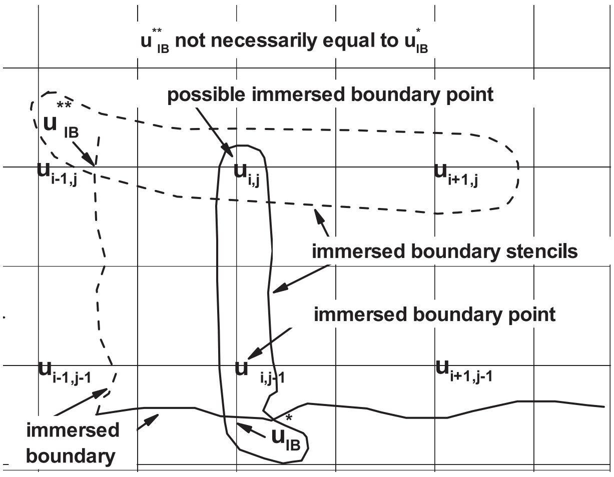
u**IB not necessarily equal to u*IB
possible immersed boundary point
u**IB
ui-1,j
ui,j
ui+1,j
immersed boundary stencils
immersed boundary point
ui-1,j-1
u i,j-1
ui+1,j-1
immersed boundary
u*IB

Figure 7. An example of a point that may be in another immersed boundary point's stencil.

NB are non-immersed boundary points. In Figure 7, for example, $u_{i,j}$, which is in the stencil for $u_{i,j-1}$, cannot itself be an immersed boundary point, as may occur if the immersed boundary includes the dotted line:

$$G=\sum^{N_u}\Delta u_{i,j}^2+\sum^{N_v}\Delta v_{i,j}^2 + \sum^{N_p}\lambda_{i,j}\left(\frac{\overbrace{\widetilde{u}_{i,j}+\Delta u_{i,j}}^{u^{n+1}_{i,j}}-\left(\overbrace{\widetilde{u}_{i-1,j}+\Delta u_{i-1,j}}^{u^{n+1}_{i-1,j}}\right)}{\Delta x_i^u}+\frac{\overbrace{\widetilde{v}_{i,j}+\Delta v_{i,j}}^{v^{n+1}_{i,j}}-\left(\overbrace{\widetilde{v}_{i,j-1}+\Delta v_{i,j-1}}^{v^{n+1}_{i,j-1}}\right)}{\Delta y_j^v}\right) \tag{31}$$

where if $u_{i,j}$ $(v_{i,j})$ is an immersed boundary point $\widetilde{u}_{i,j}$ $(\widetilde{v}_{i,j})$ is replaced with

$$L_{\text{NB}}\left(\overbrace{\widetilde{u}_{\text{NB}}+\Delta u_{\text{NB}}}^{u^{n+1}_{\text{NB}}}\right)_{i,j}+L_{\text{IB}}(u^{n+1}_{\text{IB}})_{i,j}\quad\left(L_{\text{NB}}\left(\overbrace{\widetilde{v}_{\text{NB}}+\Delta v_{\text{NB}}}^{v^{n+1}_{\text{NB}}}\right)_{i,j}+L_{\text{IB}}(v^{n+1}_{\text{IB}})_{i,j}\right)$$

using Equation (26) and $\mathbf{u}^{n+1}_{\text{NB}}=\widetilde{\mathbf{u}_{\text{NB}}}+\Delta\mathbf{u}_{\text{NB}}$.

Δx_i^u, Δy_j^v are defined in Figure 1, N_u, N_v, N_p are the number of interior u, v, p points, respectively. These interior points are points inside the computational domain, i.e. points not lying on wall boundaries.

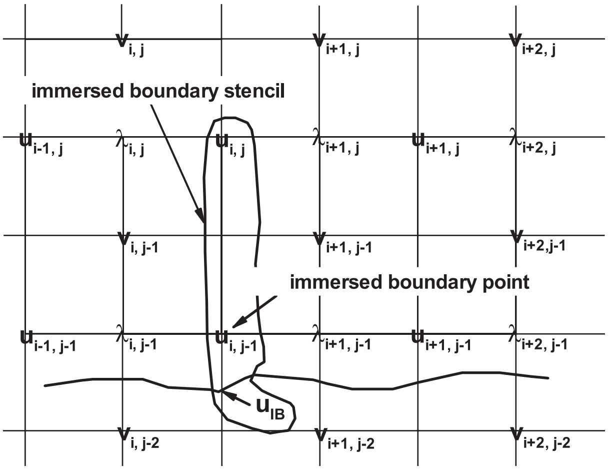

Figure 8. The point that is in an immersed boundary stencil but is not an immersed boundary point.

The equations that minimize the modified objective function (Equation (31)) are given by the following equation:

$$
\begin{aligned}
\frac{\partial G}{\partial(\Delta u_{i,j})} &= 2\Delta u_{i,j} - \frac{\lambda_{i+1,j}}{\Delta x^u_{i+1}} + \frac{\lambda_{i,j}}{\Delta x^u_i} + H_{i,j} = 0\\
\frac{\partial G}{\partial(\Delta v_{i,j})} &= 2\Delta v_{i,j} - \frac{\lambda_{i,j+1}}{\Delta y^v_{j+1}} + \frac{\lambda_{i,j}}{\Delta y^v_j} + I_{i,j} = 0\\
\frac{\partial G}{\partial(\lambda_{i,j})} &= \frac{\Delta u_{i,j} - \Delta u_{i-1,j}}{\Delta x^u_i} + \frac{\Delta v_{i,j} - \Delta v_{i,j-1}}{\Delta y^v_j} + \frac{\widetilde{u}_{i,j} - \widetilde{u}_{i-1,j}}{\Delta x^u_i} + \frac{\widetilde{v}_{i,j} - \widetilde{v}_{i,j-1}}{\Delta y^v_j} = 0
\end{aligned}
\tag{32}
$$

where if $u_{i,j}$ $(v_{i,j})$ is an immersed boundary point $\widetilde{u}_{i,j}$ $(\widetilde{v}_{i,j})$ is replaced with

$$
L_{\mathrm{NB}}\left(\overbrace{\widetilde{u}_{\mathrm{NB}} + \Delta u_{\mathrm{NB}}}^{u^{n+1}_{\mathrm{NB}}}\right)_{i,j} + L_{\mathrm{IB}}(u^{n+1}_{\mathrm{IB}})_{i,j} \left(L_{\mathrm{NB}}\left(\overbrace{\widetilde{v}_{\mathrm{NB}} + \Delta v_{\mathrm{NB}}}^{v^{n+1}_{\mathrm{NB}}}\right)_{i,j} + L_{\mathrm{IB}}(v^{n+1}_{\mathrm{IB}})_{i,j} \right)
$$

using Equation (26) and $\mathbf{u}^{n+1}_{\mathrm{NB}} = \widetilde{\mathbf{u}_{\mathrm{NB}}} + \Delta\mathbf{u}_{\mathrm{NB}}$.

For non-immersed boundary points appearing in an immersed boundary interpolation operator, the partial derivative of the modified objective function with respect to the $\Delta u_{i,j}$ or $\Delta v_{i,j}$ at those points will contain additional terms $(H_{i,j}, I_{i,j})$ which cause the partial derivative to contain more than the two ‘basic’ Lagrange multipliers that appear if the problem contains no immersed boundary. For example, in Figure 8, if $\Delta u_{i,j}$ (at a non-immersed boundary point) appears in a linear interpolation stencil, then the part of the modified objective function G that contains $\Delta u_{i,j}$

will be the following equation (where a and b are the weights of the interpolation operator):

$$\begin{aligned}
\mathscr{G} &= \Delta u_{i,j}^2 + \lambda_{i,j}\left(\frac{\overbrace{\widetilde{u}_{i,j}+\Delta u_{i,j}}^{u_{i,j}^{n+1}} - \overbrace{(\widetilde{u}_{i-1,j}+\Delta u_{i-1,j})}^{u_{i-1,j}^{n+1}}}{\Delta x_i^u}\right)\\
&\quad + \lambda_{i+1,j}\left(\frac{\overbrace{\widetilde{u}_{i+1,j}+\Delta u_{i+1,j}}^{u_{i+1,j}^{n+1}} - \overbrace{(\widetilde{u}_{i,j}+\Delta u_{i,j})}^{u_{i,j}^{n+1}}}{\Delta x_{i+1}^u}\right) + \mathscr{H}_{i,j}\\
\mathscr{H}_{i,j} &= \lambda_{i,j-1}\left(\frac{\overbrace{a(\widetilde{u}_{i,j}+\Delta u_{i,j}) + bu_{\mathrm{IB}}^{n+1}}^{u_{i,j-1}^{n+1}} - \overbrace{(\widetilde{u}_{i-1,j-1}+\Delta u_{i-1,j-1})}^{u_{i-1,j-1}^{n+1}}}{\Delta x_i^u}\right)\\
&\quad + \lambda_{i+1,j-1}\left(\frac{\overbrace{\widetilde{u}_{i+1,j-1}+\Delta u_{i+1,j-1}}^{u_{i+1,j-1}^{n+1}} - \overbrace{(a(\widetilde{u}_{i,j}+\Delta u_{i,j}) + bu_{\mathrm{IB}}^{n+1})}^{u_{i,j-1}^{n+1}}}{\Delta x_{i+1}^u}\right)
\end{aligned} \tag{33}$$

where for clarity $\mathscr{G}$ is just the parts of Equation (31), which contain $\Delta u_{i,j}$.

