Oral presentation | Numerical methods

Numerical methods-V

Wed. Jul 17, 2024 4:30 PM - 6:30 PM Room A

[9-A-01] A Mass-Conservative Immersed Boundary Method for Compressible Flows

Kasturirangan Madabushi[1], *Santanu Ghosh[1] (1. Indian Institute of Technology Madras)

Keywords: Immersed boundary method, Face-based forcing, Mass conservation, Sharp Interface method

Twelfth International Conference on Computational Fluid Dynamics (ICCFD12), Kobe, Japan, July 14-19, 2024

A Mass-Conservative Immersed Boundary Method for Compressible Flows

Kasturirangan Madabushi*, Santanu Ghosh*
Corresponding author: sghosh1@smail.iitm.ac.in
* Indian Institute of Technology, Madras.

Abstract: In this study, the immersed-boundary method developed in "Kasturi Rangan, M. L. N. V., and Ghosh, S. A Face-Based Immersed Boundary Method for Compressible Flows Using a Uniform Interpolation Stencil. Frontiers in Mechanical Engineering, Vol. 8, No. October, 2022, pp. 1–17" for compressible viscous flows is further improved to account for better mass conservation. In this method, a face-based solution forcing approach is considered to achieve local conservation in the control volumes near the embedded boundary leveraging the finite-volume framework. However, in this approach, there is a lack of mass (or any convective flux) conservation on two accounts: Lack of geometric conservation and a source-sink effect due to net mass inflow/outflow in interior cells near the immersed boundary. In this work, we look at a two-fold approach to address these issues. Firstly, a cut-cell type approach is proposed to determine the effective wetted area of a cell face intersected by the immersed boundary to improve geometric conservation. Secondly, the mass (and other convective) flux at cell faces where solution forcing is done is corrected to account for the mass source/sink effect. The proposed method(s) are validated against standard inviscid and viscous compressible flow test cases. The results are compared with simulations from literature using contours of flow properties, surface pressure. Finally net mass balance in the computational domain is determined for the different test cases and compared with the baseline formulation to check for improvement in mass conservation.

Keywords: Immersed boundary method, Face-based forcing, Mass conservation, Sharp Interface method, Computational Fluid Dynamics.

1 Introduction

Even with the advent of high-performance computing and parallelization with CPU/GPUs, there is still a need to mitigate the excessive use of computational resources that add to the cost. The computational cost increases according to the geometry's complexity and when the body is in motion. This can be attributed to the grid generation process involved. Grid generation over a complex geometry is challenging, and regenerating the grid at every time step for moving body problems is computationally exhaustive. Therefore, it is preferred to devise methods that do not involve repetitive grid generation without compromising the accuracy and efficacy of the solvers involved. Immersed boundary methods (IBMs) have emerged as powerful tools to address this challenge by enabling the simulation of fluid flow around arbitrarily shaped boundaries without the need for complex mesh generation. Originally developed by Peskin [1, 2, 3] for modeling heart valves, IBMs have since evolved into a diverse family of techniques that find application across various fields, from bio-fluid dynamics to aerospace engineering. These methods aim to achieve the accuracy offered by existing body-fitted grid algorithms while reducing the computational burden. Fundamentally, an immersed boundary method represents the embedded boundary as a collection of points/markers, or connectted line segments (2D) or facets (3D) that move/ remain independant of the underlying Cartesian grid. This decoupling of the boundary representation from the (volume) grid allows for efficient simulations of flows past intricate geometries, including moving or deforming boundaries.

Immersed boundary methods are broadly categorized into two types: continuous forcing and direct-forcing methods. Continuous forcing methods involve using a forcing function to incorporate the *"body's effects"* into the fluid domain. The earliest use of this method is found in the work by Peskin [1] and subsequently used by Goldstein et al.[4]. In contrast, direct-forcing methods directly modify variables near the immersed boundary to enforce the boundary conditions. These methods excel at capturing sharp object boundaries that are often diffused in continuous forcing methods due to the smoothing effect of the forcing function. Moreover, unlike continuous forcing methods, which are constrained by time-step limitations[5, 6], these methods do not encounter such issues. Direct-forcing methods have gained prominence in immersed boundary modeling, with numerous variants proposed in the literature [7, 8, 9, 10, 11]. However, direct forcing methods [12] encounter challenges with mass conservation, primarily due to lack of geometric conservation and solution forcing (instead of being obtained from solving the governing equations) in some cells. Efforts such as those by Kim et al. [8] have proposed augmenting direct forcing methods with mass source/sink terms in the continuity equation to mitigate non-physical flux across immersed boundaries in incompressible flow scenarios. This issue of lack of conservation is however not present in cut-cell methods, a type of Cartesian grid approach wherein the cell boundaries in the vicinity of the IB are reconstructed to adhere to the embedded geometry. However, these methods are computationally intensive and the formation of low-volume split cells pose a significant issue that various methods have attempted to address using techniques like cell-merging and cell-linking[13, 14, 15, 16, 17]. On the other hand, direct forcing methods are favored for their straightforward formulation and implementation, making them popular for simulating complex geometries and moving bodies.

This work extends an immersed boundary method [18] developed by the authors in a finite-volume framework that uses a face-based solution forcing in the vicinity of the immersed surface. Solution forcing at cell faces allows the integration of the discretized equations in all the fluid cells, thus allowing better adherence to the conservation of mass, momentum, and energy. The extension of this face-based approach involves two main aspects. Firstly, a cut-cell-type approach is proposed to determine the effective wetted area of a cell face intersected by the immersed boundary, aiming to enhance geometric conservation. Secondly, corrections are applied to the mass (and other convective) flux(es) at cell faces where solution forcing occurs, accounting for mass source/sink effects. This flux correction, often referred to as "virtual mixing" in cut-cell approaches, is implemented in neighboring immersed boundary cells. The mixing technique, originally introduced by Hu et al. [19] and subsequently adopted by Meyer [20] and Seo et al. [21], is adapted in our solver. The exact mixing approach used in our work differs from those reported in the literature but aims to be easy to implement and improve the conservation properties of the existing solver. This integration aims to leverage the advantages of the cut-cell method within the context of the direct-forcing approach. The proposed method, as such, can be considered a hybrid of cut-cell and flow reconstruction methods.

This paper is structured into sections that delineates in a sequential manner the proposed IBM and its approach to improve mass conservation. The outline of the rest of the paper is as follows. The section(s) [2.1,2.2] provides an overview of the governing equations and details about the flow solver. Following this, a concise summary of the existing immersed boundary methodology is presented in the section 2.3. The subsequent section 2.4 is divided into two main parts: one focusing on geometric conservation (see section 2.4.1) and the other on flux redistribution (see section 2.4.2). Results from test cases investigated as part of this paper are presented and discussed in the subsequent section 3. Section 3.6 presents a quantitative analysis of the proposed IB methods in terms of net mass balance in the domain and the comparison of quantities like Mach number, stagnation pressure, lift, and drag coefficients with literature. Finally, conclusions drawn from the test cases are presented in section 4.

2 Computational Framework

In this work, the Navier–Stokes equations are discretized using a finite volume method and solved on structured grids. The governing equations, overview of the solver and the IB methodology are briefly discussed in this section.

2.1 Governing equations

In this work, the Navier-Stokes equations are discretized using a finite-volume method and solved on structured grids. The governing equations and an overview of the solver are briefly discussed in this section.

$$\frac{\partial \vec{q_c}}{\partial t} + \nabla \cdot (\vec{F_I} - \vec{F_V}) = 0 \tag{1}$$

In the above equation, $\vec{F_I}$ constitutes the inviscid fluxes and $\vec{F_V}$ constitutes viscous fluxes. The vector $\vec{q_c}$ consists of the conservative variables, which include mass, momentum, and energy per unit volume.

$$\vec{q}_c = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ \rho w \\ \rho E_t \end{pmatrix} \tag{2}$$

$$\vec{F}_I = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho uw \\ \rho u H_t \end{pmatrix} \hat{i} + \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ \rho vw \\ \rho v H_t \end{pmatrix} \hat{j} + \begin{pmatrix} \rho w \\ \rho uw \\ \rho vw \\ \rho w^2 + p \\ \rho w H_t \end{pmatrix} \hat{k} \tag{3}$$

$$\vec{F}_V = \begin{pmatrix} 0 \\ t_{xx} \\ t_{xy} \\ t_{xz} \\ ut_{xx} + vt_{xy} + wt_{xz} - q_x \end{pmatrix} \hat{i} + \begin{pmatrix} 0 \\ t_{yx} \\ t_{yy} \\ t_{yz} \\ ut_{yx} + vt_{yy} + wt_{yz} - q_y \end{pmatrix} \hat{j} + \begin{pmatrix} 0 \\ t_{zx} \\ t_{zy} \\ t_{zz} \\ ut_{zx} + vt_{zy} + wt_{zz} - q_z \end{pmatrix} \hat{k} \tag{4}$$

In the equations above, u, v, and w are the Cartesian components of velocity along the X, Y, and Z directions, respectively, ρ is the fluid density, p is the fluid pressure and H_t is the total specific enthalpy. Also, $\tilde{\tilde{t}}$ is the laminar stress tensor and $\vec{q}$ is the laminar heat flux. In order to close the equations, the equation of state for an ideal gas is used:

$$P = \rho RT \tag{5}$$

where R is the specific gas constant for air. Molecular viscosity is modelled with Sutherland's law, and a laminar Prandtl number of 0.72 is used for air.

