Towards Aerodynamic Shape Optimization Using an Immersed Boundary Overset Grid Method

Brandon M. Lowe*, Chase P. Ashby†, James R. L. Koch‡, David A. Craig Penner§, Jeffrey A. Housman¶, and Jared C. Duensing‖
NASA Ames Research Center, Moffett Field, California, 94035

Traditional Reynolds-averaged Navier-Stokes grid methods applied to aerodynamic shape optimization can struggle with the deformation of surface and volume grids at component intersections, such as at wing-fuselage junctions. To overcome this, we propose an approach which utilizes curvilinear overset grids for the discretization of the domain, with the presence of the body modeled using an immersed boundary method. This approach handles complex geometries without the need for their explicit integration into the grid. The goal of this approach is to reduce grid generation time and allow for greater geometric freedom for component-based aerodynamic shape optimization. Two different methods are presented: a source-term-based and a ghost-node-based immersed boundary method. Flow analyses and adjoint solutions obtained using the proposed methods show promising comparisons with standard body-fitted grid methods. Preliminary aerodynamic shape optimization results obtained using one of the immersed boundary methods are also presented.

I. Introduction

The curvilinear overset grid method for computational fluid dynamics (CFD) was originally introduced as an approach, in part, to simplify the process of grid generation around complex geometric configurations [1, 2]. The overset methodology involves using several overlapping grids in space, with CFD analyses performed by exchanging data between grids to construct a flow field over the entire domain. Though grid generation is simplified with the overset approach, combining multiple geometric components into a single configuration requires the addition of specialized grids, known as collar and cap grids, which faithfully represents the geometry of each component near junctions and intersections [3]. Generation of collar and cap grids is a bottleneck in the grid generation process, usually requiring significant time to construct. Additionally, grid cells present in a collar grid are often highly skewed and low quality. Deformation of these grids to conform to new geometries for shape optimization can easily lead to dubious flow field results or negative cell volumes near the junction. We hypothesize that the need for collar grids at component intersections can be superseded by an immersed boundary method. Using this method in conjunction with a curvilinear overset grid paradigm would allow for reduced grid generation time and greater geometric freedom for aerodynamic shape optimization.

Utilization of structured grids retains a low computational expense for flow analyses and shape optimization based on the steady-state Reynolds-averaged Navier-Stokes (RANS) equations [4–8]. While overset grids have been successfully used for aerodynamic shape optimization [9–13], early efforts lacked the ability to separately manipulate individual components of complex geometries. Such a challenge largely stems from the deformation of the surface grid at component intersections, for example, at the intersection of the wing and the fuselage. Collar grids are typically constructed at component intersections to provide a smoothly varying surface grid bridging both components. Deformation of only one of the intersecting components requires either the reconstruction or deformation of the collar grid to conform to the new geometry. An approach for the reconstruction of the collar grid was developed by Secco et al. [14], while an approach enabling component-based deformation of the collar grid was developed by Yildirim et al. [15]. Though both of these approaches have been successfully applied to the optimization of basic aircraft [6, 8], they still require the use of collar grids between components. Additionally, both approaches require well-defined feature curves on the geometry to create viable CFD grids with acceptable quality.

*Science & Technology Corporation, MS N258-2, AIAA Member, brandon.lowe@nasa.gov
†Computational Aerosciences Branch, NAS Division, MS N258-2, chase.p.ashby@nasa.gov
‡Computational Aerosciences Branch, NAS Division, MS N258-2, AIAA Member, james.r.koch@nasa.gov
§Science & Technology Corporation, MS N258-2, AIAA Member, david.a.craigpenner@nasa.gov
¶Computational Aerosciences Branch, NAS Division, MS N258-2, AIAA Senior Member, jeffrey.a.housman@nasa.gov
‖Computational Aerosciences Branch, NAS Division, MS N258-2, AIAA Member, jared.c.duensing@nasa.gov

An alternative strategy to the overset grid method, which similarly arose in part to address the challenges of CFD analysis of complex geometries, is the immersed boundary method [16–21]. The term immersed boundary generally encompasses a wide variety of numerical techniques intended to solve systems of equations on grids that do not conform with physical boundaries. One of the challenges with using the immersed boundary method in an aerodynamic design context is the potential numerical noise it introduces into the aerodynamic functionals when changes in the geometry occur. This noise inherently results in non-smooth gradients of the functionals, which can lead to difficulties during optimization. Despite this drawback, immersed boundary techniques have been successfully applied to aerodynamic shape optimization problems [22, 23]. However, these applications have relied on the use of Cartesian grids. Though the Cartesian grid approach has an inherent advantage with respect to grid generation, the curvilinear grid paradigm remains advantageous for applications of the RANS equations, the governing flow equations used for viscous aerodynamic shape optimization. This advantage stems from the curvilinear grid's inherent alignment with the near-body flow field, its cells' varied aspect ratios, and its anisotropic grid spacings. Altogether, these features enable effective capturing of flow phenomena with high gradients, such as boundary layers, shocks, and wakes. While immersed boundary methods have also been applied on curvilinear grids [24–27], their applications have been limited to modeling small complex aspects of geometries while the bulk of the configuration remains body-fitted. To the authors' knowledge, the immersed boundary method on curvilinear grids has not been applied to aerodynamic shape optimization.

Towards the goal of eliminating the need for collar grids with the application of an immersed boundary method, we present an approach which utilizes curvilinear overset grids with the geometry modeled as an immersed boundary. The generated curvilinear grids are thus not required to be strictly body conforming. Instead, they are constructed to encompass the domain both inside and outside of individually closed geometry components. Using such an approach, component intersections need not be explicitly defined within the overlapping grid system, eliminating the need for the construction and deformation of collar grids or complex multi-block interfaces. The overset curvilinear grids can still be constructed in such a way as to maintain their inherent advantages such as their alignment with the near-body flow field. In regards to aerodynamic shape optimization, application of an immersed boundary method allows for shape changes which need not be transmitted to the entire volume grid. Instead, deformations can be restricted to the near-body grid of the component being deformed, preserving the quality of the grid. In this paper, we present two different immersed boundary methods on curvilinear overset grids for applications towards aerodynamic shape optimization. The first approach, termed the source-term immersed boundary method, uses source terms at nodes that lie within the immersed domain to emulate the presence of the body in the flow. The second approach, termed the ghost-node immersed boundary method, marks certain nodes within the immersed domain as ghosts which extrapolate pressure and temperature from the flow field, and on which the velocities are specified, also to emulate the presence of the body.

The remainder of this paper is divided into the following sections. In Section II, we present the flow solver and the source-term and ghost-node immersed boundary methods applied in this work. In Section III, we discuss the curvilinear grid requirements and advantages for the immersed boundary method. Section IV describes our approach to aerodynamic shape optimization, including gradient calculations, geometry parameterization, and grid deformation. Section V compares flow analysis results obtained by the body-fitted and immersed boundary methods, while Section VI compares discrete adjoint results. Finally, Section VII presents aerodynamic shape optimization results obtained with the immersed boundary and body-fitted approaches.

II. Computational Methodology

CFD flow solutions, including aerodynamic functionals and their gradients, are computed using the Launch, Ascent, and Vehicle Aerodynamics (LAVA) framework [28]. This framework is grid agnostic in that it allows for flow solutions on unstructured arbitrary polyhedral, structured Cartesian, and structured curvilinear overset grids. This work is focused on the curvilinear overset grid paradigm in LAVA.