Since $u_{i,j-1}^{n+1}$ is an immersed boundary point, it is determined from the immersed boundary stencil $u_{i,j-1}^{n+1} = au_{i,j}^{n+1} + bu_{\mathrm{IB}} + \Delta u_{i,j-1}$; as $u_{i,j}^{n+1} = \widetilde{u}_{i,j} + \Delta u_{i,j}$, the immersed boundary stencil becomes $u_{i,j-1}^{n+1} = a(\widetilde{u}_{i,j} + \Delta u_{i,j}) + bu_{\mathrm{IB}} + \Delta u_{i,j-1}$. Therefore, the two discrete divergence (i.e. constraint) equations containing $u_{i,j-1}^{n+1}$ will also contain $\Delta u_{i,j}$, and when Equation (33) is differentiated with respect to $\Delta u_{i,j}$, the Lagrange multipliers $\lambda_{i,j-1}$ and $\lambda_{i+1,j-1}$ will appear:

$$\frac{\partial G}{\partial(\Delta u_{i,j})} = 2\Delta u_{i,j} - \frac{\lambda_{i+1,j}}{\Delta x_{i+1}^u} + \frac{\lambda_{i,j}}{\Delta x_i^u} + \overbrace{\left(-a\frac{\lambda_{i+1,j-1}}{\Delta x_{i+1}^u} + a\frac{\lambda_{i,j-1}}{\Delta x_i^u}\right)}^{H_{i,j} = \dfrac{\partial \mathscr{H}_{i,j}}{\partial(\Delta u_{i,j})}} \tag{34}$$

These Lagrange multipliers $\lambda_{i,j-1}$ and $\lambda_{i+1,j-1}$ are in addition to the 'basic' Lagrange multipliers $\lambda_{i+1,j}$, $\lambda_{i,j}$. Clearly, depending on the geometry and the immersed boundary stencil many other combinations are possible, i.e. a $\Delta u_{i,j}$ or $\Delta v_{i,j}$ may appear in many discrete divergence (i.e. constraint) equations. In general, when Equation (31) is differentiated with respect to a specific $\Delta u_{i,j}$ or $\Delta v_{i,j}$, a distinct Lagrange multiplier will appear for each associated discrete divergence equation that contains that specific $\Delta u_{i,j}$ or $\Delta v_{i,j}$. Because of the assumption that immersed boundary points do not appear in stencils for other immersed boundary points, at immersed boundary points $(H_{i,j}, I_{i,j})$ in Equation (32) are zero.

The importance of the immersed boundary conditions can be increased by minimizing $\Delta\mathbf{u}$ at immersed boundary points relative to $\Delta\mathbf{u}$ at non-immersed boundary points. This can be done by weighting (by a factor $\varphi > 1$) $\Delta\mathbf{u}$ at immersed boundary points in the objective function; this changes Equation (32) into Equation (35) (the constraint equation is unchanged). The norm (i.e. objective function) that is now minimized (subject to the divergence-free constraint) is the sum of the squares of $\Delta\mathbf{u}$ where the square of $\Delta\mathbf{u}$ is weighted by a factor $\varphi > 1$, if $\Delta\mathbf{u}$ is at an immersed boundary point. If the weighting is not done, then the solution of Equation (32) suffers the same problem of not satisfying the boundary conditions at the immersed boundary as do solutions to Equations (2) and (3) (non-immersed boundary points) and Equation (18) (immersed boundary points) for explicit time integration or Equations (2) and (4) (non-immersed boundary points) and Equation (21) (immersed boundary points) for implicit time integration.

$$\begin{aligned}\frac{\partial G}{\partial(\Delta u_{i,j})} &= 2\varphi\Delta u_{i,j} - \frac{\lambda_{i+1,j}}{\Delta x^u_{i+1}} + \frac{\lambda_{i,j}}{\Delta x^u_i} + H_{i,j} = 0\\ \frac{\partial G}{\partial(\Delta v_{i,j})} &= 2\varphi\Delta v_{i,j} - \frac{\lambda_{i,j+1}}{\Delta y^v_{j+1}} + \frac{\lambda_{i,j}}{\Delta y^v_j} + I_{i,j} = 0\end{aligned} \tag{35}$$

where $\varphi \geqslant 1$ if $u_{i,j}$ or $v_{i,j}$ is an immersed boundary point, $\varphi = 1$ otherwise where if $u_{i,j}(v_{i,j})$ is an immersed boundary point, $H_{i,j}(I_{i,j}) = 0$.

$$\frac{\partial G}{\partial(\lambda_{i,j})} = \frac{\Delta u_{i,j} - \Delta u_{i-1,j}}{\Delta x^u_i} + \frac{\Delta v_{i,j} - \Delta v_{i,j-1}}{\Delta y^v_j} + \frac{\widetilde{u}_{i,j} - \widetilde{u}_{i-1,j}}{\Delta x^u_i} + \frac{\widetilde{v}_{i,j} - \widetilde{v}_{i,j-1}}{\Delta y^v_j} = 0$$

where if $u_{i,j}$ $(v_{i,j})$ is an immersed boundary point $\widetilde{u}_{i,j}$ $(\widetilde{v}_{i,j})$ is replaced with

$$L_{\mathrm{NB}}\left(\overbrace{\widetilde{u}_{\mathrm{NB}} + \Delta u_{\mathrm{NB}}}^{u^{n+1}_{\mathrm{NB}}}\right)_{i,j} + L_{\mathrm{IB}}(u^{n+1}_{\mathrm{IB}})_{i,j} \quad \left(L_{\mathrm{NB}}\left(\overbrace{\widetilde{v}_{\mathrm{NB}} + \Delta v_{\mathrm{NB}}}^{v^{n+1}_{\mathrm{NB}}}\right)_{i,j} + L_{\mathrm{IB}}(v^{n+1}_{\mathrm{IB}})_{i,j}\right)$$

using Equation (26) and $\mathbf{u}^{n+1}_{\mathrm{NB}} = \widetilde{\mathbf{u}_{\mathrm{NB}}} + \Delta\mathbf{u}_{\mathrm{NB}}$.

The authors have found that the solution of Equation (35) with $\varphi > 1$ improves the satisfaction of the boundary conditions at the immersed boundary. However, there has been found to be a rather low limit ~ 10 on the magnitude of φ, above which the solution does not converge. The reason for this can be seen by considering Equations (37) and (36) which are obtained by substituting for $\Delta u_{i,j}$ (as found from Equation (35)) in Equation (27) or (26) for non-immersed boundary and immersed boundary points, respectively:

$$2(u^{n+1}_{i,j} - \widetilde{u}_{i,j}) = \frac{\lambda_{i+1,j}}{\Delta x^u_{i+1}} - \frac{\lambda_{i,j}}{\Delta x^u_i} + H_{i,j} \tag{36}$$

$$2\varphi(u^{n+1}_{i,j} - (L_{\mathrm{NB}}(u^{n+1}_{\mathrm{NB}})_{i,j} + L_{\mathrm{IB}}(u^{n+1}_{\mathrm{IB}})_{i,j})) = \frac{\lambda_{i+1,j}}{\Delta x^u_{i+1}} - \frac{\lambda_{i,j}}{\Delta x^u_i} \tag{37}$$

Assume that as $\varphi \to \infty$, the velocity converges and becomes independent of φ. Therefore, as $u^{n+1}_{i,j} - (L_{\mathrm{NB}}(u^{n+1}_{\mathrm{NB}})_{i,j} + L_{\mathrm{IB}}(u^{n+1}_{\mathrm{IB}})_{i,j}) = \text{constant}$ (since the velocity converges), we conclude that

DOI: 10.1002/fld

the magnitude of the right-hand side must increase as φ; therefore, either $\lambda_{i+1,j}$ or $\lambda_{i,j}$ must increase as φ. As a result, if solving Equation (36) at a non-immersed boundary point near an immersed boundary, Equation (36) may, due to $H_{i,j}$, contain a λ which increases as φ. Therefore, as the left-hand side of Equation (36) does not contain a φ the velocity must increase as the λ appearing in $H_{i,j}$ increases. This problem can be avoided by dropping the additional terms $(H_{i,j},I_{i,j})$ in Equation (35).

As in the previous section, it is desired to use the transformation $p^{n+1}=-\lambda/2\Delta t$ to recover Equations (5) and (6), however, if the grid is not even, then $1/\Delta x^u_{i+1} - 1/\Delta x^u_i \neq 0$ and the resulting gradient approximation of p^{n+1} is not 0th order accurate. Therefore, the substitution $p^{n+1}=-\lambda/2\Delta t$ cannot be made. In order to recover Equations (5) and (6), the weights for λ are replaced with the ones for pressure in Equation (6), after the substitution $p^{n+1}=-\lambda/2\Delta t$ the result is the following equation:

$$\frac{\mathbf{u}^{n+1}-(L_{\mathrm{NB}}(\mathbf{u}^{n+1}_{\mathrm{NB}})+L_{\mathrm{IB}}(\mathbf{u}^{n+1}_{\mathrm{IB}}))}{\Delta t/\varphi}=-\nabla p^{n+1} \tag{38}$$

The assumption that no immersed boundary points appear in the interpolation stencil for other immersed boundary points is removed; for the stencil and geometries used in the present work, this situation occurs very rarely if at all. These changes have modified the solution of the minimization problem, as has the dropping of the additional terms $(H_{i,j},I_{i,j})$ in Equation (35), but it will be shown by numerical examples that the result is a numerical method that is an improvement over the other methods evaluated. This method involves the solution of Equation (3) (explicit time integration) or Equation (4) (implicit time integration) at non-immersed boundary points, and the solution of Equation (38) at immersed boundary points, along with the solution of the discretized continuity equation (Equation (5)). In the present work, this method is referred to by the acronym 'CV' from 'constrained velocity'. Note that for the implicit scheme, the error involved in the splitting of the momentum equations (see Equation (19)) is avoided by solving Equation (4) instead of Equation (19) at non-immersed boundary points. Also, note that the time integration scheme plays no role in the derivation of Equation (38); this equation can be used with both the explicit and implicit time integration schemes. By comparing Equation (26) with Equation (38), the correction term $\Delta\mathbf{u}$ is shown to be given by $\Delta\mathbf{u}=-(\Delta t/\varphi)\nabla p^{n+1}$.

A comparison of Equations (38) and (22) shows that decreasing the time step at immersed boundary points is closely related to increasing the weighting factor of the square of $\Delta\mathbf{u}$ in the objective function if $\Delta\mathbf{u}$ is at an immersed boundary point (i.e. penalizing changes from the interpolation stencil, Equation (11)). Solving Equations (38) and (22) for $\mathbf{u}^{n+1}$ and subtracting results in:

$$\overbrace{\mathbf{u}^{n+1}}^{\text{from Equation (38)}} - \overbrace{\mathbf{u}^{n+1}}^{\text{from Equation (22)}} = L_{\mathrm{NB}}(\mathbf{u}^{n+1}_{\mathrm{NB}}) - \left(\frac{\Delta t(L_{\mathrm{NB}}(\boldsymbol{\Upsilon}^{n}_{\mathrm{NB}}))}{\varphi}+L_{\mathrm{NB}}(\mathbf{u}^{n}_{\mathrm{NB}})\right)$$

As $\varphi\to\infty$, this reduces to the following equation:

$$\overbrace{\mathbf{u}^{n+1}}^{\text{from Equation (38)}} - \overbrace{\mathbf{u}^{n+1}}^{\text{from Equation (22)}} = L_{\mathrm{NB}}(\mathbf{u}^{n+1}_{\mathrm{NB}}) - L_{\mathrm{NB}}(\mathbf{u}^{n}_{\mathrm{NB}}) \tag{39}$$

This shows that for large values of φ the two methods (Equations (38) and (22)) are expected to give the same results for steady-state problems. However, if the problem is unsteady, the two

DOI: 10.1002/fld

methods differ. As an example, consider a two-dimensional domain, with periodic boundaries in both directions. In it place an immersed boundary of an $x=\text{constant}$ line across the entire domain. The initial conditions are $u=2\pi f\sin(2\pi f t_0)$, $v=0$, $p=\text{constant}$, while the immersed boundary moves with a prescribed motion of $\cos(2\pi f t)$. The solution to this problem for all time is $u=2\pi f\sin(2\pi f t)$, $v=0$, $p=\text{constant}$. This exact solution will be recovered by Equations (38) and (11), but not by Equations (22) and (23) as the left-hand side of these equations will be non-zero, and therefore ∇p^{n+1} will be non-zero. Note that in both the PVR and CV methods, the introduction of the time step reduction or weighting factor φ will change the spectral radius of A in Equation (7). If an iterative method is used to solve Equation (7), this may have an adverse effect on the convergence rate. A preconditioning matrix is a possible remedy but has not been explored in the present work.