2.2 Flow solver

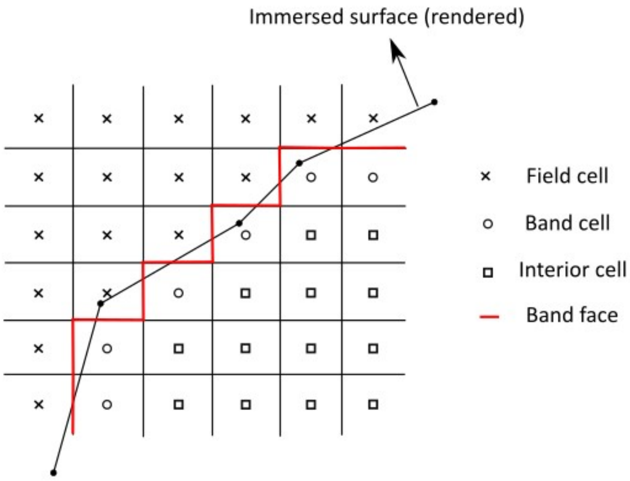
FEST-3D (Finite Volume Explicit Structured 3-Dimensional) [22], an in-house developed parallel code for structured grids based on a finite-volume framework, is used. This solver discretizes the variable density, 3D Navier-Stokes equations to simulate compressible laminar flows, and Favre-averaged Navier-Stokes equations for compressible flows. The capabilities of this solver range from laminar to turbulent regimes comprising various spatial and time-discretization schemes. In the present work, the solver is run with HLLE [23, 24] scheme for inviscid flux calculation, MUSCL for second-order reconstruction (spatial) as presented in [25] and RK4 for time integration. All the simulations presented in this work are 2D computations. Molecular viscosity is modeled with Sutherland's law, and a laminar Prandtl number of 0.72 is used for air. The scope of the work presented in this paper is specific to laminar and two-dimensional flows.

2.3 Immersed Boundary Method (Baseline Approach)

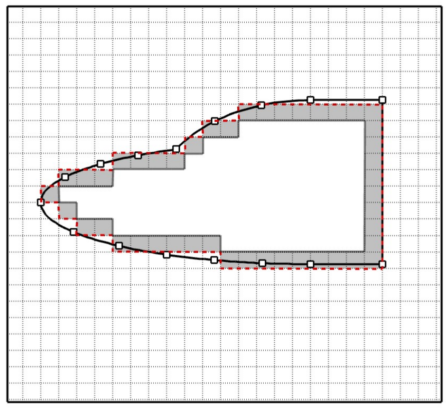
This section describes the Immersed Boundary (IB) method used in this work [26]. This is a direct-forcing-approach-based IB method involving face-based reconstruction. Face-based reconstruction is expected to achieve local conservation in control volumes by leveraging a finite-volume framework. As such, the solution for all the cells external to the IB satisfies the discretized governing equations. The face-based method takes inspiration from [27, 28], which was developed for inviscid flows and later extended to the viscous flows in the compressible regime. Details of the present method are explained in the following section(s).

2.3.1 Cell classification

This section outlines the methodology employed by IBM in this study [26]. The primary step in any IB method is classifying cells as exterior and interior. To do this, the immersed boundary is represented by a set of line segments, with outward normals specified along with the coordinates of their endpoints. This information is utilized for cell classification. Cells are categorized as field and interior cells based on a signed-distance algorithm. Band cells are then identified as interior cells sharing a face neighbor with a field cell. Furthermore, band faces are recognized as cell interfaces shared by field and interior (band) cells. The primary computations of this IB method revolve around enforcing the solution at these faces to ensure compliance with boundary conditions at the immersed surface. **Fig.** 1a illustrates the cell classification of a section of a grid with an immersed surface, which is represented by a series of line segments. Furthermore, the cell classification for the double ellipse configuration (see **Fig.** 1b investigated in this work is presented, which illustrates how the flow solver interprets the immersed surface. The cells in the entire domain are classified as field cells (white), band cells (grey), and interior cells (blanked out). Additionally, the *band* faces (colored red) are also shown in the figure.



(a) Schematic indicating cell classification.



(b) Cell classification for a double ellipse configuration.

Figure 1: Cell classification

2.3.2 Variable reconstruction

The process of solution reconstruction at a band face [26] is achieved by first constructing an interpolation point (shown in **Fig.** 2) along the normal to the immersed surface. Data is interpolated from the surrounding fluid cells onto the interpolation point using inverse distances as weights, as shown in the **Fig.** 2. In contrast to some existing immersed boundary methods[29, 28], the interpolation point is not fixed at the same predetermined distance along the normal to the immersed surface. The idea is to determine the location of the interpolation point based on the local geometry of the immersed surface and the local grid size instead of fixing it at a certain distance based on a certain algorithm, as mentioned in [26]. This ensures that more field cells are used to interpolate the primitive variables at the interpolation point. Once data is reconstructed at the interpolation point, velocity and other primitive variables are reconstructed at the band face according to the desired boundary condition on the immersed surface. The variable reconstruction at the band face is performed next and is briefly described here.

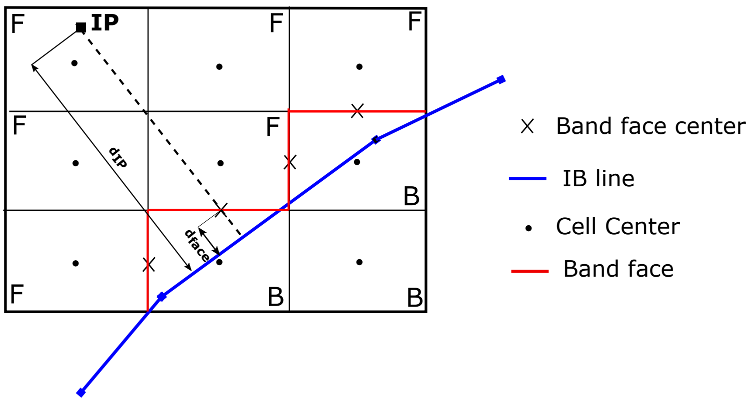


Figure 2: Stencil for determination of interpolation point.

- **Velocity reconstruction (Slip wall)**

$$\vec{u}_{face} = \vec{u}_{IP} + \left(\left[1 - \frac{d_{face}}{d_{IP}}\right](\vec{u}_{\perp,IB} - \vec{u}_{\perp,IP}) \cdot \hat{n}_{IB}\right)\hat{n}_{IB}$$

- **Velocity reconstruction (No-slip wall)**

$$\vec{u}_{face} = \vec{u}_{IB} + (\vec{u}_{IP} - \vec{u}_{IB})\frac{d_{face}}{d_{IP}}$$

- **Pressure & Temperature reconstruction**

$$\left.\frac{\partial P}{\partial n}\right|_{IB} = 0, P_{face} = P_F$$

$$\textbf{Adiabatic wall:}\, T_{face} = T_F$$

$$\textbf{Isothermal wall:}\, T_{face} = T_{wall} + (T_{IP} - T_{wall})\,\frac{d_{face}}{d_{IP}}$$

- **Gradients at band faces**

$$\vec{\nabla}\phi = \frac{2(\phi_{IP} - \phi_{face})}{-d_{IP} + d_{face}}, \phi = [\rho, u, v, P]$$

Here d_{IP} and d_{face} are the signed distances [1] from the interpolation point and the band face (center) to the immersed surface. Also, in the slip-wall velocity formulation, $u_\perp$ indicates the component of velocity normal to the immersed surface. The density at the face is reconstructed using an ideal gas equation.

2.4 Addressing mass conservation

2.4.1 Effective area method (EA-IBM)

This method is the first step towards improving mass conservation by improving geometric conservation/fidelity within the face-based direct-forcing Immersed Boundary (IB) approach. The fundamental concept of this method is to integrate the advantages of the cut-cell method, which offers better mass conservation, and the flow reconstruction methods (direct-forcing), which are relatively easy to implement.

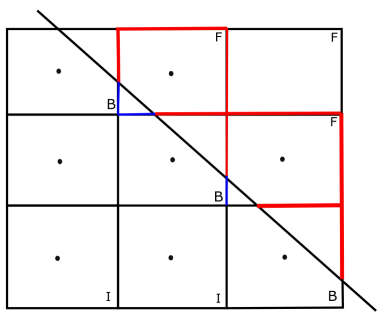


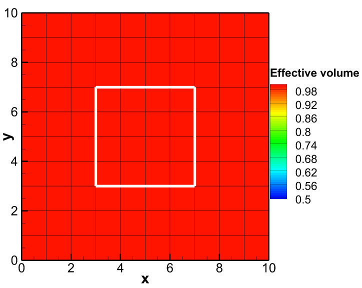
Figure 3: Cell classification with effective area highlighted

Fig. 3 shows an immersed boundary (black line) that divides the mesh into internal (marked 'I') and external (marked 'F') cells. Further, internal cells adjacent to an external cell are termed as band cells (marked 'B'). In the baseline IB method [26], the whole face area was considered while evaluating the

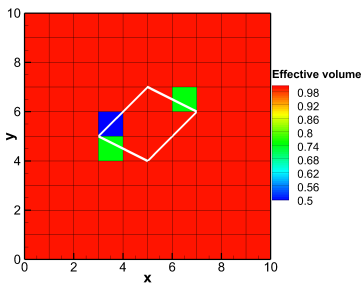
[1] In this work, the signed distance from a point to the immersed surface is considered negative if the point is external to the immersed boundary.

fluxes at a band face. Herein, we propose to use the partial area of the band face (shown in red) that physically contributes to the convective flux at that face. Specifically, the idea is to assign different areas to different flux terms based on the physical nature of the flux. To elaborate, while the combination of red and blue areas (total area of a band face), as shown in Fig. 3 is used for pressure force, viscous stress, and heat fluxes, only the areas marked in red are used for convective fluxes. It is to be noted, though, that the exact boundary of the immersed object is never reconstructed as in a cut-cell method in this approach.