A. Flow Solver

The LAVA curvilinear solver utilizes a finite-difference discretization approach for solutions to the RANS equations on structured curvilinear overset grids. Convective fluxes of the flow equations are discretized using a second-order accurate modified Roe scheme with third-order left/right state reconstructions that are slope limited using the Koren limiter [29, 30]. The negative variant Spalart-Allmaras (SA) turbulence model is applied with the convective terms discretized using a first-order upwind scheme. Viscous fluxes in the momentum, total energy, and turbulence model equations are discretized with a second-order mid-point and node scheme. An approximate Newton-Krylov method is used to solve for the steady-state solution. At each nonlinear iteration, the linear system for the mean flow and turbulence equations are decoupled and solved independently using the preconditioned generalized minimal residual (GMRES) algorithm [31]. Incomplete lower-upper factorization of a first-order approximation of the residual Jacobian is used to form the preconditioner for GMRES.

B. Immersed Boundary Methods

Two implementations of the immersed boundary method are presented. The first implementation relies on the addition of source terms to the discrete governing flow equations to emulate the presence of the body in the domain; in this paper, this method is referred to as the source-term immersed boundary approach. The second implementation directly extrapolates flow field quantities to grid nodes lying within the immersed body, while modifying the velocity at said nodes in order to emulate the body; we term this latter method the ghost-node immersed boundary approach.

1. Source-Term Immersed Boundary Method

The source-term immersed boundary method adds a penalty source term to the discretized flow equations to introduce a body force, which emulates the presence of the body in the domain [17, 24, 27, 32, 33]. In the case of the flow equations, the penalty source term is introduced into the system as

$$\boldsymbol{R}_{\text{RANS}}(\boldsymbol{Q}) + \boldsymbol{F}(\boldsymbol{Q}) = 0, \tag{1}$$

where $\boldsymbol{F}$ is the applied source term, $\boldsymbol{R}_{\text{RANS}}$ is the residual of the RANS equations after discretization in space, and $\boldsymbol{Q}$ is the vector of flow states. In this work, the source term $\boldsymbol{F}$ is formulated as a Brinkman penalization [32]. For example, in the case of the x-momentum equation, we may write the forcing term at node i as

$$\boldsymbol{F}_i^{x\text{-mom}} = \alpha_i \left(\frac{\rho_i (u_i - u_i^*)}{\Delta\tau} J^{-1} \right) \tag{2}$$

where ρ is the density, u is the flow velocity in the x-direction, $\Delta\tau$ is a user-specified penalty time scale, and J is the determinant of the Jacobian of the transformation to curvilinear coordinates. The term u_i^* is the target velocity for the flow at node i inside the immersed boundary. In this work, because the geometric bodies remain stationary at each design iteration, u_i^* is zero. The parameter α_i controls the nodes on which the force is applied. This parameter is given by

$$\alpha_i = \begin{cases} 1 & \text{node } i \text{ is in an immersed domain,} \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

Similar equations to (2) exist for the energy and turbulence model equations used for RANS. An example two-dimensional curvilinear grid with an immersed domain is shown in Fig. 1, where nodes on which the source terms are applied are marked as (◘).

The advantage of this penalty approach comes from its simplicity of implementation. The method requires no user-specified parameters other then the penalty time scale, and does not influence the stability of the time integration scheme [33]. Indeed, the source-term immersed boundary method has been successfully applied with large-eddy simulations [24–27]. However, directly setting the target velocity $u_i^* = 0$ in the penalty term (2) may lead to an incorrect representation of the body's influence on the flow unless the volume grid conforms to the surface of the body and the grid is sufficiently resolved. This can lead to effects such as penetration of flow into the body. For example, for a no-slip boundary in a node-centered scheme, if the immersed boundary lies between two nodes and the velocity is only forced to be zero at the immersed node, then the resulting velocity at the actual immersed boundary will not be zero. Previous work has attempted to reconcile this disparity in geometric fidelity by using extrapolation to determine the value of the target velocity u_i^* such that the velocity at the immersed boundary corresponds to the desired boundary condition [33]. However, in the current work, we assume the grid is *mostly* body conforming, allowing us to directly set $u_i^* = 0$ without significant loss in the geometric accuracy of the body. A significant drawback of the source-term method is that the entirety of the immersed domain must be included in the volume grid. This requirement may lead to difficulty in generating the grid, notably if the goal is to have the grid mostly conform with the immersed surface. Additionally, the governing flow equations are solved at all nodes within the immersed domain, potentially leading to higher computational costs compared to solving the equations on a traditional body-fitted grid. Moreover, the degree to which the zero velocity condition within the body is enforced is dependent on both the threshold to which (1) is satisfied and the user-specified penalty time scale $\Delta\tau$.

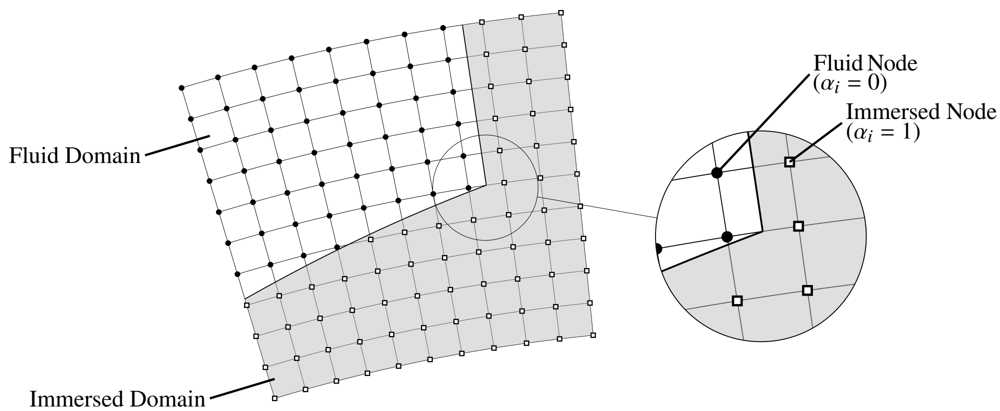


Fig. 1 Example of the application of the source-term immersed boundary method on a two-dimensional curvilinear grid. Fluid nodes are marked as (●), immersed nodes on which the source terms are applied are marked as (□).

2. Ghost-Node Immersed Boundary Method

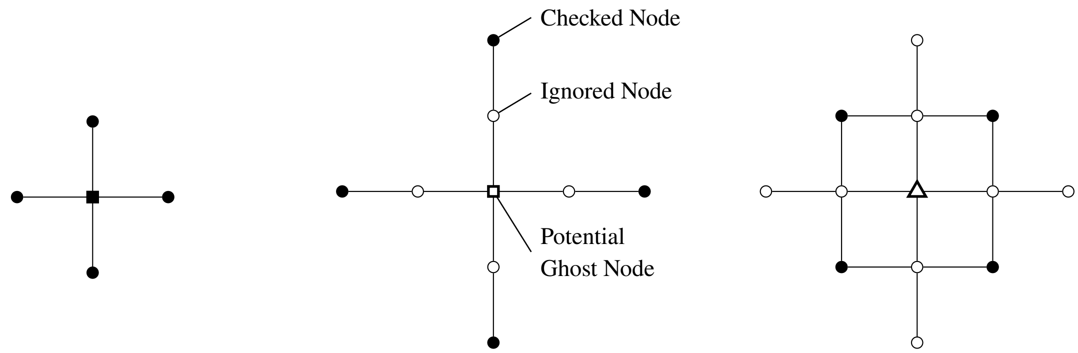
The ghost-node immersed boundary method used in this work is an extension of a simple ghost-cell method traditionally used for boundary conditions on body-fitted grids [34]. In essence, we assume that the immersed boundary interface is mostly aligned with the underlying curvilinear grid. Following this assumption, flow quantities, such as pressure and temperature, are extracted to a given ghost node from the neighboring node in the fluid most aligned with the wall normal direction. Flow velocities at the ghost nodes are specified according to a no-slip boundary condition. This approach is similar in spirit to the well-known ghost-cell approach for immersed boundary methods which uses flow quantities interpolated to an image point in the fluid domain to set the quantities at the ghost cells [20, 21, 35].

