AIAA AVIATION Forum
5-9 June 2017, Denver, Colorado
23rd AIAA Computational Fluid Dynamics Conference

10.2514/6.2017-3624

Very High-Order Accurate Sharp Immersed Interface Method: Application to Direct Numerical Simulations of Incompressible Flows

Shirzad Hosseinverdi[1], Hermann F. Fasel[2]
Department of Aerospace and Mechanical Engineering
The University of Arizona, Tucson, AZ 85721

Development of robust and very high-order accurate Immersed Interface Method (IIM) for solving the incompressible Navier-Stokes equations in the vorticity-velocity formulation on non-equidistant grids is presented. For computation of spatial derivatives on regular grid points, a seventh-order upwind Combined Compact Difference (CCD) scheme for first-derivative and sixth-order central CCD scheme for second derivative are employed. The coefficients of the CCD schemes are constructed for a non-equidistant grid instead of using a coordinate transformation. Corrections to the finite-difference schemes are used for irregular grid points near the interface of the immersed boundary to maintain high formal accuracy. For the interface points, the CCD schemes are tuned and adjusted accordingly to obtain numerically stable schemes and no jump correction will therefore be required. To demonstrate the numerical stability of the new IIM, both semi- and fully-discrete eigenvalue problems are employed for the one-dimensional pure advection (inviscid) and the pure diffusion, and advection-diffusion equations. The new IIM satisfies both necessary and sufficient conditions for numerical stability. The new IIM was first applied to two-dimensional linear advection equation to demonstrate its stability. Then the development of a new, efficient and high-order sharp-interface method for the solution of the Poisson equation in irregular domains on non-equidistant grids is presented. The underlying approach for this is based on the combination of the fourth-order compact finite difference scheme and the Multiscale Multigrid (MSMG) method. The computational efficiency of the new solution strategy for the Poisson equation is demonstrated with regard to convergence rate and required computer time, which shows that the MSMG method is equally efficient for domains with immersed boundaries and for simple domains. To validate the application of IIM for incompressible flows, the results from the new method is compared with the benchmark solutions for the flow past a circular cylinder and the propagation of Tollmien–Schlichting wave in a boundary layer.

I. Introduction

DIRECT Numerical Simulations (DNS) have become one of the major tools to study and investigate transitional and turbulent flows. However, its application has been often limited to simple geometries and low Reynolds numbers. Extending DNS to more complex geometries remains a difficult task. The primary issues are accuracy, numerical stability and computational efficiency. Traditional approaches in Computational Fluid Dynamics (CFD) use conformal body-fitted structured or unstructured grids. However, generating high-quality grids is generally cumbersome and it becomes very laborious as the complexity of the geometry increases. Furthermore, the grid generation process becomes a very difficult task, even for the simplest geometries, when more than one body is located within the domain. Grids with poor qualities (smoothness, orthogonality, aspect ratio) could negatively impact the accuracy and convergence properties of the numerical method for solving the Navier-Stokes equations.

An alternative approach is to use the so-called immersed boundary method (IBM) where the equations of interest are discretized on a fixed Cartesian grid in a bounded domain Ω which contain an arbitrary immersed body with boundary Γ, in which case the domain is divided by the curve Γ into two subdomains Ω^+ and Ω^- as shown in Fig. 1.

[1] PhD candidate, AIAA member.
[2] Professor, AIAA Associate Fellow.

Referring to Fig. 1, one would typically solve a PDE defined on the open region Ω^+ with boundary conditions on $\partial\Omega$, the outer boundary which conforms to the computational boundary, and Γ, the immersed boundary itself. The solution in the region Ω^- may or may not be of interest. In either case, the immersed boundary Γ represents a singularity, thus field variables and/or their derivatives will be discontinuous across the immersed boundary.

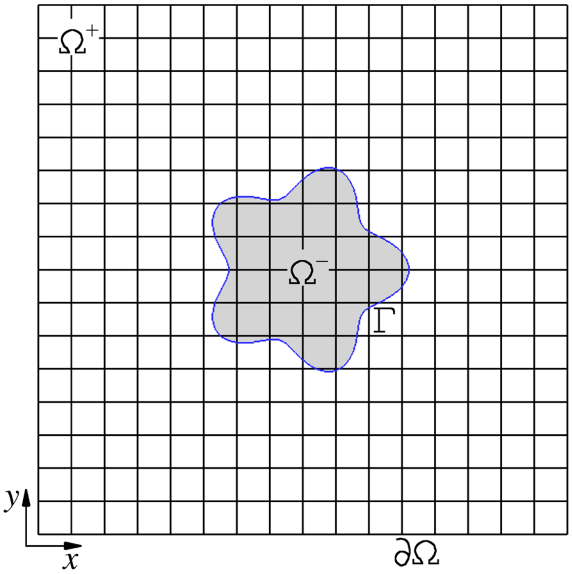

Figure 1. Schematic of an arbitrary geometry immersed in a Cartesian grid.

Immersed boundary methods have recently been demonstrated to be applicable to complex geometries while requiring significantly less computation than traditional approaches (Mittal et. al.[1], Brehm & Fasel[2], Linnick & Fasel[3], Duan et al.[4] and many others). The main advantage of the IBM lies in the use of Cartesian grids for which there is almost no grid generation cost. Moreover, many efficient numerical methods such as high-order methods, structured multigrid solvers, and many others have been developed for Cartesian grids can be used in IBMs. A disadvantage of the IBM is that the imposition of boundary conditions on the immersed boundaries is not straightforward as compared to traditional body fitted methods. Also, the influence of the boundary treatment on accuracy and conservation properties of the numerical schemes is not trivial. Several methods have been proposed in the past to handle the singularity associated with the immersed boundary. Generally, they can be classified as either continuous forcing (diffuse) approaches or discrete forcing (sharp) methods (Mittal & Iaccarino[5], Marella et al.[6],). A detailed overview of immersed boundary methods can be found in (Mittal & Iaccarino[5]).

In the continuous methods, the boundary conditions are enforced through a smooth forcing term added to the equations. Peskin[7,8] pioneered this type of IBM to handle elastic boundaries for simulating blood flow in the heart. The method was later extended to simulate flow with rigid boundaries (see for instance Lai & Peskin[9], Su et. al.[10]). One disadvantage of diffuse methods is that the effect of the boundary is distributed over a band of several grid points which smears out discontinuities across the boundary. This smearing has a detrimental effect on the accuracy of the numerical scheme.

In the other approach, the discrete IBM, the numerical discretization near the immersed boundary is modified such as to account directly for the presence of the boundary, so that the interface still remains "sharp". There are several different implementations of the sharp-interface approach. One of them is the so-called "ghost cell" approach. Ghost cells are defined as cells in the solid region that have at least one neighbor in the fluid. Boundary conditions at the immersed boundary is enforced through the ghost cells. For each ghost cell, an interpolation scheme that implicitly incorporates the boundary condition on the immersed boundary is then devised. A number of options are available for constructing the interpolation scheme (Tseng &. Ferziger[11], Mittal et. al.[1]). While the approach is well suited for achieving second-order accuracy, extension to higher-order formulations is problematic. Higher-order formulations require large interpolation stencils which could lead to robustness issues. Furthermore, obtaining the ghost cell value is not that obvious for more irregular shaped geometries, e.g. sharp corner.

Another method that falls into the sharp category is the so called "cut-cell" method, often used within the finite volume framework (Udaykumar[12,13]). Being based on the finite volume approach, this method is designed to provide better conservation properties, especially for cells around the immersed boundary. Grid cells cut by the immersed boundary are reshaped to conform to the boundary. This reshaping may in some cases result in very small grid cells with an adverse impact on numerical stability. To overcome this problem, cell-merging strategies have been successfully proposed. However, due to the many possible configurations of the irregular cut-cell, particularly in three dimensions, implementation becomes a tedious and non-trivial task. The cut-cell method is also used in the finite-difference approach, see Duan et al.[4] The basic idea behind the method is to switch to a non-uniform one-sided finite difference stencil whenever the normally used stencil will cross the immersed boundary. To overcome the small cell problem, a dropped-point approach is used. A grid point is defined as a dropped point if it is adjacent to a boundary point along a grid line with a distance smaller than a pre-specified critical distance.

An alternative to the immersed boundary method is the sharp interface immersed interface method (IIM) introduced by LeVeque & Li[14] to solve elliptic problems with discontinuous and non-smooth solutions. The basic idea behind the IIM approach, first discussed by LeVeque & Li[14], and later clarified by Wiegmann & Bube[15] is to recognize that standard finite difference techniques fail when applied to non-smooth functions because the underlying Taylor expansions upon which they are based are invalid. The key idea of the IIM is that correction terms are added to the finite difference schemes at the interface of the immersed boundary to account for any jump in the solution or its derivative in order to maintain the formal accuracy of the underlying numerical scheme. An important application of sharp IIM is the treatment of problems defined in an irregular domain where the solution inside or the outside the interface is not of interest and is trivial. In such cases, higher order discretizations can be used near the interface. The original IIM has been extended for simulating incompressible flows with solid boundaries (see e.g., Li & Wang[16]).

Brehm & Fasel[2, 17] and Brehm et al.[18] proposed a new IIM scheme to solve the incompressible and compressible Navier–Stokes equations using explicit finite difference schemes. In addition to conventional immersed schemes, which usually only consider the local truncation error at irregular grid points, they combined local Taylor-series expansion at irregular grid points with a local stability constraint as part of the design process of the boundary stencils. Their novel concept provides improved robustness to the numerical scheme.

Linnick & Fasel[3] developed a fourth-order accurate, compact Finite-Difference (FD) scheme based on the IIM. A stream-function-vorticity formulation was used to solve the incompressible Navier–Stokes Equations. They were able to accurately simulate the hydrodynamic instability (Tollmien-Schlichting waves) in a Blasius boundary-layer in addition to computing the flow past a circular cylinder, an elliptic airfoil and a backward-facing step. From the stability analysis of one-dimensional linear advection equation, however, it was found that their IIM is unstable for advection equation. For the advection-diffusion equation, it was stable for cell Reynolds numbers less than a limiting value that depends upon the location of the immersed boundary.[19]

A. Scope and Objective of the Present Investigation

Numerical solution of fluid flows dominated by a wide range of length and time scales, e.g. transitional and turbulent flows, requires stable and high-order numerical methods that can accurately represent the entire, or at least a significant portion of this range of scales. Despite the continuous progress of IBM and IIM, DNS of flows in complex geometries remains a difficult for several reasons: i) Most IBMs/IIMs use low-order and explicit finite difference schemes that have low resolution characteristics in high wavenumbers, (ii) they suffer from robustness issues when extended to higher-order and (iii) they are numerically unstable in convectively dominated flows (e.g. inviscid flows, high Reynolds number).

Apart from the accuracy and stability of IIMs/IBMs, Poisson equation represents one of the main pieces in the solution of the incompressible Navier-Stokes Equations (INSE) and the vast majority of the computational time necessary to solve the INSE is spent solving the Poisson equation. Thus, it is evident that this piece of the INSE solver must be as computationally efficient as possible. However, what has not yet been addressed is the question of how to solve the Poisson equation highly-accurately and still be as computationally efficient as for the case without immersed boundary.

The goal of the present research is the development of a robust/stable, very high-order accurate, and computationally efficient sharp IIM aimed at the DNS of transitional and turbulent incompressible flows over complex geometries. Hence, our objective in this paper is twofold:

- First, we use seventh- and sixth-order combine compact difference schemes to approximate the first and second spatial derivatives, respectively. Instead of using grid transformation, the CCD schemes are constructed on a non-equidistant grid. The CCD schemes give a higher order of accuracy and spectral resolution characteristics for the same number of grid points compared to the explicit and compact finite difference schemes. For grid points located adjacent to the immersed boundary that their finite difference stencils contain a solid point (point inside the body), stencils are adjusted and corrections to the finite difference schemes are used to maintain high formal accuracy. The stability of the new IIM is then tested for one- and two-dimensional linear advection-diffusion equation.
- Next, the development of a new, efficient and high-order method for the solution of the Poisson equation in a domain with immersed boundaries on non-uniform meshes is presented. To achieve this goal, a fourth-order compact finite difference scheme is combined with the multiscale multigrid method. Our new method falls into the sharp interface category. However, it distinguishes itself from other IIM in that the jump corrections are not required any more. The underlying feature of the new method is that the regular finite difference stencil is adjusted at the irregular grid points in the vicinity of the interfaces of the immersed bodies to obtain a solution that is sharp across the interface while keeping the fourth-order global accuracy. In the current work, the standard multigrid V-cycle technique is used to build the MSMG method whose aim is to converge to the discretization error in a fixed number of iterations which is independent of the grid size.

II. Spatial Discretization for Regular Grid Points: Combined Compact Difference Schemes

In this section, the spatial derivatives (first and second) in the Navier–Stokes equations are evaluated by combined compact difference schemes on non-uniform grids for regular points. The regular points are all grid points whose finite difference stencil does not contain any solid points (grid points inside an immersed body). The CCD schemes are more accurate than the standard Padé schemes, while using the same stencil width. By solving the first and second derivatives simultaneously, the CCD schemes have higher-order accuracy and better spectral characteristics (Mahesh[20], Chu & Fan[21]).

Although any CCD scheme simultaneously evaluate both first and second derivatives for a given variable, we use one CCD scheme to discretize and compute the first derivative $\partial\phi/\partial x$ (and disregard the second derivative obtained, $\partial^2\phi/\partial x^2$) and another CCD scheme to discretize and calculate the second derivative $\partial^2\psi/\partial x^2$ (ignoring here the first derivative approximation $\partial\psi/\partial x$). This stems from the fact that the new IIM will be applied to the INSE in vorticity-velocity formulation, in which special derivatives such as

$$\frac{\partial(u\omega_z)}{\partial x}, \frac{\partial^2\omega_z}{\partial x^2}, \tag{1}$$

need to be evaluated. Hence, the following derivatives are not needed

$$\frac{\partial^2(u\omega_z)}{\partial x^2}, \frac{\partial\omega_z}{\partial x}. \tag{2}$$

Here ϕ corresponds to the $(u\omega_z)$ term and ψ is the ω_z variable. With this technique, it is then possible to tune the first and second derivatives CCD schemes such as to maximize numerical stability.

A. Interior Schemes

Let ϕ and ψ be defined on the interval $0 \le x \le L$ where $0 = x_1 < x_2 < \cdots < x_N = L$. The following sections will describe the CCD scheme to evaluate the spatial derivatives for $3 \le i \le N-1$. When the problem is periodic, the same scheme may be used at the boundary nodes using the periodicity of the solution. The first and second derivatives of a known periodic cosine function $u = cos(4 \cdot x + 1)$ are calculated using the CCD schemes on a domain of size $L = 2\pi$ with node coordinates given by

$$x_i = k\frac{L}{N} + \frac{\alpha}{2\pi} sin\left(\alpha\frac{2\pi k}{N}\right); \quad k = i-1; \quad 1 \le i \le N \tag{3}$$

where α controls the non-uniformity of the grid. Here we choose $\alpha = 0.0$ for uniform grid and $\alpha = 0.6$ for non-uniform grid. The norms of the errors between the analytical and the numerical derivatives are calculated over the 1D domain as follows

$$L_1 = \left\|\epsilon^{(k)}\right\|_1 = \frac{1}{N}\sum_{i=1}^{N}\left|u_i^{(k)} - u_i^{(k),exact}\right|, \tag{4a}$$

$$L_2 = \left\|\epsilon^{(k)}\right\|_2 = \frac{1}{N}\sqrt{\sum_{i=1}^{N}\left(u_i^{(k)} - u_i^{(k),exact}\right)^2}, \tag{4b}$$

$$L_\infty = \left\|\epsilon^{(k)}\right\|_\infty = \max_{1<i<N}\left|u_i^{(k)} - u_i^{(k),exact}\right|. \tag{4c}$$

Here k corresponds to the kth-derivative ($k = 1$ and $k = 2$).

1. First derivative approximation

The CCD scheme used to evaluate the first derivate can be written as

$$\alpha_{i,1}^{(1)}\phi'_{i-1} + \phi'_i + \alpha_{i,2}^{(1)}\phi'_{i+1} + \alpha_{i,3}^{(1)}\phi''_{i-1} + \alpha_{i,4}^{(1)}\phi''_i + \alpha_{i,5}^{(1)}\phi''_{i+1} = c_{i,1}^{(1)}\phi_{i-2} + c_{i,2}^{(1)}\phi_{i-1} + c_{i,3}^{(1)}\phi_i, \tag{5a}$$

$$\beta_{i,1}^{(1)}\phi'_{i-1} + \beta_{i,2}^{(1)}\phi'_i + \beta_{i,3}^{(1)}\phi'_{i+1} + \beta_{i,4}^{(1)}\phi''_{i-1} + \phi''_i + \beta_{i,5}^{(1)}\phi''_{i+1} = d_{i,1}^{(1)}\phi_{i-1} + d_{i,2}^{(1)}\phi_i + d_{i,3}^{(1)}\phi_{i+1}. \tag{5b}$$

The above scheme will provide both the first and the second derivative approximations, however, we only use the first derivative approximation. The stencil consists of four grid points. Note that the above expression differs from the standard Padé schemes, in that the left-hand side contains a linear combination of the first and second derivatives. This scheme is different from the original CCD scheme proposed by Chu & Fan[21] and Mahesh[20] in that it uses some upwinding.

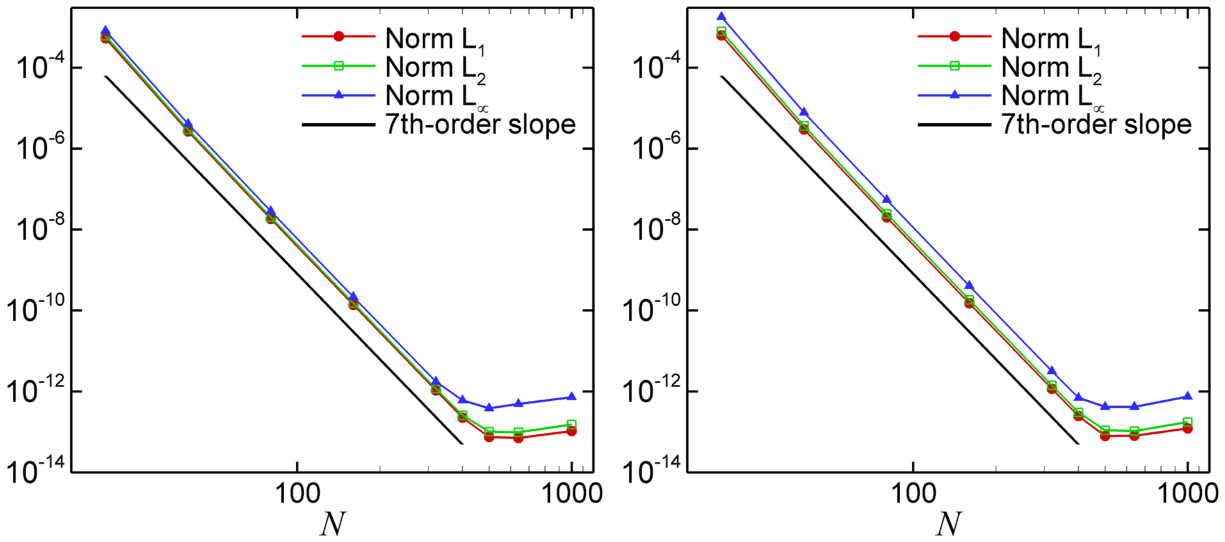

Figure 2. Norms of numerical error as a function of the number of grid points for first derivative. Left: uniform grid; right: Non-uniform grid for $\alpha = 0.6$ (see Eq. 3).

The coefficients in Eq. 5 are derived through matching the Taylor series expansions on a non-uniform grid (see Appendix A). The truncation error of the resulting scheme is determined by the first unmatched coefficient in the Taylor series which is $O(dx^7)$ for the propsed upwind CCD. An order study of the discrete spatial operator is presented in Fig. 2. It shows that a seventh-order scheme is recovered in the case of uniform and smoothly varying non-uniform mesh. The error increase at high N is due to the machine precision.

2. Second derivative approximation

The second derivative is approximated with the following three-point CCD scheme

$$\alpha_{i,1}^{(2)}\psi'_{i-1} + \psi'_i + \alpha_{i,2}^{(2)}\psi'_{i+1} + \alpha_{i,3}^{(2)}\psi''_{i-1} + \alpha_{i,4}^{(2)}\psi''_{i+1} = c_{i,1}^{(2)}\psi_{i-1} + c_{i,2}^{(2)}\psi_i + c_{i,3}^{(2)}\psi_{i+1}, \qquad (6a)$$

$$\beta_{i,1}^{(2)}\psi'_{i-1} + \beta_{i,2}^{(2)}\psi'_i + \beta_{i,3}^{(2)}\psi'_{i+1} + \beta_{i,4}^{(2)}\psi''_{i-1} + \psi''_i + \beta_{i,5}^{(2)}\psi''_{i+1} = d_{i,1}^{(2)}\psi_{i-1} + d_{i,2}^{(2)}\psi_i + d_{i,3}^{(2)}\psi_{i+1}. \qquad (6b)$$

Matching the Taylor series coefficients will determine the stencil coefficients in Eq. 6 as described in Appendix A. Figure 3 demonstrates that the above CCD scheme is 6th-order accurate on the uniform and non-uniform grids. The same test function and grid distribution described in the previous section were used.

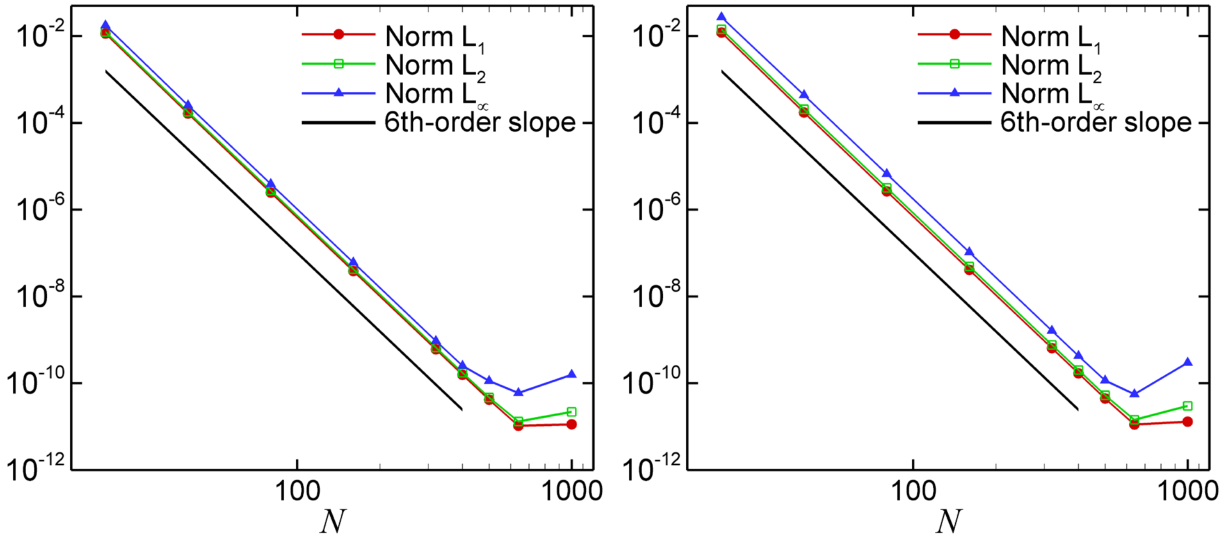

Figure 3. Norms of numerical error as a function of the number of grid points for second derivative. Left: uniform grid; right: Non-uniform grid for $\alpha = 0.6$ (see Eq. 3).