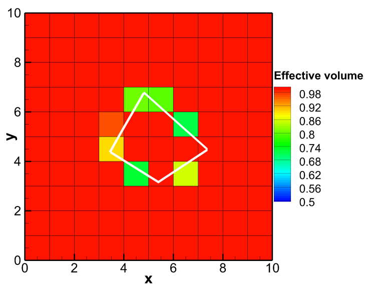
In every scenario, the key task is to identify how the cell faces intersect with the Immersed Boundary (IB), enabling the determination of the effective area for convective flux (highlighted in red). This adjustment of cell areas specifically targets band faces, and the cell volumes of the corresponding (adjacent) field cells are also adjusted. Similar to the baseline IB method, this approach doesn't evolve the solution within interior cells over time. To efficiently compute effective volumes and areas, a bounding box encompassing the immersed object is considered based on cell classification. Within this bounding box, the intersection points of the IB classifier (represented as line segments) and the faces are determined for all field cells. Subsequently, the portion of a face entirely within the fluid domain is identified. If a face is entirely within the IB, its effective area is deemed zero. Conversely, if the face lies entirely outside the IB, its effective area remains unchanged from the original. A point to note here is that cell area modifications may also be required in faces other than band faces, which is also considered here. Additionally, the volume for field cells within the bounding box is computed using the standard shoe-lace formula [30].



(a) IB (A)



(b) IB (B)



(c) IB (C)

Figure 4: Verification cases

Fig. 4 shows the contour plots of the effective volume calculated for three different immersed geometries. These verification cases are designed to mimic different scenarios encountered in determining the effective area and volume. IB (A) shown in **Fig.** 4a is a case where the geometry coincides with the grid lines. In this case, a volume change is not expected, and the same is obtained after the volume calculations. IB (B) shown in **Fig.** 4b is a scenario where the geometry goes through the diagonal of a grid cell, and IB (C) in **Fig.** 4c is a case where there are numerous single intersections on the faces and multiple (two) intersections on a particular face. All the contour plots shown in 4 show the newly calculated cell volumes as a fraction of the original cell volumes, and it is observed that the volume of any cell does not fall below 50%. The primary idea herein is to improve geometric conservation near the immersed boundary by having a better approximation of the cell face areas and volume that contribute physically to the development of the flow.

2.4.2 Flux redistribution (EA-IBM$^+$)

The approach mentioned in the previous section was aimed at improving geometric conservation in the face-based immersed boundary (IB) method developed by the authors [26]. A further development is proposed to improve the conservation properties of the present IBM, including mass conservation. The approach adopted here involves redistribution (or a correction) of the convective fluxes in the band cells to mitigate the source-sink phenomenon resulting from net inflow/outflow within interior cells. To illustrate, consider the cumulative convective fluxes at the band faces depicted in **Fig.** 5. For perfect conservation, the sum of these fluxes should be zero to prevent mass accumulation/overflow in/from the gray-shaded region of the band cell. However, this conservation is not inherently assured in the existing framework. In this paper, we attempt to address this issue by altering the computed convective fluxes [2] at these interfaces.

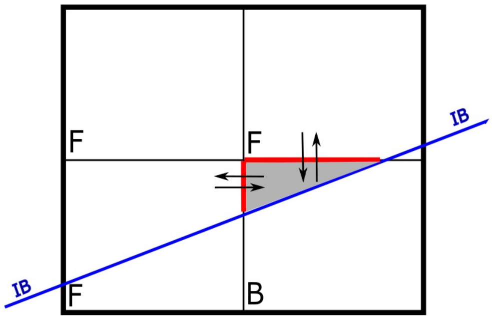


Figure 5: Schematic of mass accumulation/overflow in a band cell.

The accumulation/overflow of mass in band cells for two (most) likely scenarios are shown in **Fig.** 6. Please note that here, by accumulation, we refer to the net accumulation of mass in a band cell, which acts as a sink for the outer flow in such case. Conversely, by overflow, we refer to the net efflux of mass in a band cell, which then acts as a source for the outer flow in such case. The first scenario in **Fig.** 6 involves two band faces across which flow moves in/out to/from a band cell through the areas marked in red (wetted or 'cut' areas), which can lead to a net (rate of) accumulation/overflow of mass in/from that band cell. In contrast, the second scenario has in addition to the band faces, a face that is partially wetted, and as such allows convection of fluxes across it. This face, termed the "Pseudo" band face (shown in green), is shared by two adjacent band cells.

The approach adopted for redistribution of the accumulation/overflow of mass at each iteration for these cases is as follows: The first step in the redistribution involves determining the net residual (mass and momentum) in a band cell. The net residual within the band cell, represented by $\Re_b$, is defined as in 6. For the first situation in **Fig.** 6, it is done by summing up the convective fluxes through the band faces that enclose the band cell. For the subsequent redistribution/correction of the convective fluxes at band faces in the particular band cell, the residual $\Re_b$ is partitioned using a weighted approach. The weights are constructed by multiplying the face-normal velocity with the wetted area of each band face in the band cell. Thus, once $\Re_b$ is determined for a specific band cell; it is subsequently redistributed using the **Eqs.**(7-8).

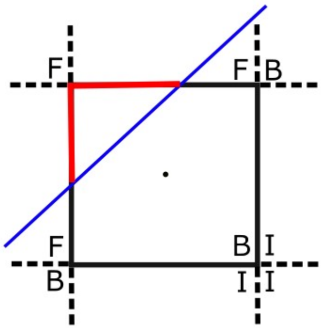
$$\Re_b = \int_S (G_i \cdot \hat{n}) dS : G = \{\rho u, \rho uu, \rho uv\} \tag{6}$$

$$w_i = |\vec{V_i} \cdot \hat{n}| \, A_i \tag{7}$$

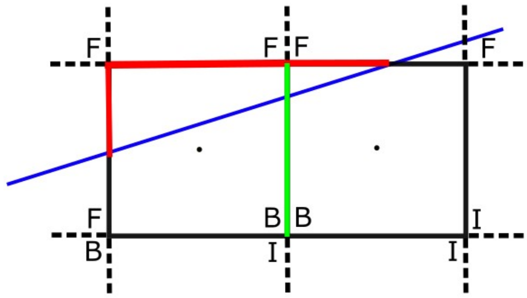
$$\mathrm{G}_i^{'} = \mathrm{G}_i \pm \frac{\Re_{b_i}}{\Sigma w_i} w_i \tag{8}$$

Here, G is the *convective* flux vector, $\hat{n}$ is the unit normal at the face, $\Re_b$ is the net residual of convective fluxes, w is a weight, $\vec{V}$ is the velocity vector, and $G^{'}$ is the corrected flux vector after the redistribution. Here, the subscript i refers to the quantity determined at a band face in the band cell. However, the process differs for the situation shown in **Fig.** 6(b) in the presence of pseudo-band faces. A point to note here is that a loop through the band cells (and not band faces) is used to correct the flux at a band face. While the flux correction at a band face is not repeated in this approach (as a band face cannot be shared by two band cells by definition), the situation with pseudo band faces is different as it is shared by adjacent band cells. As such, for cells having a pseudo band face with a non-zero wetted area, the following process is adopted. Initially, during each iteration, no flux is allocated to the pseudo band faces. Let us consider that in **Fig.** 6 (b), the band cell on the left is first visited in the loop over band cells for convective flux correction at band faces. The net residue residual in this cell is now entirely considered as a flux correction for the pseudo-band face. Subsequently, when the band cell on the right

[2]Mass and momentum flux redistribution has been considered in the present study



(a) Configuration 1



(b) Configuration 2

Figure 6: Schematic of different configurations arising in the mass source/sink redistribution.

is reached in the loop, the net residual, considering the (now non-zero) convective flux at the pseudo band face and the band face, is provided as a correction (only) to the band face.
The proposed IBM with improved conservation properties can be summarized as follows,

1. Calculate the flux vector across all faces within the fluid domain.
2. Reconstruct the primitive variables at the band faces and determine the flux vector.
3. Determining the cut-areas and volumes of cells intersected by the immersed boundary.
4. Adjust/correct flux at band faces to enforce improved conservation of mass flux.

3 Results and Discussion

The performance of the proposed methods is assessed and compared on the test cases listed in Table 1. These test cases include three inviscid and two viscous cases ranging from high subsonic to supersonic speeds. Results are compared with CFD predictions from the literature using pressure contours, Mach number contours, and surface pressure plots. In addition to these, the convergence history of two test cases - bump in a channel and double ellipse is also presented. For each of the test cases, the grid size mentioned here refers to the 'fine' grid used for the test case. In addition, the simulations were also performed on 'medium' and 'coarse' grids, obtained by successively coarsening the grids starting with the fine grid.