Let us introduce the three types of grid nodes present in the discretized domain. We denote nodes which exist inside the fluid domain simply as "fluid nodes". Following the ghost-cell approach for immersed boundary methods, nodes within the immersed domain are categorized into two groups: "ghost nodes" and "solid nodes" [21]. The categorization of a node within the immersed domain is determined based on its proximity to one or more fluid nodes in computational space: if a node in the immersed domain belongs to the computational stencil of any fluid node, then it is deemed to be a ghost node. Conversely, if a node present in the immersed domain does not belong to any fluid node stencil, it is marked as a solid node, and is omitted from the solution of the governing flow equations.

Once a node inside of the immersed domain is marked as a ghost, we must next determine its "donor nodes", i.e., the fluid nodes from which to extract flow quantities. In order to determine the best donor node for each ghost node, a three level hierarchy of searches is performed. The first level searches the nearest neighbors within the stencil along the computational directions, the second level searches the second-nearest neighbors along the directions, and the third level searches the neighboring nodes diagonal to the computational directions. These searches are exemplified in Fig. 2 for a two-dimensional domain, where (●) indicates the nodes in the stencil which are checked to determine if they are a fluid node.

If one or more fluid nodes are found at any of these levels, then the search is halted before moving onto the next level in the hierarchy, and the ghost node is labeled with the corresponding level. For example, a ghost node with one or more fluid nodes found during the level 1 search is labeled as a level 1 ghost. In our current implementation, level 1 and level 2 ghost nodes are restricted to a single donor node, selected as the fluid node most aligned with the wall normal direction. A level 3 ghost takes the average of all donor nodes found in its stencil. A two-dimensional example of a curvilinear grid which utilizes the ghost-node immersed boundary approach is shown in Fig. 3. In this figure, we illustrate that level 1 (■) and level 2 ghost nodes (□) extrapolate their flow quantities from the fluid node most aligned with the wall normal.

One advantage that the ghost-node immersed boundary method holds over its source-term-based counterpart is that the governing flow equations are not solved on the solid nodes in the immersed domain. This provides some computational savings for the approach. Furthermore, there are no additional source terms for the ghost-node method, and thus no extra stiffness added to the residual Jacobian. The most significant drawback is the extra complexity in the implementation of the method, as immersed nodes must be sorted into appropriate categories, and appropriate fluid nodes must be selected as donors. Implementation is somewhat eased by the simplification introduced above, i.e., that fluid quantities can be extracted directly from the fluid node most aligned with the wall normal.



(a) Level 1 stencil **(b) Level 2 stencil** **(c) Level 3 stencil**

Fig. 2 Two-dimensional stencils representing the three levels of search performed to check for the presence of a fluid node around an immersed boundary node. Potential level 1, 2, and 3 ghost nodes are marked as (■), (□), and (△), respectively. Nodes checked if they are fluid nodes marked as (●), ignored nodes marked as (○).

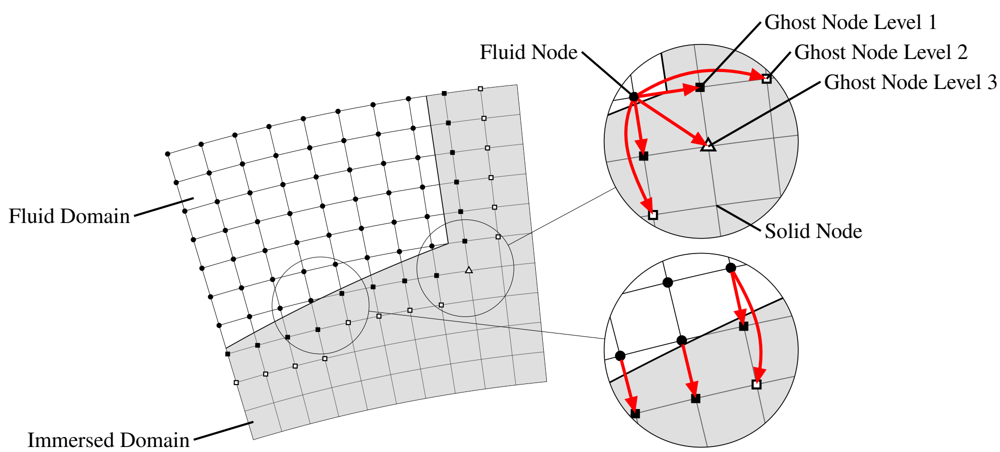


Fig. 3 Example of the application of the ghost-node immersed boundary method on a two-dimensional curvilinear grid. Fluid nodes are marked as (●), level 1 ghost nodes as (■), level 2 ghost nodes as (□) and level 3 ghost nodes as (△). Arrows indicate the fluid nodes from which the ghost nodes extract their flow field values.

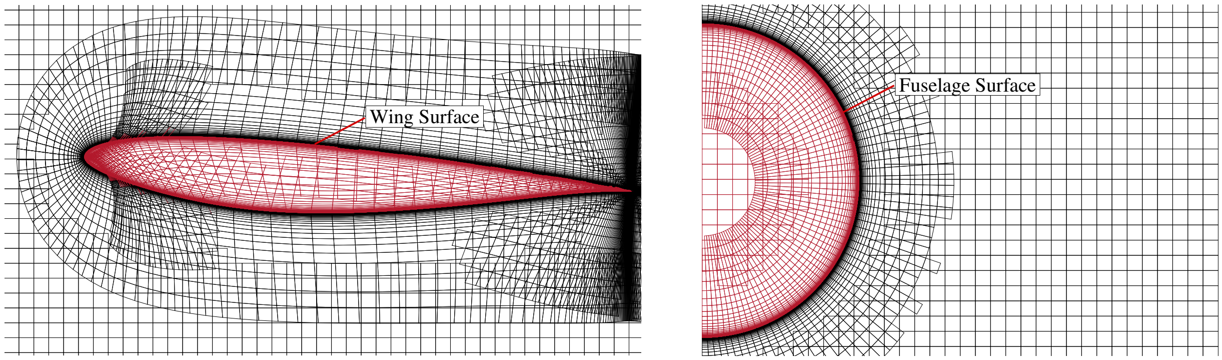
C. Computation of Aerodynamic Functionals

For the immersed boundary method, the computation of aerodynamic functionals such as lift and drag is complicated by the fact that no clear boundary exists in the grid system on which integration can be performed. Additionally, overlapping regions of the overset grid system may be present within the immersed domain, complicating the application of control-volume-based functional calculations. Thus, for the present work, aerodynamic functionals are computed by interpolating flow field quantities onto triangulated representations of the geometry and integrating over this triangulation. Specifically, trilinear interpolation is used to interpolate flow quantities from the cells in the grid to the centroids of the surface triangles. Thus, the flow quantities at a given triangle centroid $\boldsymbol{Q}_{\text{surf}}$ are computed as

$$\boldsymbol{Q}_{\text{surf}} = \sum_{i=1}^{8} w_i \boldsymbol{Q}_{i,j} \tag{4}$$

where $\boldsymbol{Q}_{i,j}$ are the flow quantities for node $i \in [1, 8]$ from cell j in the aerodynamic grid. The interpolation weights w_i are obtained based on the interpolation condition for the Cartesian coordinate of the triangle centroid $\boldsymbol{x}_{\text{surf}}$,

$$\boldsymbol{x}_{\text{surf}} - \sum_{i=1}^{8} w_i \boldsymbol{x}_{j,i} = 0 \tag{5}$$



(a) Wing grid **(b) Fuselage grid**

Fig. 4 Example grid used for the source-term immersed boundary method for a wing geometry (a) and a fuselage geometry (b). Red cells represent those that lie within the immersed domain.

where $\boldsymbol{x}_{j,i}$ is the Cartesian coordinate for node i in cell j of the aerodynamic grid.