C. Boundary Schemes

For non-periodic problems, additional expressions are needed at the boundary nodes to close the system. Let us consider a domain discretized with N grid points and consider a Dirichlet boundary condition at $i = 1$. Therefore, there are $2 \cdot (N-1)$ unknown variables ($\partial\phi_i/\partial x,\ \partial^2\phi/\partial x^2;\ i = 2,3,\cdots,N$), but there are only $2 \cdot (N-3)$ equations since the CCD schemes are used for $i = 3$ to $i = N-1$. Thus, two additional conditions are needed at each boundary ($i = 2$ & $i = N$), hence resulting in a system of $2 \cdot (N-1)$ equations that can be solved for the nodal values of $\partial\phi_i/\partial x$ and $\partial^2\phi_i/\partial x^2$. The stencils used for boundary points are shown in Fig. 4. The derivation of the boundary stencils is outlined here. The fourth-order accurate stable boundary closure at node $i = 2$ is given by

$$\phi_2' + \alpha_{b2,1}\phi_3' + \alpha_{b2,2}\phi_3'' = c_{b2,1}\phi_1 + c_{b2,2}\phi_2 + c_{b2,3}\phi_3 \tag{7a}$$

$$\beta_{b2,1}\phi_3' + \phi_2'' + \beta_{b2,2}\phi_3'' = d_{b2,1}\phi_1 + d_{b2,2}\phi_2 + d_{b2,3}\phi_3 + d_{b2,4}\phi_4 \tag{7b}$$

The fourth-order accurate stable boundary scheme at node $i = N$ is obtained from the following relations:

$$\alpha_{bN,1}\phi_{N-1}' + \phi_N' + \alpha_{bN,2}\phi_{N-1}'' = c_{bN,1}\phi_N + c_{bN,2}\phi_{N-1} + c_{bN,3}\phi_{N-2} \tag{8a}$$

$$\beta_{bN,1}\phi_{N-1}' + \beta_{bN,2}\phi_{N-1}'' + \phi_N'' = d_{bN,1}\phi_N + d_{bN,2}\phi_{N-1} + d_{bN,3}\phi_{N-2} + d_{bN,4}\phi_{N-3} \tag{8b}$$

Similar schemes can be used for second derivative ($\partial^2\psi/\partial x^2$). The coefficients in the above equations are derived through matching the Taylor series expansions on a non-uniform grid (see Appendix B).

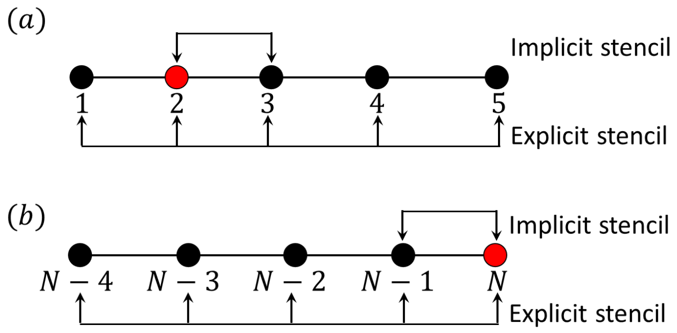

Figure 4. Stencils at boundary points. (a) $i = 2$; (b) $i = N$.

D. CCD Calculation

We can apply the interior scheme at nodes $i = 3$ to $i = N-1$, and the boundary formulation is used at nodes $i = 2$ and $i = N$. The resulting scheme takes the following matrix form

$$\begin{pmatrix} [B]_2 & [C]_2 & & & & & \\ [A]_3 & [B]_3 & [C]_3 & & & & \\ & \ddots & \ddots & \ddots & & & \\ & & [A]_i & [B]_i & [C]_i & & \\ & & & \ddots & \ddots & \ddots & \\ & & & & [A]_{N1} & [B]_{N1} & [C]_{N1} \\ & & & & & [A]_N & [B]_N \end{pmatrix} \begin{pmatrix} \{du\}_2 \\ \{du\}_3 \\ \vdots \\ \{du\}_i \\ \vdots \\ \{du\}_{N1} \\ \{du\}_N \end{pmatrix} = \begin{pmatrix} \{R\}_2 \\ \{R\}_3 \\ \vdots \\ \{R\}_i \\ \vdots \\ \{R\}_{N1} \\ \{R\}_N \end{pmatrix} \tag{9}$$

In the above system of equations $N1 = N - 1$. Each individual element of the left-hand side matrix is a (2×2) block matrix. The first (Eq. 10a) and second derivative (Eq. 10b) stencils for the interior of the domain are given by

$$[A]_i = \begin{bmatrix} \alpha_{i,1}^{(1)} & \alpha_{i,3}^{(1)} \\ \beta_{i,1}^{(1)} & \beta_{i,4}^{(1)} \end{bmatrix}; \quad [B]_i = \begin{bmatrix} 1 & \alpha_{i,4}^{(1)} \\ \beta_{i,2}^{(1)} & 1 \end{bmatrix}; \quad [C]_i = \begin{bmatrix} \alpha_{i,2}^{(1)} & \alpha_{i,5}^{(1)} \\ \beta_{i,3}^{(1)} & \beta_{i,5}^{(1)} \end{bmatrix}, \tag{10a}$$

$$[A]_i = \begin{bmatrix} \alpha_{i,1}^{(2)} & \alpha_{i,3}^{(2)} \\ \beta_{i,1}^{(2)} & \beta_{i,4}^{(2)} \end{bmatrix}; \quad [B]_i = \begin{bmatrix} 1 & 0 \\ \beta_{i,2}^{(2)} & 1 \end{bmatrix}; \quad [C]_i = \begin{bmatrix} \alpha_{i,2}^{(2)} & \alpha_{i,4}^{(2)} \\ \beta_{i,3}^{(2)} & \beta_{i,5}^{(2)} \end{bmatrix}. \tag{10b}$$

Elements along the $i - th$ row of vector $\{R\}_i$ are given by

$$\{R\}_i = \begin{bmatrix} c_{i,1}^{(1)}\phi_{i-2} + c_{i,2}^{(1)}\phi_{i-1} + c_{i,3}^{(1)}\phi_i \\ d_{i,1}^{(1)}\phi_{i-1} + d_{i,2}^{(1)}\phi_i + d_{i,3}^{(1)}\phi_{i+1} \end{bmatrix}, \tag{11a}$$

$$\{R\}_i = \begin{bmatrix} c_{i,1}^{(2)}\psi_{i-1} + c_{i,2}^{(2)}\psi_i + c_{i,3}^{(2)}\psi_{i+1} \\ d_{i,1}^{(2)}\psi_{i-1} + d_{i,2}^{(2)}\psi_i + d_{i,3}^{(2)}\psi_{i+1} \end{bmatrix}, \tag{11b}$$

where Eq. 11a is used for the first derivative (see Eq. 5) and Eq. 11b for the second derivative (see Eq. 6). The unknown vector $\{du\}_i$ can be written as

$$\{du\}_i = \begin{bmatrix} \phi_i' \\ \phi_{ii}'' \end{bmatrix}. \tag{12a}$$

$$\{du\}_i = \begin{bmatrix} \psi_i' \\ \psi_{ii}'' \end{bmatrix}. \tag{12b}$$

The boundary stencils given in Eqs. 7 & 8 are included in the modified (2×2) block matrices at $i = 2$ and $i = N$ to form a complete system. The above system can be solved very efficiently using Block Tridiagonal Matrix Algorithm (Block TDMA).

E. Fourier Analysis of the Differencing Errors

The spectral resolution characteristics of the CCD schemes can be studied by computing the dispersion and dissipation errors using a Fourier analysis and the notion of "modified wavenumber" (see Lele[22] and Mahesh[20]). For the purpose of Fourier analysis, consider a test function on a periodic domain of length 2π as

$$u_j = e^{ikx_j} \tag{13}$$

It is convenient to introduce the scaled wave number $\omega = kh$ where $h = \max\left(x_{j+1} - x_j, x_j - x_{j-1}, x_{j-1} - x_{j-2}\right)$. The exact first and second derivatives of this trial function at nodes x_j are

$$u_j' = \frac{i\omega}{h} e^{i\omega\frac{x_j}{h}}, \tag{14a}$$

$$u_j'' = -\left(\frac{\omega}{h}\right)^2 e^{i\omega\frac{x_j}{h}}. \tag{14b}$$

The numerical approximations of the derivatives obtained from the difference schemes can be written as

$$\left(u_j'\right)_{fd} = \frac{i\omega'}{h} e^{i\omega\frac{x_j}{h}}, \tag{15a}$$

$$\left(u_j''\right)_{fd} = -\left(\frac{\omega''}{h}\right)^2 e^{i\omega\frac{x_j}{h}}, \tag{15b}$$

where $\omega' = \omega'(\omega)$ and $\omega'' = \omega''(\omega)$. The difference between $\omega'(\omega)$ and ω, and $\omega''^2(\omega)$ and ω^2 is then a measure of the error in the first and second derivative approximations, respectively. The effect of grid non-uniformity are examined by considering a simple stretched grid given by

$$x_j = x_{j-1} + \Delta r^{j-2} \tag{16}$$

Where $\Delta = x_2 - x_1$ and r is the stretching constant such that:

- $r = 1$: uniform grid,
- $r > 1$: increasingly coarser grid, and
- $r < 1$: increasingly finer grid.

1. First derivative

Applying the first derivative scheme given in Eq. 5 to the test function (Eq. 13) leads to the finite difference scheme in Fourier space:

$$\begin{bmatrix} \left(\alpha_{i,1}^{(1)} e^{-i\omega\alpha_b} + 1 + \alpha_{i,2}^{(1)} e^{i\omega\alpha_f}\right) & -\left(\alpha_{i,3}^{(1)} e^{-i\omega\alpha_b} + \alpha_{i,4}^{(1)} + \alpha_{i,5}^{(1)} e^{i\omega\alpha_f}\right) \\ \left(\beta_{i,1}^{(1)} e^{-i\omega\alpha_b} + \beta_{i,2}^{(1)} + \beta_{i,3}^{(1)} e^{i\omega\alpha_f}\right) & -\left(\beta_{i,4}^{(1)} e^{-i\omega\alpha_b} + 1 + \beta_{i,5}^{(1)} e^{i\omega\alpha_f}\right) \end{bmatrix} \begin{bmatrix} \frac{i\omega'}{h} \\ \left(\frac{\omega''}{h}\right)^2 \end{bmatrix} = \begin{bmatrix} c_{i,1}^{(1)} e^{-i\omega\alpha} + c_{i,2}^{(1)} e^{-i\omega\alpha_b} + c_{i,3}^{(1)} \\ d_{i,1}^{(1)} e^{-i\omega\alpha_b} + d_{i,2}^{(1)} + d_{i,3}^{(1)} e^{i\omega\alpha_f} \end{bmatrix} \tag{17}$$

where α_f, α_b, and α are given by

$$\alpha_f = \frac{x_{j+1} - x_j}{h}, \tag{18a}$$

$$\alpha_b = \frac{x_j - x_{j-1}}{h}, \tag{18b}$$

$$\alpha = \frac{x_j - x_{j-1}}{h}. \tag{18c}$$

The modified wavenumber for the first derivative, $\omega'(\omega)$, can now be found by solving Eq. 17. Figure 5 shows the real and imaginary parts of $\omega'(\omega) = k'h$ for different grids. The dispersive and dissipative errors are given by the real and the imaginary part of $(k' - k)$, respectively. For comparison, the standard 6th-order compact scheme from Lele[22] is included in the figure. Although the upwind CCD stencil is smaller, Fig. 5 shows that the CCD scheme exhibits better spectral resolution than Lele's[22] sixth-order scheme at high wave numbers.

Central compact schemes with spectral-like resolution are also prone to aliasing error. Thus, while designing higher-order accurate schemes, one should pay close attention to control aliasing error. Fig. 5 (right) illustrates that the present upwind CCD scheme has non-zero dissipation error that decreases for stretched grids. One should note that the numerical diffusion is restricted to the high wavenumber region $kh > 3\pi/4$. It is worth mentioning that the modified wavenumber kh is equal to the number of points per wavelength. Therefore, for waves with 4 or more points per wavelength, the proposed upwind CCD scheme has zero dissipation error.

2. Second derivative

Application of the second derivative approximation given by Eq. 6 to the test function yields the following (2×2) system of equations in Fourier space

$$\begin{bmatrix} \left(\alpha_{i,1}^{(2)} e^{-i\omega\alpha_b} + 1 + \alpha_{i,2\ i}^{(2)} e^{i\omega\alpha_f}\right) & -\left(\alpha_{i,3}^{(2)} e^{-i\omega\alpha_b} + \alpha_{i,4}^{(2)} e^{i\omega\alpha_f}\right) \\ \left(\beta_{i,1}^{(2)} e^{-i\omega\alpha_b} + \beta_{i,2}^{(2)} + \beta_{i,3}^{(2)} e^{i\omega\alpha_f}\right) & -\left(\beta_{i,4}^{(2)} e^{-i\omega\alpha_b} + 1 + \beta_{i,5}^{(2)} e^{i\omega\alpha_f}\right) \end{bmatrix} \begin{bmatrix} \frac{i\omega'}{h} \\ \left(\frac{\omega''}{h}\right)^2 \end{bmatrix} = \begin{bmatrix} c_{i,1}^{(2)} e^{-i\omega\alpha_b} + c_{i,2}^{(2)} + c_{i,3}^{(2)} e^{i\omega\alpha_f} \\ d_{i,1}^{(2)} e^{-i\omega\alpha_b} + d_{i,2}^{(2)} + d_{i,3}^{(2)} e^{i\omega\alpha_f} \end{bmatrix} \tag{19}$$

The real (dispersion) and imaginary (dissipation) parts of k''^2h^2 are plotted versus kh for various stretched grids in Fig. 6. The plots show that the sixth-order CCD scheme has better spectral resolution than Lele's[22] scheme. Recall that the CCD scheme has the same stencil width as the fourth-order Padé.

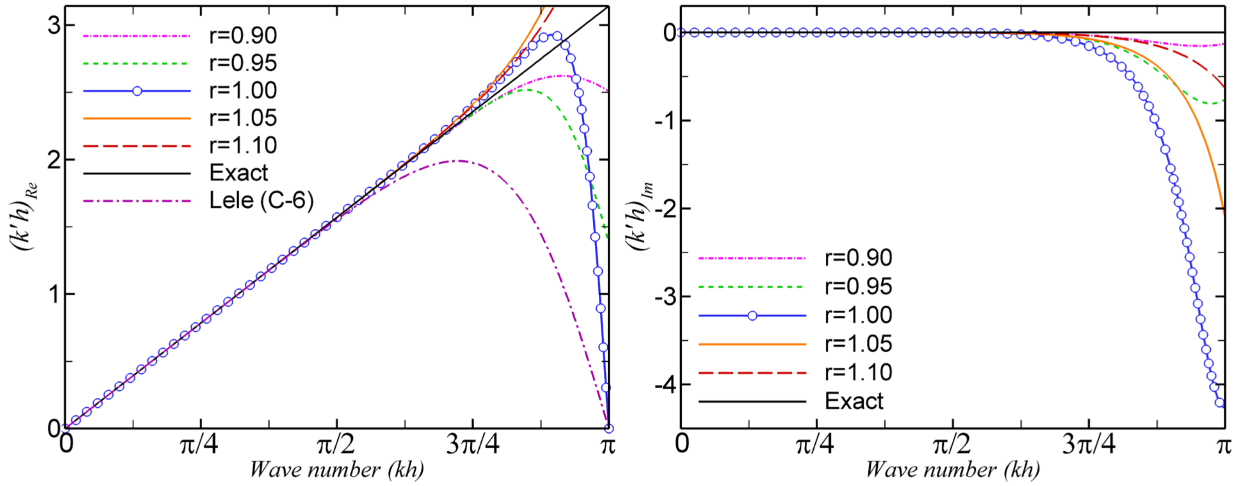

Figure 5. Comparison of modified wavenumber for first derivative for various grid stretching parameter, r. Left: Real part of ω'; right: Imaginary part of ω'.

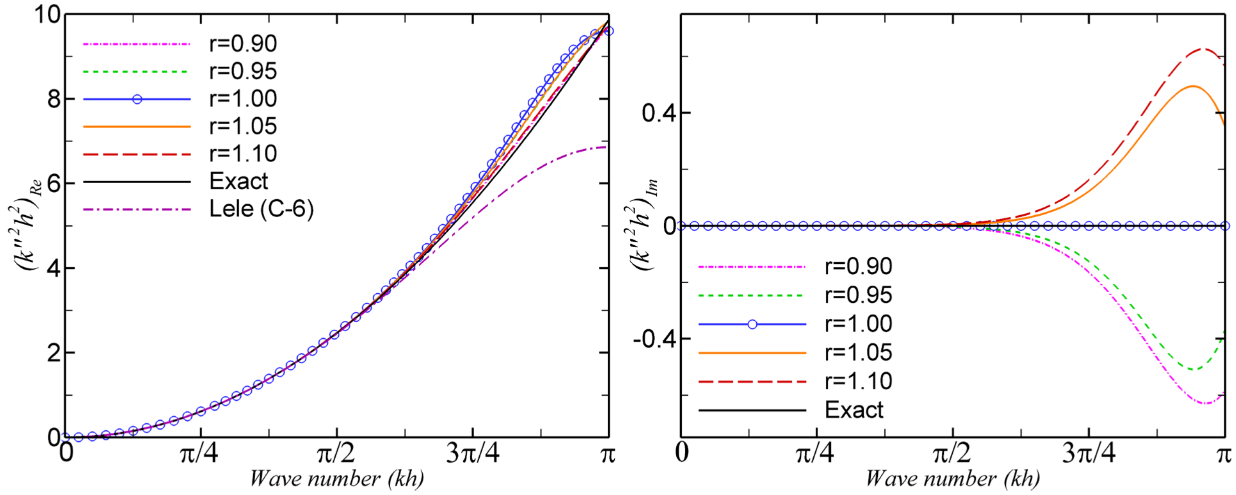

Figure 6. Comparison of modified wavenumber for second derivative for various grid stretching parameter, r. Left: Real part of ω''^2; right: imaginary part of ω''^2.

III. High-Order Sharp Interface Immersed Method

The key idea of the IIM is that when the interface passes through the difference stencil, the difference scheme needs to be modified by incorporating the jump condition for the variable and its derivatives. This section outlines the process for obtaining the jump corrections and jump-corrected CCD schemes.

A. Corrected Taylor Series

Taylor expansions are not valid when applied to non-smooth or discontinuous functions. A correction term needs to be added at any point where the function or its derivatives have a jump discontinuity.

Lemma 1 *suppose that the function u(x) is analytic everywhere except at the interface* x_α *where it lies between two arbitrary grid points* x_i *and* x_{i+1}, $x_i < x_\alpha < x_{i+1}$. *Let* $dx = x_{i+1} - x_i$ *and* $h^+ = x_{i+1} - x_\alpha$, *then we can write the following corrected Taylor series to expand* $u(x_{i+1})$ *about* $u(x_i)$*:*

$$u(x_{i+1}) = u(x_i) + \Delta x\, u'(x_i) + \frac{\Delta x^2}{2!}u''(x_i) + \cdots + \frac{\Delta x^p}{p!}u^{(p)}(x_i) + J_{\alpha 0} + O(\Delta x^{p+1})\,, \tag{20}$$

The proof of Lemma 1 can be found in Wiegmann & Bube.[15] The correction term $J_{\alpha 0}$ is defined as

$$J_{\alpha 0} = [u]_\alpha + h^+ \, [u']_\alpha + \frac{(h^+)^2}{2!} [u'']_\alpha + \cdots + \frac{(h^+)^p}{p!} \left[u^{(p)}\right]_\alpha . \tag{21}$$

The term $[u]_\alpha$ represents the jump in the value of u at $x = x_\alpha$ that is

$$[u]_\alpha = \lim_{x \to x_\alpha^+} u(x) - \lim_{x \to x_\alpha^-} u(x) , \tag{22}$$

$[u']_\alpha$ is the jump in the value of the function first derivative, and so on. We can use Lemma 1 to derive the Taylor corrected series for the first and second derivatives of $u(x_{i+1})$ as

$$u'(x_{i+1}) = u'(x_i) + \Delta x \, u''(x_i) + \frac{\Delta x^2}{2!} u^{(3)}(x_i) + \cdots + \frac{\Delta x^{(p-1)}}{(p-1)!} u^{(p)}(x_i) + J_{\alpha 1} + O(\Delta x^p) , \tag{23a}$$

$$u''(x_{i+1}) = u''(x_i) + \Delta x \, u^{(3)}(x_i) + \frac{\Delta x^2}{2!} u^{(4)}(x_i) + \cdots + \frac{\Delta x^{(p-2)}}{(p-2)!} u^{(p)}(x_i) + J_{\alpha 2} + O(\Delta x^p) , \tag{23b}$$

where $J_{\alpha 1}$ and $J_{\alpha 2}$ are given by

$$J_{\alpha 1} = [u']_\alpha + h^+ \, [u'']_\alpha + \cdots + \frac{(h^+)^{p-1}}{(p-1)!} \left[u^{(p)}\right]_\alpha , \tag{24a}$$

$$J_{\alpha 2} = [u'']_\alpha + h^+ \left[u^{(3)}\right]_\alpha + \cdots + \frac{(h^+)^{p-2}}{(p-2)!} \left[u^{(p)}\right]_\alpha . \tag{24b}$$

B. Computing the Jump Correction Terms

If the interface occurs at $x_i < x_\alpha < x_{i+1}$, then one has $h = x_{i+1} - x_\alpha$

$$J_{\alpha k} = \sum_{n=k}^{N_d} \frac{h^{(n-k)}}{(n-k)!} \left[u^{(n)}\right]_\alpha . \tag{25}$$

If the interface occurs at $x_{i-1} < x_\alpha < x_i$, then one has $h = x_\alpha - x_{i-1}$

$$J_{\alpha k} = \sum_{n=k}^{N_d} (-1)^{(n-k+1)} \frac{h^{(n-k)}}{(n-k)!} \left[u^{n)}\right]_\alpha . \tag{26}$$

In Eqs. 25 and 26, k corresponds to the correction in *kth*-derivative ($k = 0$, 1 and 2) and N_d is the highest derivative that has been sought for computing the jump correction. The jump in the *nth*-derivative of the function is computed as

$$\left[u^{(n)}\right]_\alpha = u^{(n)}_{F.D^+} - u^{(n)}_{F.D^-} , \tag{27}$$

where

$$\begin{cases} u^{(n)}_{F.D^+} = c_0^{(n)} u_\alpha^+ + \displaystyle\sum_{k=1}^{N_p} c_k^{(n)} u_{i+k-1} , & \theta_{cr} < \theta^+ < 1.0 \\ u^{(n)}_{F.D^+} = c_0^{(n)} u_\alpha^+ + \displaystyle\sum_{k=1}^{N_p} c_k^{(n)} u_{i+k} , & 0.0 < \theta^+ \leq \theta_{cr} \end{cases} \tag{28a}$$

$$\begin{cases} u^{(n)}_{F.D^-} = c^{(n)}_0 u^-_\alpha + \sum_{k=1}^{N_p} c^{(n)}_k u_{i-k+1}, & \theta_{cr} < \theta^- < 1.0 \\ u^{(n)}_{F.D^-} = c^{(n)}_0 u^-_\alpha + \sum_{k=1}^{N_p} c^{(n)}_k u_{i-k}, & 0.0 < \theta^- \leq \theta_{cr} \end{cases} \qquad (28b)$$

Here N_p is the number of points used to compute the one-sided nth-derivative and $N_p \geq N_d$. The $+$ and $-$ superscripts indicate respectively right and left limits at x_α. The parameters θ^+ and $\theta^-$are

$$\theta^+ = \frac{(x_i - x_\alpha)}{(x_i - x_{i-1})}, \qquad (29a)$$

$$\theta^- = \frac{(x_\alpha - x_i)}{(x_{i+1} - x_i)}. \qquad (29b)$$

To maintain the order of accuracy, each of the one-sided stencils used to compute the jump correction terms in Eqs. 25 & 26 contain at least five points ($N_p \geq 5, N_d = 5$). The pre-specified critical ratio, θ_{cr}, is used to decide whether a grid point adjacent to the interface x_α can be used to compute the jump correction terms and because of this, the time step restriction has been avoided. In the present work, θ_{cr} is selected to be $0.5 \leq \theta_{cr} \leq 1$.