Table 1: Test cases

Validation cases				
Body	M_∞ **and** α	Re	**Flow**	**Grid size**
Bump in a channel	0.675, 0°	N/A	Transonic (Inviscid)	192×64, (Non-uniform)
Cylinder	3,0°	N/A	Supersonic (Inviscid)	200×800, (Uniform)
Double ellipse	8.15, 30°	N/A	Hypersonic (Inviscid)	200×200, (Uniform)
NACA0012	0.8, 0°	500	Transonic (Viscous)	648×1024, (Non-uniform)
Cylinder	2, 0°	300	Supersonic (Viscous)	632×432, (Non-uniform)

3.1 Bump in a channel

This is an inviscid flow past a bump in the channel. The flow is transonic with the inlet Mach number, pressure and temperature are equal to 0.675, 1.0e5 Pa and 300 K respectively. The channel is 3.0 m long and 1.0 m in height. The bump is located halfway along the length on the lower wall. The thickness-to-chord ratio of the bump is 10%. The grid used is of the size 192 × 64 and the spacing same as in [31]. The schematic of the boundary conditions applied and the non-uniform grid used is shown in the **Fig.** 7.

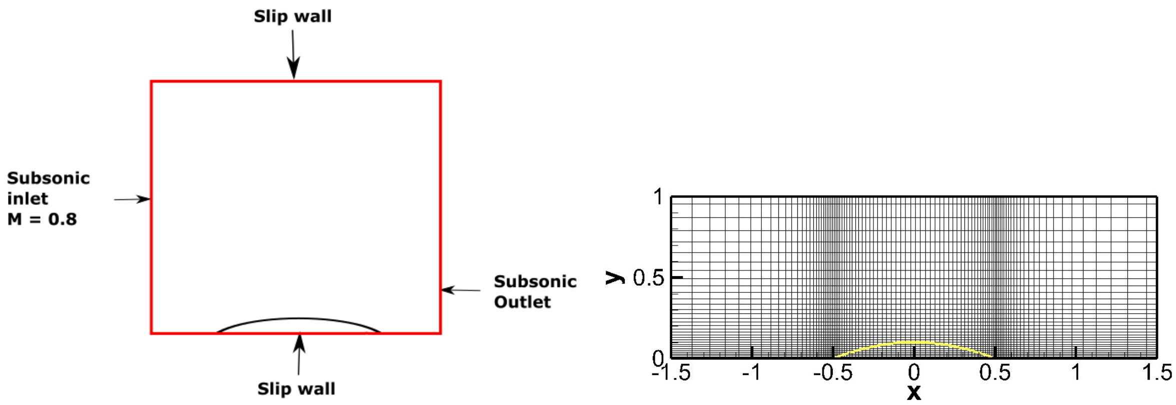
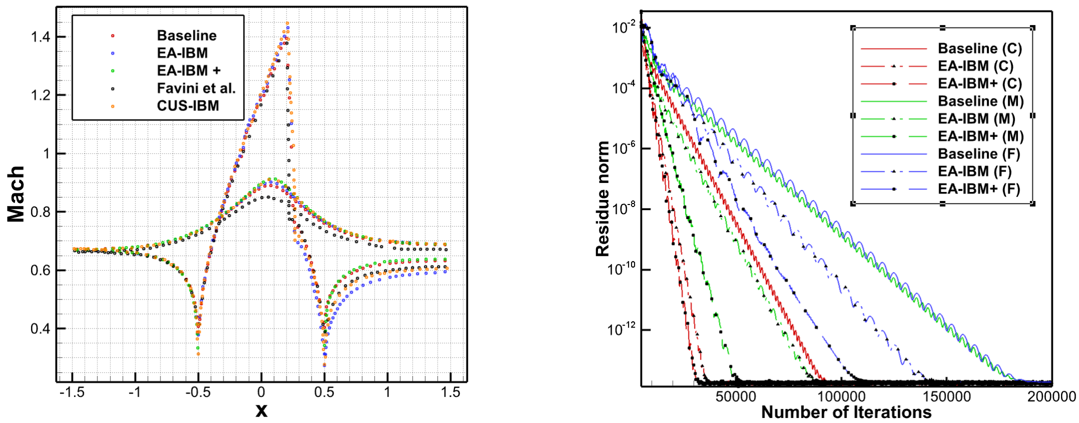


Figure 7: Schematic of domain with boundary conditions indicated (left) & Non-uniform mesh of size 192 x 64 (right) (showing alternate grid lines in both directions); IB shown by yellow line. - Transonic flow over a bump (inviscid).

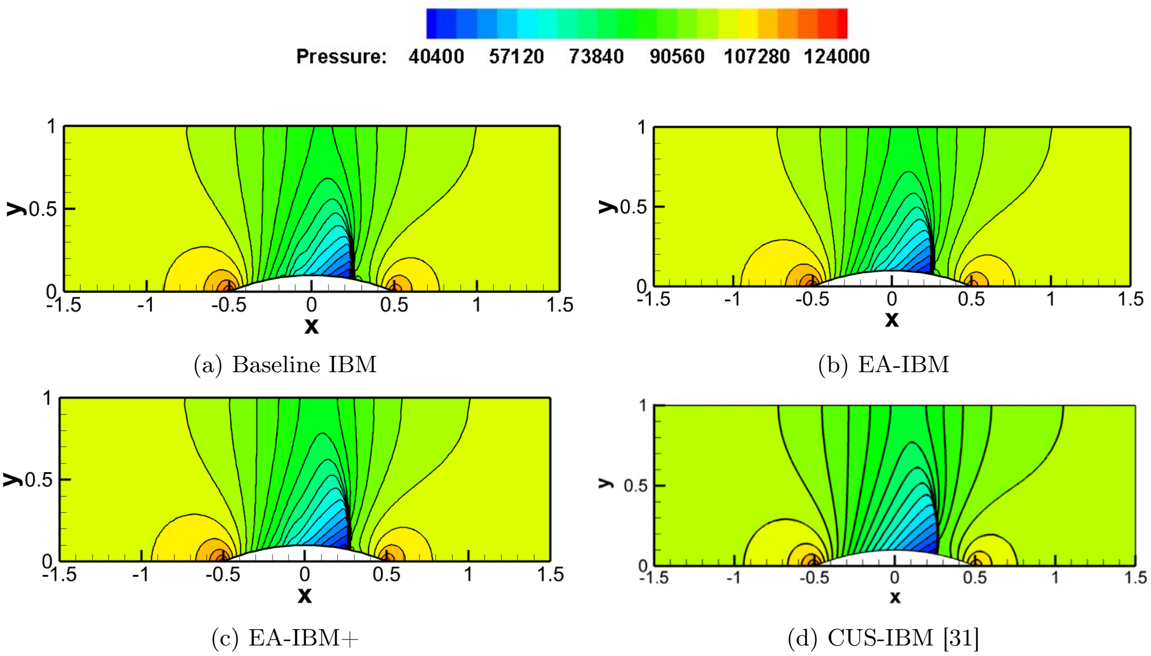
The Mach number distribution plots shown in **Fig.** 8a provide insights into the performance of the IB methods. With the use of EA-IBM and EA-IBM+, the Mach number predicted at the ends of the bump is lower compared with the body-fitted simulations. Moreover, the predictions on the upper surface of the channel compare favorably with CUS-IBM from the literature; however, the baseline method remains closer to the results obtained from body-fitted grid simulations for this test case. The residue norm comparisons shown in the **Fig.** 8 shows that EA-IBM and the EA-IBM+ improves the convergence across the grid levels.



(a) (b)

Figure 8: Mach number distribution (left). Residue norm comparison of the IB methods across grid levels - Transonic flow over a bump (inviscid).

The pressure contours are compared in **Fig.** 9. These plots demonstrate that all IB methods accurately predict the shock structure and shock location compared to the literature. No significant qualitative difference was observed across the IB methods in these contour plots.



(a) Baseline IBM (b) EA-IBM

(c) EA-IBM+ (d) CUS-IBM [31]

Figure 9: Pressure contours - Transonic flow over a bump (inviscid).

3.2 Inviscid supersonic flow past a circular cylinder

This case simulates inviscid flow past a cylinder with intel mach $M_\infty = 3.0$, $P_\infty = 103320.0Pa$ and $T = 300K$ This is a external flow computation past a circular cylinder. A supersonic flow of $M_{inlet} = 3$, P_{inlet} =103320 Pa and $T = 300$ K over a half cylinder is simulated. The computational domain is $[-1\text{m}, 0\text{m}] \times [-2\text{m}, 2\text{m}]$, and the cylinder is centered at $(0\text{m}, 0\text{m})$ with the radius of 0.5m. A uniform cartesian mesh of 200×800 cells along x and y directions is used. Boundary conditions implemented for the domain and the uniform mesh (IB rendered as a yellow curve) are presented in the **Fig.** 10.