6.2.3. Force density. Similarly as for the PVR method, the force density in the CV method is given by Equation (40) (explicit time integration) and Equation (41) (implicit time integration):

$$\mathbf{F}^{n+1}=-\boldsymbol{\Upsilon}^n+\frac{(L_{\text{NB}}(\mathbf{u}_{\text{NB}}^{n+1})+L_{\text{IB}}(\mathbf{u}_{\text{IB}}^{n+1}))-\mathbf{u}^{n+1}}{\Delta t/\varphi}+\frac{\mathbf{u}^{n+1}-\mathbf{u}^n}{\Delta t} \tag{40}$$

$$\mathbf{F}^{n+1}=-\beta\left(\frac{\nabla^2\mathbf{u}^{n+1}}{Re}+(\mathbf{u}\cdot\nabla\mathbf{u})^{n+1}\right)-\boldsymbol{\Phi}^n+\frac{L_{\text{NB}}(\mathbf{u}_{\text{NB}}^{n+1})+L_{\text{IB}}(\mathbf{u}_{\text{IB}}^{n+1})-\mathbf{u}^{n+1}}{\Delta t/\varphi}+\frac{\mathbf{u}^{n+1}-\mathbf{u}^n}{\Delta t} \tag{41}$$

7. COMPUTATION AND BEHAVIOR OF FORCE DENSITY

The concept of the immersed boundary method is that the stress which the surface of an immersed boundary exerts on a fluid is transformed into a force density which is then added to the Navier–Stokes equations to represent the presence of the immersed boundary. The integral of this stress (denoted $\mathbf{f}^{n+1}$) over the surface (S) of the immersed boundary yields the force of the immersed boundary on the fluid and should equal the integral of the force density over the entire volume (V) of fluid:

$$\int_S \mathbf{f}^{n+1}\,\mathrm{d}S=\int_V \mathbf{F}^{n+1}\,\mathrm{d}V=\text{total force of immersed boundary on fluid} \tag{42}$$

If the volume in which the force density is added decreases, Equation (42) implies that the force density must increase at the same rate for the total force of the immersed boundary on the fluid to remain constant. With the criteria used to select immersed boundary points at which the force density is added, as the grid spacing Δx decreases, the volume in which the force density is added decreases as Δx^2 (two dimensions) and Δx^3 (three dimensions). Therefore, the force density increases as Δx^{-2} (two dimensions) and Δx^{-3} (three dimensions) as the grid spacing decreases and is not bounded. However, as the product of the force density and the discrete volume goes to a constant, this suggests that the local stress on the surface of the immersed boundary may be obtained from the force density.

7.1. Computation of force density

The modeling of objects in a fluid by the addition of a force density to the momentum equations to represent the force of the objects on the fluid can be useful if the force of the objects on the fluid or work done by the objects on the fluid is of interest. Once the force density is known, the force of the immersed boundary on the fluid can be computed by summing up the product of the force densities with the volumes of the respective cells of the immersed boundary points at which the force density is added (in Figure 1, the volume of the cell for the i, jth x momentum equation is $\Delta x_i \Delta y_j^v$). The method of computing the force density (Equations (24)–(41)) is termed the 1st method in the present work. Note that the force density can also be defined using Equation (8) (explicit time integration) or the equivalent for implicit time integration, however, the form given by Equations (24)–(41) is useful to show the explicit dependence of the force density on various quantities.

In the context of an iterative method, the above definitions of the force density are not necessarily correct. In any iterative method, the change in the velocity and pressure at each iteration is found from $B\Delta\mathbf{q}=-\mathbf{R}$, where B is the matrix defining the iterative method, $\Delta\mathbf{q}$ is the change in the velocity and pressure and $\mathbf{R}$ the residual. Matrix B and residual $\mathbf{R}$ take into account the changes in the momentum equations at immersed boundary points which define the PV, PVR and CV methods. To find the force density that is added, the change in velocity and pressure obtained from the iterative method with a matrix A and a residual $\mathbf{R}^*$ (both unmodified by the PV, PVR or CV methods) and with the added force density term, must be equated with the change in the velocity and pressure obtained from the iterative method using the PV, PVR or CV methods:

$$\Delta\mathbf{q}=-B^{-1}\mathbf{R}=-A^{-1}(\mathbf{R}^*-\mathbf{F}^{n+1})$$

Solving for the force density $\mathbf{F}^{n+1}$ yields the following equation and is termed the 2nd method in the present work:

$$\mathbf{F}^{n+1}=\mathbf{R}^*-AB^{-1}\mathbf{R} \tag{43}$$

Note that the force density in Equation (43) depends on the residual ($\mathbf{R}$) of the equations being solved; this dependence is a key to a correct computation of the force density. Figure 9 (from the vibrating structure problem discussed later) shows the volume integral of the x component of force density over all immersed boundary points using the 1st and 2nd methods for computing the force density at different residual levels. The 2nd method is seen to be well behaved even with very high levels of residual, while the 1st method results in unbounded behavior as $\log_2(\varphi)$ increases. The 1st method behaves better as the residual ($\mathbf{R}$ in Equation (43)) decreases. As $\mathbf{R}$ goes to zero, the force density becomes equal to that defined by the immersed boundary concept (i.e. Equation (8) or equivalently Equations (24)–(41)) and the 1st and 2nd methods give identical force densities. The dependence of the 1st method on $\log_2(\varphi)$ results from the definition of the residual of the momentum equations. Using the CV method as an example, the residual ($\mathbf{R}$ in Equation (43)) used by the iterative method is defined by $R=\varphi\Delta t^{-1}(\mathbf{u}^{n+1}-(L_{\mathrm{NB}}(\mathbf{u}_{\mathrm{NB}}^{n+1})+L_{\mathrm{IB}}(\mathbf{u}_{\mathrm{IB}}^{n+1})))+\nabla p^{n+1}$ while the residual monitored (and used to distinguish the lines in Figure 9) is defined by $(\mathbf{u}^{n+1}-(L_{\mathrm{NB}}(\mathbf{u}_{\mathrm{NB}}^{n+1})+L_{\mathrm{IB}}(\mathbf{u}_{\mathrm{IB}}^{n+1})))+\varphi^{-1}\Delta t\nabla p^{n+1}$. This difference in the residual definition is because as $\varphi\rightarrow\infty$, $\mathbf{u}^{n+1}-(L_{\mathrm{NB}}(\mathbf{u}_{\mathrm{NB}}^{n+1})+L_{\mathrm{IB}}(\mathbf{u}_{\mathrm{IB}}^{n+1}))\rightarrow$ constant and $\nabla p^{n+1}\rightarrow\infty$ and the residual becomes large. Scaling the residual by $\varphi^{-1}\Delta t$ has been found to result in a good measure of convergence for the iterative method, the residual used in the iterative solver ($\mathbf{R}$ in Equation (43)) is of course unscaled.

Note that to compute the force density using Equation (43), B must not be rank deficient. However, as shown in Section 3.4, for the BI method the global matrix may be rank deficient, which leaves open the possibility of B being rank deficient. Matrix B will indeed be rank deficient

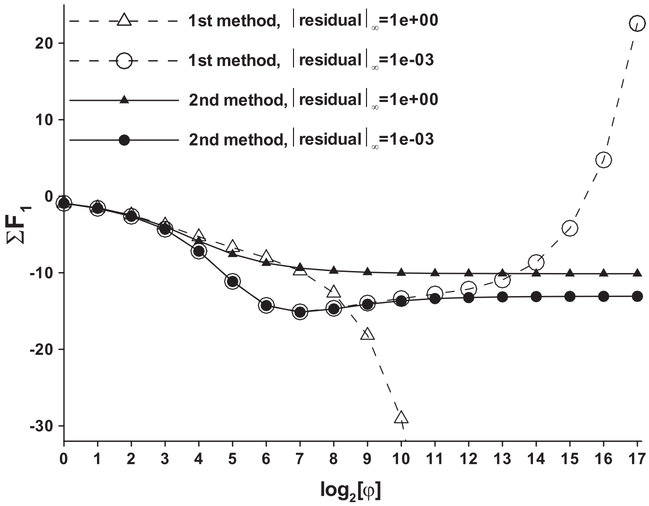

Figure 9. A comparison of the 1st and 2nd methods of computing force, implicit time integration, vibrating structure, 74 × 38 grid.

if all four velocities surrounding a main (pressure) grid point are immersed boundary points (i.e. the four velocities, $u_{i-1,j}$, $u_{i,j}$, $v_{i,j-1}$, $v_{i,j}$ surrounding $p_{i,j}$ in Figure 2). The rank deficiency arises because all the terms describing the influence of $p_{i,j}$ are zero.

7.2. Behavior of force density as $\Delta t \to 0$ or $\varphi \to \infty$

Equations (24)–(41) contain a term of the form $\varphi(L_{\mathrm{NB}}() + L_{\mathrm{IB}}(\mathbf{u}_{\mathrm{IB}}^{n+1}) - \mathbf{u}^{n+1}/\Delta t)$ and, therefore, the force density will not be bounded as $\Delta t \to 0$ or $\varphi \to \infty$. This behavior is shown in Figure 10 for both the explicit and implicit schemes. However, in the present work, it has been found that if the 2nd method is used to compute the force density, then the volume integral of the force density over the fluid (i.e. the total force of the immersed boundary on the fluid) is bounded and converges as $\Delta t \to 0$ or $\varphi \to \infty$ regardless of the residual remaining upon termination of the iterative process (if the 1st method is used, this will be true if the residual is reduced to machine zero). The convergence of this volume integral of the force density is to be expected for the existence of bounded solutions, however, there is no such restriction on the force density at each immersed boundary point. The specific unbounded behavior of the force density at each immersed boundary point is that it approaches a linear relation to $1/\Delta t$ as $\Delta t \to 0$ or to φ as $\varphi \to \infty$. The result of this unbounded behavior is that the force density cannot be used to compute the local stress on the immersed boundary and, therefore, deformations of the immersed boundary, but can be used to compute the entire force acting on the immersed boundary as a rigid body.