C. Jump Corrected Combined Compact Differences

As explained in section II, we use combined compact difference schemes to compute spatial derivatives. If we consider a line $y = const$ along which one would like to numerically approximate spatial derivatives and if that line passes through the immersed boundary, then two different scenarios are possible: (i) $x_i < x_\alpha < x_{i+1}$ and (ii) $x_{i-1} < x_\alpha < x_i$.

1. Case (i)

If the interface location is located between x_i and x_{i+1}, $x_i < x_\alpha < x_{i+1}$, as shown in Fig. 7, then the first derivative is approximated using

$$\alpha^{(1)}_{i,1}\phi'_{i-1} + \phi'_i + \alpha^{(1)}_{i,2}\phi'_{i+1} + \alpha^{(1)}_{i,3}\phi''_{i-1} + \alpha^{(1)}_{i,4}\phi''_i + \alpha^{(1)}_{i,5}\phi''_{i+1} = c^{(1)}_{i,1}\phi_{i-2} + c^{(1)}_{i,2}\phi_{i-1} + c^{(1)}_{i,3}\phi_i + \Gamma^{(1)}_\alpha, \qquad (30a)$$

$$\beta^{(1)}_{i,1}\phi'_{i-1} + \beta^{(1)}_{i,2}\phi'_i + \beta^{(1)}_{i,3}\phi'_{i+1} + \beta^{(1)}_{i,4}\phi''_{i-1} + \phi''_i + \beta^{(1)}_{i,5}\phi''_{i+1} = d^{(1)}_{i,1}\phi_{i-1} + d^{(1)}_{i,2}\phi_i + d^{(1)}_{i,3}\phi_{i+1} + \Gamma^{(1)}_{\alpha\alpha}, \qquad (30b)$$

$$\Gamma^{(1)}_\alpha = \left[\alpha^{(1)}_{i,2}J_{\alpha 1} + \alpha^{(1)}_{i,5}J_{\alpha 2}\right], \qquad (30c)$$

$$\Gamma^{(1)}_{\alpha\alpha} = \left[\beta^{(1)}_{i,3}J_{\alpha 1} + \beta^{(1)}_{i,5}J_{\alpha 2} - d^{(1)}_{i,3}J_{\alpha 0}\right]. \qquad (30d)$$

And the second derivative is evaluated from the following equations

$$\alpha^{(2)}_{i,1}\psi'_{i-1} + \psi'_i + \alpha^{(2)}_{i,2}\psi'_{i+1} + \alpha^{(2)}_{i,3}\psi''_{i-1} + \alpha^{(2)}_{i,4}\psi''_{i+1} = c^{(2)}_{i,1}\psi_{i-1} + c^{(2)}_{i,2}\psi_i + c^{(2)}_{i,3}\psi_{i+1} + \Gamma^{(2)}_\alpha, \qquad (31a)$$

$$\beta^{(2)}_{i,1}\psi'_{i-1} + \beta^{(2)}_{i,2}\psi'_i + \beta^{(2)}_{i,3}\psi'_{i+1} + \beta^{(2)}_{i,4}\psi''_{i-1} + \psi''_i + \beta^{(2)}_{i,5}\psi''_{i+1} = d^{(2)}_{i,1}\psi_{i-1} + d^{(2)}_{i,2}\psi_i + d^{(2)}_{i,3}\psi_{i+1} + \Gamma^{(2)}_{\alpha\alpha}, \qquad (31b)$$

$$\Gamma^{(2)}_\alpha = \left[\alpha^{(2)}_{i,2}J_{\alpha 1} + \alpha^{(2)}_{i,4}J_{\alpha 2} - c^{(2)}_{i,3}J_{\alpha 0}\right], \qquad (31c)$$

$$\Gamma^{(2)}_{\alpha\alpha} = \left[\beta^{(2)}_{i,3}J_{\alpha 1} + \beta^{(3)}_{i,5}J_{\alpha 2} - d^{(3)}_{i,3}J_{\alpha 0}\right]. \qquad (31d)$$

The jump correction terms in the above equations are determined using the schemes presented in the previous section.

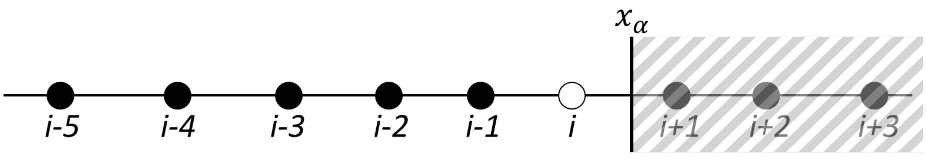

Figure 7. Schematic of a finite difference stencil at an irregular grid point where $x_i < x_\alpha < x_{i+1}$.

2. Case (ii)

In this case, the interface location is located between x_{i-1} and x_i, $x_{i-1} < x_\alpha < x_i$ (see Fig. 8). The process to obtain the first derivative at nodes i and j is outline here. At node i, we have

$$\alpha^{(1)}_{i,1}\phi'_{i-1} + \phi'_i + \alpha^{(1)}_{i,2}\phi'_{i+1} + \alpha^{(1)}_{i,3}\phi''_{i-1} + \alpha^{(1)}_{i,4}\phi''_i + \alpha^{(1)}_{i,5}\phi''_{i+1} = c^{(1)}_{i,1}\phi_{i-1} + c^{(1)}_{i,2}\phi_i + c^{(1)}_{i,3}\phi_{i+1} + \Gamma^{(1)}_\alpha, \tag{32a}$$

$$\beta^{(1)}_{i,1}\phi'_{i-1} + \beta^{(1)}_{i,2}\phi'_i + \beta^{(1)}_{i,3}\phi'_{i+1} + \beta^{(1)}_{i,4}\phi''_{i-1} + \phi''_i + \beta^{(1)}_{i,5}\phi''_{i+1} = d^{(1)}_{i,1}\phi_{i-1} + d^{(1)}_{i,2}\phi_i + d^{(1)}_{i,3}\phi_{i+1} + \Gamma^{(1)}_{\alpha\alpha}\,, \tag{32b}$$

$$\Gamma^{(1)}_\alpha = \left[\alpha^{(1)}_{i,1}J_{\alpha 1} + \alpha^{(1)}_{i,3}J_{\alpha 2} - c^{(1)}_{i,1}J_{\alpha 0}\right], \tag{32c}$$

$$\Gamma^{(1)}_{\alpha\alpha} = \left[\beta^{(1)}_{i,1}J_{\alpha 1} + \beta^{(1)}_{i,4}J_{\alpha 2} - d^{(1)}_{i,1}J_{\alpha 0}\right]. \tag{32d}$$

At node j

$$\alpha^{(1)}_{j,1}\phi'_{j-1} + \phi'_j + \alpha^{(1)}_{j,2}\phi'_{j+1} + \alpha^{(1)}_{j,3}\phi''_{j-1} + \alpha^{(1)}_{j,4}\phi''_j + \alpha^{(1)}_{j,5}\phi''_{j+1} = c^{(1)}_{j,1}\phi_{j-2} + c^{(1)}_{j,2}\phi_{j-1} + c^{(1)}_{i,3}\phi_j - \left[c^{(1)}_{j,1}J_{\alpha 0}\right], \tag{33a}$$

$$\beta^{(1)}_{j,1}\phi'_{j-1} + \beta^{(1)}_{j,2}\phi'_j + \beta^{(1)}_{j,3}\phi'_{j+1} + \beta^{(1)}_{j,4}\phi''_{j-1} + \phi''_j + \beta^{(1)}_{j,5}\phi''_{j+1} = d^{(1)}_{j,1}\phi_{j-1} + d^{(1)}_{j,2}\phi_j + d^{(1)}_{j,3}\phi_{j+1}\,. \tag{33b}$$

The second derivative is obtained at node i using

$$\alpha^{(2)}_{i,1}\psi'_{i-1} + \psi'_i + \alpha^{(2)}_{i,2}\psi'_{i+1} + \alpha^{(2)}_{i,3}\psi''_{i-1} + \alpha^{(2)}_{i,4}\psi''_{i+1} = c^{(2)}_{i,1}\psi_{i-1} + c^{(2)}_{i,2}\psi_i + c^{(2)}_{i,3}\psi_{i+1} + \Gamma^{(2)}_\alpha, \tag{34a}$$

$$\beta^{(2)}_{i,1}\psi'_{i-1} + \beta^{(2)}_{i,2}\psi'_i + \beta^{(2)}_{i,3}\psi'_{i+1} + \beta^{(2)}_{i,4}\psi''_{i-1} + \psi''_i + \beta^{(2)}_{i,5}\psi''_{i+1} = d^{(2)}_{i,1}\psi_{i-1} + d^{(2)}_{i,2}\psi_i + d^{(2)}_{i,3}\psi_{i+1} + \Gamma^{(2)}_{\alpha\alpha}, \tag{34b}$$

$$\Gamma^{(2)}_\alpha = \left[\alpha^{(2)}_{i,1}J_{\alpha 1} + \alpha^{(2)}_{i,3}J_{\alpha 2} - c^{(2)}_{i,1}J_{\alpha 0}\right], \tag{34c}$$

$$\Gamma^{(2)}_{\alpha\alpha} = \left[\beta^{(2)}_{i,1}J_{\alpha 1} + \beta^{(3)}_{i,4}J_{\alpha 2} - d^{(3)}_{i,1}J_{\alpha 0}\right]. \tag{34d}$$

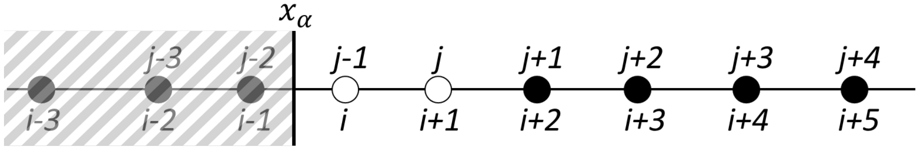

Figure 8. Schematic of a finite difference stencil at an irregular grid point where $x_{i-1} < x_\alpha < x_i$.

D. CCD Schemes for Interface-Matched-Grid Node

When the interface of the immersed boundary, x_α, falls exactly on a grid-point, the CCD scheme stencil needs to be adjusted accordingly and no jump correction will therefore be needed. If $x_\alpha = x_{i+1}$ and x_{i+2} falls inside the immersed boundary, as shown in Fig. 9, the following fifth-order formulations are used to evaluate the first derivative at node i

$$\alpha^{(1)}_{i,1}\phi'_{i-1} + \phi'_i + \alpha^{(1)}_{i,2}\phi''_{i-1} = c^{(1)}_{i,1}\phi_{i+1} + c^{(1)}_{i,2}\phi_i + c^{(1)}_{i,3}\phi_{i-1} + c^{(1)}_{i,4}\phi_{i-2}, \tag{35a}$$

$$\beta^{(1)}_{i,1}\phi'_{i-1} + \beta^{(1)}_{i,4}\phi''_{i-1} + \phi''_i = d^{(1)}_{i,1}\phi_{i+1} + d^{(1)}_{i,2}\phi_i + d^{(1)}_{i,3}\phi_{i-1} + d^{(1)}_{i,4}\phi_{i-2} + d^{(1)}_{i,5}\phi_{i-3}\,. \tag{35b}$$

The second derivative is approximated at node i using the following fifth-order scheme

$$\alpha^{(2)}_{i,1}\psi'_{i-1} + \psi'_i + \alpha^{(2)}_{i,4}\psi''_{i+1} = c^{(2)}_{i,1}\psi_{i+1} + c^{(2)}_{i,2}\psi_i + c^{(2)}_{i,3}\psi_{i-1} + c^{(2)}_{i,4}\psi_{i-2}, \tag{36a}$$

$$\beta^{(2)}_{i,1}\psi'_{i-1} + \beta^{(2)}_{i,4}\psi''_{i-1} + \psi''_i = d^{(2)}_{i,1}\psi_{i+1} + d^{(2)}_{i,2}\psi_i + d^{(2)}_{i,3}\psi_{i-1} + d^{(2)}_{i,4}\psi_{i-2} + d^{(2)}_{i,5}\psi_{i-3}. \tag{36b}$$

Note that we do not evaluate the derivatives at node $i + 1$ since a Dirichlet boundary condition was used there.

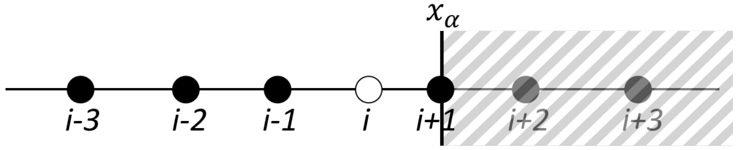

Figure 9. Computational setup near the interface location for $x_\alpha = x_{i+1}$.

When the interface of the immersed boundary coincides with $x_\alpha = x_{i-1}$ and x_{i-2} is inside the immersed boundary (see Fig. 10), the first derivative at node i is obtained from the following expressions

$$\phi_i' + \alpha_{i,1}^{(1)}\phi_{i+1}' + \alpha_{i,2}^{(1)}\phi_{i+1}'' = c_{i,1}^{(1)}\phi_{i-1} + c_{i,2}^{(1)}\phi_i + c_{i,3}^{(1)}\phi_{i+1} + c_{i,4}^{(1)}\phi_{i+2}, \tag{37a}$$

$$\phi_i'' = d_{i,1}^{(1)}\phi_{i-1} + d_{i,2}^{(1)}\phi_i + d_{i,3}^{(1)}\phi_{i+1} + d_{i,4}^{(1)}\phi_{i+2} + d_{i,5}^{(1)}\phi_{i+3} + d_{i,6}^{(1)}\phi_{i+4}, \tag{37b}$$

And the second derivative at node i is evaluated using

$$\psi_i' + \alpha_{i,1}^{(2)}\psi_{i+1}' + \alpha_{i,4}^{(2)}\psi_{i+1}'' = c_{i,1}^{(2)}\psi_{i-1} + c_{i,2}^{(2)}\psi_i + c_{i,3}^{(2)}\psi_{i+1} + c_{i,4}^{(2)}\psi_{i+2}, \tag{38a}$$

$$\beta_{i,1}^{(2)}\psi_{i+1}' + \psi_i'' + \beta_{i,4}^{(2)}\psi_{i+1}'' = d_{i,1}^{(2)}\psi_{i-1} + d_{i,2}^{(2)}\psi_i + d_{i,3}^{(2)}\psi_{i+1} + d_{i,4}^{(2)}\psi_{i+2} + d_{i,5}^{(2)}\psi_{i+3}, \tag{38b}$$

The first and second derivatives are fifth-order-accurate and the coefficients for Eqs.35-38 are derived in appendix C.

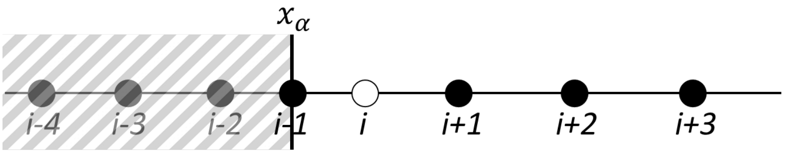

Figure 10. Computational setup near the interface location for $x_\alpha = x_{i-1}$.

E. Validation of IIM: Analysis of Jump Corrected CCD Scheme

The feasibility of combining the CCD schemes and the high-order IIM is investigated in this section. The computational domain is from 0 to 2π with N nodes non-uniformly distributed according to Eq. 3 with $\alpha = 0.6$. The test function described in section II(A) is used to investigate the IIM and its effect on the order of the spatial discretization.

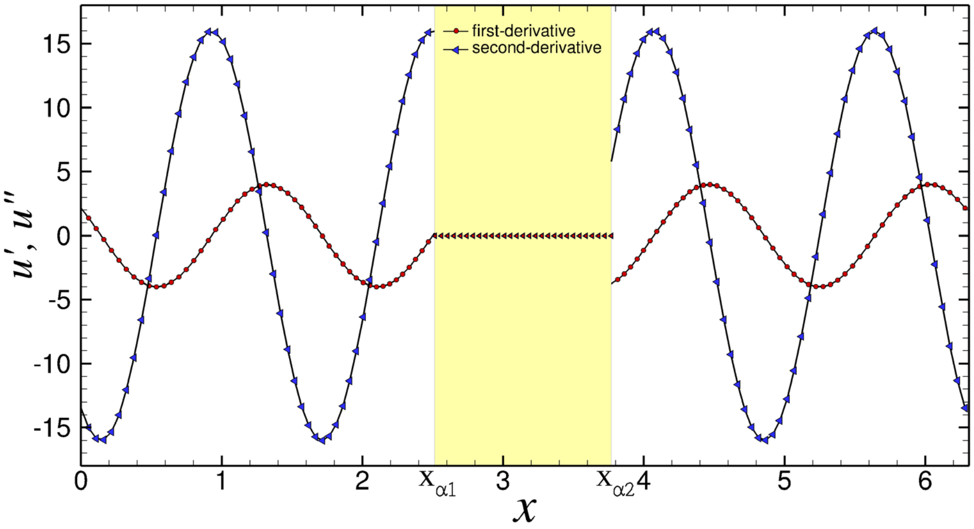

Figure 11. Comparison of numerical vs exact solution for first and second derivatives. Sold lines represent the exact solutions and symbols are for numerical approximations with $N = 101$ points.

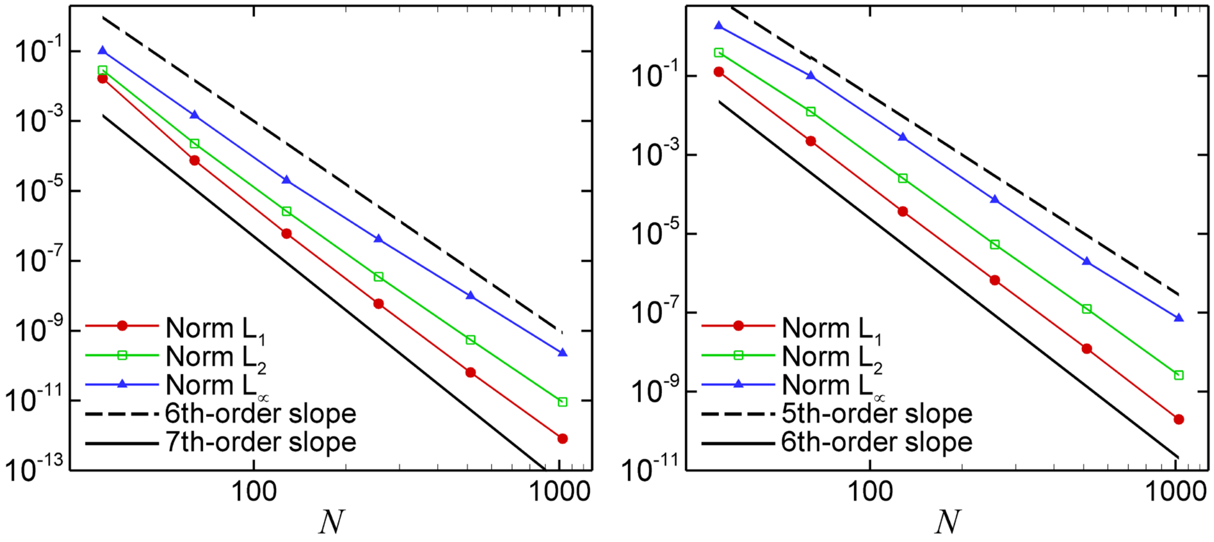

Figure 12. Error study of the spatial operator using CCD schemes and IIM. Left: First derivative, right: Second derivative.

The interface locations are fixed at $x_{\alpha 1} = 0.8\pi$ and $x_{\alpha 2} = 1.2\pi$ as shown in Fig. 11. The solution is identically set to zero, $u = 0$ for $x_{\alpha 1} < x < x_{\alpha 2}$. At interior points, the first derivative is approximated using a seventh-order upwind CCD scheme as discussed in section II(A) while a sixth-order CCD scheme is used for the second derivative. At irregular points, the jump corrected CCD schemes are implemented.

An order study of the discrete spatial operator is presented in Fig. 12. It demonstrates that the IIM has negligible effect on the order of the spatial discretization.

IV. Stability Analysis: Semi- and Fully-Discrete Eigenvalue Problem

In most investigations of IBMs and IIMs, the discretization error of the scheme is generally discussed in detail. However, the stability characteristics of the scheme, which is just as important, is often not investigated specifically. Numerical simulations of the Navier-Stokes equations require long-time integration and for this reason the numerical schemes must be strictly stable in time. The numerical stability of the finite difference schemes with IIM and boundary closures can be analyzed through the eigenvalue spectrum of the spatial discretization matrix (semi-discrete) and also, more thoroughly, by analyzing the eigenvalue spectrum of the combined spatio-temporal operator (fully-discrete).

The approach is demonstrated for the following one-dimensional linear PDE

$$\frac{\partial u}{\partial t} = [\mathcal{L}]\{u\} + \{B\}, \tag{39}$$

where $[\mathcal{L}]$ could represent the linear combination of the first and second derivative discretization matrices. In the case of semi-discrete approximations, strict stability implies that for a fixed mesh size dx, all eigenvalues of the spatial discretization matrix of the corresponding system of ordinary differential equations must have a non-positive real part. This is a necessary condition but it is not always sufficient. For a sufficient condition, one has to investigate the eigenvalue spectrum of the combined spatio-temporal operator (fully discrete). In the fully-discrete approach, stability requires that all eigenvalues of the full discretization matrix lie inside the unit circle.