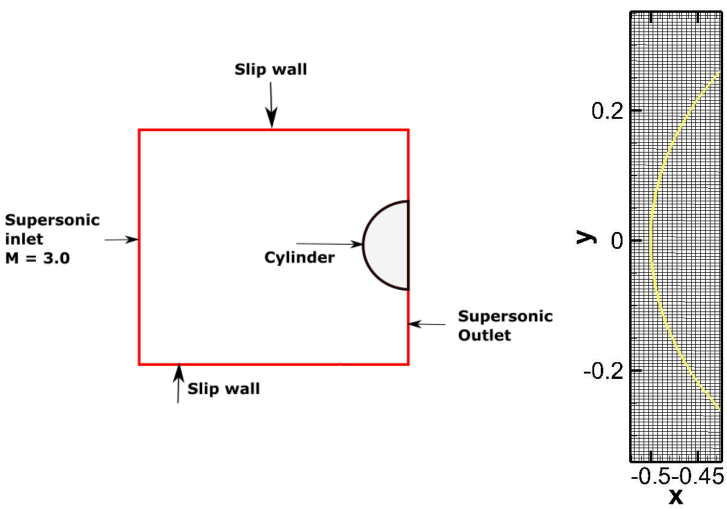


Figure 10: Schematic of domain with boundary conditions indicated (left) & Near view of the grid showing every other pair of grid lines (right);IB shown by yellow line.

The surface pressure plot shown in the **Fig.** 11 compares the predictions of the different IB methods with the literature. It can be observed that all the methods agree excellently, and there are no irregularities in the surface data using different methods. Additionally, mach number contours are shown in the **Fig.** 12 for qualitative representation of the solution. Contour plots predict the flow structures properly and compare well with the literature.

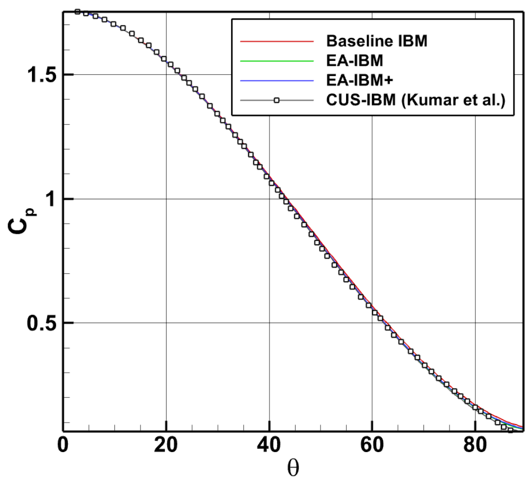
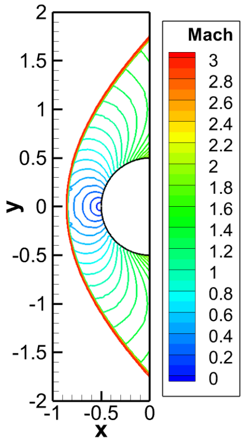
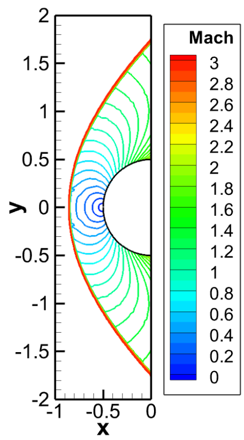


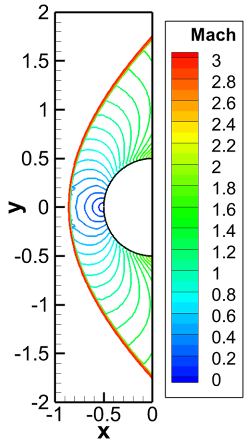
Figure 11: Comparison of surface pressure distribution - Flow past a cylinder (inviscid).



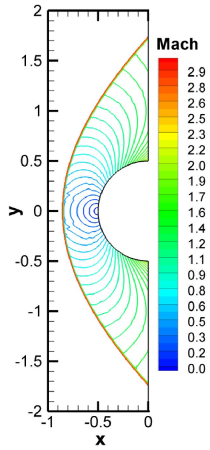
(a) Baseline IBM



(b) EA-IBM



(c) EA-IBM+

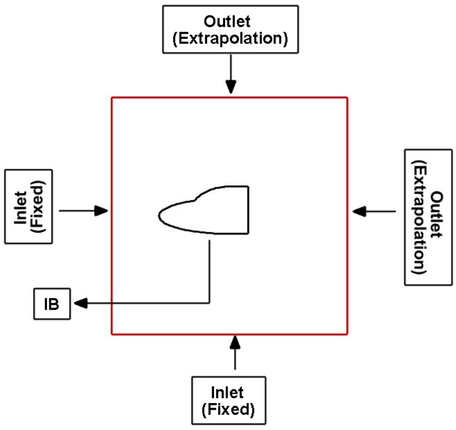


(d) CUS-IBM [31]

Figure 12: Mach number contour - Flow past a cylinder (inviscid).

3.3 Flow past a double ellipse

This test case is considered to determine the performance of the IB solver for the flow at a high angle of attack. The free stream properties are $M_\infty = 8.15$ at a 30° angle of attack with respect to horizontal, $P_\infty = 101325.0$, $T_\infty = 273.0K$. The dimensions of the double ellipse geometry are the same as in [32]. The domain considered is of the size [-0.1,0.1]x[-0.1,0.1] with the double ellipse immersed in the domain.



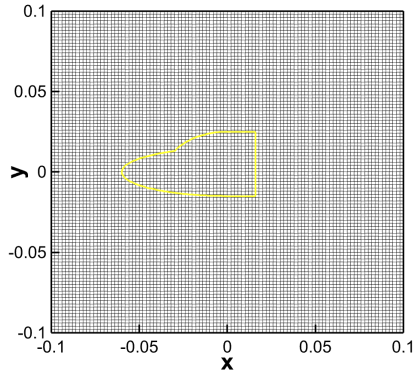
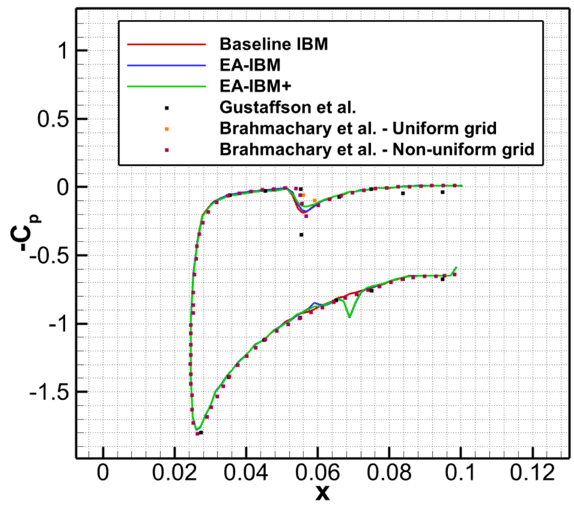
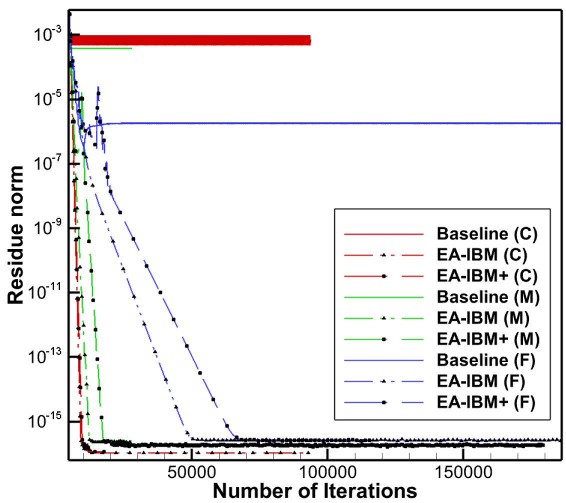


Figure 13: Schematic of domain with boundary conditions indicated (left) & Uniform mesh of size 200 x 200 (right) (showing alternate grid lines in both directions); IB shown by yellow line. - Flow past a double ellipse (inviscid).

A uniform grid with 200 $\times$ 200 cells in both x and y directions has been used in our work, and the mesh and boundary conditions used are shown in the **Fig.** 13. The results are compared with Cartesian grid IB simulations conducted by [33] using a similar number of control volumes. **Fig.**14a presents a comparison of the surface pressure coefficient, demonstrating that all three IB methods align closely with the literature and notably lean towards the results obtained from the body-fitted and non-uniform grid solution used in the comparison. However, minor deviations in the surface pressure are observed with the EA-IBM+.