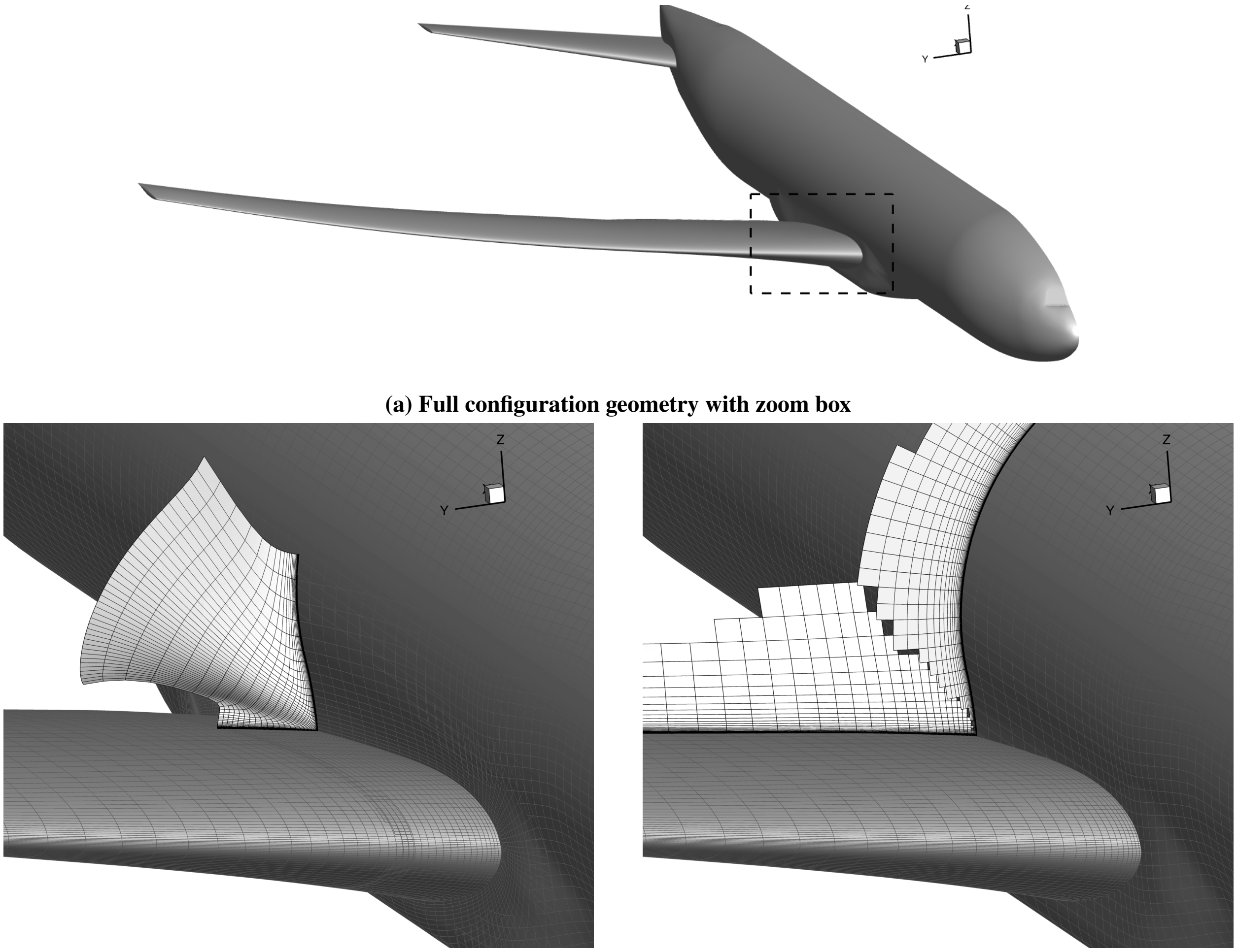
For applications of the ghost-node immersed boundary method, a given triangle centroid may not be located within a valid grid cell from which to interpolate flow quantities. This occurs due to the fact that no flow quantities exist at the solid nodes within the immersed domain. This is particularly problematic for regions where the surface triangulation is not well aligned with the volume grid, notably around regions of high curvature. To alleviate this problem, surface pressures and temperatures are interpolated at points slightly above the surface within the fluid domain. These points are offset from the triangulated surface following the outward pointing normal of the surface at each triangle centroid. This procedure follows from the assumption that the pressure and temperature gradients along the normals of the surface are near zero close to the wall. Thus, interpolating these quantities some small distance away from the wall should lead to relatively small error.

Due to the fact that the discrete representation of the immersed body need not be aligned with the volume grid, there may not exist a layer of nodes immediately off of the surface that can be used to compute the shear stress at the wall. To allow for the computation of shear stress, we rely on the application of an algebraic wall model. The wall model applied in the current work uses the SA wall function of Berger and Aftosmis [36]. For application of the model, flow velocities are interpolated from the volume grid to a "matching" point a small distance away from the surface triangulation. The wall function is fit to the flow velocity at this matching point by solving a small nonlinear equation. The shear stress at the wall is then obtained from the wall function once the fitting procedure has converged. In this work, the matching point coincides with the interpolation point used for the pressure and temperature interpolations. This point is chosen such that an average $y+$ distance over the surface between 30 and 100 is obtained in order for the matching point to reside in the log-layer region of the local boundary layer.

III. Grid Generation

The meshing requirements for the two immersed boundary methods described above are slightly different. The ghost-node immersed boundary approach only necessitates two layers of grid within the immersed domain. Conversely, for the source-term immersed boundary approach, the entirety of the immersed domain must be included in the grid. Figure 4 shows slices of example wing and fuselage component grids for the source-term approach. From Fig. 4, we observe that each grid follows the curvature of the geometry with small off-wall spacing near the surface. In this way, we say that these grids are nearly aligned with their respective geometries.

At component intersections, the grids used for the source-term and ghost-node immersed boundary methods look essentially identical in the fluid domain. As shown in Fig. 5, the immersed boundary methods do not require a collar grid. In fact, for the immersed boundary approach, an overset region is present in lieu of a collar grid. Thanks to our application of implicit hole cutting [13, 37]—used to determine node blanking and the interpolations required between the various grids in the overset system—each component in the immersed boundary approach retains a high quality grid in the boundary layer. This is because grid nodes associated with smaller cell volumes are prioritized for retention. In Fig. 5c, we are only showing the nodes on which the governing flow equations are solved. Thus, we see that grid nodes near the surface remain in the discrete system.



(a) Full configuration geometry with zoom box

(b) Body-fitted grid

(c) Immersed boundary grid

Fig. 5 Example of the grid at a wing-fuselage junction for an overset body-fitted approach (b) and an overset immersed boundary approach (c). Also shown is a zoom box on the full configuration (a). Only a subset of nodes on which the governing flow equations are solved are shown in (c).