The upwind CCD scheme proposed for the first derivative can be implemented in matrix form as

$$[A_1]\{u'\} + [B_1]\{u''\} = [R_1]\{u\} + \{C_1\}\,, \tag{40a}$$

$$[A_2]\{u'\} + [B_2]\{u''\} = [R_2]\{u\} + \{C_2\}\,, \tag{40b}$$

where $[A_1]$, $[B_1]$, $[A_2]$, $[B_2]$, $[R_1]$, and $[R_2]$ are the coefficient matrices of the upwind CCD scheme and $\{C_1\}$ and $\{C_2\}$ represent the vectors of boundary values. Note that the effect of the IIM (cf. jump corrections) are included in $[R_1]$ and $[R_2]$. After some math, the discretization matrix for the first derivative can be expressed as

$$\{u'\} = [M_1]\{u\} + \{G_1\}\,, \tag{41}$$

where

$$[M_1] = [ML_1^{-1}MR_1]\,, \tag{42a}$$

$$\{G_1\} = [ML_1^{-1}]\{C_1 - B_1B_2^{-1}C_2\}\,, \tag{42b}$$

$$[ML_1] = [A_1 - B_1B_2^{-1}A_2]\,, \tag{42c}$$

$$[MR_1] = [R_1 - B_1B_2^{-1}R_2]\,, \tag{42d}$$

The central CCD scheme proposed for the second derivative can be implemented in matrix form as follows

$$[\overline{A_1}]\{u'\} + [\overline{B_1}]\{u''\} = [\overline{R_1}]\{u\} + \{\overline{C_1}\}\,, \tag{43a}$$

$$[\overline{A_2}]\{u'\} + [\overline{B_2}]\{u''\} = [\overline{R_2}]\{u\} + \{\overline{C_2}\}\,. \tag{43b}$$

The spatial second derivative CCD scheme for either uniform or non-uniform grids can be formally written in a general matrix form, as

$$\{u''\} = [M_2]\{u\} + \{G_2\}\,, \tag{44}$$

where

$$[M_2] = [ML_2^{-1}MR_2]\,, \tag{45a}$$

$$\{G_2\} = [ML_2^{-1}]\left\{\overline{C_2} - \overline{A_2}\,\overline{A_1}^{-1}\overline{C_1}\right\}, \tag{45b}$$

$$[ML_2] = \left[\overline{B_2} - \overline{A_2}\,\overline{A_1}^{-1}\overline{B_1}\right], \tag{45c}$$

$$[MR_2] = \left[\overline{R_2} - \overline{A_2}\,\overline{A_1}^{-1}\overline{R_1}\right], \tag{45d}$$

The operator $[\mathcal{L}]$ in Eq. 39 represents the linear combination of matrices $[M_1]$ and $[M_2]$. The stability analysis is investigated for one-dimensional advection, diffusion and advection-diffusion equations for a domain of length 2π. The equations are discretized on a uniform grid using $N = 101$ points. Interior points are discretized using the CCD schemes discussed in section II(A) with fourth-order boundary schemes for the first and second derivatives applied at $i = 2$ and $i = N$. The interfaces locations are fixed at $x = x_{\alpha1}$ and $x = x_{\alpha2}$ according to Eq. 46 where θ_1 & θ_2 determine the distance between each interface and the irregular grid points, $i = 41$ and $i = 61$ (see Fig. 13).

$$x_{\alpha1} = x_{41} + \theta_1(x_{42} - x_{41}), \tag{46a}$$

$$x_{\alpha2} = x_{61} - \theta_2(x_{61} - x_{60}). \tag{46b}$$

The solution is identically set to zero, $u = 0$ for $x_{\alpha1} < x < x_{\alpha2}$. At irregular points, jump corrected CCD schemes are implemented. The advection speed is kept constant and set equal to $c = 1$ while the diffusion coefficient is chosen to be $v = 0.001$. The stability analysis is performed for different interfaces distances from the irregular points, θ_1 & θ_2.

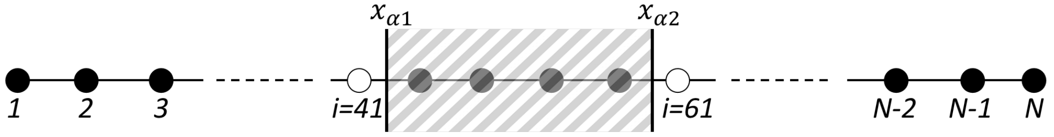

Figure 13. 1D schematic describing the interfaces locations and irregular grid points.

A. Advection Equation

The following discussion will focus on the stability of the IIM for pure advection equation which is a good model for inviscid flows.

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0\,. \tag{47}$$

Applying the first derivative discretization matrix, Eq. 41, to the advection equation leads to

$$\frac{\partial\{u\}}{\partial t} = -c\cdot[M_1]\{u\} - c\cdot\{G_1\}\,. \qquad (48$$

At this point, the time discretization has not been addressed yet, and only the semi-discrete problem is considered. Seeking normal mode solutions of the form $u = \hat{u}e^{\sigma t}$ to the homogeneous part of the above equation results in

$$\sigma\{\hat{u}\} = -c\cdot[M_1]\{\hat{u}\}\,. \tag{49}$$

Rearranging this equation yields,

$$\omega\{\hat{u}\} = -dx_{min}\cdot[M_1]\{\hat{u}\}\,, \tag{50}$$

where the eigenvalue $\omega = \sigma dx_{min}/c$. Stable numerical solutions will require that the real part of ω lies in the left half of the complex ω-plane, ($Re(\omega) \leq 0$). To consider the full-discretization scheme, the temporal discretization matrix must be combined with the spatial discretization matrix. In this paper, a fourth-order Runge-Kutta (RK4) method is chosen for time integration. The RK4 method can be written in the following manner for a linear differential equation:

$$\frac{\{u^{n+1}\}}{\{u^n\}} = \sum_{k=0}^{4}\frac{(\Delta t\cdot[Q])^k}{k!}\;, \tag{51}$$

where $\{u^n\}$ is the $(N\times1)$ vector of the variable at time n, and $[Q]$ is the spatial discretizaion matrix. Hence the full discretization matrix including the RK4 time integration scheme may be written in the following form

$$\frac{\{u^{n+1}\}}{\{u^n\}} = \sum_{k=0}^{4}\frac{(-\Delta t\cdot c\cdot[M_1])^k}{k!}\,, \tag{52a}$$

$$\frac{\{u^{n+1}\}}{\{u^n\}} = \sum_{k=0}^{4}\frac{(-CFL\cdot dx_{min}\cdot[M_1])^k}{k!} = [\mathcal{A}_a]\,, \tag{52b}$$

where $CFL = c\,\Delta t/dx_{min}$. For the numerical stability of the fully-discrete problem, the eigenvalues of the matrix $[\mathcal{A}_a]$ must lie inside the unit circle in the complex plane.

B. Diffusion Equation

A similar analysis can be carried out for the pure diffusion equation

$$\frac{\partial u}{\partial t} = \nu\frac{\partial^2 u}{\partial x^2}\,. \tag{53}$$

Applying the second derivative discretization matrix to the above equation yields

$$\frac{\partial\{u\}}{\partial t} = \nu\cdot[M_2]\{u\} + \nu\cdot\{G_2\}\,. \tag{54}$$

Similar to the advection equation, we look for normal mode solutions of the form $u = \hat{u}e^{\sigma t}$ to the homogeneous part of the equation, which results in

$$\sigma\{\hat{u}\} = \nu \cdot [M_2]\{\hat{u}\}\,. \tag{55}$$

After some manipulations, we get

$$\omega\{\hat{u}\} = dx_{min}^2[M_2]\{\hat{u}\}\,, \tag{56}$$

where $\omega = \sigma dx_{min}^2/\nu$. The necessary condition for stability is that these eigenvalues (cf. ω) must lie in the left half of the complex plane. For a fully discrete problem with a RK4 time advancement scheme, the resulting eigenvalue problem reduces to

$$\frac{\{u^{n+1}\}}{\{u^n\}} = \sum_{k=0}^{4} \frac{(\Delta t \cdot \nu \cdot [M_2])^k}{k!}\,, \tag{57a}$$

$$\frac{\{u^{n+1}\}}{\{u^n\}} = \sum_{k=0}^{4} \frac{(DFL \cdot dx_{min}^2 \cdot [M_2])^k}{k!} = [\mathcal{A}_d]\,, \tag{57b}$$

where $DFL = \nu\Delta t/dx_{min}^2$. Now a stability analysis can be carried out by extracting the eigenvalues of the matrix $[\mathcal{A}_d]$.

C. Advection-Diffusion Equation

Finally, an analysis of the combined advection-diffusion equation

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2}\,, \tag{58}$$

yields the system

$$\omega\{\hat{u}\} = (dx_{min}^2 \cdot [M_2] - Re \cdot dx_{min} \cdot [M_1])\{\hat{u}\}\,, \tag{59}$$

where $\omega = \sigma dx_{min}^2/\nu$ are the eigenvalues and $Re = c\ dx_{min}/\nu$ is the cell Reynolds number. The full discretization matrix is

$$\frac{\{u^{n+1}\}}{\{u^n\}} = \sum_{k=0}^{4} \frac{(DFL \cdot dx_{min}^2 \cdot [M_2] - CFL \cdot dx_{min} \cdot [M_1])^k}{k!} = [\mathcal{A}_{ad}]\,. \tag{60}$$

The calculated eigenvalues for the semi- and fully-discrete problem are shown in Fig. 14 for pure advection, pure diffusion and advection-diffusion equations for the case without immersed boundary and the case with $\theta_1 = 0.5$ and $\theta_2 = 0.5$. Furthermore, the stability analysis results for different distances between the interface and the irregular grid point, $0 \leq \theta_1, \theta_2 < 1$, are illustrated in Fig. 15. For all cases, we choose $CFL = 0.45$ and $DFL = 0.25$. It should be noted that the focus here is not to find the maximum CFL and DFL numbers beyond which the schemes are unstable.

Semi-discrete investigations demonstrate that the IIM is stable in that all the eigenvalues are located in the left-half plane of the complex plane, $\omega_{r,max} < 0$. Fully-discrete analysis (discretization in space and time) shows that all the eigenvalues of the full discretization matrix lie inside the unit circle ($|\lambda|_{max} < 1$) and therefore, the IIM fulfills the necessary and sufficient conditions for stability. The results from the stability analysis are very promising in that the new IIM is stable in the inviscid limit and it could be used for DNS of high Reynolds number flows.

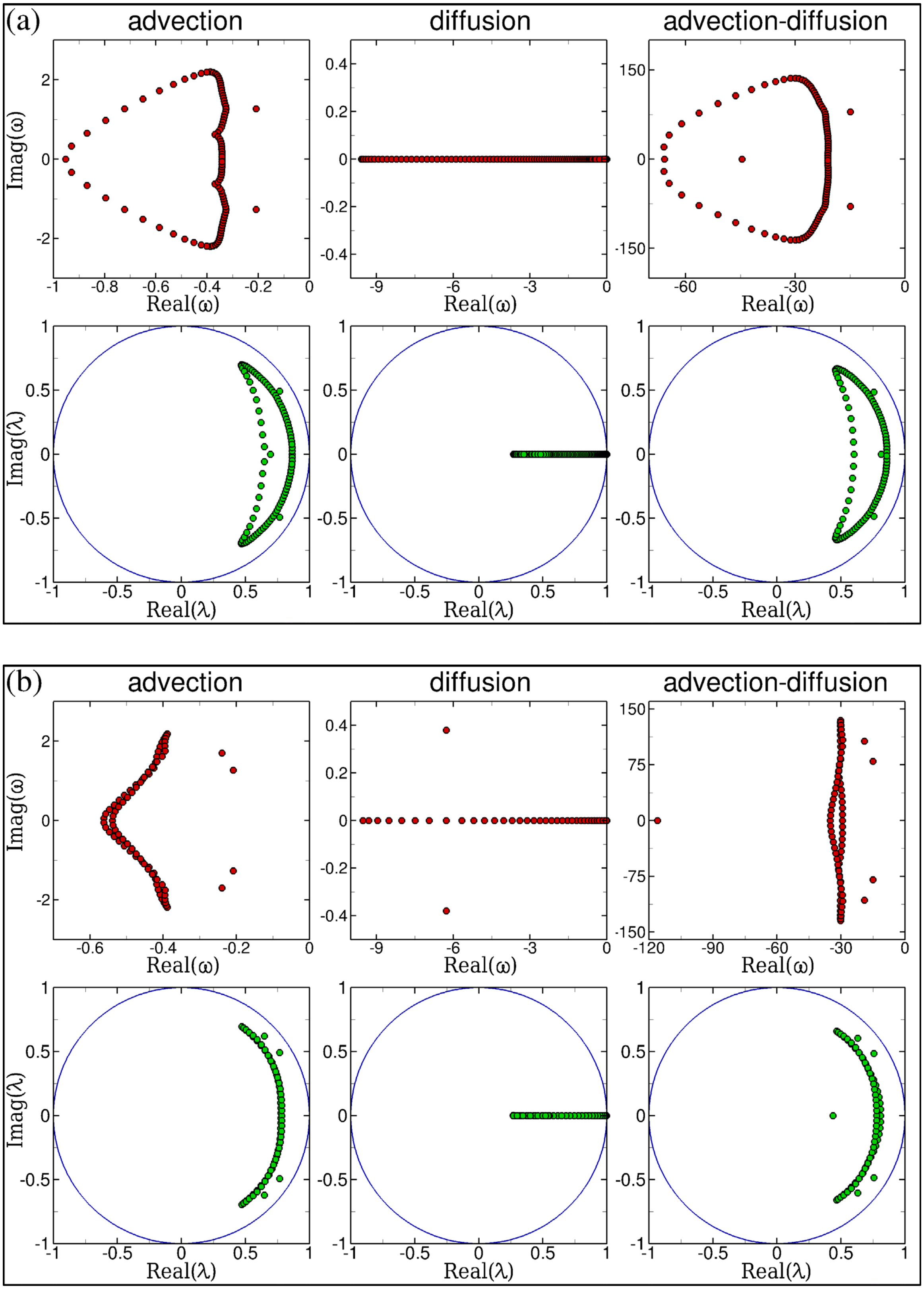

Figure 14. Eigenvalue spectrum for the semi- and fully-discrete cases for the pure advection, pure diffusion and advection-diffusion equations. (a) No immersed boundary, (b) $\theta_1 = 0.5, \theta_2 = 0.5$.

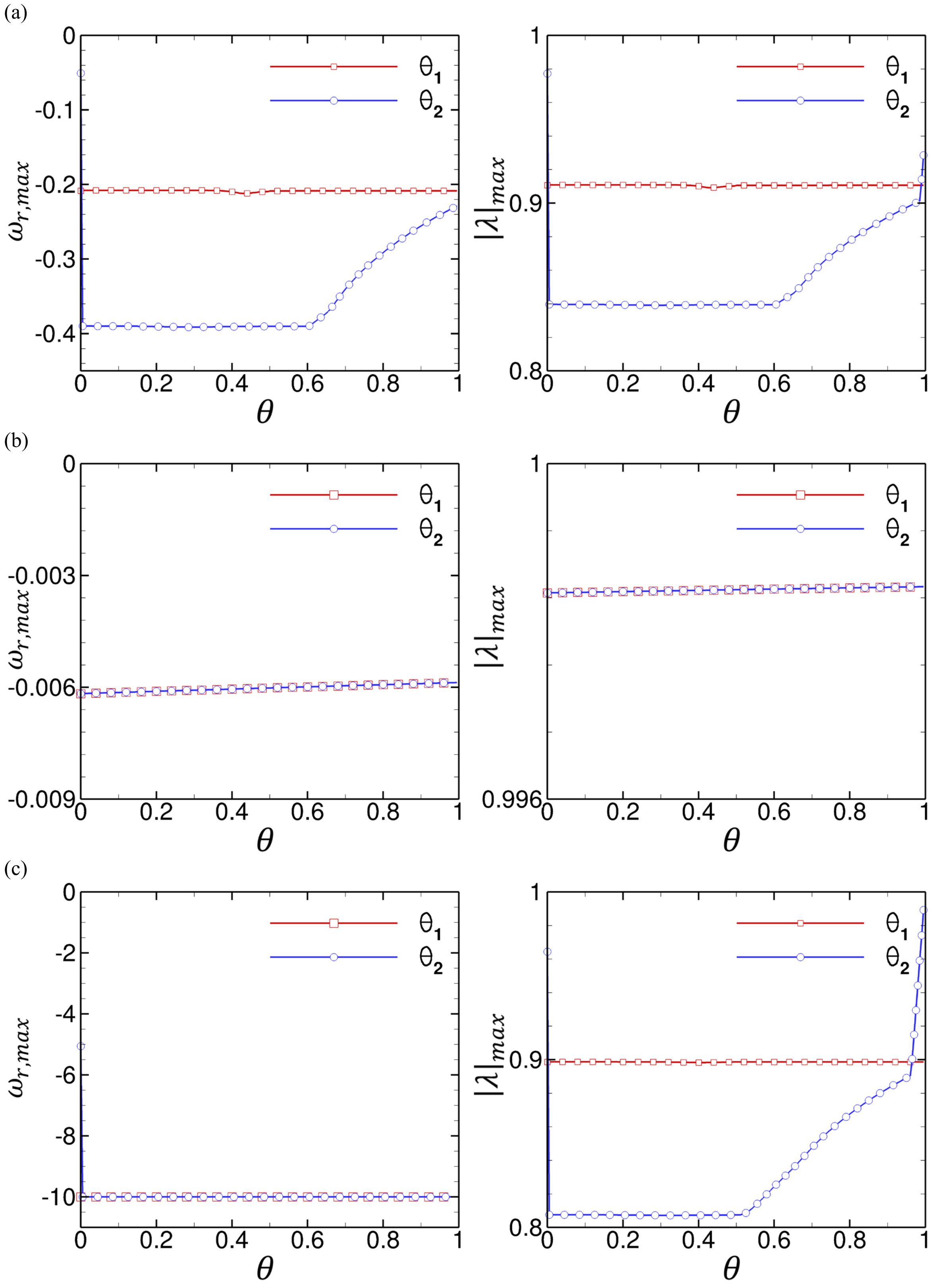

Figure 15. Maximum real eigenvalue ($\omega_{r,max}$) of the spatial discretization matrix and the maximum eigenvalue ($|\lambda|_{max}$) of the combined spatial-temporal discretizations matrix for $0 \leq \theta_1, \theta_2 < 1$. (a) advection, (b) diffusion and (c) advection-diffusion equation.

D. Validation of IIM: Stability Analysis for Two-Dimensional Hyperbolic Equation

To demonstrated the numerical stability of the IIM, two-dimensional advection equation is solved numerically for two different cases:

$$\frac{\partial u}{\partial t}+c_x\frac{\partial u}{\partial x}+c_y\frac{\partial u}{\partial y}=0, \quad for\ x,y\ \in[0,2\pi]^2\ \cap\ \Omega^+, \tag{61}$$

where the domain $\Omega^+$is defined as $\Omega^+=\{x,y\colon(x-\pi)^2+(y-\pi)^2\geq 0.5^2\}$ for case (1) as shown in Fig. 16 (left). For case (2), the domain Ω^+ is the region inside the boundary Γ defined by the following parametric equations:

$$x_\Gamma(\theta)=[0.47\sin^2(2\theta)+0.2\cos^2(2\theta)+2]\,cos(\theta), \tag{62a}$$

$$y_\Gamma(\theta)=[0.47\sin^2(2\theta)+0.2\cos^2(2\theta)+2]\,sin(\theta), \tag{62b}$$

for $0\leq\theta\leq 2\pi$, see Fig. 16 (right). The time-dependent boundary condition is specified along the interface Γ for each case. The exact solution is

$$u(x,y,t)=\sin\bigl(2(x-c_xt)\bigr)+\sin\Bigl(2\bigl(y-c_yt\bigr)\Bigr)\,, \tag{63}$$

where $c_x=c_y=1$. We use 101×101 equidistance Cartesian grid points for both cases. The fourth-order Runge-Kutta method is used for the time integration and the CFL number set to $CFL=0.25$. Figure 17 shows the evolution of the error, $\varepsilon=u_{num}-u_{exact}$, measured in L_2- and $L_\infty-$norm over time. For both cases, a stable time-integration is achieved, thereby indicating the numerical stability of the new sharp IIM.

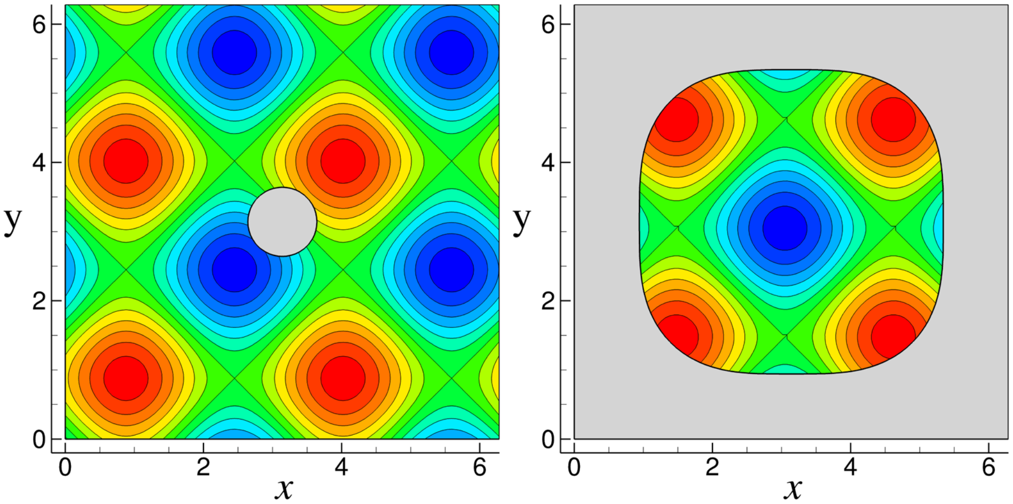

Figure 16.Numerical solutions computed by the new IIM for two-dimensional linear advection equation. Left: Case (1), right: Case (2).

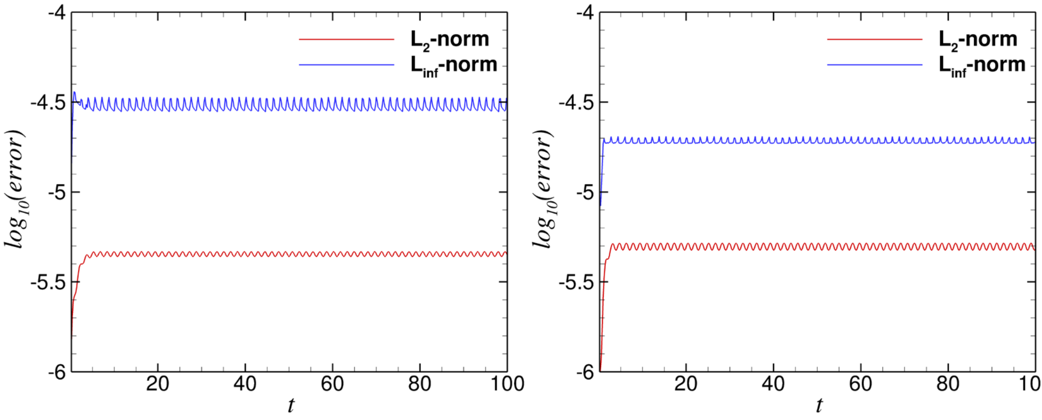

Figure 17. Illustration of the asymptotic stability of the IIM in L_2- and $L_\infty-$norm. Left: Case (1), right: Case (2).