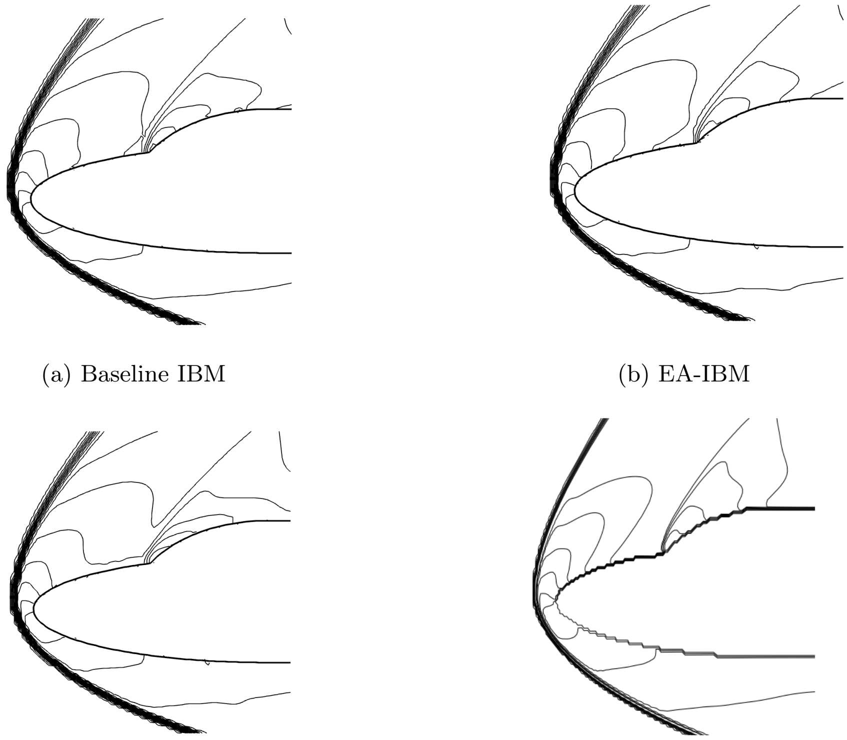
(a)



(b)

Figure 14: Comparison of pressure coefficient(left).Residual comparison of the IB methods across the grid levels(right) - Flow past a double ellipse (inviscid).

Additionally, the residual norms show improvement with both the EA-IBM and EA-IBM+ compared to the baseline method, which experiences convergence issues at coarse and medium grid resolutions. Overall, faster convergence is observed with EA-IBM and EA-IBM+. Regarding Mach number contour plots, all IB methods accurately capture the detached shock at the leading edge of the nose. While the canopy shock is well captured by all methods without grid clustering, it appears slightly smeared in the case of EA-IBM+. This observation is evident in the Mach contour plot (see **Fig.** 15), which shows satisfactory resolution of both the bow shock and canopy shock.



(a) Baseline IBM (b) EA-IBM

(c) EA-IBM+ (d) IB method [33]

Figure 15: Mach number contour - Flow past a double ellipse.

3.4 Transonic viscous flow past NACA0012 airfoil

This is a laminar flow simulation with a large separation vortex near the trailing edge of the airfoil. Free stream flow parameters are: $M_\infty = 0.8$, $P_\infty = 103320\,\mathrm{Pa}$, $T_\infty = 300\,\mathrm{K}$, $Re_\infty = 500$ and the angle of attack is $10°$. This test case is chosen as the flow is transonic and involves flow separation with a recirculation bubble forming on the suction side of the airfoil. This tests whether IBM can accurately predict flow separation, which is crucial for the reliable prediction of viscous flows. The computations for this simulation are carried out on a non-uniform grid of size 648×1024 in x and y directions, respectively. The grid spacings used are the same as in [34].

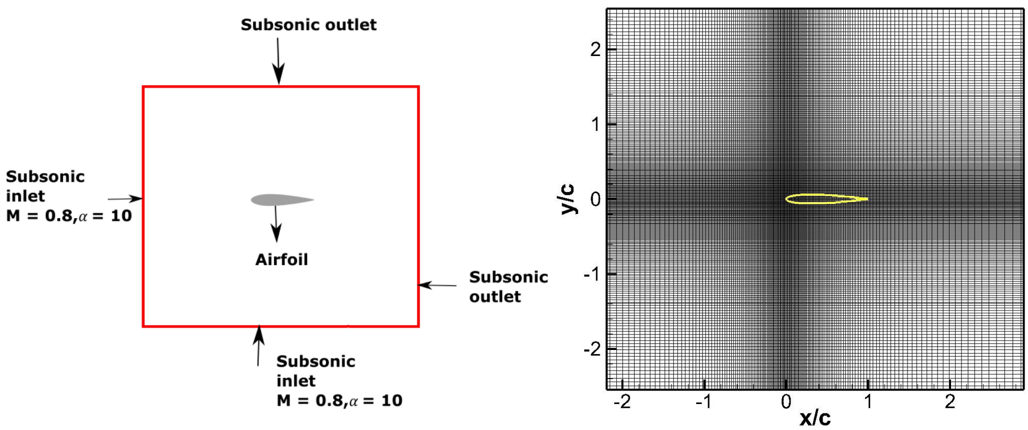


Figure 16: Schematic of domain with boundary conditions indicated: Flow past airfoil NACA0012 (left); & Closeup view of 648 x 1024 non uniform mesh (right) (showing alternate grid lines in both directions); IB shown by yellow line.

In **Fig.** 17, the surface pressure coefficient predictions from different immersed boundary (IB) methods

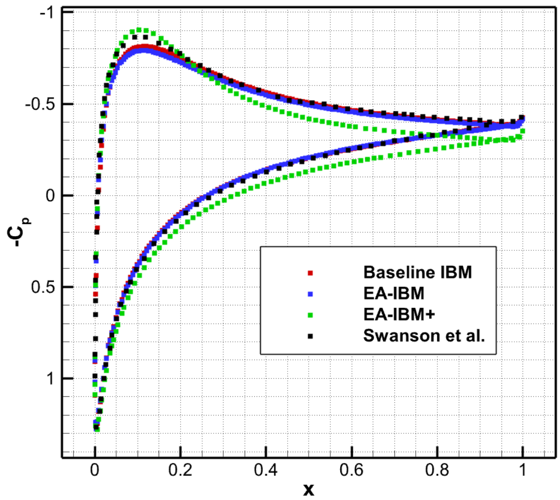
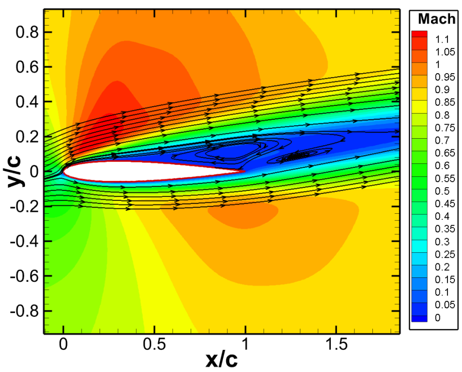
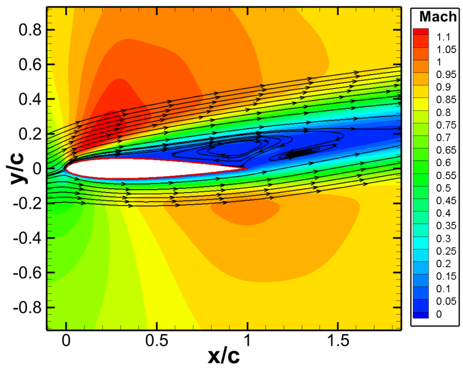


Figure 17: Comparison of surface pressure distribution - Transonic flow past NACA0012 airfoil (viscous).

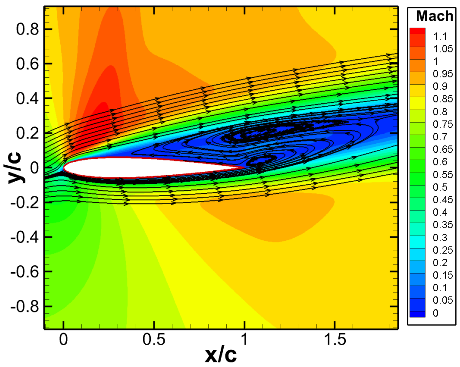
align well with the literature. The baseline IBM and EA-IBM show consistent agreement with literature across most regions of the airfoil, although minor deviations are evident at the airfoil's leading edge. EA-IBM+ follows the overall trend but exhibits deviations both at the leading edge and on the upper surface where flow separation occurs. These observations are also evident in the Mach number contour plots shown in the **Fig.** 18. Specifically, EA-IBM+ does not capture the recirculation bubble beyond the trailing edge, whereas other methods accurately reproduce the recirculation zones and bubbles, same as in the literature [34].



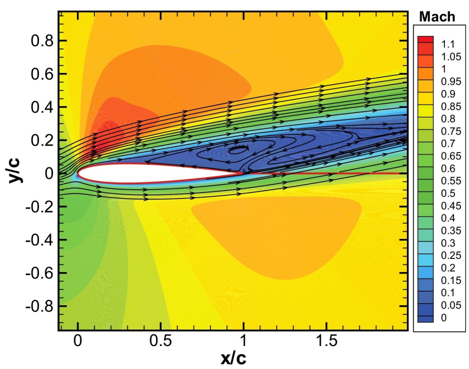
(a) Baseline IBM



(b) EA-IBM



(c) EA-IBM+



(d) [34]

Figure 18: Comparison of mach number contours - - Transonic flow past NACA0012 airfoil (viscous).

3.5 Supersonic viscous flow past circular cylinder

Flow conditions for this external flow simulation are: $M_\infty = 2.0$, $P_\infty = 103320$ Pa, $T_\infty = 300$ K, and $Re = 300$. The domain extent is 60 $D \times 40$ D (D =1m), same as in [31] [35] and the cylinder is centered at (24,20). Non-uniform mesh with the grid size 632×432 along x and y directions was used. There is a uniform mesh region around the cylinder of size 1.7 $D \times 1.7$ D with grid spacing 0.025D same as in [36]. This grid spacing of the finest grid used is considered sufficient to capture the boundary layer effects for the specific Reynolds number. The domain with the boundary conditions employed and the grid are as shown in the **Fig.** 19.