IV. Aerodynamic Shape Optimization

Consider an optimization problem where we aim to minimize the value of an aerodynamic functional $\mathcal{J}$ with respect to geometric shape and aerodynamic design variables $\boldsymbol{V}$. This can be formulated as follows,

$$\min_{\boldsymbol{V}} \mathcal{J}(\boldsymbol{V}, \boldsymbol{Q}, \boldsymbol{W}). \tag{6}$$

where $\boldsymbol{Q}$ is the vector of flow variables and $\boldsymbol{W}$ is the vector of interpolation weights w_i from (5). In this work, the flow variables $\boldsymbol{Q}$ are obtained by solving the discrete aerodynamic equations,

$$\boldsymbol{R}_{\text{aero}}(\boldsymbol{V}, \boldsymbol{Q}) = 0, \tag{7}$$

where $\boldsymbol{R}_{\text{aero}}$ is the residual of the steady-state RANS and SA turbulence model equations. For application of the source-term immersed boundary method, $\boldsymbol{R}_{\text{aero}}$ includes the source terms as in (1). To obtain the interpolation weight vector $\boldsymbol{W}$, the interpolation condition of (5) is solved. Here, we rewrite it in residual form for all triangles on the surface,

$$\boldsymbol{R}_{\text{interp}}(\boldsymbol{V}, \boldsymbol{W}) = 0. \tag{8}$$

A. Optimization and Gradient Calculations

Due to the high costs of computing aerodynamic functionals, gradient-based optimization is deemed the best approach to minimize the number of functional evaluations. In this work, we use the Sparse Nonlinear Optimizer (SNOPT) [38], a sparse sequential quadratic programming algorithm. The interface to SNOPT is provided by the open source package pyOptSparse [39]. The discrete adjoint method is used to compute the necessary functional gradients. This can be written as follows,

$$\frac{d\mathcal{J}}{d\boldsymbol{V}} = \frac{\partial \mathcal{J}}{\partial \boldsymbol{V}} - \boldsymbol{\Lambda}^T \frac{\partial \boldsymbol{R}_{\text{interp}}}{\partial \boldsymbol{V}} - \boldsymbol{\Psi}^T \frac{\partial \boldsymbol{R}_{\text{aero}}}{\partial \boldsymbol{V}}, \tag{9}$$

where $\boldsymbol{\Psi}$ is the discrete flow adjoint vector obtained by solving the adjoint equation

$$\frac{\partial \boldsymbol{R}_{\text{aero}}}{\partial \boldsymbol{Q}}^T \boldsymbol{\Psi} = \frac{\partial \mathcal{J}}{\partial \boldsymbol{Q}}^T. \tag{10}$$

Similarly, $\boldsymbol{\Lambda}$ is the discrete interpolation adjoint vector obtained by solving the adjoint equation

$$\frac{\partial \boldsymbol{R}_{\text{interp}}}{\partial \boldsymbol{W}}^T \boldsymbol{\Lambda} = \frac{\partial \mathcal{J}}{\partial \boldsymbol{W}}^T. \tag{11}$$

Equations (9) to (11) require the computation of the partial derivatives of the flow functional $\mathcal{J}$, aerodynamic residual $\boldsymbol{R}_{\text{aero}}$, and interpolation residual $\boldsymbol{R}_{\text{interp}}$. These derivatives have been implemented using a combination of automatic differentiation and hand differentiation. The Tapenade automatic differentiation tool [40] is used to differentiate basic single purpose subroutines in the LAVA framework such as flux calculations and surface integrations. Hand differentiation is used in large part to implement the function call sequences and parallelization of the differentiated code. Reverse-mode differentiation is used to construct subroutines which provide the action of the transposed residual Jacobians $(\partial \boldsymbol{R}/\partial \boldsymbol{Q})^T$, $(\partial \boldsymbol{R}/\partial \boldsymbol{W})^T$ and $(\partial \boldsymbol{R}/\partial \boldsymbol{V})^T$ on a vector. Equation (10) is solved using the preconditioned GMRES algorithm with an incomplete lower-upper factorization of a first-order approximation of the residual Jacobian as preconditioner. Equation (11) represents a set of small independent linear equations for each surface triangle, each of which is solved directly.

B. Grid Deformation and Geometric Parameterization

The optimization algorithm controls a series of user-specified design variables which, in turn, drive the deformation of the geometry. In the current work, the geometry is parameterized using the free-form deformation (FFD) volume approach [41] applied through the open source package pyGeo [42]. The design variables controlled by the optimizer are user-specified functions which control the movement of the control points of the FFD volumes. Deformation of the geometry must be propagated to the volume grid to perform aerodynamic analyses on each new design iteration. In this work, grid deformation is accomplished using an in-house implementation of the approximate inverse distance weighting approach described by Secco et al. [43]. This approach is applicable to the overset immersed boundary methods presented herein as no connectivity information is required between the surface of the geometry and the volume grid. The volume grid is essentially treated as a point cloud, with each node's deformation computed by approximating the influence of each surface point based on the inverse distance.

V. Flow Analysis Results

In this section, we present flow analysis results obtained using the traditional body-fitted approach and the source-term and ghost-node immersed boundary methods previously discussed. Results are presented for two geometries: a wing-only geometry, and a wing-body-tail configuration. Throughout all figures shown in this section, the results obtained using the source-term immersed boundary method are labeled as "IB Source", and results obtained using the ghost-node method are labeled "IB Ghost".

A. Wing-Only Flow Analysis

Results presented in this section focus on the application and comparison of the body-fitted and immersed boundary methods for the flow analysis of a wing. The geometry selected is the NASA Common Research Model (CRM) wing used in the 4$^{\text{th}}$ case of the Aerodynamic Design Optimization Discussion Group (ADODG) suite [44]. All results are obtained for a freestream Mach number of 0.85 and a Reynolds number of 5 million. Three families of computational

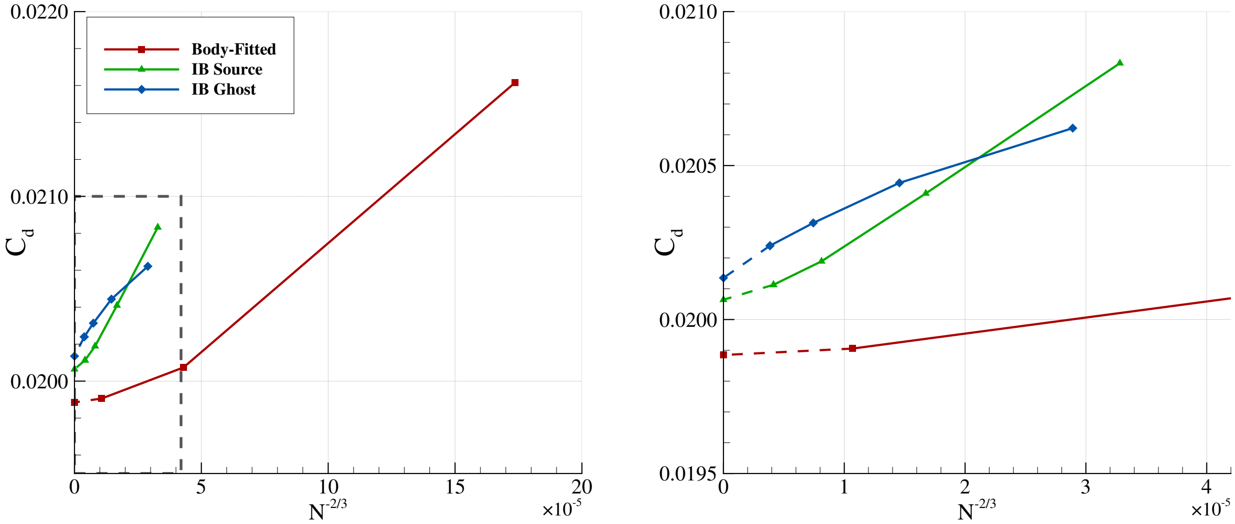


Fig. 6 Grid convergence study of the various approaches for a wing-only geometry. The left plot shows all results with a zoom box for the right plot, which allows for clearer visualization of the immersed boundary results. *N* is the number of nodes on which the governing flow equations are solved.

grids were used for this test case. A standard curvilinear body-fitted multi-block 3-member grid family originally used by Lyu *et al*. [45] with 0.5, 4, and 28 million nodes is used as a baseline for comparisons. Results for the source-term immersed boundary method are obtained using a 4-member grid family which includes the immersed domain on the interior of the wing; these grids have 6, 16, 46, and 126 million nodes. Results for the ghost-node immersed boundary method are obtained using a 4-member grid family which only includes a small potion of the immersed domain near the interface; these grids have 10, 29, 79, and 213 million nodes.