V. High-Order Poisson Solver

A critical issue in the numerical solution of the INSE with the immersed boundary is the question of how to solve the Poisson equation highly-accurately and still be computationally efficient. It is important because a considerable amount of CPU time is spent to solve the Poisson equation. If the flow field is assumed to be periodic in the spanwise z-direction, then the Poisson equation takes the form

$$\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}-\gamma^2 u=\rho\,, \tag{64}$$

where γ is the spanwise wavenumber. This section presents a new, highly efficient and high-order method to solve Eq. 64 in a domain with immersed boundaries on a non-uniform mesh. To achieve this goal, a fourth-order compact finite difference scheme is combined with the multiscale multigrid method. Our new method falls into the sharp interface category. However, it distinguishes itself from other IIM in that the jump corrections are not required any more. The underlying feature of the new method is that the regular finite difference stencil is adjusted at the irregular grid points in the vicinity of the interfaces of the immersed bodies to obtain a solution that is sharp across the interface while keeping the fourth-order global accuracy.

A. Discretization: Fourth-Order Compact Difference Scheme on Non-Uniform Grid

Before we proceed with the details of the discretization, different type of grid points need to be defined. A grid point is called a solid point if it lies inside the immersed body. A grid point is said to be regular if all the 8 neighboring grid points are outside the immersed body, otherwise, it is defined as irregular point. Furthermore, the locations where the immersed body intersects with the grid are called body intercepted (BI) points. The BI points are the locations where the boundary conditions can be enforced. All types of grid points are illustrated in Fig. 18.

1. Compact finite difference stencil for regular points

First, we present the discretization for regular points so that the compact difference scheme is well-defined and valid. The discretization of Eq. 64 is based on two 1-D, fourth-order compact finite difference stencils for second derivatives in x and y:

$$(L_{xx}u_{xx})_j=(R_{xx}u)_j, \tag{65a}$$

$$\left(L_{yy}u_{yy}\right)_i=\left(R_{yy}u\right)_i, \tag{65b}$$

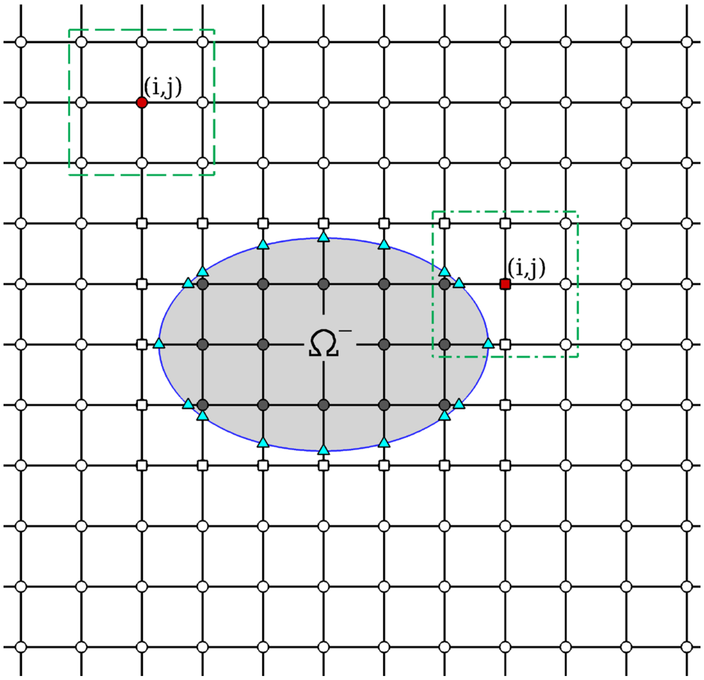

Figure 18. Representation of the different type of grid points: The regular grid points (marked by ○), irregular grid points (marked by □), the solid points (marked by ●), and the body intercepted points (▲).

where the finite difference (FD) operators are given by

$$(L_{xx}u_{xx})_j = a_i^x \cdot u_{xx_{i-1,j}} + b_i^x \cdot u_{xx_{i,j}} + c_i^x \cdot u_{xx_{i+1,j}} \,, \tag{66a}$$

$$\left(L_{yy}u_{yy}\right)_i = a_j^y \cdot u_{yy_{i,j-1}} + b_j^y \cdot u_{yy_{i,j}} + c_j^y \cdot u_{yy_{i,j+1}}, \tag{66b}$$

$$(R_{xx}u)_j = ar_i^x \cdot u_{i-1,j} + br_i^x \cdot u_{i,j} + cr_i^x \cdot u_{i+1,j} \,, \tag{66c}$$

$$\left(R_{yy}u\right)_i = ar_j^y \cdot u_{i,j-1} + br_j^y \cdot u_{i,j} + cr_j^y \cdot u_{i,j+1} \,. \tag{66d}$$

Here, u_{xx} and u_{yy} represent numerical approximations to the second partial derivatives in x and y directions, respectively. Coefficients in Eq. 66 are obtained by matching the coefficients in the Taylor expansion about $u_{i,j}$ in the x and y directions (see Appendix D). Combining Eqs. 65a and 65b at three consecutive x and y locations centered at point (i,j) lead to

$$\begin{aligned} a_j^y(L_{xx}u_{xx})_{j-1} + b_j^y(L_{xx}u_{xx})_j + c_j^y(L_{xx}u_{xx})_{j+1} + a_i^x\left(L_{yy}u_{yy}\right)_{i-1} + b_i^x\left(L_{yy}u_{yy}\right)_i + c_i^x\left(L_{yy}u_{yy}\right)_{i+1} \\ = a_j^y(R_{xx}u)_{j-1} + b_j^y(R_{xx}u)_j + c_j^y(R_{xx}u)_{j+1} + a_i^x\left(R_{yy}u\right)_{i-1} + b_i^x\left(R_{yy}u\right)_i + c_i^x\left(R_{yy}u\right)_{i+1}, \end{aligned} \tag{67}$$

Applying the FD operators in Eq. 67 and using Eq. 64 leads to a nine-points, fourth-order compact scheme at the regular grid point (i,j) inside the computational domain as follows:

$$\begin{aligned} A_{i,j}^{SW}\, u_{i-1,j-1} + A_{i,j}^{S}\, u_{i,j-1} + A_{i,j}^{SE}\, u_{i+1,j-1} + A_{i,j}^{W}\, u_{i-1,j} + A_{i,j}^{P}\, u_{i,j} + A_{i,j}^{E}\, u_{i+1,j} + A_{i,j}^{NW}\, u_{i-1,j+1} + A_{i,j}^{N}\, u_{i,j+1} \\ + A_{i,j}^{NE}\, u_{i+1,j+1} = Q_{i,j} \,, \end{aligned} \tag{68}$$

where RHS is given by

$$\begin{aligned} Q_{i,j} = Q_{i,j}^{SW}\, \rho_{i-1,j-1} + Q_{i,j}^{S}\, \rho_{i,j-1} + Q_{i,j}^{SE}\, \rho_{i+1,j-1} + Q_{i,j}^{W}\, \rho_{i-1,j} + Q_{i,j}^{P}\, \rho_{i,j} + Q_{i,j}^{E}\, \rho_{i+1,j} + Q_{i,j}^{NW}\, \rho_{i-1,j+1} \\ + Q_{i,j}^{N}\, \rho_{i,j+1} + Q_{i,j}^{NE}\, \rho_{i+1,j+1}. \end{aligned} \tag{69}$$

The coefficients in Eqs. 68 and 69 are given in Appendix D.

2. *Treatment of irregular points*

In this section, the method to determine the coefficients of the compact scheme stencil at an irregular grid point is presented. The key aspect of the new method is to modify the compact finite difference scheme, Eq. 65, when the three-point stencil intersects an immersed boundary. For the illustration, we consider the nine-point 2-D stencil centered at grid point (i,j) as shown in Fig. 19. Lines $y_{j-1} = const$, $y_j = const$ and $x_{i-1} = const$ do intersect the immersed boundary and the FD operators along these lines need to be adjusted to take into account the immersed boundary while maintaining the formal fourth-order accuracy.

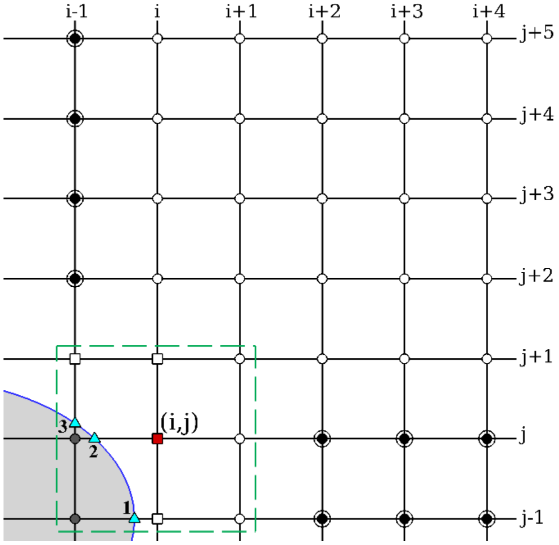

Figure 19. Intersection of compact FD nine-point stencil at irregular grid point *(i,j)* with an immersed boundary. Grid points marked by (◉) are additional points used in the modified finite difference stencils to maintain the formal fourth-order accuracy.

We rewrite Eq. 67 here

$$\begin{aligned} a_j^y(\overline{L_{xx}u_{xx}})_{j-1} + b_j^y(\overline{L_{xx}u_{xx}})_j + c_j^y(L_{xx}u_{xx})_{j+1} + a_i^x\left(\overline{L_{yy}u_{yy}}\right)_{i-1} + b_i^x\left(L_{yy}u_{yy}\right)_i + c_i^x\left(L_{yy}u_{yy}\right)_{i+1} \\ = a_j^y(\overline{R_{xx}u})_{j-1} + b_j^y(\overline{R_{xx}u})_j + c_j^y(R_{xx}u)_{j+1} + a_i^x\left(\overline{R_{yy}u}\right)_{i-1} + b_i^x\left(R_{yy}u\right)_i + c_i^x\left(R_{yy}u\right)_{i+1}. \quad (70) \end{aligned}$$

In the above equation, the FD operators with the overbar are modified, while the other operators are the same as those given in Eq. 66. A key point is that the coefficients used in the modified FD operators on the LHS of Eq. 70 must be kept the same as those given by Eq. D.1, however, the solid grid points (inside the body) used in the FD operators have to be dropped. Therefore, to maintain the formal fourth order accuracy, additional grid points are needed to determine the coefficients for the modified FD operators on the RHS of Eq. 70. Along the line $\mathrm{y_{j-1}} = \mathrm{const}$, we have $(\overline{L_{xx}u_{xx}})_{j-1} = (\overline{R_{xx}u})_{j-1}$ where

$$(\overline{L_{xx}u_{xx}})_{j-1} = b_i^x \cdot u_{xx_{i,j-1}} + c_i^x \cdot u_{xx_{i+1,j-1}}, \tag{71a}$$

$$(\overline{R_{xx}u})_{j-1} = \overline{br_i^x} \cdot u_{i,j-1} + \overline{cr_i^x} \cdot u_{i+1,j-1} + \psi_1\, u_{i+2,j-1} + \psi_2\, u_{i+3,j-1} + \psi_3\, u_{i+4,j-1} + \psi_b\, u_{BI_1}. \quad (71b)$$

In Eq. 71b, $u_{BI,1}$ is the known function value at the body intercept location 1, BI_1. For the right-hand side operator, Eq. 71b, we use 3 additional points (see Fig. 19) to keep the fourth-order accuracy. We use Taylor series expansions about $u_{i,j-1}$ to find the coefficients which are obtained by solving the following system of equations

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & h_0 & h_1 & h_2 & h_3 & h_b \\ 0 & h_0^2 & h_1^2 & h_2^2 & h_3^2 & h_b^2 \\ 0 & h_0^3 & h_1^3 & h_2^3 & h_3^3 & h_b^3 \\ 0 & h_0^4 & h_1^4 & h_2^4 & h_3^4 & h_b^4 \\ 0 & h_0^5 & h_1^5 & h_2^5 & h_3^5 & h_b^5 \end{pmatrix} \begin{pmatrix} \overline{\mathrm{br_i^x}} \\ \overline{\mathrm{cr_i^x}} \\ \psi_1 \\ \psi_2 \\ \psi_3 \\ \psi_b \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \mathrm{b_i^x} + \mathrm{c_i^x} \\ 3! \cdot h_0 \cdot \mathrm{c_i^x} \\ 4!/2! \cdot h_0^2 \cdot \mathrm{c_i^x} \\ 5!/3! \cdot h_0^3 \cdot \mathrm{c_i^x} \end{pmatrix}, \tag{72}$$

where $h_0 = x_{i+1} - x_i, h_1 = x_{i+2} - x_i, h_2 = x_{i+3} - x_i, h_3 = x_{i+4} - x_i,$ and $h_b = x_{BI_1} - x_i$. Note that the coefficients on the RHS in Eq. 72 are given by Eq. D.1 (see Appendix D). For the stencil passing through the $\mathrm{y_j} = \mathrm{const}$, we have $(\overline{L_{xx}u_{xx}})_j = (\overline{R_{xx}u})_j$ in which the modified FD operators are given by

$$(\overline{L_{xx}u_{xx}})_j = b_i^x \cdot u_{xx_{i,j}} + c_i^x \cdot u_{xx_{i+1,j}}, \tag{73a}$$

$$(\overline{R_{xx}u})_j = \overline{\overline{br_i^x}} \cdot u_{i,j} + \overline{\overline{cr_i^x}} \cdot u_{i+1,j} + \alpha_1\, u_{i+2,j} + \alpha_2\, u_{i+3,j} + \alpha_3\, u_{i+4,j} + \alpha_b\, u_{BI_2}, \quad (73b)$$

where u_{BI_2} is the known boundary value at BI location 2, BI_2. Matching the Taylor series coefficients around $u_{i,j}$, the coefficients in Eq. 73b can be found from

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & h_0 & h_1 & h_2 & h_3 & h_b \\ 0 & h_0^2 & h_1^2 & h_2^2 & h_3^2 & h_b^2 \\ 0 & h_0^3 & h_1^3 & h_2^3 & h_3^3 & h_b^3 \\ 0 & h_0^4 & h_1^4 & h_2^4 & h_3^4 & h_b^4 \\ 0 & h_0^5 & h_1^5 & h_2^5 & h_3^5 & h_b^5 \end{pmatrix} \begin{pmatrix} \overline{\overline{\mathrm{br_i^x}}} \\ \overline{\overline{\mathrm{cr_i^x}}} \\ \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \alpha_b \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \mathrm{b_i^x} + \mathrm{c_i^x} \\ 3! \cdot h_0 \cdot \mathrm{c_i^x} \\ 4!/2! \cdot h_0^2 \cdot \mathrm{c_i^x} \\ 5!/3! \cdot h_0^3 \cdot \mathrm{c_i^x} \end{pmatrix}, \tag{74}$$

where $h_0 = x_{i+1} - x_i, h_1 = x_{i+2} - x_i, h_2 = x_{i+3} - x_i, h_3 = x_{i+4} - x_i,$ and $h_b = x_{BI_2} - x_i$. Finally, for the stencil along the $x_{i-1} = const$, the FD scheme takes the form $\left(\overline{L_{yy}u_{yy}}\right)_{i-1} = \left(\overline{R_{yy}u}\right)_{i-1}$ where

$$\left(\overline{L_{yy}u_{yy}}\right)_{i-1} = c_j^y \cdot u_{yy_{i-1,j+1}}, \tag{75a}$$

$$\left(\overline{R_{yy}u}\right)_{i-1} = \overline{cr_j^y} \cdot u_{i-1,j+1} + \theta_1\, u_{i-1,j+2} + \theta_2\, u_{i-1,j+3} + \theta_3\, u_{i-1,j+4} + \theta_4\, u_{i-1,j+5} + \theta_b\, u_{BI_3}, \quad (75b)$$

where u_{BI_3}denotes the boundary value at BI location 3, BI_3. One should note that the Taylor expansion around $u_{i-1,j+1}$ is used to find the coefficients in Eq. 75b as follows

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & h_0 & h_1 & h_2 & h_3 & h_b \\ 0 & h_0^2 & h_1^2 & h_2^2 & h_3^2 & h_b^2 \\ 0 & h_0^3 & h_1^3 & h_2^3 & h_3^3 & h_b^3 \\ 0 & h_0^4 & h_1^4 & h_2^4 & h_3^4 & h_b^4 \\ 0 & h_0^5 & h_1^5 & h_2^5 & h_3^5 & h_b^5 \end{pmatrix} \begin{pmatrix} \overline{\mathrm{cr}_\mathrm{j}^\mathrm{y}} \\ \theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_\mathrm{b} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \mathrm{c}_\mathrm{j}^\mathrm{y} \\ 0 \\ 0 \\ 0 \end{pmatrix}, \tag{76}$$

where $h_0 = y_{j+2} - y_{j+1}$; $h_1 = y_{j+3} - y_{j+1}$; $h_2 = y_{j+4} - y_{j+1}$; $h_3 = y_{j+5} - y_{j+1}$, and $h_b = y_{BI_3} - y_{j+1}$. Applying the above equations to Eq.70, we get the modified compact scheme stencil at the irregular grid point (i, j)

$$\begin{aligned} &\overline{A_{i,j}^{S}}\, u_{i,j-1} + \overline{A_{i,j}^{SE}}\, u_{i+1,j-1} + \overline{A_{i,j}^{P}}\, u_{i,j} + \overline{A_{i,j}^{E}}\, u_{i+1,j} + \overline{A_{i,j}^{NW}}\, u_{i-1,j+1} + A_{i,j}^{N}\, u_{i,j+1} + A_{i,j}^{NE}\, u_{i+1,j+1} + \mathcal{C}_{i,j} \\ &\qquad = Q_{i,j}^{S}\, \rho_{i,j-1} + Q_{i,j}^{SE}\, \rho_{i+1,j-1} + Q_{i,j}^{P}\, \rho_{i,\mathrm{j}} + Q_{i,j}^{E}\, \rho_{i+1,j} + Q_{i,j}^{NW}\, \rho_{i-1,j+1} + Q_{i,j}^{N}\, \rho_{i,j+1} + Q_{i,j}^{NE}\, \rho_{i+1,j+1} \\ &\qquad + \mathfrak{B}_{i,j}\,, \end{aligned} \tag{77}$$

where the modified coefficients are given in Appendix D. In Eq. 77, $\mathcal{C}_{i,j}$ contains the additional points used to keep the fourth-order formal accuracy

$$\begin{aligned} \mathcal{C}_{\mathrm{i,j}} = {} & \mathrm{a}_\mathrm{j}^\mathrm{y}\left(\psi_1\, u_{i+2,j-1} + \psi_2\, u_{i+3,j-1} + \psi_3\, u_{i+4,j-1}\right) + \mathrm{b}_\mathrm{j}^\mathrm{y}\left(\alpha_1\, u_{i+2,j} + \alpha_2\, u_{i+3,j} + \alpha_3\, u_{i+4,j}\right) \\ & + a_i^x\left(\theta_1\, u_{i-1,j+2} + \theta_2\, u_{i-1,j+3} + \theta_3\, u_{i-1,j+4} + \theta_4\, u_{i-1,j+5}\right), \end{aligned} \tag{78}$$

and $\mathfrak{B}_{i,j}$ includes the known function values at the boundary intercept points,

$$\mathfrak{B}_{\mathrm{i,j}} = -\left(\mathrm{a}_\mathrm{j}^\mathrm{y}\psi_b \mathrm{u}_{\mathrm{b}1} + \mathrm{b}_\mathrm{j}^\mathrm{y}\alpha_b \mathrm{u}_{\mathrm{b},2} + \mathrm{a}_\mathrm{i}^\mathrm{x}\theta_b\, u_{b,3}\right). \tag{79}$$

The last point to note is that the coefficients without the overbar in Eq. 77 are the same as the coefficients obtained for the regular grid points. One should note that the coefficients for the solid grid points are not used in Eq. 77 since the function value is zero at those grid points.

B. Solution Strategy: Multiscale Multigrid Method (MSMG)

The compact finite difference discretization of Eq. 64 at all grid points forms a large sparse algebraic system. The multigrid method is among the fastest and most efficient iterative methods for solving systems of linear equations arising from the discretization of partial differential equations. Multigrid methods typically converge at a rate independent of the grid size for a prescribed accuracy for elliptic problems. They will thus provide a solution for a specified accuracy in a fixed number of iterations, independent of how fine the mesh is. However, more iterations are needed to exploit the higher accuracy expected for the finer mesh.

One approach to improve to performance of the multigrid method is by providing a better initial guess to start the multigrid iteration. This can be achieved by a *nested iteration*, where low-cost computations on the coarser level is used to provide an initial guess on the finest grid. In this technique, the system is first solved (iteratively or directly) on the coarsest grid, then the results are interpolated to the next finer grid, where a few inexpensive multigrid V-cycles are applied and so on, until the finest grid is reached. The combination of the nested iteration and the multigrid is called full multigrid (FMG) method.

In this paper, we use a modified version of FMG method, the so-called multiscale multigrid (MSMG) method proposed by Wang & Zhang.[23] They used the MSMG method on a simple rectangular domain with uniform mesh to obtain higher-order accurate solution from two lower-order solutions by using a multiscale strategy and Richardson extrapolation. Here, our goal is to obtain a solution, whose accuracy is only limited by the discretization error, in a fixed number of iterations (independent of the grid size) by using the MSMG method for irregular boundaries on uniform and non-uniform grids. Figure 20 illustrates the structure of the MSMG method. It is similar to the FMG technique, but the computation does not start from the coarsest grid. In this example, we have 6 different grid levels and 4 different multigrid V-cycles are applied. The MSMG method starts with a standard multigrid V-cycle on $\Omega_{8\Delta}$ and then on $\Omega_{4\Delta}$ and so on until it reaches the finest level Ω_{Δ}. In this technique, the solution from $\Omega_{8\Delta}$ is interpolated on $\Omega_{4\Delta}$ and used as initial approximation; the solution on $\Omega_{4\Delta}$ is used as the initial guess for $\Omega_{2\Delta}$; and finally, the solution from $\Omega_{2\Delta}$ provides the initial approximation for Ω_{Δ}. Typically, just one or two multigrid V-cycles are necessary at each level of the hierarchy. Here Ω_{Δ} denotes the finest grid level and $\Omega_{2\Delta}$ corresponds to a grid one level coarser than grid Ω_{Δ} and so on. Grid $\Omega_{2\Delta}$ is formed by the standard coarsening strategy in which every second grid point of grid Ω_{Δ} is considered.