8. NUMERICAL EXAMPLES

8.1. Vibrating structure

A test problem consisting of a two-dimensional structure constructed using infinitely thin walls, one of which vibrates, is used to evaluate the behavior of the methods on unsteady

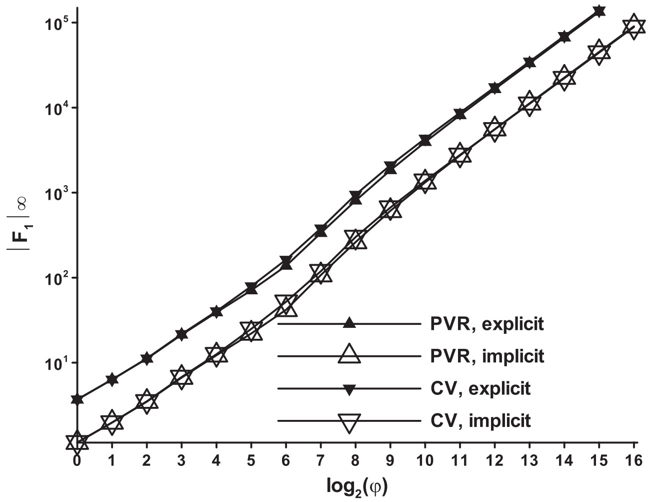

Figure 10. L_∞ norm of the x component of force density, two-dimensional obstacle, $Re=10^6$, 161×81 grid.

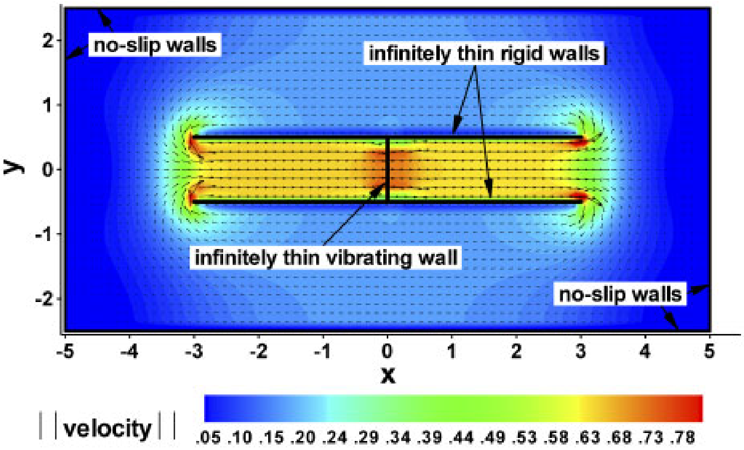

Figure 11. Vibrating structure schematic, contours are of velocity magnitude.

problems. The x coordinate of the vertical wall in Figure 11 moves with a motion of $x(\Delta t(n-1))=(\Delta t M/2\pi)\sin(2\pi\Delta t(n-1)/\Delta t M)$ (where n is the time index), the other two walls are rigid with no-slip velocity boundary conditions; the immersed boundary method is used to model all three walls. No-slip velocity boundary conditions are applied on all four sides of the computational grid. The parameter M is the number of time steps per cycle; as M increases the vibration frequency decreases. Beginning with a quiescent fluid, the simulation is run for $2M$ time steps. From a test of the effect of the residual level, it was determined that the L_∞ norm of the divergence has to be reduced to 1×10^{-3}; therefore, it was reduced to this level at each time step for all methods (except the BI method, for which this is not possible). To make a fair comparison,

DOI: 10.1002/fld

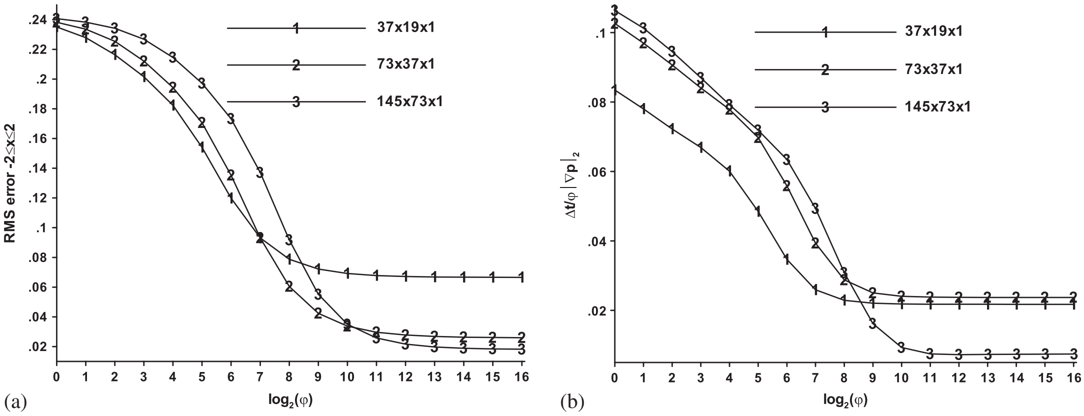

Figure 12. Effect of φ, vibrating structure, CV method, explicit time integration, $M=8$, even grid: (a) mass flow and (b) correction term.

however, the BI method uses the same number of iterations of the solver at each time step as required by the CV method to reduce the L_∞ norms of the residuals $\leqslant 1\times 10^{-3}$. The problem is non-dimensionalized with the maximum velocity of the vibrating vertical wall and the wall's height; the Reynolds number being 20 000 based on these quantities. Results are obtained on five grids with dimensions 37×19, 73×37, 145×73, 289×145, 577×289 (numbered on plots as grids $1,2,3,4,5$, respectively). The time step is $(0.05,0.3)/2^{(k-1)}$ (where k is the grid number) for the explicit time integration scheme and the implicit Adams–Moulton time integration scheme, respectively. Results are obtained for each set of grid dimensions on both even and stretched grids; on the stretched grid stretching is done to reduce the grid spacing near the immersed boundaries. The time step is reduced by half when moving from one grid to the next finest grid; this doubles the frequency while keeping the number of time steps per cycle constant. The exact mass flow between the two rigid walls for this problem is given by $\cos(2\pi\Delta t(n-1)/\Delta tM)$ and is used to define the error of the methods. The error is collapsed in space and time by determining the root mean square over $-2\leqslant x\leqslant 2$ and averaging over all time steps. The effect of φ, which is the weighting factor in the CV method and the local reduction in time step in the PVR method, on the solution is shown in Figure 12. When plotted against $\log_2(\varphi)$, the error in the mass flow has the same shape on the first three grids (due to computational expense, the 289×145 and 577×289 grids were not used in this part of the study). A solution independent of φ occurs by $\log_2(\varphi)\approx 13$ on all three grids for both the explicit and implicit time integration schemes. Figure 12(a) also shows that as φ increases the boundary conditions are better satisfied until they asymptote at $\log_2(\varphi)\approx 13$. If so, the L_2 (not necessarily the L_∞) norm of the correction term ($\Delta\mathbf{u}=-(\Delta t/\varphi)\nabla p$) at immersed boundary points should decay to a constant (at the same value of $\log_2(\varphi)$) as φ increases and indeed this behavior is shown in Figure 12(b). Results are only shown for the explicit CV method as the implicit and the PVR method (explicit and implicit) behave similarly.

The value (2^{13}) at which φ asymptotes (as in Figure 12) is then used in the PVR and CV methods to compare them with the BI method. Figure 13 shows that on an even grid the CV method has the lowest error for both the explicit and implicit schemes. The error of the CV method decays to zero as the grid spacing decreases while that of the other methods does not. The BI method has less error than the PV and PVR methods, particularly for the implicit scheme, although still

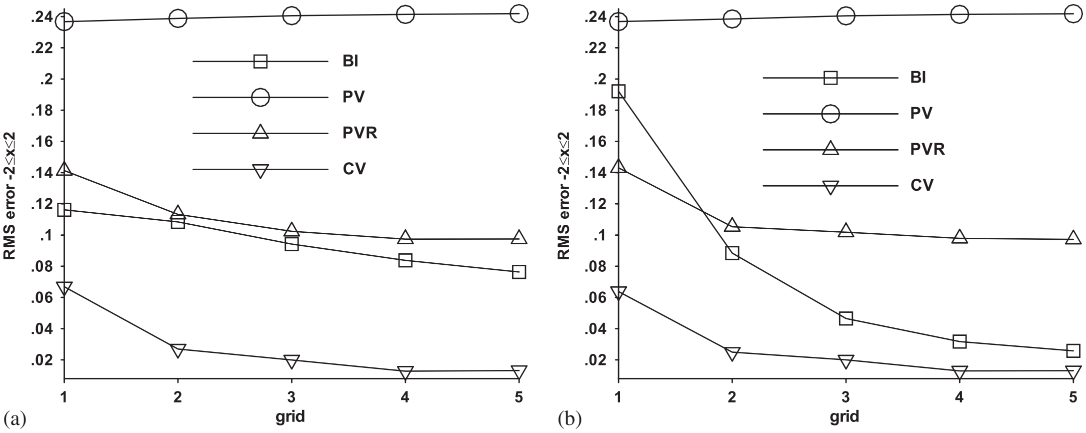

Figure 13. Spatial and temporal average of mass flow error, vibrating structure, $M=8$: (a) explicit convection–diffusion and (b) implicit convection–diffusion.

significantly more error than the CV method. The reason why the error in the PVR method does not decay to zero as the grid spacing is reduced can be seen in Figure 14(a) and (b), where a large lagging phase error in the mass flow is evident for the PVR method. The mass flow in Figure 14(a) and (b) is the mass flow in the x direction at $x=0$ and $-0.5\leqslant y\leqslant 0.5$. This phase error is a result of the force density (and, therefore, the boundary condition at the immersed boundary) not depending on the velocity on the Cartesian grid at the current $(n+1)$ time step. Figure 14(c) and (d) shows that the force density of the PVR method lags that of the CV method with the same phase error as the mass flow (the force density in Figure 14(c) and (d) is the x component of the force density representing the entire immersed boundary, not just the force density representing the vibrating part). The PV method is seen to give very poor results for the mass flow, as on top of the phase error, there is a large amplitude error. This large amplitude error is particularly evident in the force density. The force density of the BI method is not shown since, as mentioned in Section 7.1, matrix B in Equation (43) is rank deficient. While not shown, the further away from the origin ($x=0$, around which the center part vibrates) that the mass flow is computed for the PV method, the greater the amplitude error in the mass flow. As global mass is conserved with the PV method, the lost mass, represented by the amplitude error, leaks through the two rigid walls. Figure 14(a) indicates very little errors in the BI method which appears to conflict with the large error for the method shown in Figure 13. This is because the error in Figure 13 is averaged spatially over $-2\leqslant x\leqslant 2$ while Figure 14(a) is plotted at $x=0$; when the mass flow is plotted further away from $x=0$, a significant phase error appears in the results from the BI method. A stretched grid, which concentrated points near the origin, was also used to obtain solutions for all the methods. The results are almost identical to those obtained using the even grid, which validates the change of the weights for λ in Equation (35) to those used for the pressure gradient.