We first present the grid convergence results in Fig. 6 for the drag coefficient C_D obtained for a fixed lift coefficient $C_L = 0.5$. We observe that the body-fitted C_D has converged to a value of 198.9 drag counts, a result consistent with Lyu et al. [45] and Lee et al. [46], who obtained 199.0 and 199.1 drag counts, respectively. Taking the body-fitted results as our truth, we observe both the immersed boundary methods to be converging to higher drag values. The source term immersed boundary converges to 200.6 drag counts, while the ghost-node method converges to 201.4. Notably, we observe large errors at coarser grid levels for the immersed boundary methods. Such discrepancies at these coarse levels may originate from the imperfect representation of the body in the flow field. Recall that for both immersed boundary methods, the velocity at the nodes near the interface—or within the entire immersed domain in the case of the source-term approach—are set to zero. This simplification may lead to non-zero velocity at the actual surface of the body unless the volume grid exactly conforms with the surface geometry. A more sophisticated approach is likely required to determine the value of the velocity at the immersed nodes near the interface to properly impose the desired wall boundary conditions.

To further identify discrepancies in the results between methods, we analyze the pressure and skin friction distributions at several stations along the span of the wing. Station locations are shown in Fig. 7, along with the pressure distribution obtained using the body-fitted method. The pressure and skin friction distributions for these stations are plotted in Fig. 8 and 9, respectively. The pressure distributions across the wing are remarkably similar for all methods, though some slight discrepancies in shock location are observed at stations E and F. More significant differences in results are observed for the skin friction distributions.

Notably, the skin friction distributions obtained from the immersed boundary methods at the leading edge of the wing show the starkest difference with the body-fitted results. Two potential sources could be causing such discrepancies: the method by which surface quantities are obtained for the immersed boundary method, and differences in the flow field itself. Recall that surface quantities for the immersed boundary methods are obtained by interpolating flow field quantities at points just off the surface within the fluid domain. Additionally, the shear stress at the surface is computed using the wall model of Berger and Aftosmis [36]. To help determine the largest source of error, we perform this same interpolation and wall model application using the results obtained from the body-fitted method. A comparison of skin

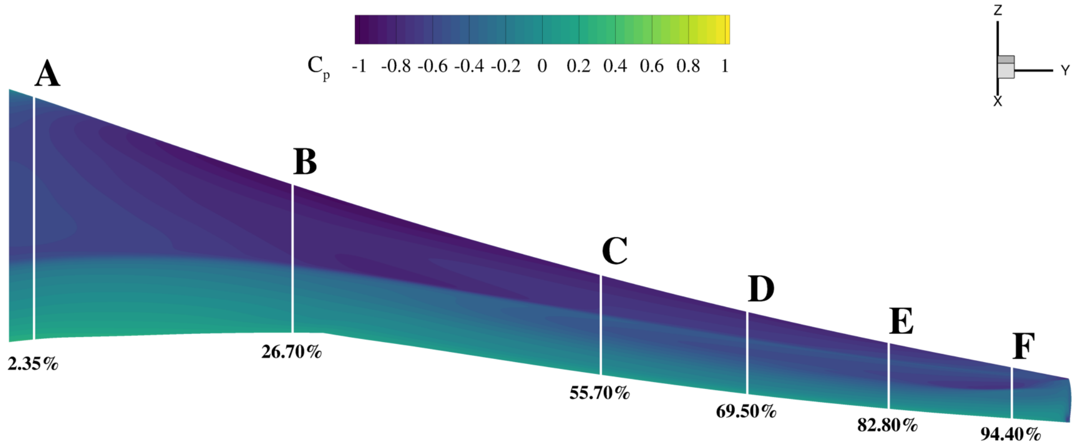


Fig. 7 Spanwise station locations on the wing-only geometry for C_p and C_{fx} slices. Contours show the pressure coefficient on the wing obtained from the body-fitted approach.

friction distribution obtained using this method is labeled as "BF Wall Model" in Fig. 10, and is plotted along with the results from the immersed boundary methods. Interestingly, we observe a much higher peak in skin friction at the leading edge of the wing using the wall model on the body-fitted results. This difference between the traditional body-fitted skin friction and that obtained from the wall model is likely due to the fact that the boundary layer thickness at the leading edge is small. This small thickness is such that the chosen matching point used for the wall model lies beyond the log-layer where the wall function is no longer accurate [47]. Some slight differences remain present between the body-fitted and immersed boundary distributions, indicating that there also exist differences in the flow field at the leading edge. As previously discussed, this may be due to the inaccurate representation of the body in the flow from the immersed boundary approaches leading to non-zero velocities at the surface.

Finally, we present a computational cost comparison between the source-term and ghost-node immersed boundary methods. Shown in Fig. 11 is the residual norm convergence history for a flow solve using each approach, along with the residual norm of the linear system at each Newton iteration. For this comparison, both flow solves were performed on the same 6 million node grid using 480 Intel E5-2680v3 processors capable of executing AVX2 instruction sets. All solver parameters—for example, preconditioner fill level, GMRES Krylov subspace size—were identical. From the comparison, we see that the ghost-node approach provides faster convergence of the residual. This is in part due to the fact that the governing flow equations are not solved on the solid nodes inside the immersed boundary. In fact, this grid in particular contains 610 000 nodes within the immersed domain, 515 000 of which are solid nodes for the ghost-node approach. This means that the source-term approach solved the governing flow equations on 9% more grid nodes than the ghost-node approach for this grid.

To investigate any potential difference in the behavior of the linear solver for each Newton iteration of the flow solve, we show the linear residual convergence achieved at each Newton iteration in Fig. 11b. We note that at each Newton iteration, the linear system is solved using GMRES to either a tolerance of 0.1 or until a Krylov subspace size of 40 vectors has been reached, whichever is attained first. For this flow solve, we see that in fact the convergence of the linear system for the ghost-node immersed boundary approach is, on average, worse than that of the source-term approach. Despite this slightly worse convergence of the linear system, it seems that the reduced number of computational nodes still provides the ghost-node immersed boundary method a lower wall-clock time.

B. Wing-Body-Tail Flow Analysis

Results presented in this section focus on the application and comparison of the body-fitted and immersed boundary methods for the flow analysis of a wing-body-tail configuration. The geometry is that of the CRM for the 4$^{\text{th}}$ AIAA CFD Drag Prediction Workshop [48]. All results are obtained for a freestream Mach number of 0.85 and a Reynolds number of 5 million. Two families of computational grids were used for this test case. An overset body-fitted 3-member grid family with 11, 23, and 55 million nodes is used as a baseline for comparisons. Results for both the source-term and ghost-node immersed boundary methods were obtained using a 3-member grid family with 16, 38, and 110 million nodes.