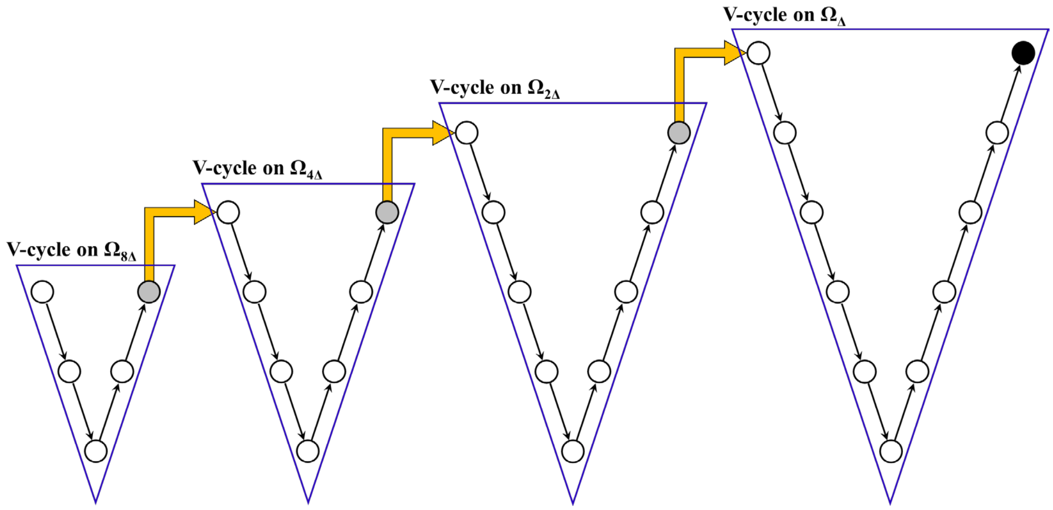
V-cycle on $\Omega_{8\Delta}$
V-cycle on $\Omega_{4\Delta}$
V-cycle on $\Omega_{2\Delta}$
V-cycle on Ω_{Δ}

Figure 20. Representation of multiscale multigrid method. Down arrows represent the restriction operator, up arrows correspond to the prolongation operator and right bent arrows interpolate the solution to the next multigrid V-cycle.

Each of the standard multigrid V-cycle in the MSMG method consists of smoothing the errors using a relaxation technique, restricting the residuals to the next coarser grid and prolongating the corrections back to the finer grid. Note finally that a one-dimensional, 4th-order finite difference interpolation is used to interpolate the solution from one multigrid V-cycle to the next one (shown as right-bent arrows in Fig. 20).

The relaxation operator (the smoother) is a very important operator in the multigrid method. The main task of the relaxation operator is to damp out the high frequency components of the errors on the current grid while leaving the low frequency components to be removed by the coarser grids. The discretization of the Poisson equation without immersed boundaries leads to a system of equations $\boldsymbol{\mathcal{A}}\boldsymbol{u} = \boldsymbol{q}$ where the matrix $\boldsymbol{\mathcal{A}}$ has a 9-diagonal structure. This system is solved iteratively using the modified strongly implicit (MSI) procedure (Schneider & Zedan[24]). The MSI method has very good convergence properties, even for highly stretched grids and has shown to outperform the ILLU and Gauss–Seidel methods (see Schneider & Zedan[24]). The basic idea in the MSIP method is to replace the matrix $\boldsymbol{\mathcal{A}}$ with approximated lower and upper diagonal matrices, $\boldsymbol{\mathcal{A}} = \boldsymbol{L}\boldsymbol{U}$. With this, a single-grid iterative method for solving $\boldsymbol{\mathcal{A}}\boldsymbol{u} = \boldsymbol{q}$ with starting guess $\boldsymbol{u}^n$ is given by:

(1) $\boldsymbol{r}^n = \boldsymbol{q} - \boldsymbol{\mathcal{A}}\boldsymbol{u}^n$
(2) $\mathbf{L}\boldsymbol{w}^{n+1} = \boldsymbol{r}^n$
(3) $\mathbf{U}\boldsymbol{\delta}^{n+1} = \boldsymbol{w}^{n+1}$
(4) $\boldsymbol{u}^{n+1} = \boldsymbol{u}^n + \boldsymbol{\delta}^{n+1}$

A considerable amount of detail has been left out of the present description regarding the MSI algorithm and the reader is referred to Schneider & Zedan.[24] The extension to problems with immersed boundaries is considered next. As discussed in the previous section, the finite difference schemes are modified when the 9-point stencil intersects with the immersed boundary. As a result, this introduces additional points to the 9-point compact discretization which are solution dependents. Note that some of the coefficients of the 9-point stencil are modified while some of them remain unchanged, see Eq. 77. Therefore, the matrix $\boldsymbol{\mathcal{A}}$ is replaced by the modified matrix $\bar{\boldsymbol{\mathcal{A}}}$ and to $\bar{\boldsymbol{\mathcal{A}}}$ is added a matrix $\boldsymbol{\mathcal{C}}$ as follows,

$$(\bar{\boldsymbol{\mathcal{A}}} + \boldsymbol{\mathcal{C}})\boldsymbol{u}^n = \boldsymbol{q} + \boldsymbol{b}. \tag{80}$$

In the above equation, vector $\boldsymbol{b}$ includes the known function values at the body intercept points. The modified matrix $\bar{\boldsymbol{\mathcal{A}}}$ has the same structure as $\boldsymbol{\mathcal{A}}$ (9-diagonal matrix), however, matrix $\boldsymbol{\mathcal{C}}$ adds irregular entries. To handle these irregular entries, the following strategy was developed:

(1) $\boldsymbol{r}^n = (\boldsymbol{q} - \bar{\boldsymbol{\mathcal{A}}}\boldsymbol{u}^n) + (\boldsymbol{b} - \boldsymbol{\mathcal{C}}\boldsymbol{u}^n)$
(2) $\bar{\mathbf{L}}\boldsymbol{w}^{n+1} = \boldsymbol{r}^n$
(3) $\bar{\mathbf{U}}\boldsymbol{\delta}^{n+1} = \boldsymbol{w}^{n+1}$
(4) $\boldsymbol{u}^{n+1} = \boldsymbol{u}^n + \boldsymbol{\delta}^{n+1}$

In the above equation, $\bar{\boldsymbol{L}}$ and $\bar{\boldsymbol{U}}$ are the lower and upper diagonal matrices obtained for the modified matrix $\bar{\boldsymbol{\mathcal{A}}}$ according to the MSI algorithm. The procedure just outlined was empirically determined, and was found to converge in all cases in which it was employed to solve the Poisson equation with immersed boundaries. It is a convenient algorithm that allows the MSI procedure to be performed on the regular matrix $\bar{\boldsymbol{\mathcal{A}}}$, yet it easily accommodates the irregularly located entries introduced by immersed boundaries.

The restriction and prolongation operators are designed for non-uniform meshes using area-ratio law and are adjusted near the immersed boundary.

C. Validation

To demonstrate the accuracy of the new method and efficiency of the MSMG solver on a non-uniform grid, we consider solving the following Poisson equation with Dirichlet boundary conditions:

$$u_{xx} + u_{yy} = 12(10r^2 - 1)e^{-10r^2} + \sum_{k=1}^{4} 40(10r_k^2 - 1)e^{-10r_k^2}, \quad in\ \Omega^+, \tag{81a}$$

$$u(x,y) = 0.3e^{-10r^2} + \sum_{k=1}^{4} e^{-10r_k^2}, \qquad on\ \Gamma\ \&\ \partial\Omega\,, \tag{81b}$$

where $r = \sqrt{x^2 + y^2}$ and each r_k is given by

$$r_1 = \sqrt{x^2 + (y - 0.45)^2}, r_2 = \sqrt{x^2 + (y + 0.45)^2}, r_3 = \sqrt{(x - 0.45)^2 + y^2}, \text{and } r_4 = \sqrt{(x + 0.45)^2 + y^2}. \tag{82}$$

We solve Eq. 81 for two cases: (i) simple domain without immersed boundary and (ii) the domain with immersed boundary. For the latter case, the domain $\Omega^+$is the region outside the immersed boundary Γ defined by the parametric equations:

$$x_\Gamma(\theta) = (0.305 + 0.117\,cos(4\theta))\,cos(\theta), \tag{83a}$$

$$y_\Gamma(\theta) = (0.305 + 0.117\,cos(4\theta))\,sin(\theta), \tag{83b}$$

for $0 \le \theta \le 2\pi$. The grid stretching parameter is the same for both directions and chosen to be $\alpha = 0.7$ (see Eq. 3). Computed solutions and the corresponding errors, $\varepsilon(x,y) = |u(x,y) - u^{exact}(x,y)|$ are shown in Figs. 21 and 22, respectively, for both cases on a 257×257 grid. The reader's attention is drawn to the sharp interface in the solution obtained, and the relative smoothness of the error distribution in the vicinity of the immersed boundary.

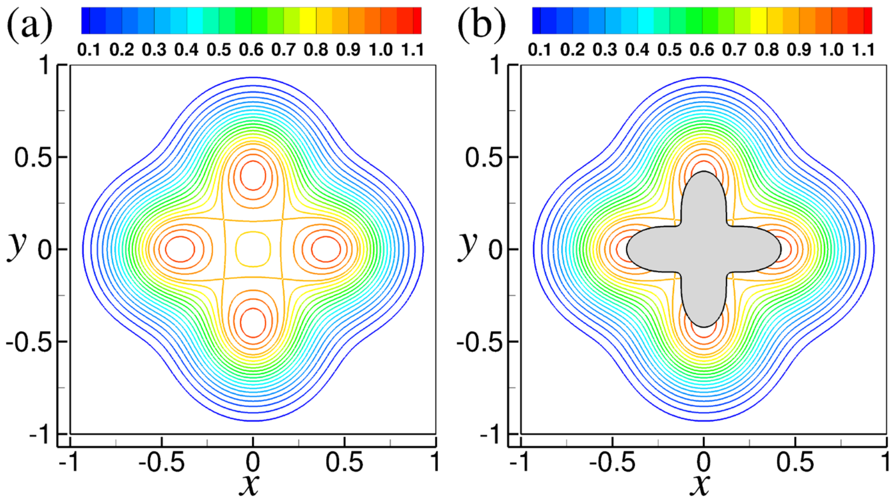

Figure 21. Numerical solutions of Eq. 81 on the computational domain with 257×257 non-uniform grid points for two different cases: (a) simple domain without immersed boundary, (b) domain with immersed boundary.

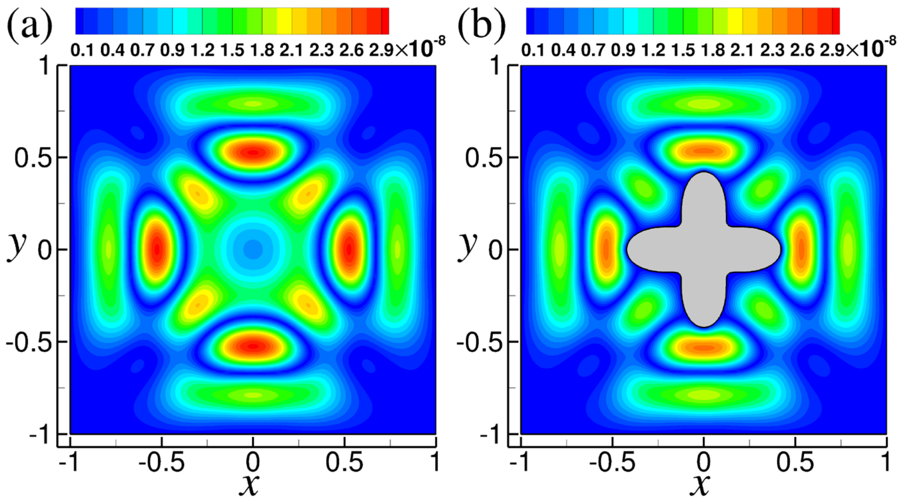

Figure 22. Errors correspond to numerical solutions of Eq. 81 on the computational domain with 257×257 non-uniform grid points: (a) simple domain without immersed boundary, (b) domain with immersed boundary.

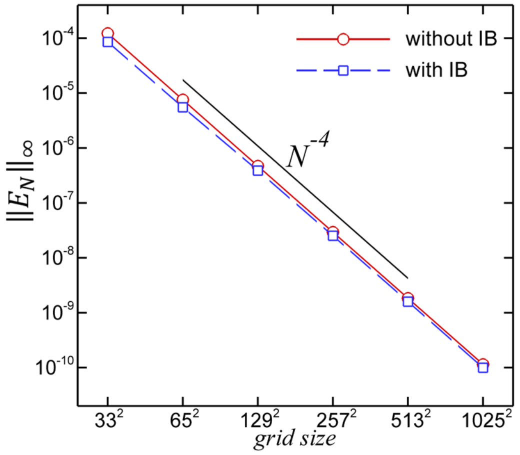

Figure 23. Numerical error in the infinity-norm as a function of grid points for solutions of Eq. 81 for domain with immersed boundary (IB) and without IB.

The solution and the corresponding error is computed on several grids, and the errors in the infinity-norm, $\|E_N\|_\infty$, is plotted as a function of grid size in Fig. 23. It confirms the fourth-order accuracy of the new method for both the simple domain and the domain with the immersed boundary. Furthermore, no loss of accuracy for the domain with immersed boundary is observed.

Another important aspect of the solution of Poisson equation with the MSMG method is the time penalty associated with adding immersed boundaries. Table 1 presents run times to compute 2 V-cycles in the last stage of the MSMG method for the problem with and without immersed boundary. For the problem sizes of interest, Table 1 shows that the extra computational time due to the presence of immersed boundary is negligible for the MSMG solver. For the largest grid size investigated, the extra CPU time is less than 1%. The relaxation operator constitutes the most expensive part of the MSMG method. With a proper design of this operator which accommodates the immersed boundary without any significant loss of performance of the original algorithm, the new solution strategy is proven to be equally efficient for domains with immersed boundaries as for simple domains.

Grid size →	129×129	257×257	513×513	1025×1025
simple domain	0.0143	0.0720	0.3385	1.4043
with immersed boundary	0.0146	0.0741	0.3454	1.4137
(%) increase	2.945	2.916	2.056	0.674

Table 1. Comparison of computation times between the problem with immersed boundary and the problem without immersed boundary with the MSMG method.

VI. Application to the Incompressible Navier–Stokes Equations

We consider the INSE for the unsteady two-dimensional flows in in vorticity-velocity formulation

$$\frac{\partial \omega_z}{\partial t}+\frac{\partial (u\omega_z)}{\partial x}+\frac{\partial (v\omega_z)}{\partial y}=\frac{1}{Re}\left(\frac{\partial^2 \omega_z}{\partial x^2}+\frac{\partial^2 \omega_z}{\partial x^2}\right), \tag{84}$$

where u, v are the velocity components in the streamwise (x) and the wall-normal (y) directions, respectively, and ω_z is the spaniwse vorticity. The wall-normal and streamwise velocity components are obtained from

$$\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial x^2}=-\frac{\partial \omega_z}{\partial x}. \tag{85a}$$

$$\frac{\partial u}{\partial x}=-\frac{\partial v}{\partial y}. \tag{85b}$$

In the above equations, all quantities are non-dimensionalized by the free-stream velocity, U_∞^*, a reference length, L_∞^*, and the kinematic viscosity, ν_∞^*. The global Reynolds number is defined as $Re = U_\infty^* L_\infty^* / \nu_\infty^*$. Asterisk is used to denote dimensional quantities.

The solution of the INSE vorticity-velocity formulation consists mainly of the two computational components discussed above, specifically, the time integration of a convection–diffusion equation, the vorticity-transport equation, Eq. 83, the solution of a Poisson equation for the v-velocity, Eq. 85a.

The algorithm starts at time t_n with discrete variables u, v and ω_z known everywhere in the domain. The vorticity-transport equation, Eq. 84, can be used to advance to vorticity in time from t_n to t_{n+1}, using a fourth-order Runge-Kutta time integration method. Next, the Poisson equation, Eq. 84a, is solved using the scheme discussed in Section V to obtain the v-velocity at time t_{n+1}. Then Eq. 85b is used to calcite the u-velocity. Once the velocity field is determined, the wall-vorticity on the immersed boundary at t_{n+1} is computed using the definition of spaniwse vorticity.

A. Uniform Flow Past a Circular Cylinder

In this section, the immersed boundary code is used to compute the solutions for the steady and unsteady flow around a circular cylinder placed in a uniform free-stream. Reynolds numbers based on the cylinder diameter (D) in the range $Re = 20$ to $Re = 200$ are considered. The inflow and outflow boundaries are located 10D and 13D from the cylinder, respectively. The distance from the cylinder to the lower and upper free-stream boundaries is set to 8D. At the outflow, a buffer domain technique by Meitz & Fasel[28] is used to ramp the vorticity down to zero.The same grid resolution, 513×257, is used for all cases.

Results for the steady regime of Reynolds numbers, $Re = 20$ and $Re = 40$, are shown in Fig. 24. The attached, steady, symmetric standing eddy behind the cylinder is seen to grow in length as Re increases. Table 2 presents a comparison of the present results with those published in the literature. For all quantities of interest, excellent agreement is found within the scatter of the data.

Flows with higher Reynolds numbers, $Re = 100$ and $Re = 200$, are shown in Fig. 25. As the time progresses, an unsteady vortex shedding sets in. The Strouhal number of $f = 0.168$ and $f = 0.2$ is calculated for $Re = 100$ and $Re = 200$, respectively, which is very close with the previous studies of Linnick & Fasel[3] and Shukla et al.[25]

	$Re = 20$					$Re = 40$				
	L	a	b	$\theta(°)$	C_D	L	a	b	$\theta(°)$	C_D
Coutanceau and Bouard[26]	0.93	0.33	0.46	45.0	-	2.13	0.76	0.59	53.8	-
Brehm and Fasel[27]	0.94	0.36	0.43	43.5	2.08	2.29	0.72	0.6	52.4	1.57
Linnick and Fasel[3]	0.93	0.36	0.43	43.5	2.06	2.28	0.72	0.6	53.6	1.54
Shukla et al.[25]	0.92	-	-	43.2	2.07	2.34	-	-	52.7	1.55
Present	**0.92**	**0.36**	**0.43**	**43.1**	**2.07**	**2.21**	**0.71**	**0.6**	**53.1**	**1.57**

Table 2. Steady flow past a circular cylinder: Length L of standing eddy behind cylinder, locations a and b correspond to the x- and y-location of the vortex centers, respectively, separation angle θ, and drag coefficient C_D for $Re = 20$ and $Re = 40$.

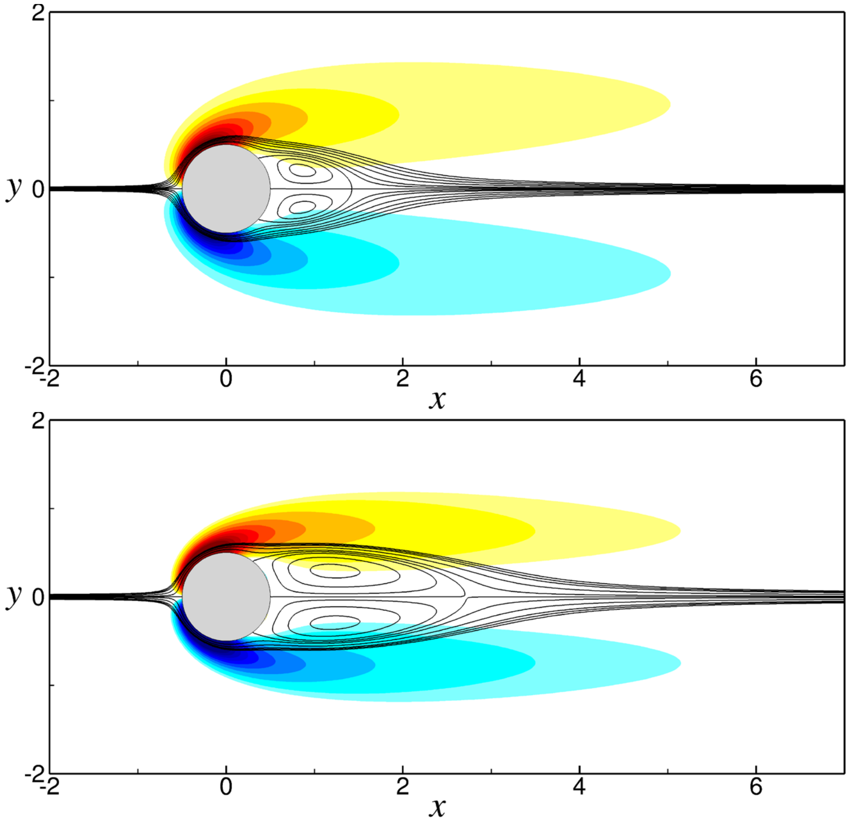

Figure 24. Streamlines and Contours of spanwise vorticity for steady flow past circular cylinders at $Re = 20$ (top) and $Re = 40$ (bottom).

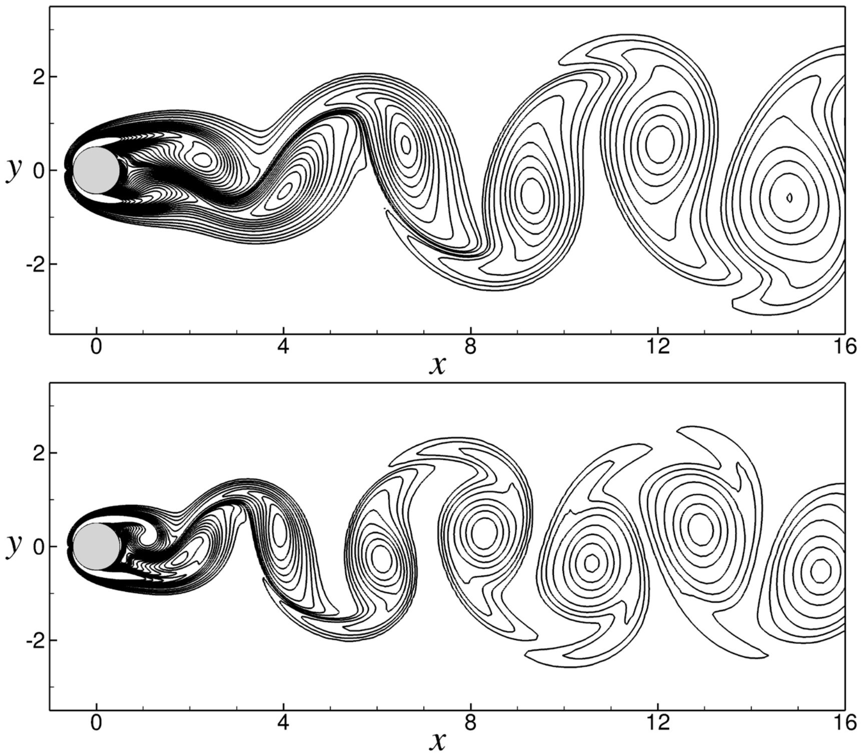

Figure 25. Contour lines of spanwise vorticity for flow past circular cylinders at $Re = 100$ (top) and $Re = 200$ (bottom).

B. Tollmien–Schlichting Wave in a Flat-Plate Boundary-Layer

Under conditions of low environment disturbances, the laminar-turbulent transition process in boundary-layer flows is known to be initiated by the spatial growth of Tollmien-Schlichting (TS) waves. In this section, the application of the new sharp IIM to compute the propagation of TS waves in a zero pressure-gradient flat-plate boundary-layer is investigated. The purpose of this test-case is to determine whether the current IIM is accurate enough to be used for transition research.

The results obtained from the present IIM are compared against the results computed from a body-fitted code (the temporally and spatially fourth-order accurate code nst2d of Meitz & Fasel[28]). The setup is shown in Fig. 26 where the immersed flat plate wall runs from the inflow to the outflow boundary. The immersed wall does not lie on the computational grid, but rather half-way between grid points $j = 4$ and $j = 5$ in the y-direction. Computational parameters are given in Table 3. Near the outflow boundary the buffer domain technique proposed by Meitz & Fasel[28] was applied in order to smoothly dampen out the fluctuations generated inside the domain.