8.2. Two-dimensional obstacle

The two-dimensional obstacle problem of Section 5 is revisited using the two new methods. The effect of the weighting factor φ on the boundary conditions (specifically the mass flow) at the immersed boundary is dramatically seen in Figure 15(a) for the CV method for both the

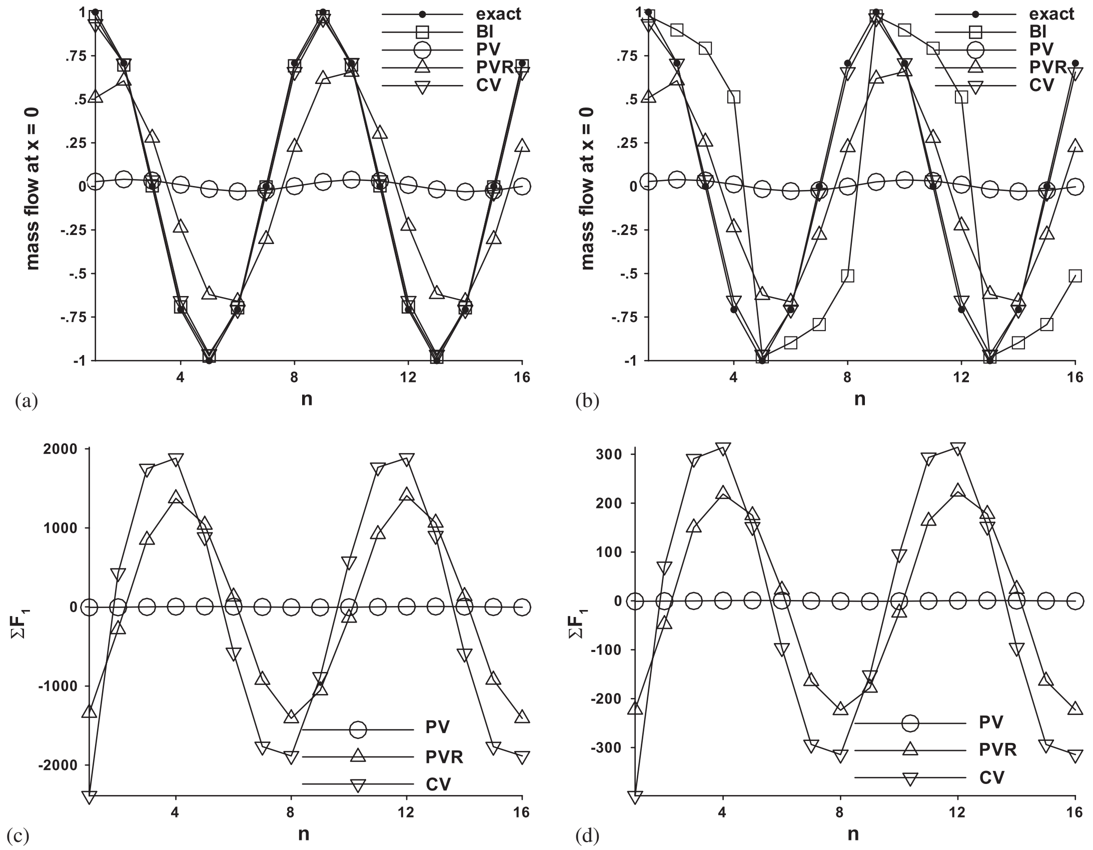

Figure 14. Mass flow at $x=0$ and total force density as a function of time, vibrating structure, $M=8$, even 577×289 grid: (a) mass flow explicit convection–diffusion; (b) mass flow implicit convection–diffusion; (c) force density explicit convection–diffusion; and (d) force density implicit convection–diffusion.

explicit and implicit time integration schemes. The explicit and implicit schemes behave the same for both the PVR and CV methods, however, the CV method asymptotes faster than the PVR method. This drastic improvement in the satisfaction of the boundary conditions at the immersed boundary, indicated by convergence toward the correct mass flow through the opening as φ is increased, suggests that the correction term ($\Delta\mathbf{u}=-(\Delta t/\varphi)\nabla p$) which is added to Equation (26) is minimized in some sense as φ is increased. This is seen in Figure 15(b) in which the L_2 norm of the correction term becomes smaller as φ is increased. The reason for the dip in the curves in Figure 15(b), in which the minimum of the correction term does not coincide with the asymptotic improvement in the boundary conditions at the immersed boundary in Figure 15(a), may be due to the simplifications (the dropping of the additional terms ($H_{i,j}, I_{i,j}$) in Equation (35), the change in the weights of the Lagrange multipliers, the assumption that no immersed boundary points appear in interpolation stencils for other immersed boundary points) made in the process of solving the minimization problem given by Equation (32).

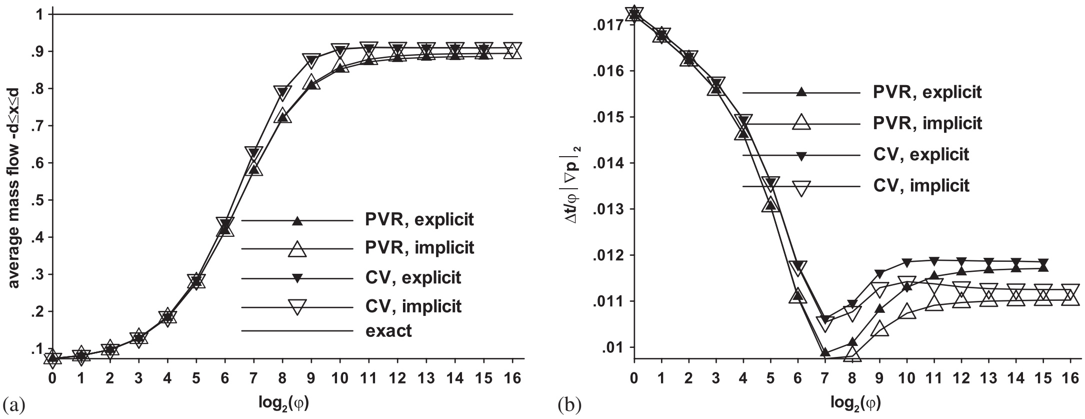

Figure 15. Effect of φ, two-dimensional obstacle, $Re = 10^6$, 161×81 grid: (a) mass flow and (b) correction term.

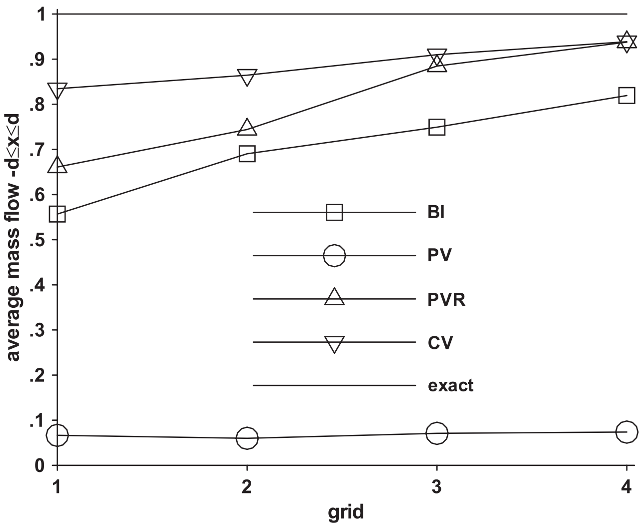

Figure 16. Two-dimensional obstacle, $Re = 10^6$, explicit convection–diffusion time integration.

Choosing a value for φ for the PVR and CV methods at which the solution asymptotes (see Figure 15), the methods are applied to the problem using the same four grids described in Section 5.1. Figure 16 shows that for the explicit time integration scheme (the implicit scheme behaves almost identically) the PVR and CV methods converge toward the correct mass flow of one, performing better on all grids than the BI method. The reason for the poorer results of the PVR method compared with the CV method on the coarser grids is the independence of the force density in the PVR method from the velocity at the current time step. On the coarsest grid, on which one time step is taken, the force density at the final time step in the PVR method depends on the velocity at the previous time step (i.e. the initial conditions). On the next finest grid, on which two time steps are taken (the same length of time is simulated on each grid), the force density at the final time step in the PVR method does not depend on the initial conditions, but instead depends on the velocity at the previous time step which is expected to be closer to the velocity at the final time step than the velocity at the previous time step is for the coarsest grid.