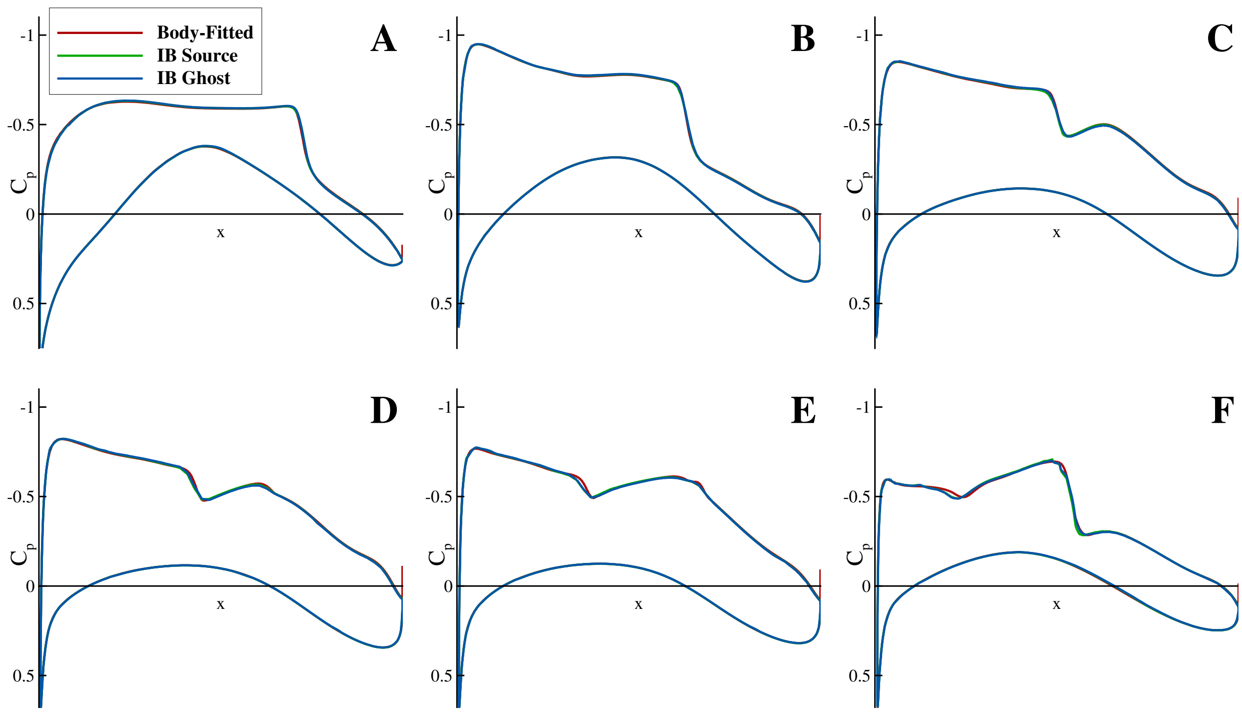


Fig. 8 Pressure coefficient C_p distributions for the wing-only geometry obtained using the body-fitted and immersed boundary methods. Station locations are shown in Fig. 7.

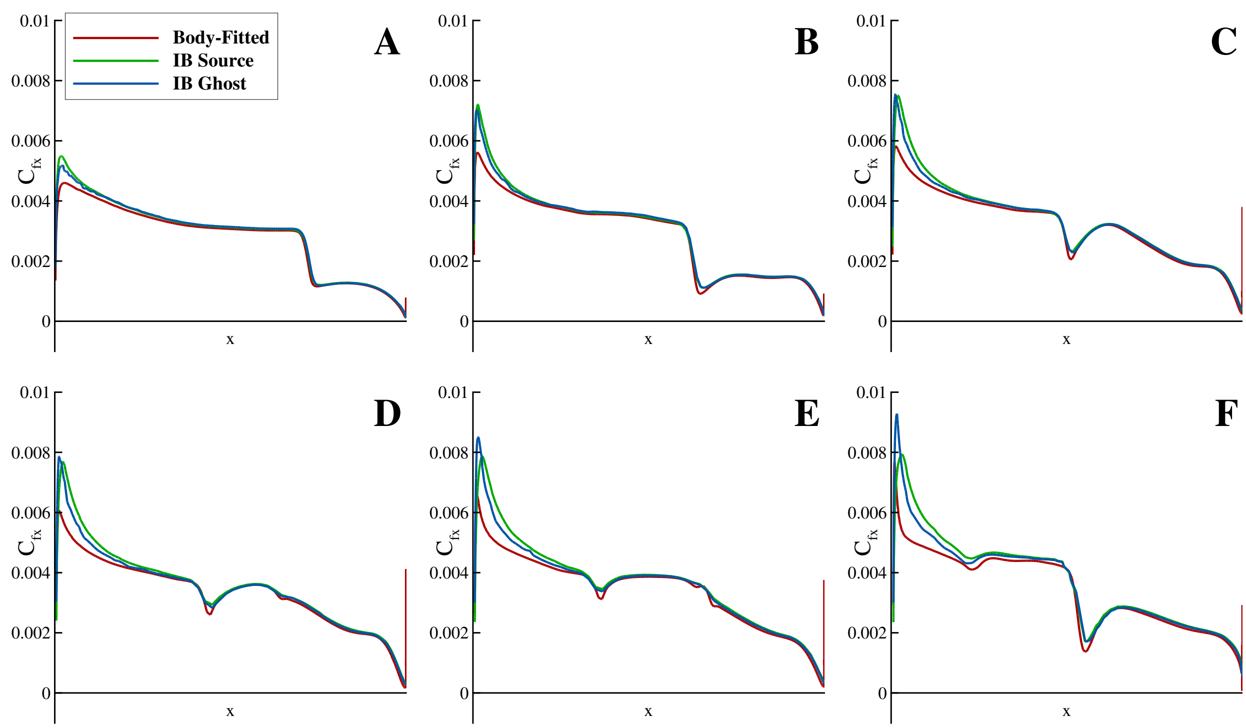


Fig. 9 Skin friction coefficient C_{fx} distributions (upper surface only) for the wing-only geometry obtained using the body-fitted and immersed boundary methods. Station locations are shown in Fig. 7.