As a first step, the undisturbed flat-plate boundary layer is computed, starting with the Blasius similarity solution as initial condition and converging to a steady-state. Figure 27 compares the resulting u-velocity and ω_z-voritcty profiles at various x-locations along the flat plate. Excellent agreement between the body-fitted and immersed boundary code is found.

Next, small, time-harmonic disturbances are introduced into the steady boundary layer. Low amplitude, two-dimensional disturbance waves are generated near the inflow boundary by specifying a wall-normal velocity component across a narrow blowing and suction slot at the immersed wall. The forcing function for the harmonic blowing and suction (HBS) simulations has the form

$$v_f(x,t) = B \cdot S(x) \cdot \sin(2\pi f), \tag{86}$$

where B is the maximum forcing amplitude and f represent the forcing frequency. The shape function, $S(x)$, is a polynomial which is zero outside the suction/blowing slot such that we have smooth derivatives near boundaries of the suction/blowing slot. Furthermore, it guarantees zero net volume flow through the disturbance slot at any time:

$$S(x) = \frac{1}{48}\begin{cases} +729\psi^5 - 1701\psi^4 + 972\psi^3, & \psi = \dfrac{x - x_s}{x_f - x_s}; \quad x_s \le x \le x_f \\ -729\psi^5 + 1701\psi^4 - 972\psi^3, & \psi = \dfrac{x_e - x}{x_e - x_f}; \quad x_f \le x \le x_e \end{cases}, \tag{87}$$

The downstream locations x_s, x_e and x_f correspond the beginning, the end, and the center of the slot, respectively. The same technique was used for the body fitted code.

The spatial development of the TS wave is demonstrated in Fig. 28 which shows the contours of disturbance u-velocity and ω_z-vorticity. Downstream of forcing location the disturbances are decaying first until they encounter the lower branch of neutral instability ($x \approx 1.57$), and then, they exhibit the strong amplification as they travel downstream up to the upper branch of neutral instability ($x \approx 4$) at which they begin to decay.

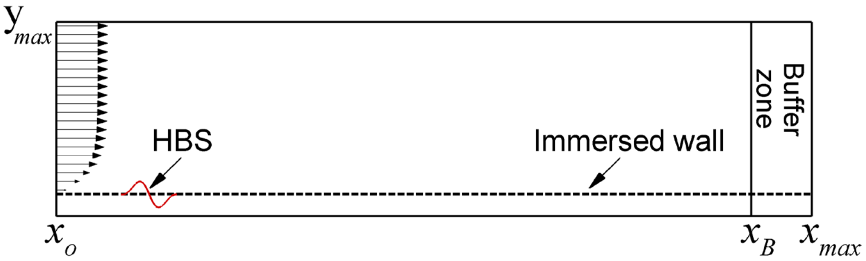

Figure 26. Computational setup for a flat plate immersed in a rectangular domain.

Parameter	Value
Re	10^5
Domain size ($x_0 \times x_B \times x_{max} \times y_{max}$)	$0.4 \times 5.36 \times 6.16 \times 0.3$
Grid size ($N_x \times N_y$)	577×129
Forcing ($B,\ 2\pi f/Re,\ x_s \times x_e$)	$10^{-4}, 1.2 \times 10^{-4}, 0.6 \times 0.75$

Table 3. Parameters for the TS wave computation

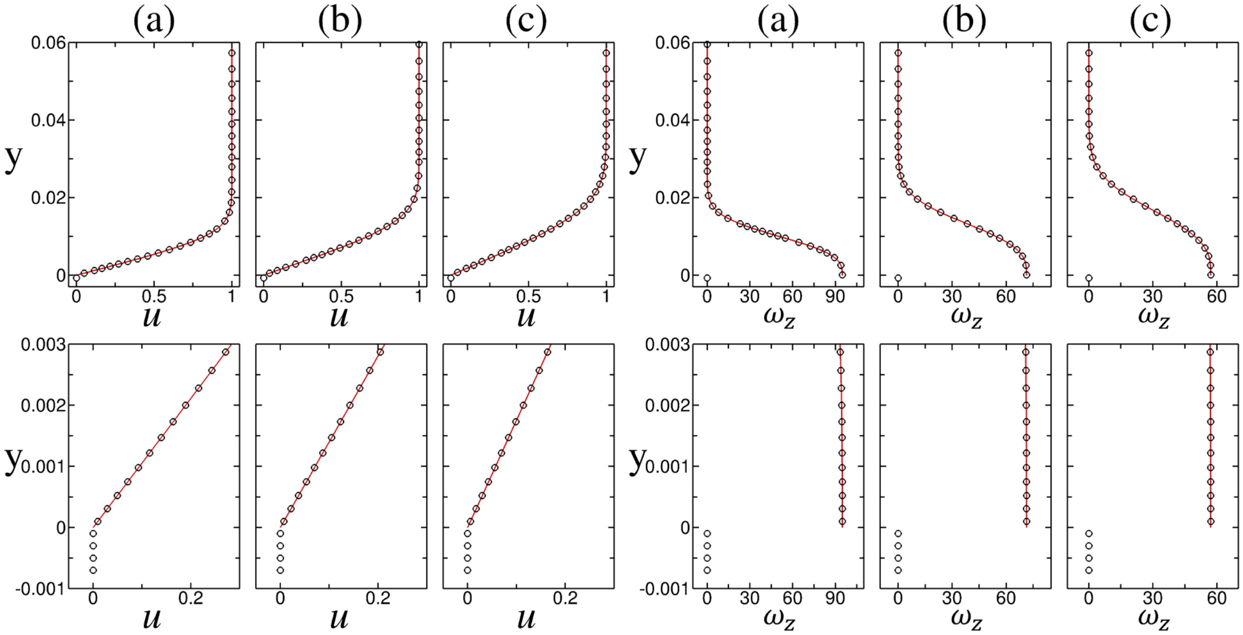

Figure 27. Comparison of u-velocity (left) and ω_z-vorticity (right) profiles at several downstream locations. (a) $x = 1.22$ ($Re_{\delta_1} = 600$); (b) $x = 2.16$ ($Re_{\delta_1} = 800$); (c) $x = 3.38$ ($Re_{\delta_1} = 1000$). Symbols indicate values computed using an immersed wall located at $y = 0$, and solid lines results from the code nst2d of Meitz and Fasel.[28] Lower figure is a zoom-in of the upper figure near $y = 0$.

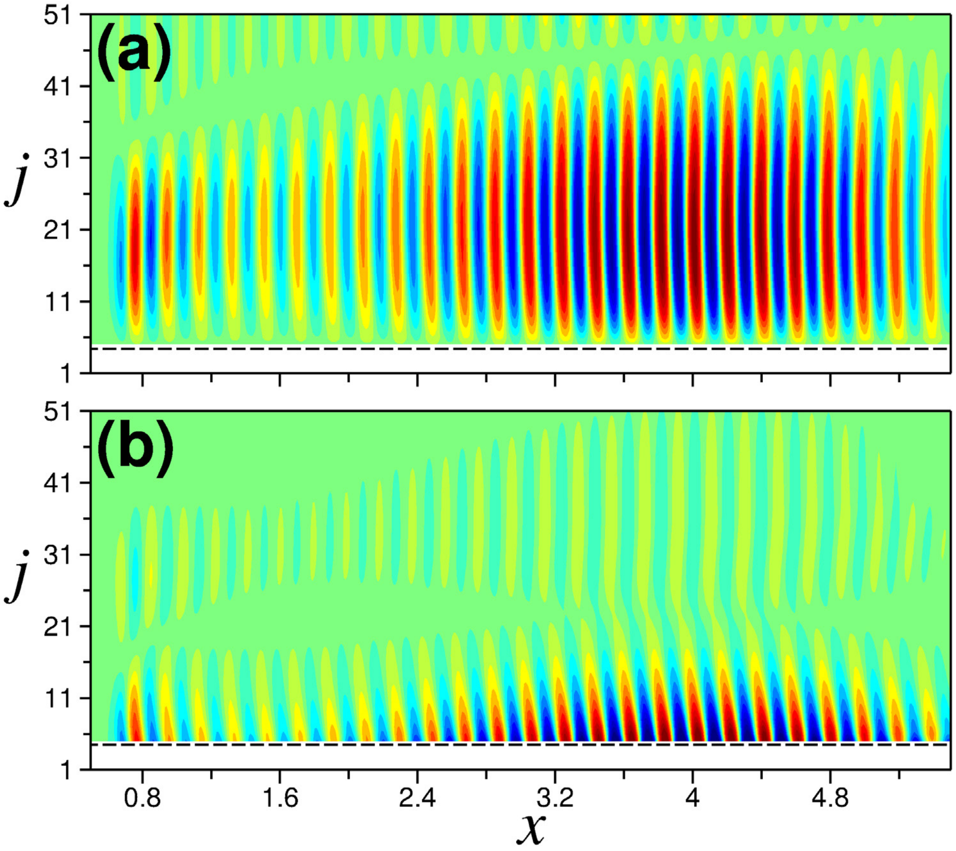

Figure 28. (a) contutors of disturbance streamwise velocity. (b) contours of disturbance spanwise vorticity. Dashed lines correspond to an immersed wall located at $y = 0$.

The wall-normal Fourier amplitude distribution of the streamwise disturbance velocity and the spanwise disturbance vorticity are shown in Fig. 29. Again, the agreement between the body-fitted results and the immersed boundary results is excellent. Once more it is noted that the sharp interface in ω_z-amplitude is captured perfectly by the IIM. Finally, amplification curves, normalized by their respective minimum at the first branch of the unstable region, are presented in Fig. 30. The results show excellent agreement between the new IIM and the body-fitted code.

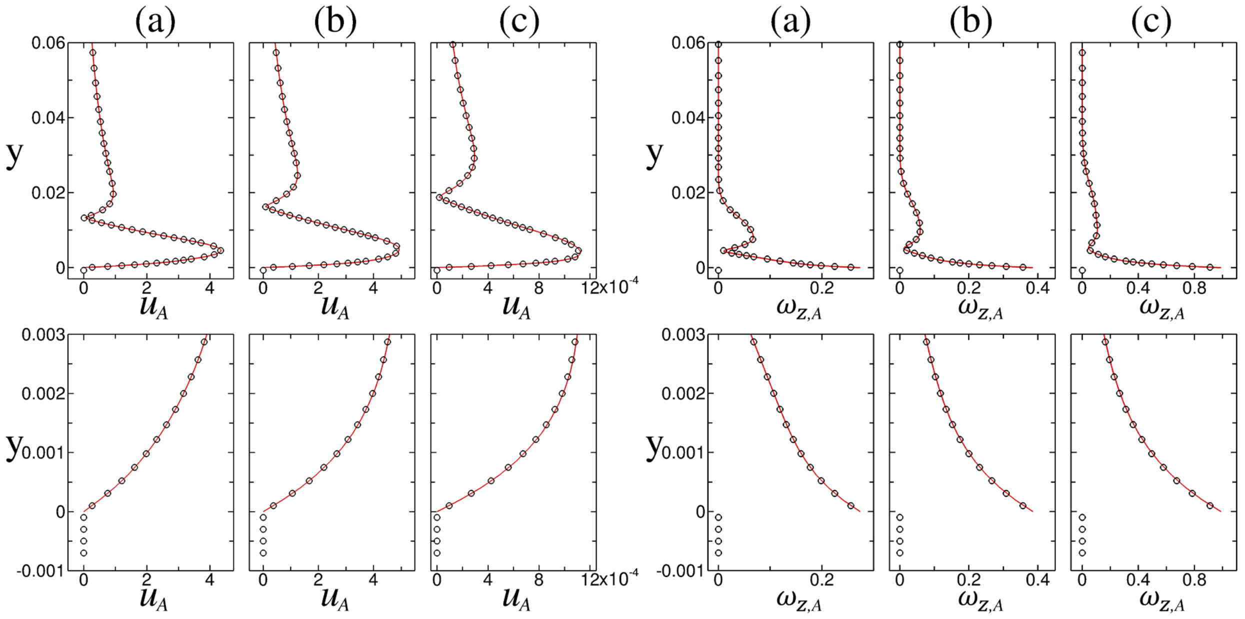

Figure 29. Comparison of wall-normal amplitude distribution of disturbance u-velocity (left) and disturbance ω_z-vorticity (right) profiles at several downstream locations. (a) $x = 1.22$ ($Re_{\delta_1} = 600$); (b) $x = 2.16$ ($Re_{\delta_1} = 800$); (c) $x = 3.38$ ($Re_{\delta_1} = 1000$). Symbols indicate values computed using an immersed wall located at $y = 0$, and solid lines results from the code nst2d of Meitz and Fasel.[28] Lower figure is a zoom-in of the upper figure near $y = 0$.

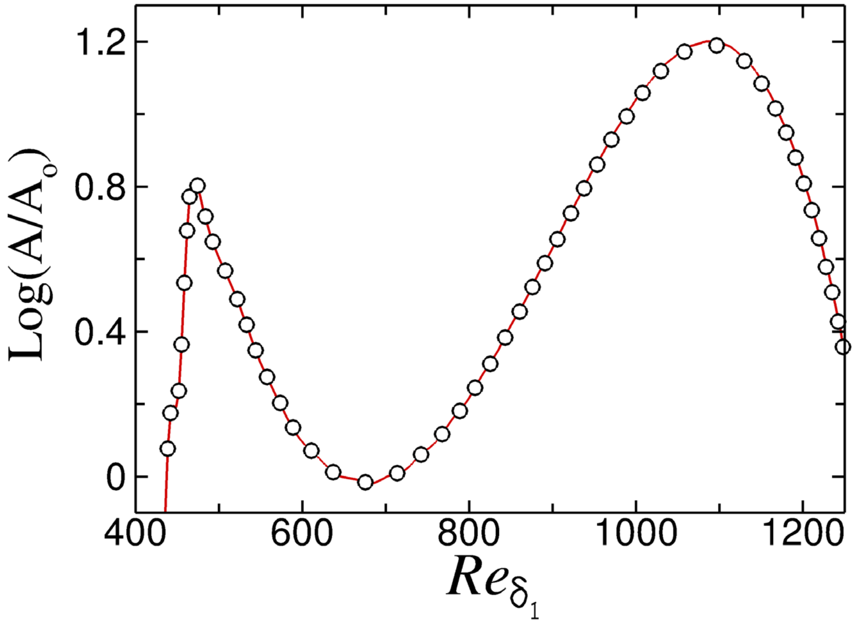

Figure 30. Comparison of the growth rate of a 2D TS wave. Symbols indicate values computed using an immersed wall located at $y = 0$, and solid line represents results from the code nst2d of Meitz and Fasel.[28] Here A represnts the wall-normal maximum of the disturabcne u-velcoity.

VII. Conclusions

In this paper, the development of very high-order (sixth-order-accuatre) sharp Immersed Interface Method for the computation of unsteady, incompressible flows is discussed in detail. By separately using the convection-diffusion and Poisson equations as model equations, the accuracy and stability of the new IIM has been demonstrated.

Combined Compact Difference (CCD) schemes are used to approximate the first and second derivatives for regular grid points. Finite-difference stencils for derivatives are constructed for a non-equidistant grid instead of using a coordinate transformation. A numerical analysis of the truncation errors and a Fourier analysis of the dispersive and dissipation errors demonstrates the superiority of the CCD schemes compared to other explicit and compact difference schemes. To avoid discontinuities across the interface of the immersed boundary, a sharp interface method is introduced where the numerical scheme takes such discontinuities into consideration in its design. Jump-correction terms are added to the finite difference stencils near the interface of the immersed boundary to maintain high formal accuracy. For the interface-matched grid points, the CCD schemes are tuned and adjusted accordingly to obtain numerically stable schemes and no jump correction will therefore be required.

Detailed stability calculations in terms of both semi- and fully-discrete eigenvalue problems are presented for the pure advection, pure diffusion and advection-diffusion equations. These calculations show that the new IIM satisfies the necessary and sufficient conditions for stability. To demonstrate the numerical stability, the new IIM is applied to the two-dimensional linear advection equation in a square domain with an immersed body.

In addition, in the present paper a sharp-interface high-order method for solving the Poisson equation on a non-uniform grid with immersed boundaries is presented. Fourth-order compact difference schemes are used to discretize the Poisson equation on the non-uniform mesh resulting in nine-point FD regular stencils. The sharp representation of the interface of the immersed body is accomplished by modifying the finite-difference stencils at irregular grid points near the immersed boundary. In particular, additional grid points are added to the standard nine-point FD stencils to obtain a sharp solution across the interface while retaining the fourth-order formal accuracy.

A new solution strategy is designed to solve the large sparse algebraic system that arises from the discretization of Poisson equation. This new strategy is based on a multiscale multigrid method that reduces the error to the level of discretization error in a fixed number of iterations which is independent of the grid size. A standard multigrid V-cycle technique is used to build the MSMG method and the corresponding operators are properly derived and constructed on stretched grids with irregular boundaries. Specifically, the relaxation operator is modified to easily accommodate the irregularly located entries introduced by immersed boundaries without losing the performance of the original algorithm. The accuracy and the efficiency of the new approach is demonstrated for a domain with and without an immersed boundary on a non-uniform mesh. The numerical results show that the solutions are sharp across the interface and that the new method is fourth-order-accurate in the maximum-norm including the irregular grid points near the interface of the immersed boundary. In addition, the solution technique is found to be very efficient in the sense that the same computational efficiency is obtained for problems with or without irregular boundaries.

Finally, the new IIM is used to solve the incompressible Navier-Stokes equations in vorticity-velocity formulation. Very good agreement was found between the present IIM results and previous benchmark solutions for a uniform flow past a circular cylinder. The second validation case was the propagation of a Tollmien-Schlichting wave in a zero-pressure-gradient flat-plate boundarylayer. The agreement between the results from the body-fitted code and the IIM results was excellent. This test case demonstrates that new IIM can be used for transition research where the near wall accuracy is critically important to accurately predict the stability characteristics of wall-bounded flows.

Appendix A

The coefficients for the CCD scheme for first derivative (Eq. 5) on a non-uniform grid are obtained by solving the following system of equations:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ -1 & -1 & 0 & 0 & 0 & dx & dx_b & 0 \\ -dx_b & -dx_f & -1 & -1 & -1 & \frac{dx^2}{2!} & \frac{dx_b^2}{2!} & 0 \\ -\frac{dx_b^2}{2!} & -\frac{dx_f^2}{2!} & -dx_b & 0 & -dx_f & \frac{dx^3}{3!} & \frac{dx_b^3}{3!} & 0 \\ -\frac{dx_b^3}{3!} & -\frac{dx_f^3}{3!} & -\frac{dx_b^2}{2!} & 0 & -\frac{dx_f^2}{2!} & \frac{dx^4}{4!} & \frac{dx_b^4}{4!} & 0 \\ -\frac{dx_b^4}{4!} & -\frac{dx_f^4}{4!} & -\frac{dx_b^3}{3!} & 0 & -\frac{dx_f^3}{3!} & \frac{dx^5}{5!} & \frac{dx_b^5}{5!} & 0 \\ -\frac{dx_b^5}{5!} & -\frac{dx_f^5}{5!} & -\frac{dx_b^4}{4!} & 0 & -\frac{dx_f^4}{4!} & \frac{dx^6}{6!} & \frac{dx_b^6}{6!} & 0 \\ -\frac{dx_b^6}{6!} & -\frac{dx_f^6}{6!} & -\frac{dx_b^5}{5!} & 0 & -\frac{dx_f^5}{5!} & \frac{dx^7}{7!} & \frac{dx_b^7}{7!} & 0 \end{bmatrix} \begin{bmatrix} \alpha_{i,1}^{(1)} \\ \alpha_{i,2}^{(1)} \\ \alpha_{i,3}^{(1)} \\ \alpha_{i,4}^{(1)} \\ \alpha_{i,5}^{(1)} \\ c_{i,1}^{(1)} \\ c_{i,2}^{(1)} \\ c_{i,3}^{(1)} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{A.1a}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ -1 & -1 & -1 & 0 & 0 & dx_b & 0 & dx_f \\ -dx_b & 0 & -dx_f & -1 & -1 & \frac{dx_b^2}{2!} & 0 & \frac{dx_f^2}{2!} \\ -\frac{dx_b^2}{2!} & 0 & -\frac{dx_f^2}{2!} & -dx_b & -dx_f & \frac{dx_b^3}{3!} & 0 & \frac{dx_f^3}{3!} \\ -\frac{dx_b^3}{3!} & 0 & -\frac{dx_f^3}{3!} & -\frac{dx_b^2}{2!} & -\frac{dx_f^2}{2!} & \frac{dx_b^4}{4!} & 0 & \frac{dx_f^4}{4!} \\ -\frac{dx_b^4}{4!} & 0 & -\frac{dx_f^4}{4!} & -\frac{dx_b^3}{3!} & -\frac{dx_f^3}{3!} & \frac{dx_b^5}{5!} & 0 & \frac{dx_f^5}{5!} \\ -\frac{dx_b^5}{5!} & 0 & -\frac{dx_f^5}{5!} & -\frac{dx_b^4}{4!} & -\frac{dx_f^4}{4!} & \frac{dx_b^6}{6!} & 0 & \frac{dx_f^6}{6!} \\ -\frac{dx_b^6}{6!} & 0 & -\frac{dx_f^6}{6!} & -\frac{dx_b^5}{5!} & -\frac{dx_f^5}{5!} & \frac{dx_b^7}{7!} & 0 & \frac{dx_f^7}{7!} \end{bmatrix} \begin{bmatrix} \beta_{i,1}^{(1)} \\ \beta_{i,2}^{(1)} \\ \beta_{i,3}^{(1)} \\ \beta_{i,4}^{(1)} \\ \beta_{i,5}^{(1)} \\ d_{i,1}^{(1)} \\ d_{i,2}^{(1)} \\ d_{i,3}^{(1)} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{A.1b}$$

Similarly, for the second derivative (Eq. 6), the coefficients are obtained using the following system of equations:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ -1 & -1 & 0 & 0 & dx_b & 0 & dx_f \\ -dx_b & -dx_f & -1 & -1 & \frac{dx_b^2}{2!} & 0 & \frac{dx_f^2}{2!} \\ -\frac{dx_b^2}{2!} & -\frac{dx_f^2}{2!} & -dx_b & -dx_f & \frac{dx_b^3}{3!} & 0 & \frac{dx_f^3}{3!} \\ -\frac{dx_b^3}{3!} & -\frac{dx_f^3}{3!} & -\frac{dx_b^2}{2!} & -\frac{dx_f^2}{2!} & \frac{dx_b^4}{4!} & 0 & \frac{dx_f^4}{4!} \\ -\frac{dx_b^4}{4!} & -\frac{dx_f^4}{4!} & -\frac{dx_b^3}{3!} & -\frac{dx_f^3}{3!} & \frac{dx_b^5}{5!} & 0 & \frac{dx_f^5}{5!} \\ -\frac{dx_b^5}{5!} & -\frac{dx_f^5}{5!} & -\frac{dx_b^4}{4!} & -\frac{dx_f^4}{4!} & \frac{dx_b^6}{6!} & 0 & \frac{dx_f^6}{6!} \end{bmatrix} \begin{bmatrix} \alpha_{i,1}^{(2)} \\ \alpha_{i,2}^{(2)} \\ \alpha_{i,3}^{(2)} \\ \alpha_{i,4}^{(2)} \\ c_{i,1}^{(2)} \\ c_{i,2}^{(2)} \\ c_{i,3}^{(2)} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{A.2a}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \\ -1 & -1 & -1 & 0 & 0 & dx_b & 0 & dx_f \\ -dx_b & 0 & -dx_f & -1 & -1 & \frac{dx_b^2}{2!} & 0 & \frac{dx_f^2}{2!} \\ -\frac{dx_b^2}{2!} & 0 & -\frac{dx_f^2}{2!} & -dx_b & -dx_f & \frac{dx_b^3}{3!} & 0 & \frac{dx_f^3}{3!} \\ -\frac{dx_b^3}{3!} & 0 & -\frac{dx_f^3}{3!} & -\frac{dx_b^2}{2!} & -\frac{dx_f^2}{2!} & \frac{dx_b^4}{4!} & 0 & \frac{dx_f^4}{4!} \\ -\frac{dx_b^4}{4!} & 0 & -\frac{dx_f^4}{4!} & -\frac{dx_b^3}{3!} & -\frac{dx_f^3}{3!} & \frac{dx_b^5}{5!} & 0 & \frac{dx_f^5}{5!} \\ -\frac{dx_b^5}{5!} & 0 & -\frac{dx_f^5}{5!} & -\frac{dx_b^4}{4!} & -\frac{dx_f^4}{4!} & \frac{dx_b^6}{6!} & 0 & \frac{dx_f^6}{6!} \\ -\frac{dx_b^6}{6!} & 0 & -\frac{dx_f^6}{6!} & -\frac{dx_b^5}{5!} & -\frac{dx_f^5}{5!} & \frac{dx_b^7}{7!} & 0 & \frac{dx_f^7}{7!} \end{bmatrix} \begin{bmatrix} \beta_{i,1}^{(2)} \\ \beta_{i,2}^{(2)} \\ \beta_{i,3}^{(2)} \\ \beta_{i,4}^{(2)} \\ \beta_{i,5}^{(2)} \\ d_{i,1}^{(2)} \\ d_{i,2}^{(2)} \\ d_{i,3}^{(2)} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{A.2b}$$

where $dx_f = x_{i+1} - x_i$, $dx_b = x_{i-1} - x_i$ and $dx = x_{i-2} - x_i$.