8.3. Cylinder in crossflow

A commonly used flow for validation purposes is flow over a cylinder. For this reason, the methods are applied to steady laminar flow over a circular cylinder at a Reynolds number of 40 (based on the diameter and crossflow velocity). The computational domain in the x direction extends $20d$ upstream and $80d$ downstream of the cylinder while the boundaries in the y direction extend $35d$ above and below the cylinder. At this Reynolds number, the flow is two-dimensional and, therefore, the z direction is not resolved. Dirichlet boundary conditions for the velocity are used at $x=-20d$ and at $y=\pm 35d$; a convective outflow boundary condition is used at $x=80d$. Using the implicit time integration scheme, results are obtained on six grids with dimensions 63×45, 125×89, 249×177, 497×353, 993×705, 1985×1409 (denoted on plots as grids $1, 2, 3, 4, 5, 6$, respectively). These grids are subsets of each other with every point on a grid contained in its next finest grid; every grid's next coarsest is determined by removing every other grid point in each direction. The grids are stretched in both directions to concentrate grid points near the cylinder. Using the fully implicit time integration scheme with a time step of $(0.24)/2^{(k-1)}$ (where k is the grid number), the solution is advanced 3840 units of non-dimensional time toward steady state while keeping the L_∞ norm of the residual less than 10^{-1} at each time step. Upon termination of the time integration, the L_∞ norm of the residual is less than 2×10^{-4} and the steady-state solution has been obtained. A value for φ of 256 is used with the PVR and CV methods. A metric to quantify the results of the different methods is the mass flow in the x direction across the centerline of the cylinder (at $x=0$); as the cylinder is a closed surface this mass flow should be zero. Figure 17(a) shows that using the PVR and CV methods, this mass flow approaches zero as the grid spacing is reduced, while using the PV method the mass flow asymptotes to a constant value of ~ 0.3. As global mass is conserved, this mass flow must exit the cylinder. As it exits the cylinder it changes the wake, causing flow separation to

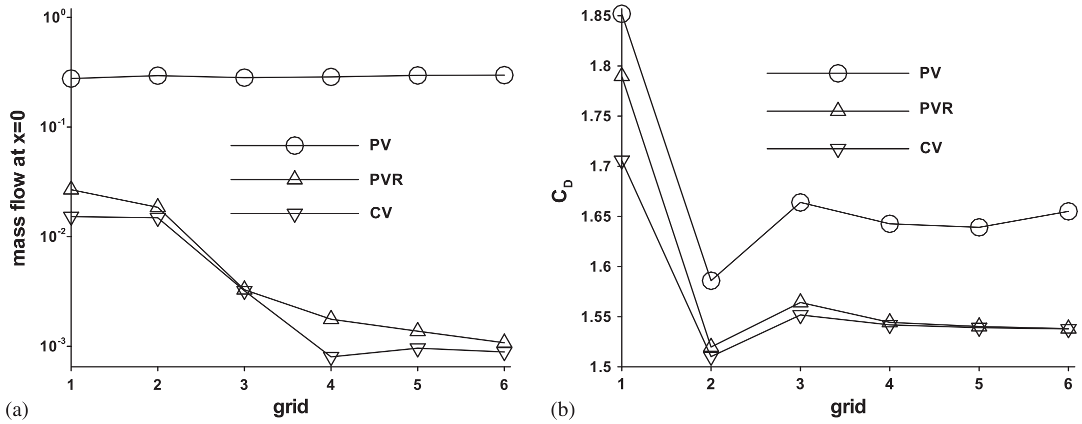

Figure 17. A cylinder in crossflow, $Re=40$: (a) mass flow in the x direction through the center of cylinder and (b) the drag coefficient.

DOI: 10.1002/fld

Table I. Drag coefficient, a cylinder in crossflow, $Re = 40$.

	PV	PVR	CV	[15]	[25]	[24]
Cd	1.65	1.54	1.54	1.51	1.55	1.5

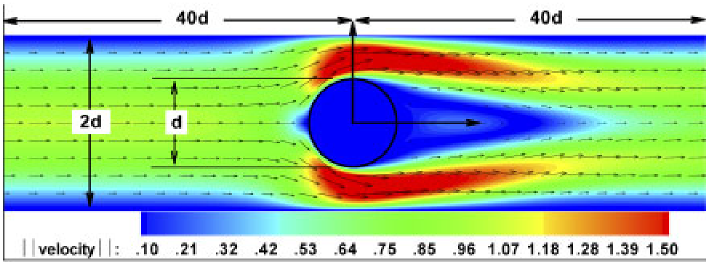

Figure 18. Schematic of a cylinder in a channel, the CV method, 2433 × 161 grid, every 16th vector shown, contours are of velocity magnitude.

occur earlier which increases the drag coefficient of the PV method *vs* the CV method (the PVR method behaves almost identical to the CV method, as is predicted by Equation (39)). The drag coefficient (obtained using the 2nd method) converges as the grid spacing is reduced, and on the sixth grid the drag coefficient can be considered to be grid independent, although there is a slight (0.9%) difference between the results on the fifth and sixth grid for the PV method. Table I shows the drag coefficient of the three methods along with the results of previous works. The PVR and CV methods predict values within the range found in the previous works. The PV method slightly (7%) overpredicts the drag coefficient relative to the PVR and CV methods. For this flow, while the PV method allows a significant flow leakage through the immersed boundary, it still predicts the drag coefficient relatively well.

8.4. Cylinder in channel

The PV, PVR and CV methods are applied to a cylinder confined in a channel; a schematic, dimensions and the coordinate system origin are shown in Figure 18. Dirichlet boundary conditions of $u = 1 - y^2$, $v = 0$ are used at $x = -40d$, $u = 0$, $v = 0$ at $y = \pm d$ and a convective outflow boundary condition for both u and v is used at $x = 40d$. Solutions are obtained at Reynolds numbers of 10, 20, 40 and 100 (based on the cylinder diameter and maximum inflow velocity). At these Reynolds numbers, the flow is steady and two-dimensional and, therefore, the z direction is not resolved. Using the fully implicit time integration scheme, results are obtained on five grids with dimensions 153×11, 305×21, 609×41, 1217×81, 2433×161 (denoted on plots as grids 1, 2, 3, 4, 5, respectively). The maximum difference over all three methods and Reynolds numbers for the drag coefficient (obtained using the 2nd method) between the fourth and fifth grids is 0.7% and, therefore, the solution obtained on the fifth grid is considered to be grid independent. Results are compared with those obtained by [24], who used a body-fitted finite volume scheme to solve the identical problem. Using a value for φ of 256 (at which the solution becomes independent of φ), the drag coefficients obtained from the PVR and CV methods match very closely at all Reynolds numbers those found by [24], while the PV method greatly underpredicts the drag coefficient,

DOI: 10.1002/fld

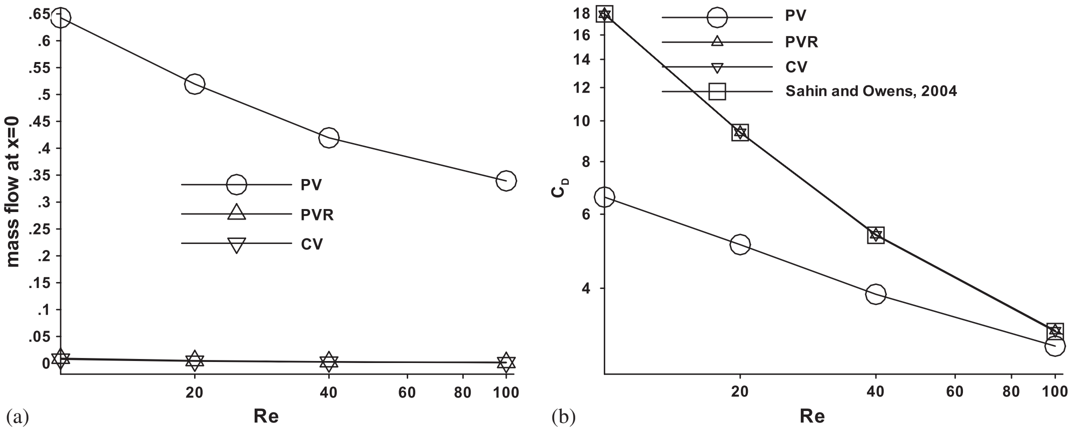

Figure 19. A cylinder in a channel, 2433 × 161 grid: (a) mass flow in the x direction through the center of a cylinder and (b) the drag coefficient.

particularly at low Reynolds numbers (Figure 19(b)). The mass flow in the x direction through the center of the cylinder at $x=0$ is shown in Figure 19(a); if the boundary conditions at the immersed boundary are enforced properly, this mass flow should be zero. The PVR and CV methods have essentially zero mass flow, while the PV method has very high values, particularly at low Reynolds numbers.

9. SPATIAL ORDER OF ACCURACY

In general, any numerical approximation based on polynomial expansions will, if the solution has no discontinuities and with constant grid spacing (Δx) the same in each spatial direction, contain truncation error terms of the form:

$$\boldsymbol{\varepsilon}(\Delta x)=c_n\Delta x^n+c_{n+1}\Delta x^{n+1}+\cdots+c_\infty\Delta x^\infty$$

where n is the spatial order of accuracy of the scheme and $c_n,\ c_{n+1}+\cdots+c_\infty$ are functions of partial derivatives. If it is assumed that the error is dominated by the leading-order error terms of power n (i.e. $c_n\Delta x^n$), there are two methods which can be used to determine the approximate value of n. In the following, Q is the exact solution, $q(\Delta x)$ the approximate solution obtained on a grid with grid spacing of Δx, α a parameter that controls the relative grid spacing between the different grids on which solutions are obtained and $\|\|$ indicates either the L_1, L_2 or L_∞ norms. The L_1 norm of a vector β is defined as $\sum|\beta|/\text{size}[\beta]$, the L_2 norm is defined as $(\sum(\beta^2/\text{size}[\beta]))^{1/2}$.

9.1. Method 1

In this method, the exact solution, if not known, is approximated by the solution (Q_{finest}) obtained on the finest grid. Using solutions obtained on two different (coarser) grids, an expression

DOI: 10.1002/fld

(Equation (44)) for the order of accuracy can be obtained:

$$\frac{\|\boldsymbol{\varepsilon}(\Delta x)\|}{\|\boldsymbol{\varepsilon}(\alpha\Delta x)\|}=\frac{\|q(\Delta x)-Q_{\text{finest}}\|}{\|q(\alpha\Delta x)-Q_{\text{finest}}\|}=\frac{\|c_n(\Delta x)^n\|}{\|c_n(\alpha\Delta x)^n\|}=\frac{1}{\alpha^n}\rightarrow n=\log_\alpha\left(\frac{\|q(\alpha\Delta x)-Q_{\text{finest}}\|}{\|q(\Delta x)-Q_{\text{finest}}\|}\right) \qquad (44)$$

As higher-order terms are neglected in obtaining Equation (44) and the exact solution is approximated by the solution obtained on the finest grid (Q_{finest}), the order of accuracy determined will not be exact.