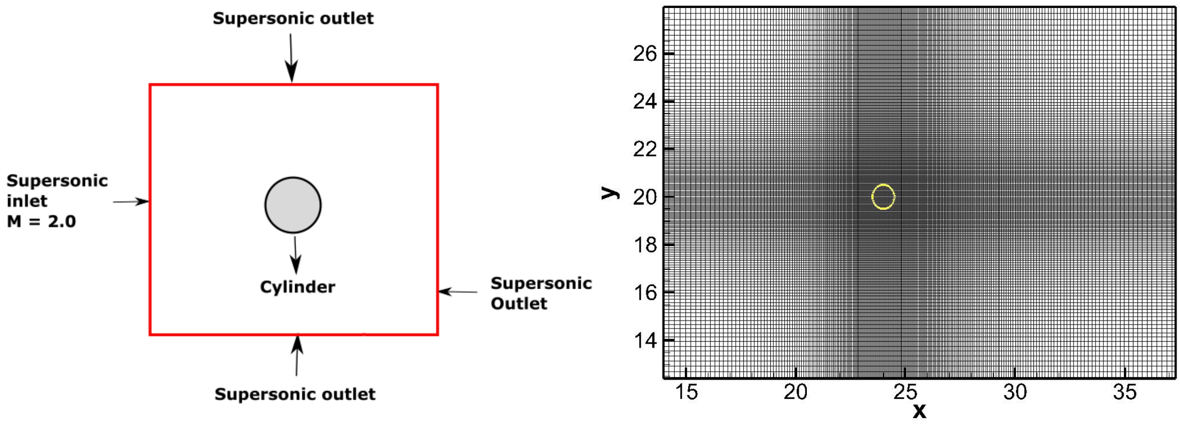


Figure 19: Schematic of domain with boundary conditions indicated (left) and close-up view of non-uniform grid (right) (showing alternate grid line in both directions);IB shown by yellow line.

The pressure distribution plot shown in the **Fig.** 20 compares the different IB methods with the literature. It can be observed that there are no irregularities in the surface pressure predictions from any of the IB methods, and they compare well with the literature. Also, no significant difference across the methods can be observed. Similar conclusion can be drawn from the density contour plots shown in the **Fig.**21. Qualitatively, all the methods predict the bow shock accurately and compare well with the literature.

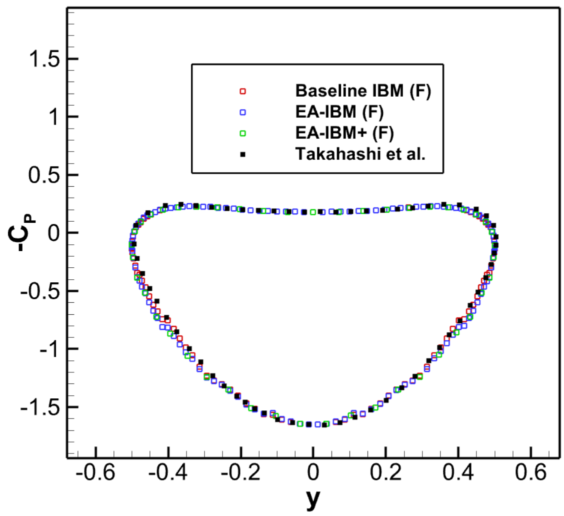
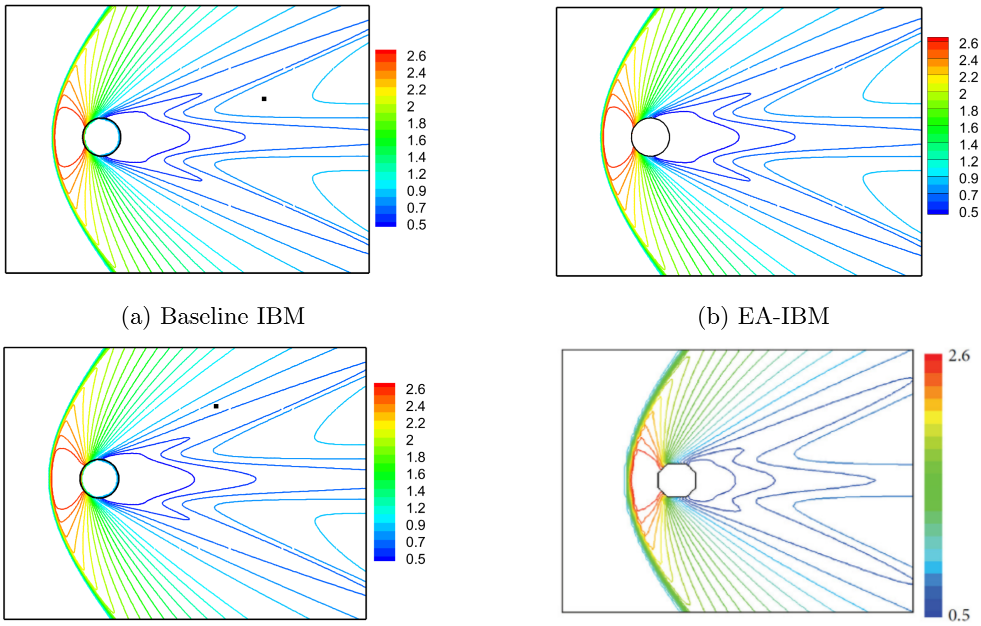


Figure 20: Comparison of surface pressure distribution - Supersonic flow over a cylinder (viscous).



(a) Baseline IBM (b) EA-IBM

(c) EA-IBM+ (d) [36]

Figure 21: Density contour plots - Supersonic flow over a cylinder (viscous).

3.6 Performance study

To quantitatively assess the effectiveness of the baseline-IBM, EA-IBM, and EA-IBM+ in terms of mass conservation, the net mass efflux is computed. This metric quantifies the mass source/sink effect introduced by the immersed boundary, calculated as the difference between net outflow and net inflow across all boundaries. The computed values using baseline IBM, EA-IBM, and EA-IBM+ are presented in the table below.

Table 2: Comparison of net mass balance in the domain obtained using baseline IBM, EA-IBM and EA-IBM+.

Test case	BSL	EA-IBM	EA-IBM+	Grid level
Bump in a channel (Inviscid)	4.31E-03	4.81E-03	1.46E-15	**Grid - C**
	3.39E-03	1.42E-04	4.18E-16	**Grid - M**
	4.61E-04	7.30E-06	6.27E-16	**Grid - F**
Cylinder (Inviscid)	6.68E-04	1.29E-03	6.66E-06	**Grid - C**
	2.58E-04	8.62E-04	7.98E-04	**Grid - M**
	1.58E-05	4.83E-04	1.52E-06	**Grid - F**
Double Ellipse (Inviscid)	6.06E-03	7.87E-03	1.65E-14	**Grid - C**
	7.11E-03	3.74E-03	2.38E-16	**Grid - M**
	1.73E-03	2.37E-03	3.65E-10	**Grid - F**
NACA 0012 (Viscous)	1.76E-05	9.40E-06	2.37E-07	**Grid - C**
	3.76E-05	4.40E-05	4.03E-05	**Grid - M**
	6.35E-04	6.48E-04	1.46E-03	**Grid - F** [3]
Cylinder (Viscous)	1.29E-04	7.75E-05	1.33E-06	**Grid - M**
	5.07E-04	5.21E-04	1.04E-06	**Grid - F**

Note: Here, Grid-F is the finest grid for the specific test case. Grid-M and Grid-C are subsequently generated from Grid-F by excluding alternate grid points—Grid-M skips every second grid point in both x and y directions, whereas Grid-C skips every fourth grid point in the same directions.

From Table 2, it can be observed that the use EA-IBM+ improves the net mass balance, showing significant improvement in inviscid cases (approaches machine epsilon). However, the effectiveness of these methods in improving the net mass balance in viscous cases appears limited. Given that viscous fluxes dominate near the IB, the implementation of effective area and subsequent flux redistribution seem to have less effects in these scenarios. For all the test cases considered, the improvement in the net mass balance from the baseline IBM to EA-IBM+ is most notable for the coarse grid simulations In summary, the EA-IBM and EA-IBM+ methods maintain or improve the net mass balance (at least by order of one) across most grid levels.

Table 3 compares flow quantities like Mach number, stagnation pressure, lift and drag coefficients predicted by the different IB methods investigated in this work. Quantitative comparison with results from literature/ theoretical values shows that the maximum errors in the present method are less than 7 %.

Table 3: Summary of the quantitative comparison of specific quantities obtained using baseline IBM, EA-IBM and EA-IBM+ with the literature.