Appendix B

The coefficients for the fourth-order-accurate boundary closure at node $i = 2$ (Eq. 7) are given by

$$\begin{bmatrix} 0 & 0 & 1 & 1 & 1 \\ -1 & 0 & dx_b & 0 & dx_f \\ -dx_f & -1 & \frac{dx_b^2}{2!} & 0 & \frac{dx_f^2}{2!} \\ -\frac{dx_f^2}{2!} & -dx_f & \frac{dx_b^3}{3!} & 0 & \frac{dx_f^3}{3!} \\ -\frac{dx_f^3}{3!} & -\frac{dx_f^2}{2!} & \frac{dx_b^4}{4!} & 0 & \frac{dx_f^4}{4!} \end{bmatrix} \begin{bmatrix} \alpha_{b2,1}^{(1)} \\ \alpha_{b2,2}^{(1)} \\ c_{b2,1}^{(1)} \\ c_{b2,2}^{(1)} \\ c_{b2,3}^{(1)} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{B.1a}$$

$$\begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 1 \\ -1 & 0 & dx_b & 0 & dx_f & dx_1 \\ -dx_f & -1 & \frac{dx_b^2}{2!} & 0 & \frac{dx_f^2}{2!} & \frac{dx_1^2}{2!} \\ -\frac{dx_f^2}{2!} & -dx_f & \frac{dx_b^3}{3!} & 0 & \frac{dx_f^3}{3!} & \frac{dx_1^3}{3!} \\ -\frac{dx_f^3}{3!} & -\frac{dx_f^2}{2!} & \frac{dx_b^4}{4!} & 0 & \frac{dx_f^4}{4!} & \frac{dx_1^4}{4!} \\ -\frac{dx_f^4}{4!} & -\frac{dx_f^3}{3!} & \frac{dx_b^5}{5!} & 0 & \frac{dx_f^5}{5!} & \frac{dx_1^5}{5!} \end{bmatrix} \begin{bmatrix} \beta_{b2,1}^{(1)} \\ \beta_{b2,2}^{(1)} \\ d_{b2,1}^{(1)} \\ d_{b2,2}^{(1)} \\ d_{b2,3}^{(1)} \\ d_{b2,4}^{(1)} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{B.1b}$$

where $dx_f = x_{i+1} - x_i$ and $dx_b = x_{i-1} - x_i$ and $dx_1 = x_{i+2} - x_i$ with $i = 2$. The following linear system of equations are solved to obtain the coefficients for fourth-order-accurate boundary closure at $i = N$ in Eq. 8

$$\begin{bmatrix} 0 & 0 & 1 & 1 & 1 \\ -1 & 0 & 0 & dx_1 & dx_2 \\ -dx_1 & -1 & 0 & \frac{dx_1^2}{2!} & \frac{dx_2^2}{2!} \\ -\frac{dx_1^2}{2!} & -dx_1 & 0 & \frac{dx_1^3}{3!} & \frac{dx_2^3}{3!} \\ -\frac{dx_1^3}{3!} & -\frac{dx_1^2}{2!} & 0 & \frac{dx_1^4}{4!} & \frac{dx_2^4}{4!} \end{bmatrix} \begin{bmatrix} \alpha_{bN,1}^{(1)} \\ \alpha_{bN,2}^{(1)} \\ c_{bN,1}^{(1)} \\ c_{bN,2}^{(1)} \\ c_{bN,3}^{(1)} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{B.2a}$$

$$\begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 1 \\ -1 & 0 & 0 & dx_1 & dx_2 & dx_3 \\ -dx_1 & -1 & 0 & \frac{dx_1^2}{2!} & \frac{dx_2^2}{2!} & \frac{dx_3^2}{2!} \\ -\frac{dx_1^2}{2!} & -dx_1 & 0 & \frac{dx_1^3}{3!} & \frac{dx_2^3}{3!} & \frac{dx_3^3}{3!} \\ -\frac{dx_1^3}{3!} & -\frac{dx_1^2}{2!} & 0 & \frac{dx_1^4}{4!} & \frac{dx_2^4}{4!} & \frac{dx_3^4}{4!} \\ -\frac{dx_1^4}{4!} & -\frac{dx_1^3}{3!} & 0 & \frac{dx_1^5}{5!} & \frac{dx_2^5}{5!} & \frac{dx_3^5}{5!} \end{bmatrix} \begin{bmatrix} \beta_{bN,1}^{(1)} \\ \beta_{bN,2}^{(1)} \\ d_{bN,1}^{(1)} \\ d_{bN,2}^{(1)} \\ d_{bN,3}^{(1)} \\ d_{bN,4}^{(1)} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{B.2b}$$

where $dx_1 = x_{i-1} - x_i$, $dx_2 = x_{i-2} - x_i$ and $dx_3 = x_{i-3} - x_i$ with $i = N$. The coefficients for boundary closures are derived through matching the Taylor series expansions on a non-uniform grid.

Appendix C

The coefficients for interface-matched grid points in Eq. 35 are obtained by solving the following linear system of equations:

$$\begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 1 \\ -1 & 0 & dx_f & 0 & dx_b & dx_1 \\ -dx_b & -1 & \frac{dx_f^2}{2!} & 0 & \frac{dx_b^2}{2!} & \frac{dx_1^2}{2!} \\ -\frac{dx_b^2}{2!} & -dx_b & \frac{dx_f^3}{3!} & 0 & \frac{dx_b^3}{3!} & \frac{dx_1^3}{3!} \\ -\frac{dx_b^3}{3!} & -\frac{dx_b^2}{2!} & \frac{dx_f^4}{4!} & 0 & \frac{dx_b^4}{4!} & \frac{dx_1^4}{4!} \\ -\frac{dx_b^4}{4!} & -\frac{dx_b^3}{3!} & \frac{dx_f^5}{5!} & 0 & \frac{dx_b^5}{5!} & \frac{dx_1^5}{5!} \end{bmatrix} \begin{bmatrix} \alpha_{i,1}^{(1)} \\ \alpha_{i,2}^{(1)} \\ c_{i,1}^{(1)} \\ c_{i,2}^{(1)} \\ c_{i,3}^{(1)} \\ c_{i,4}^{(1)} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{C.1a}$$

$$\begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ -1 & 0 & dx_f & 0 & dx_b & dx_1 & dx_2 \\ -dx_b & -1 & \frac{dx_f^2}{2!} & 0 & \frac{dx_b^2}{2!} & \frac{dx_1^2}{2!} & \frac{dx_2^2}{2!} \\ -\frac{dx_b^2}{2!} & -dx_b & \frac{dx_f^3}{3!} & 0 & \frac{dx_b^3}{3!} & \frac{dx_1^3}{3!} & \frac{dx_2^3}{3!} \\ -\frac{dx_b^3}{3!} & -\frac{dx_b^2}{2!} & \frac{dx_f^4}{4!} & 0 & \frac{dx_b^4}{4!} & \frac{dx_1^4}{4!} & \frac{dx_2^4}{4!} \\ -\frac{dx_b^4}{4!} & -\frac{dx_b^3}{3!} & \frac{dx_f^5}{5!} & 0 & \frac{dx_b^5}{5!} & \frac{dx_1^5}{5!} & \frac{dx_2^5}{5!} \\ -\frac{dx_b^5}{5!} & -\frac{dx_b^4}{4!} & \frac{dx_f^6}{6!} & 0 & \frac{dx_b^6}{6!} & \frac{dx_1^6}{6!} & \frac{dx_2^6}{6!} \end{bmatrix} \begin{bmatrix} \beta_{i,1}^{(1)} \\ \beta_{i,2}^{(1)} \\ d_{i,1}^{(1)} \\ d_{i,2}^{(1)} \\ d_{i,3}^{(1)} \\ d_{i,4}^{(1)} \\ d_{i,5}^{(1)} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{C.1b}$$

where $dx_f = x_{i+1} - x_i$, $dx_b = x_{i-1} - x_i$, $dx_1 = x_{i-2} - x_i$ and $dx_2 = x_{i-3} - x_i$. The coefficients in Eq. 36 are the same as Eq. 35 and given by Eq. C.1. The coefficients in Eq. 37 are given by

$$\begin{bmatrix} 0 & 0 & 1 & 1 & 1 & 1 \\ -1 & 0 & dx_b & 0 & dx_f & dx_1 \\ -dx_f & -1 & \frac{dx_b^2}{2!} & 0 & \frac{dx_f^2}{2!} & \frac{dx_1^2}{2!} \\ -\frac{dx_f^2}{2!} & -dx_f & \frac{dx_b^3}{3!} & 0 & \frac{dx_f^3}{3!} & \frac{dx_1^3}{3!} \\ -\frac{dx_f^3}{3!} & -\frac{dx_f^2}{2!} & \frac{dx_b^4}{4!} & 0 & \frac{dx_f^4}{4!} & \frac{dx_1^4}{4!} \\ -\frac{dx_f^4}{4!} & -\frac{dx_f^3}{3!} & \frac{dx_b^5}{5!} & 0 & \frac{dx_f^5}{5!} & \frac{dx_1^5}{5!} \end{bmatrix} \begin{bmatrix} \alpha_{i,1}^{(1)} \\ \alpha_{i,2}^{(1)} \\ c_{i,1}^{(1)} \\ c_{i,2}^{(1)} \\ c_{i,3}^{(1)} \\ c_{i,4}^{(1)} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{C.2a}$$

$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \\ dx_b & 0 & dx_f & dx_1 & dx_2 & dx_3 \\ \frac{dx_b^2}{2!} & 0 & \frac{dx_f^2}{2!} & \frac{dx_1^2}{2!} & \frac{dx_2^2}{2!} & \frac{dx_3^2}{2!} \\ \frac{dx_b^3}{3!} & 0 & \frac{dx_f^3}{3!} & \frac{dx_1^3}{3!} & \frac{dx_2^3}{3!} & \frac{dx_3^3}{3!} \\ \frac{dx_b^4}{4!} & 0 & \frac{dx_f^4}{4!} & \frac{dx_1^4}{4!} & \frac{dx_2^4}{4!} & \frac{dx_3^4}{4!} \\ \frac{dx_b^5}{5!} & 0 & \frac{dx_f^5}{5!} & \frac{dx_1^5}{5!} & \frac{dx_2^5}{5!} & \frac{dx_3^5}{5!} \end{bmatrix} \begin{bmatrix} d_{i,1}^{(1)} \\ d_{i,2}^{(1)} \\ d_{i,3}^{(1)} \\ d_{i,4}^{(1)} \\ d_{i,5}^{(1)} \\ d_{i,6}^{(1)} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{C.2b}$$

where $dx_f = x_{i+1} - x_i$, $dx_b = x_{i-1} - x_i$, $dx_1 = x_{i+2} - x_i$ and $dx_2 = x_{i+3} - x_i$, $dx_3 = x_{i+4} - x_i$.

Appendix D

Coefficients for the compact difference scheme in Eqs. 66a and 66c, obtained by matching the coefficients in the Taylor expansion about $u_{i,j}$ in the x direction on a non-uniform grid, are given by

$$
\begin{aligned}
a^x_i &= \frac{dx_f(dx_b^2 + dx_b dx_f - dx_f^2)}{(dx_f^3 + 4dx_f^2 dx_b + 4dx_f dx_b^2 + dx_b^3)},\\
b^x_i &= 1,\\
c^x_i &= \frac{dx_b(dx_f^2 + dx_b dx_f - dx_b^2)}{(dx_f^3 + 4dx_f^2 dx_b + 4dx_f dx_b^2 + dx_b^3)},\\
ar^x_i &= \frac{12dx_f}{(dx_f^3 + 4dx_f^2 dx_b + 4dx_f dx_b^2 + dx_b^3)},\\
br^x_i &= \frac{-12}{(dx_f^2 + 3dx_f dx_b + dx_b^3)},\\
cr^x_i &= \frac{12dx_b}{(dx_f^3 + 4dx_f^2 dx_b + 4dx_f dx_b^2 + dx_b^3)},
\end{aligned}
\tag{D.1}
$$

where $dx_f = x_{i+1} - x_i$ and $dx_b = x_i - x_{i-1}$. The coefficients in y-direction in Eqs. 66b and 66d are the same except that we replace x with y. The coefficients for LHS in Eq. 68 are given by

$$
\begin{aligned}
A^{SW}_{i,j} &= ar^x_i a^y_j + a^x_i ar^y_j - \gamma^2 a^x_i a^y_j, & A^{S}_{i,j} &= br^x_i a^y_j + b^x_i ar^y_j - \gamma^2 b^x_i a^y_j, & A^{SE}_{i,j} &= cr^x_i a^y_j + c^x_i ar^y_j - \gamma^2 c^x_i a^y_j,\\
A^{W}_{i,j} &= ar^x_i b^y_j + a^x_i br^y_j - \gamma^2 a^x_i b^y_j, & A^{P}_{i,j} &= br^x_i b^y_j + b^x_i br^y_j - \gamma^2 b^x_i b^y_j, & A^{E}_{i,j} &= cr^x_i b^y_j + c^x_i br^y_j - \gamma^2 c^x_i b^y_j;\\
A^{NW}_{i,j} &= ar^x_i c^y_j + a^x_i cr^y_j - \gamma^2 a^x_i c^y_j, & A^{N}_{i,j} &= br^x_i c^y_j + b^x_i cr^y_j - \gamma^2 b^x_i c^y_j, & A^{NE}_{i,j} &= cr^x_i c^y_j + c^x_i cr^y_j - \gamma^2 c^x_i c^y_j.
\end{aligned}
\tag{D.2}
$$

The coefficients for the RHS in Eq. 69 are

$$
\begin{aligned}
Q^{SW}_{i,j} &= a^x_i a^y_j, & Q^{S}_{i,j} &= b^x_i a^y_j, & Q^{SE}_{i,j} &= c^x_i a^y_j,\\
Q^{W}_{i,j} &= a^x_i b^y_j, & Q^{P}_{i,j} &= b^x_i b^y_j, & Q^{E}_{i,j} &= c^x_i b^y_j,\\
Q^{NW}_{i,j} &= a^x_i c^y_j, & Q^{N}_{i,j} &= b^x_i c^y_j, & Q^{NE}_{i,j} &= c^x_i c^y_j.
\end{aligned}
\tag{D.3}
$$

Finally, the modified coefficients in Eq. 77 are

$$
\begin{aligned}
\overline{A^{S}_{i,j}} &= \overline{br^x_i} a^y_j + b^x_i ar^y_j - \gamma^2 b^x_i a^y_j,\\
\overline{A^{SE}_{i,j}} &= \overline{cr^x_i} a^y_j + c^x_i ar^y_j - \gamma^2 c^x_i a^y_j,\\
\overline{A^{P}_{i,j}} &= \overline{\overline{br^x_i}} b^y_j + b^x_i br^y_j - \gamma^2 b^x_i b^y_j,\\
\overline{A^{E}_{i,j}} &= \overline{\overline{cr^x_i}} b^y_j + c^x_i br^y_j - \gamma^2 c^x_i b^y_j,\\
\overline{A^{NW}_{i,j}} &= ar^x_i c^y_j + a^x_i \overline{cr^y_j} - \gamma^2 a^x_i c^y_j.
\end{aligned}
\tag{D.4}
$$

Acknowledgments

This work was supported by the Air Force Office of Scientific Research (AFOSR) under grant number FA9550-14-1-0184, with Dr. Douglas Smith serving as the program manager.

References

[1]Mittal, R., Bozkurttas, D. M., Najjar, F., Vargas, A., and von Loebbecke, A., “A versatile sharp interface immersed boundary method for incompressible flows with complex boundaries,” *J. Comp. Phys*. 227, 4825–4852, 2008.

[2]Brehm, C., and Fasel, H. F., “Immersed interface method for solving the incompressible Navier–Stokes equations with moving boundaries,” *AIAA-2011-0758*, 49th AIAA Aerospace Sciences Meeting, 2011.

[3]Linnick, M., and Fasel, H. F., “A high-order immersed interface method for simulating unsteady incompressible flows on irregular domains,” *J. Comput. Phys*. 204, 157–192, 2004.

[4]Duan, L., Wang, X., and Zhong, X., “A high-order cut-cell method for numerical simulation of hypersonic boundary-layer instability with surface roughness,” *J.Comput. Phys*. 229, 7207–7237, 2010.

[5]Mittal, R., and Iaccarino,G., “Immersed boundary methods,” *Annu. Rev. Fluid Mech*. 37, 239–261, 2005.

[6]Marella, S., Krishnan, S., Liu, H., and Udaykumar, H. S., “Sharp interface Cartesian grid method I: an easily implemented technique for 3D moving boundary computations,” *J. Comput. Phys*. 210 (1), 1–31, 2005.

[7]Peskin, C. S., “Flow patterns around heart valves: a numerical method,” *J. Comput. Phys*. 10 (2), 252–271, 1972.

[8]Peskin, C. S., “Numerical analysis of blood flow in the heart,” *J. Comput. Phys*. 25 (3), 220–252, 1977.

[9]Lai, M. C., and Peskin, C. S., “An immersed boundary method with formal second-order accuracy and reduced numerical viscosity,” *J. Comput. Phys*. 160 (2) , 705–719, 2000.

[10]Su, S. W., Lai, M. C., Lin, C. A., “An immersed boundary technique for simulating complex flows with rigid boundary,” Comput. Fluids 36 (2), 313–324, 2007.

[11]Tseng, Y. H., and Ferziger, J. H., “A ghost-cell immersed boundary method for flow in complex geometry,” *J. Comput. Phys*. 192 (2), 593–623, 2003.

[12]Udaykumar, H. S, Mittal, R., Rampunggoon, P., and Khanna, A., “A sharp interface Cartesian grid method for simulating flows with complex moving boundaries,” *J. Comput. Phys*. 174:345–80, 2001.

[13]Udaykumar, H. S, Mittal, R., and Rampunggoon, P., “Interface tracking finite volume method for complex solid-fluid interactions on fixed meshes,” *Commun. Numer. Methods Eng*. 18:89–97, 2002.

[14]Leveque, R. J., and Li, Z., “The immersed interface method for elliptic equations with discontinuous coefficients and singular sources,” *SIAM J. Numer. Anal*. 31 (4),1019–1044, 1994.

[15]Wiegmann, A., and Bube, K. P., “The explicit-jump immersed interface method: finite difference methods for PDEs with piecewise smooth solutions,” *SIAM J. Numer. Anal*. 37 (3), 827–862, 2000.

[16]Li, C. W., and Wang, L. L., “An immersed boundary finite difference method for LES of ow around blu_ shapes,” *Int. J. Numer. Meth. Fluids*, 46(1):85, 2004.

[17]Brehm, C., and Fasel, H. F., “A novel concept for the design of immersed interface methods,” *J. Comput. Phys*. 242, 234–267, 2013.

[18]Brehm, C., Hader, C., and Fasel, H. F., “A locally stabilized immersed boundary method for the compressible Navier–Stokes equations”, *J. Comp. Phys*., 295, 475–504, 2015.

[19]Linnick, M., “A High-Order Immersed Boundary Method for Unsteady Incompressible Flow Calculations,” *PhD Thesis*, University of Arizona, 2003.

[20]Mahesh, K., “A Family of High Order Finite Difference Schemes with Good Spectral Resolution,” *J of Comp Physics*. 145, 332–358, 1998.

[21]Chu., P. C., and Fan, C., “A Three-Point Combined Compact Difference Scheme,” *J. Comput. Phys*. 140, 370–399, 1998.

[22]Lele, K., “Compact finite difference schemes with spectral-like resolution,” *J. Comput. Phys*. 103, 16, 1992.

[23]Wang. Y and Zhang. J., “Sixth order compact scheme combined with multigrid method and extrapolation technique for 2D Poisson equation,” *J. Comp. Phys*., 228(1), 137-146, 2009.

[24]Schneider, G.E and Zedan, M., “A Coupled Strongly Implicit Procedure For Velocity And Pressure Computation In Fluid Flow Problems,” J. Num Heat Transfer, Vol. 8,1985.

[25]Shukla, R.K., Tatineni, M and Zhong, X., “Very high-order compact finite difference schemes on non-uniform grids for incompressible Navier–Stokes equations,” *J. Comp. Phys*., 224(2), 1064-1094, 2007.

[26]Coutanceau, M., and Bouard, R., “Experimental Determination of the Main Features of the Viscous Flow in the Wake of a Circular Cylinder in Uniform Translation. Part 1, Steady Flow.,” *J. Fluid Mech*. 79, 231–256, 1977.

[27]Brehm, C., “Novel Immersed Interface Method for Solving the Incompressible Navier-Stokes Equations,” *PhD Thesis*, University of Arizona, 2011.

[28]Meitz, H. & Fasel, H. F., “A compact-difference scheme for the Navier-Stokes equations in vorticity-velocity formulation,” *J. Comp. Phys*. 157, 371–403, 2000.