9.2. Method 2

The previous method is appropriate if the exact solution is not known or is well approximated by the solution obtained on the finest grid, if this is not the case then there is another method, requiring solutions on three different grids, in which the exact solution cancels out:

$$\frac{\|q(\alpha\Delta x)-q(\Delta x)\|}{\|q(\alpha\alpha\Delta x)-q(\alpha\Delta x)\|}=\frac{\|Q+c_n(\alpha\Delta x)^n-(Q+c_n(\Delta x)^n)\|}{\|Q+c_n(\alpha\alpha\Delta x)^n-(Q+c_n(\alpha\Delta x)^n)\|}=\frac{\alpha^n-1}{(\alpha\alpha)^n-\alpha^n}$$

Simplifying

$$-\frac{\|q(\alpha\Delta x)-q(\Delta x)\|}{\|q(\alpha\alpha\Delta x)-q(\alpha\Delta x)\|}\alpha^n\alpha^n+\left(1+\frac{\|q(\alpha\Delta x)-q(\Delta x)\|}{\|q(\alpha\alpha\Delta x)-q(\alpha\Delta x)\|}\right)\alpha^n-1=0$$

The solution to this is $\alpha^n=(1,\|q(\alpha\alpha\Delta x)-q(\alpha\Delta x)\|/\|q(\alpha\Delta x)-q(\Delta x)\|)$. We reject the first possibility because this would require a zeroth-order numerical scheme. Therefore, we conclude that

$$n=\log_\alpha\left(\frac{\|q(\alpha\alpha\Delta x)-q(\alpha\Delta x)\|}{\|q(\alpha\Delta x)-q(\Delta x)\|}\right) \qquad (45)$$

9.3. Cylinder in driven cavity

To eliminate errors resulting from the temporal discretization, a two-dimensional steady-state problem of a cylinder centered in a square-driven cavity is solved. No slip velocity boundary conditions are applied on all four sides of the square cavity, with the exception of the top wall which moves in the positive x direction with a velocity of 1; the immersed boundary method is used to apply no slip velocity boundary conditions on the cylinder. The problem is non-dimensionalized with the diameter (D) of the circular cylinder and the velocity of the moving top lid of the driven cavity. The Reynolds number based on this non-dimensionalization is 20; the dimension of the square cavity is $2D$. Constant grid spacing, the same in both dimensions, is used and beginning with the coarsest grid of 11×11, the grid spacing is halved in turn on each of the five successive finer grids. Using the fully implicit convection diffusion scheme, the equations are integrated for 150 non-dimensional units of time until the steady-state solution is reached. The time step used on the coarsest (11×11) grid is 0.1 and is halved in turn on each of the five successive finer grids. A value of 256 is used for φ for both the PVR and CV methods.

The order of accuracy of both components of velocity is found using Equations (44) and (45) for each component of velocity separately; the order of accuracy shown in Figure 20 is the arithmetic mean of the orders of accuracy of each velocity component. The error norms are taken

DOI: 10.1002/fld

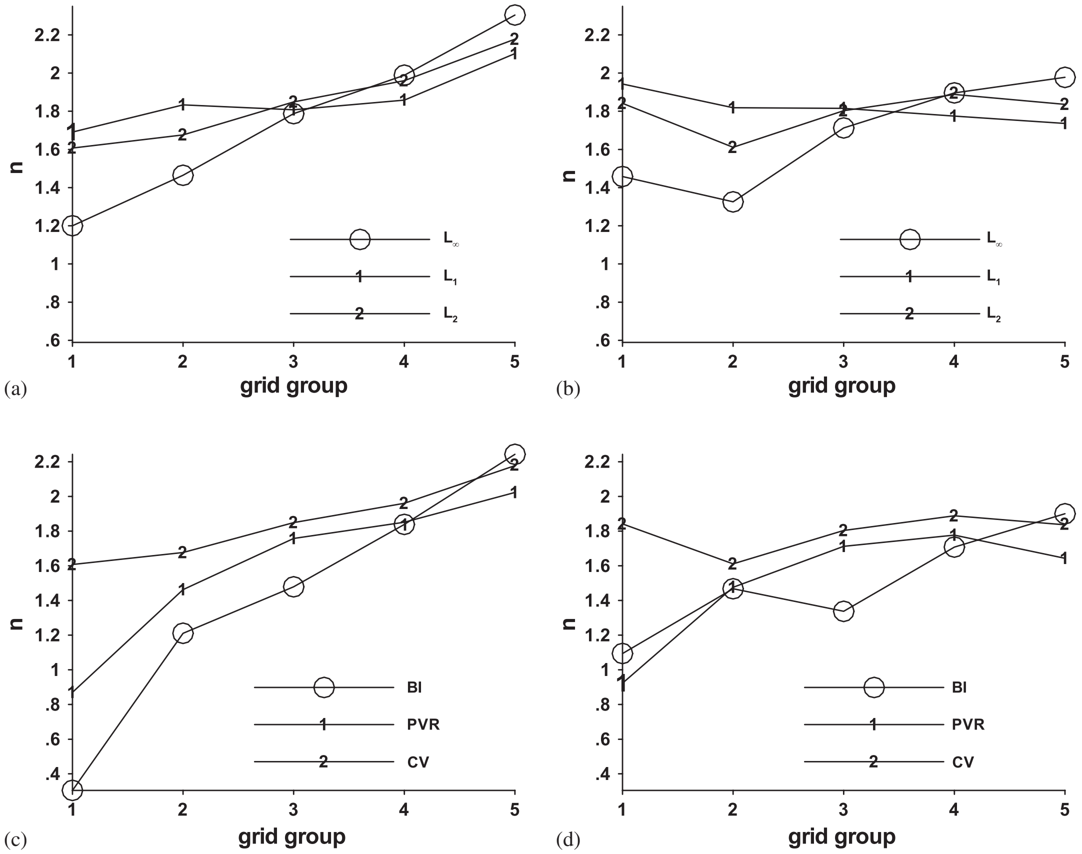

Figure 20. Spatial order of accuracy, cylinder in driven cavity, $Re = 20$: (a) order of accuracy Method 1, CV method; (b) order of accuracy Method 2, CV method; (c) order of accuracy Method 1, L_2 norm; and (d) order of accuracy Method 2, L_2 norm.

over all interior grid points which lie on or outside the cylinder. As the finer grids contain the coarser grids, the error norms are defined on the grid points on the coarsest grid. The solution obtained on the finest (641^2) grid is used to represent the exact solution in Equation (44). The x axis 'grid group' in Figure 20 are the grids used; for Method 1 they are $(11^2, 21^2, 641^2)$, $(21^2, 41^2, 641^2)$, $(41^2, 81^2, 641^2)$, $(81^2, 161^2, 641^2)$, $(81^2, 321^2, 641^2)$; for Method 2 they are $(11^2, 21^2, 41^2)$, $(21^2, 41^2, 81^2)$, $(41^2, 81^2, 161^2)$, $(81^2, 161^2, 321^2)$, $(161^2, 321^2, 641^2)$.

Using Method 1, the CV method is found to be greater than second-order accurate (Figure 20(a)) while using Method 2, the CV method is found to be nearly second-order accurate (Figure 20(b)). Figure 20(c) and (d) shows the PVR method to have a slightly lower order of accuracy than either the BI or CV method. The CV method has the highest order of accuracy. However, all methods converge toward second-order accuracy, indicating that these immersed boundary methods have maintained the second-order spatial accuracy of the numerical scheme.

10. PRESSURE

At immersed boundary points, the modified momentum equations contain a term of the form $\mathbf{u}^{n+1}-(L_{\mathrm{NB}}()+L_{\mathrm{IB}}(\mathbf{u}_{\mathrm{IB}}^{n+1}))=\Delta t(S(\mathbf{u}^{n+1},\mathbf{u}^{n}\ldots)-\nabla p^{n+1})$ (PV method) or $\mathbf{u}^{n+1}-(L_{\mathrm{NB}}()+L_{\mathrm{IB}}(\mathbf{u}_{\mathrm{IB}}^{n+1}))=\Delta t/\varphi(S(\mathbf{u}^{n+1},\mathbf{u}^{n}\ldots)-\nabla p^{n+1})$ (PVR and CV methods). Assume that as $\Delta t\rightarrow 0$ or $\varphi\rightarrow\infty$, the velocity converges and becomes independent of φ and Δt. Therefore, the only term on the right-hand side that is significant is $-\Delta t\nabla p^{n+1}$ or $-(\Delta t/\varphi)\nabla p^{n+1}$. Furthermore, since the velocity converges and, therefore, $\mathbf{u}^{n+1}-(L_{\mathrm{NB}}()+L_{\mathrm{IB}}(\mathbf{u}_{\mathrm{IB}}^{n+1}))$ converges to a non-zero constant, the pressure gradient at immersed boundary points must increase as $1/\Delta t$ or φ. Figure 21 shows the bounded behavior of the L_∞ norm of the pressure gradient at non-immersed boundary points and the unbounded behavior of the L_∞ norm of the pressure gradient at immersed boundary points when either $\Delta t\rightarrow 0$ in the PV method or $\varphi\rightarrow\infty$ in the CV method (the PVR method behaves like the CV method). As a result of the criteria used to select immersed boundary points, each immersed boundary point belongs to at least one mass conservation cell in which all velocity points are immersed boundary points. As a two-point stencil is used to discretize the pressure gradient, the pressure, in mass conservation cells in which all velocity points are immersed boundary points, is only used in computing the pressure gradient at immersed boundary points. Therefore, it can be concluded that norms of the pressure in mass conservation cells in which all velocity points are immersed boundary points also do not asymptote as $\Delta t\rightarrow 0$ in the PV method or $\varphi\rightarrow\infty$ in the PVR or CV methods. For this reason, the authors believe the pressure at these points is unphysical. This unphysical behavior is shown in Figure 22 as the magnitude of the pressure increases as Δt decreases in the PV method or φ increases in the CV method (the PVR method behaves similar to the CV method). However, as norms of the pressure gradient at non-immersed boundary points asymptote, the pressure in mass conservation cells in which not all velocity points are immersed boundary points must also asymptote. Therefore, if pressure is desired at mass conservation cells in which all velocity points are immersed boundary points, it could be obtained by interpolating from the pressure in mass conservation cells in which not all velocity points are immersed boundary points.

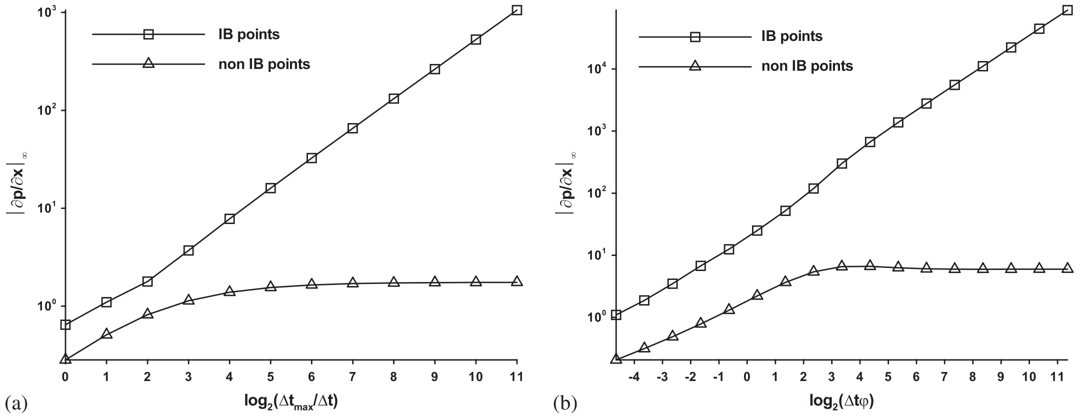

Figure 21. The pressure gradient at immersed boundary and non-immersed boundary points, a two-dimensional obstacle: (a) the PV method, explicit convection–diffusion time integration, $Re=20$, 40×20 grid and (b) the CV method, implicit convection–diffusion time integration, $Re=10^6$, 161×81 grid.