Test case	**Quantity**	**BSL**	**EA-IBM**	**EA-IBM+**	**Reference**
Bump in a channel (Inviscid)	Exit Mach (bottom)	0.632	0.594	0.637	0.615 [37]
	Exit Mach (top)	0.686	0.689	0.689	0.67 [37]
Cylinder (Inviscid)	Stagnation pressure (× 1e5 pa)	12.464	12.467	12.446	12.462 [Analytical]
Double ellipse (Inviscid)	Stagnation pressure (× 1e6 pa)	8.669	8.652	8.638	8.712 [Analytical]
NACA00012 (Viscous)	c_l	0.4172	0.4175	0.4415	0.4363 [34]
	c_d	0.2769	0.2749	0.2615	0.2752 [34]
Cylinder (Viscous)	c_d	1.5456	1.5597	1.5551	1.5477 [36]

4 Conclusions

An extension of an existing immersed-boundary method – developed earlier by the authors –aimed at improving mass conservation is presented here. Specifically, efforts have been made to improve geometric conservation (EA-IBM) and address mass source/sink effects (EA-IBM+) in the IBM. These methods are tested against three inviscid test cases and two viscous test cases with speeds ranging from high subsonic to supersonic regime. A comparison was made using contour plots and surface data from the literature. Overall, no significant difference in the contour plots can be observed across the methods. Surface parameters closely matched literature values, with slight deviations noted in specific cases like the double ellipse (inviscid) and NACA0012 airfoil (viscous) scenarios using EA-IBM+. Furthermore, the flow metrics such as exit Mach number, stagnation pressure, lift, and drag coefficients were compared against literature values, with the errors in the present method below 7%. Also, the net mass balance in the domain is compared to gauge the improvement in the mass conservation. It has been found that there is a significant improvement in the inviscid cases, with the values approaching theoretical

[3] These results are not fully converged at the time of reporting.

zero for two test cases. However, effectiveness in viscous cases was somewhat constrained due to the predominant influence of viscous fluxes near the immersed boundary, whereas these methods primarily addressed convective fluxes. In summary, a comparison with the baseline IBM indicates that EA-IBM and EA-IBM+ maintain or improve net mass balance by at least an order of magnitude across various grid sizes. Future research will focus on flux redistribution within the energy equation (in addition to continuity and momentum) and testing the performance of these methods in problems involving moving bodies.

References

[1] Charles S. Peskin. Flow patterns around heart valves: A numerical method. *Journal of Computational Physics*, 10(2):252–271, 1972.

[2] Charles S. Peskin. Numerical analysis of blood flow in the heart. *Journal of Computational Physics*, 25(3):220–252, 1977.

[3] Charles S. Peskin and David M. McQueen. A three-dimensional computational method for blood flow in the heart I. Immersed elastic fibers in a viscous incompressible fluid. *Journal of Computational Physics*, 81(2):372–405, 1989.

[4] D. Goldstein, R. Handler, and L. Sirovich. Modeling a No-Slip Flow Boundary with an External Force Field. 105(2):354–366, 1993.

[5] Ming-Chih Lai and Charles S. Peskin. An immersed boundary method with formal second-order accuracy and reduced numerical viscosity. *Journal of Computational Physics*, 160(2):705–719, 2000.

[6] John M. Stockie and Brian R. Wetton. Analysis of stiffness in the immersed boundary method and implications for time-stepping schemes. *Journal of Computational Physics*, 154(1):41–64, 1999.

[7] E. A. Fadlun, R. Verzicco, P. Orlandi, and J. Mohd-Yusof. Combined Immersed-Boundary Finite-Difference Methods for Three-Dimensional Complex Flow Simulations. *Journal of Computational Physics*, 2000.

[8] Jungwoo Kim, Dongjoo Kim, and Haecheon Choi. An Immersed-Boundary Finite-Volume Method for Simulations of Flow in Complex Geometries. *Journal of Computational Physics*, 171(1):132–150, 2001.

[9] Yu Heng Tseng and Joel H. Ferziger. A ghost-cell immersed boundary method for flow in complex geometry. *Journal of Computational Physics*, 192(2):593–623, 2003.

[10] Rajat Mittal and Gianluca Iaccarino. Immersed Boundary Methods. *Annual Review of Fluid Mechanics*, 37(1):239–261, 2005.

[11] Santanu Ghosh, Jung Il Choi, and Jack R. Edwards. Numerical simulations of effects of micro vortex generators using immersed-boundary methods. *AIAA Journal*, 48(1):92–103, 2010.

[12] J. Mohd-Yusof. Combined immersed boundary/B-spline methods for simulation of flow in complex geometries. *Center for Turbulence Research Annual Research Briefs1*, pages 317–328, 1997.

[13] T. Ye, R. Mittal, H. S. Udaykumar, and W. Shyy. An Accurate Cartesian Grid Method for Viscous Incompressible Flows with Complex Immersed Boundaries. *Journal of Computational Physics*, 156(2):209–240, 1999.

[14] M.P. Kirkpatrick, S.W. Armfield, and J.H. Kent. A representation of curved boundaries for the solution of the navier–stokes equations on a staggered three-dimensional cartesian grid. *Journal of Computational Physics*, 184(1):1–36, 2003.

[15] Motoshi Harada, Yoshiharu Tamaki, Yuichi Takahashi, and Taro Imamura. *A Novel Simple Cut-Cell Method for Robust Flow Simulation on Cartesian Grids.*

[16] Balaji Muralidharan and Suresh Menon. A high-order adaptive Cartesian cut-cell method for simulation of compressible viscous flow over immersed bodies. *Journal of Computational Physics*, 321:342–368, 2016.

[17] Balaji Muralidharan and Suresh Menon. Simulation of moving boundaries interacting with compressible reacting flows using a second-order adaptive Cartesian cut-cell method. *Journal of Computational Physics*, 357:230–262, 2018.

[18] MLNV Kasturi Rangan and Santanu Ghosh. *2D Immersed Boundary Solver for Compressible Laminar Flows.* 2021.

[19] X.Y. Hu, B.C. Khoo, N.A. Adams, and F.L. Huang. A conservative interface method for compressible flows. *Journal of Computational Physics*, 219(2):553–578, 2006.

[20] M. Meyer, A. Devesa, S. Hickel, X. Y. Hu, and N. A. Adams. A conservative immersed interface method for Large-Eddy Simulation of incompressible flows. *Journal of Computational Physics*, 229(18):6300–6317, 2010.

[21] Jung Hee Seo and Rajat Mittal. A sharp-interface immersed boundary method with improved mass conservation and reduced spurious pressure oscillations. *Journal of Computational Physics*, 230(19):7347–7363, 2011.

[22] Jatinder Pal Singh Sandhu, Anant Girdhar, Rakesh Ramakrishnan, R. D. Teja, and Santanu Ghosh. Fest-3d: Finite-volume explicit structured 3-dimensional solver. *Journal of Open Source Software*, 5(46):1555, 2020.

[23] Amiram Harten, Peter D. Lax, and Bram van Leer. On upstream differencing and godunov-type schemes for hyperbolic conservation laws. *SIAM Review*, 25(1):35–61, 1983.

[24] Bernd Einfeldt. On godunov-type methods for gas dynamics. *SIAM Journal on Numerical Analysis*, 25(2):294–318, 1988.

[25] Charles Hirsch. *Numerical Computation of Internal and External Flows: The Fundamentals of Computational Fluid Dynamics*. Butterworth-Heinemann, 2 edition, 2007.

[26] M. L. N. V. Kasturi Rangan and Santanu Ghosh. A face-based immersed boundary method for compressible flows using a uniform interpolation stencil. *Frontiers in Mechanical Engineering*, 8, 2022.

[27] Francesco Capizzano. A compressible flow simulation system based on cartesian grids with anisotropic refinements. *Collection of Technical Papers - 45th AIAA Aerospace Sciences Meeting*, 24(January):17036–17057, 2007.

[28] Francesco Capizzano. Turbulent wall model for immersed boundary methods. *AIAA Journal*, 49(11):2367–2381, 2011.

[29] Cheng Chi, Bok Jik Lee, and Hong G. Im. An improved ghost-cell immersed boundary method for compressible flow simulations. *International Journal for Numerical Methods in Fluids*, 83(2):132–148, 2017.

[30] Eugene L. Allgower and Phillip H. Schmidt. Computing Volumes of Polyhedra. *Mathematics of Computation*, 46(173):171, 1986.

[31] Vinay Kumar, Atul Sharma, and R. K. Singh. Central upwind scheme based immersed boundary method for compressible flows around complex geometries. *Computers and Fluids*, 196:104349, 2020.

[32] Emre Kara, Ahmet İhsan Kutlar, and M. Haluk Aksel. Çift elips yapısı etrafında çözüm uyarlamalı navier-stokes çözücüsü kullanarak yüksek reynolds sayılı akış analizi. *Dicle Üniversitesi Mühendislik Fakültesi Mühendislik Dergisi*, 11(2):563 – 573, 2020.

[33] Shuvayan Brahmachary, Ganesh Natarajan, Vinayak Kulkarni, and Niranjan Sahoo. A sharp-interface immersed boundary framework for simulations of high-speed inviscid compressible flows. *International Journal for Numerical Methods in Fluids*, 86:770–791, 2018.

[34] R.C. Swanson and S. Langer. Steady-state laminar flow solutions for naca 0012 airfoil. *Computers Fluids*, 126:102–128, 2016.

[35] Y.L. Qiu, C. Shu, J. Wu, Y. Sun, L.M. Yang, and T.Q. Guo. A boundary condition-enforced immersed boundary method for compressible viscous flows. *Computers Fluids*, 136:104–113, 2016.

[36] Shun Takahashi, Taku Nonomura, and Kota Fukuda. A Numerical Scheme Based on an Immersed Boundary Method for Compressible Turbulent Flows with Shocks : Application to Two-Dimensional Flows around Cylinders. 2014, 2014.

[37] B. Favini, R. Broglia, and A. Di Mascio. Multigrid acceleration of second-order eno schemes from low subsonic to high supersonic flows. *International Journal for Numerical Methods in Fluids*, 23(6):589–606, 1996.