DOI: 10.1002/fld

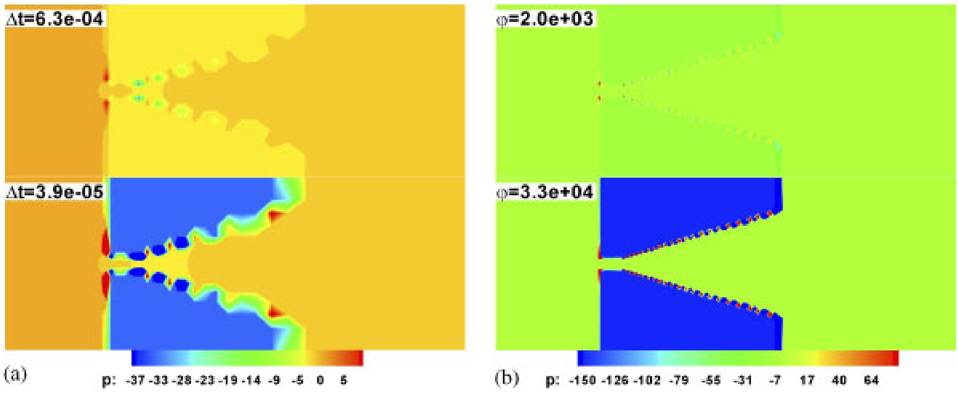

Figure 22. Pressure contours, a two-dimensional obstacle: (a) the PV method, explicit convection–diffusion time integration, $Re = 20$, 40×20 grid and (b) CV method, implicit convection–diffusion time integration, $Re = 10^6$, 161×81 grid.

The behavior in which the pressure increases without bound as $\Delta t \rightarrow 0$ in the PV method or $\varphi \rightarrow \infty$ in the PVR or CV method in one mass conservation cell while remaining bounded in a neighboring cell gives rise to the oscillitory pressure fields shown in Figure 22. It should be emphasized though that the velocity does not exhibit this same behavior, but remains bounded throughout the computational domain. This unphysical behavior of pressure near the immersed boundary has also been noticed by [9]; in this work, the pressure was corrected by solving an additional Poisson equation after the velocity field has been obtained. Note that for the BI method, the pressure at points in which all four surrounding velocities are immersed boundary points is completely decoupled from all other discrete velocities and pressure, and cannot be considered physical.

11. CONCLUSION

The performance of commonly used schemes of the immersed boundary method is examined from the perspective of mass conservation and satisfaction of the boundary conditions on the immersed boundary. The use of the velocity in the method of [5] to define the force density is shown to result in an inconsistent set of discrete equations. With the iterative solver used in the present work, this inconsistency results in an inability to obtain a divergence-free velocity, although the boundary conditions resulting from the immersed boundary are satisfied. If the pseudo-velocities are used to define the force density, the problem of an inconsistent set of equations is removed and a divergence-free velocity can be obtained. However, the use of the pseudo-velocities to define the force density in the method of [5] is shown to be inaccurate as the boundary conditions on the immersed boundary are poorly satisfied. If the pseudo-velocities are used to define the force density, the boundary conditions on the immersed boundary are shown to be strongly dependent on the time step and are better satisfied as the time step is reduced; however, this reduction of the time step results in a large computational expense.

A new method (PVR) that reduces the time step only at immersed boundary points is presented. This method greatly improves satisfaction of the boundary conditions on the immersed boundary without increasing the computational expense. However, for problems in which the immersed boundary is moving, this method can perform poorly. In order to remedy this, a second new method (CV) which performs well on moving immersed boundary problems is presented. This method is based on the concept of remaining as close as possible to the boundary conditions defined by the immersed boundary stencil, while maintaining a divergence-free velocity. This method, which is based on constrained optimization, is shown to be related to the PVR method. The methods are demonstrated on four different flows (a two-dimensional obstacle, a structure with a vibrating wall, a cylinder in crossflow and a cylinder in a channel), on which grid-independent results are shown and compared to either an analytical solution or the results of others. Second-order convergence in the grid spacing of the new methods is shown through numerical experiments. As the overall spatial accuracy of the discretized Navier–Stokes equations is also second order, this shows that the spatial order of accuracy is not adversely affected by the new methods. While the example problems in the present work are two dimensional for computational convenience, the new methods have also been applied to three-dimensional flows without encountering any additional difficulties or necessary modifications.

The concept of the immersed boundary method is that a force density is added to the Navier–Stokes equations to satisfy the boundary conditions at the immersed boundary. In the method of [5], the force density is not explicitly computed and added to the Navier–Stokes equations, however, the force density, which represents the force of the immersed boundary on the fluid, can be very useful information. The unbounded behavior of the local force density and the bounded behavior of the integral of the force density over the entire region in which the Navier–Stokes equations are solved is shown. The value of this integral is shown to depend on the iterative method used to solve the discretized Navier–Stokes equations and the residual of the iterative process. A method of computing the force density which is independent of the residual of the iterative method is presented. This method is shown to allow accurate computations of the total force of the fluid on the immersed boundary regardless of the residual level, which is particularly important for problems where the motion of the immersed boundary is determined by the force of the fluid.

An explanation is given for the observed unphysical and unbounded behavior of the pressure at certain locations near the immersed boundary. This behavior is shown to occur in all the methods examined in the present work.

ACKNOWLEDGEMENTS

This work was supported by the Office of Naval Research and the Center for Computation and Technology at Louisiana State University. The simulations were run on workstations at the Center for Computation and Technology at Louisiana State University, the Linux Xeon clusters ‘SuperMike’ and ‘Helix’ at Louisiana State University, the IBM machines Casper and Pelican at Louisiana State University and an IBM SP4 at the Arctic Region Supercomputing Center. The authors are indebted to Major (USAF-Ret.) Oliver Muldoon for a careful proofread and helpful suggestions for improving the clarity of this work.

REFERENCES

1. Peskin CS. Numerical analysis of blood flow in the heart. *Journal of Computational Physics* 1977; **25**:220–252.
2. Goldstein D, Handler R, Sirovich L. Modeling a no-slip flow boundary with an external force field. *Journal of Computational Physics* 1993; **105**:354–366.

3. Fadlun EA, Verzicco R, Orlandi P, Mohd-Yusof J. Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations. *Journal of Computational Physics* 2000; **161**:35–60.
4. Beyer RP. A computational model of the cochlea using the immersed boundary method. *Journal of Computational Physics* 1992; **98**:145–162.
5. Mohd-Yusof J. *Combined Immersed-boundary/b-Spline Methods for Simulations of Flow in Complex Geometries*, Center for Turbulence Research Annual Research Briefs, 1997; 317–327.
6. Zhang N, Zheng Z. An improved direct-forcing immersed-boundary method for finite difference applications. *Journal of Computational Physics* 2007; **221**(1):250–268.
7. Gilmanov A, Sotiropoulos F. A hybrid Cartesian/immersed boundary method for simulating flows with 3D, geometrically complex, moving bodies. *Journal of Computational Physics* 2005; **207**:457–492.
8. Tyagi M, Acharya S. Large eddy simulation of turbulent flows in complex and moving rigid geometries using the immersed boundary method. *International Journal for Numerical Methods in Fluids* 2005; **48**:691–722.
9. Kang S, Iaccarino G, Moin P. *Accurate and Efficient Immersed-boundary Interpolations for Viscous Flows*, Center for Turbulence Research Annual Research Briefs, 2004; 31–43.
10. Tseng Y-H, Ferziger JH. A ghost-cell immersed boundary method for flow in complex geometry. *Journal of Computational Physics* 2003; **192**:593–623.
11. Cui J, Patel VC, Lin C-L. Prediction of turbulent flow over rough surfaces using a force field in large eddy simulation. *Journal of Fluids Engineering* 2003; **125**:2–9.
12. Tyagi M. Large eddy simulation of complex turbulent flows. *Ph.D. Thesis*, Louisiana State University, 2003.
13. Iaccarino G, Verzicco R. Immersed boundary technique for turbulent flow simulations. *Applied Mechanics Review* 2003; **56**:331–347.
14. Verzicco R, Fatica M, Iaccarino G, Orlandi P. *Flow in an Impeller-stirred Tank using an Immersed-boundary Method*, Center for Turbulence Research Annual Research Briefs, 2000; 251–261.
15. Kim J, Kim D, Choi H. An immersed-boundary finite volume method for simulations of flow in complex geometries. *Journal of Computational Physics* 2001; **79**:317–327.
16. Mohd-Yusof J. *Development of Immersed Boundary Methods for Complex Geometries*, Center for Turbulence Research, Annual Research Briefs, 1998; 325–326.
17. Muldoon F. Numerical methods for the unsteady incompressible Navier–Stokes equations and their application to the direct numerical simulation of turbulent flows. *Ph.D. Thesis*, Louisiana State University, 2004.
18. Vanka SP. A calculation procedure for three-dimensional steady recirculating flows using multigrid methods. *Computer Methods in Applied Mechanics and Engineering* 1986; **55**:321–338.
19. Iaccarino G, Ham F. *Automatic Mesh Generation for LES in Complex Geometries*, Center for Turbulence Research Annual Research Briefs, 2005; 19–30.
20. Chorin AJ. Numerical solution of the Navier–Stokes equations. *Mathematics of Computation* 1968; **22**:745–762.
21. Issa RI. Solution of the implicitly discretized flow equations by operator splitting. *Journal of Computational Physics* 1986; **62**:40–65.
22. Patankar SV. *Numerical Heat Transfer and Fluid Flow*. Hemisphere: Washington, DC, 1980.
23. Chorin AJ. A numerical method for solving incompressible viscous flow problems. *Journal of Computational Physics* 1967; **2**:12–26.
24. Sahin M, Owens RG. A numerical investigation of wall effects up to high blockage ratios on two-dimensional flow past a confined cylinder. *Physics of Fluids* 2004; **16**:1305–1320.
25. Silva ALFLE, Silveira-Neto A, Damasceno JJR. Numerical simulation of two-dimensional flows over a circular cylinder using the immersed boundary method. *Journal of Computational Physics* 2003; **189**:351–370.

DOI: 10.1002/fld