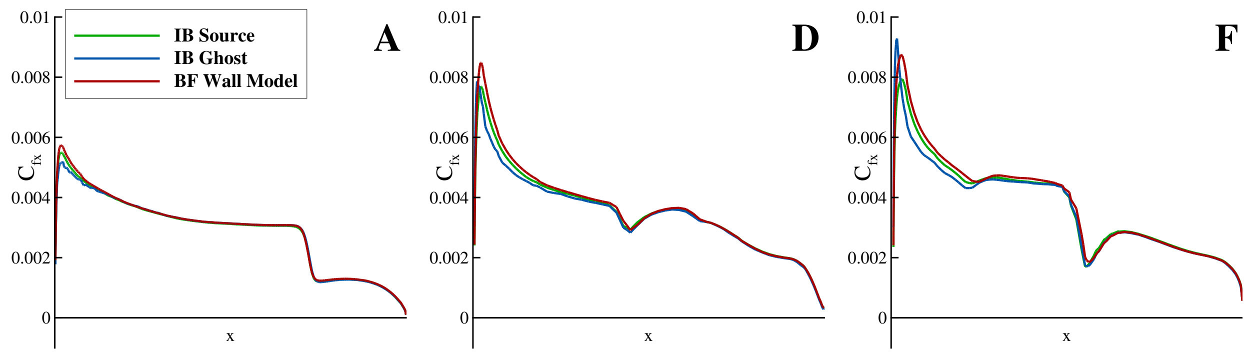
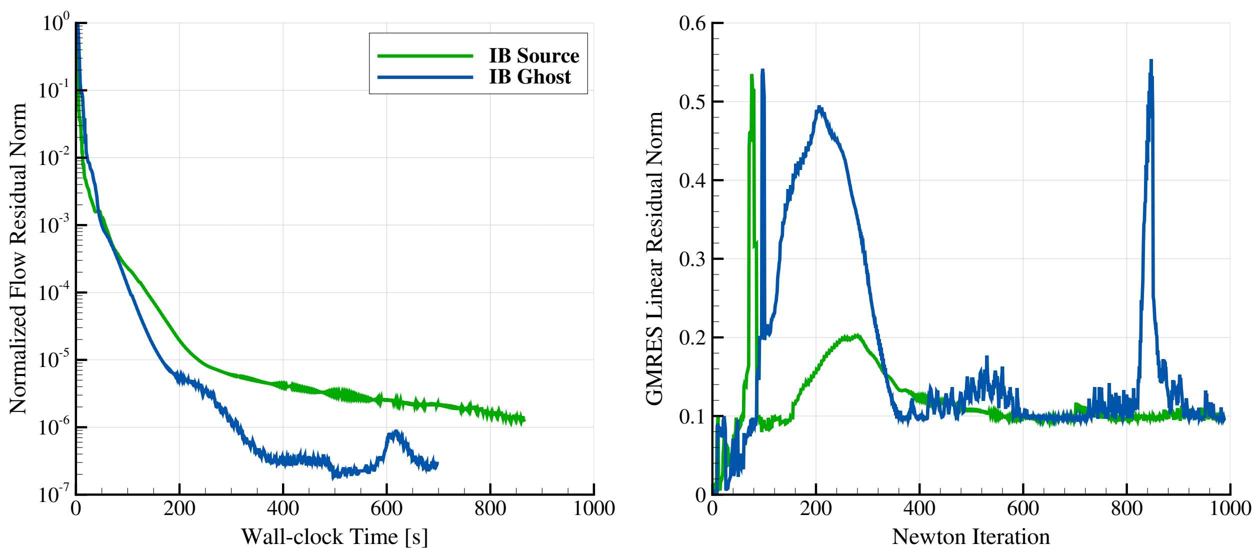


Fig. 10 Skin friction coefficient C_{fx} distributions (upper surface only) for the wing-only geometry obtained using the wall model applied to the flow fields from the body-fitted and immersed boundary methods. Station locations are shown in Fig. 7.



(a) Flow residual convergence history **(b) Linear residual convergence**

Fig. 11 Flow residual convergence history against wall-clock time (a) and GMRES linear residual convergence for each Newton iteration (b) for the source-term and ghost-node immersed boundary methods applied to a 6 million node grid for the wing-only geometry.

We first present the grid convergence results in Fig. 12 for the drag coefficient C_D obtained for a fixed lift coefficient $C_L = 0.5$. From these results, we observe that the body-fitted and source-term immersed boundary approaches converge to similar values, notably to 270.6 and 271.1 drag counts. This is close to the mean of 270.1 drag counts for results from all participants of the 4th AIAA Drag Prediction Workshop [49]. The ghost-node immersed boundary method, however, converges to a continuum value of 274.7 drag counts, 4 counts higher than expected. The fact that this approach converges to a drag value above that of the other approaches is consistent with the results obtained for the wing-only geometry. The simplification used in the ghost-node approach of extrapolating the fluid quantities from the nearest fluid node most aligned with the surface normal is likely the root cause.

As before, we investigate pressure and skin friction distributions along the geometry. Wing slice locations are shown in Fig. 13, with the contours in this plot showing the pressure distribution obtained using body-fitted method. The pressure and skin friction distributions from the slice locations are shown in Fig. 14 and 15, respectively. Focusing first on the body-fitted results, we note that oscillations are present in surface pressure and skin friction distributions on the

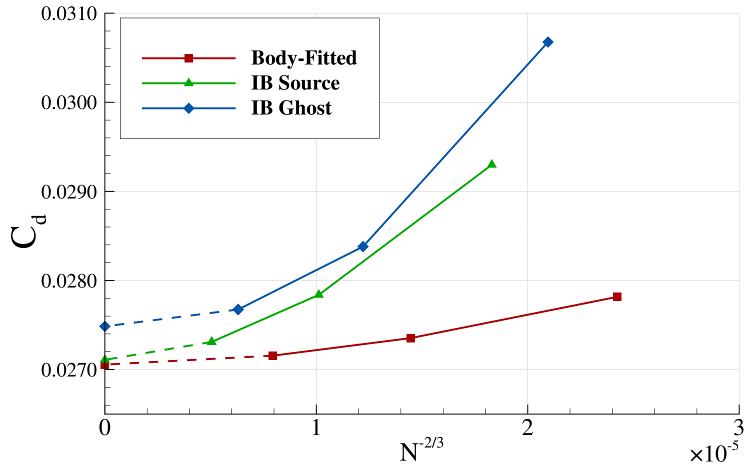


Fig. 12 Grid convergence study of the various approaches for the wing-body-tail geometry.

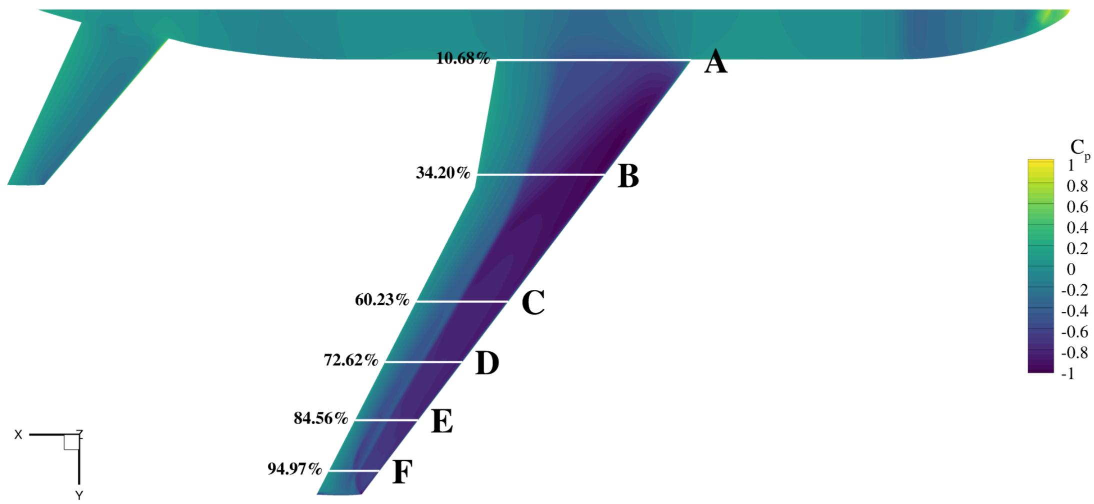


Fig. 13 Spanwise station locations on wing-body-tail geometry for C_p and C_fx slices. Contours show the pressure coefficient on the configuration obtained from the body-fitted approach.

body-fitted grid at slice F near the tip of the wing. This behavior is due to the presence of an overset region at this location on the geometry, where the oscillations are due to the interpolation of the solution between grids. Focusing now on the comparison between immersed boundary and body-fitted surface quantities, we observe a similar story as that of the wing-only results. We see that the pressure distributions largely agree well between all approaches, but significant differences are seen in the skin friction. Focusing first on the pressure distribution, we see some differences in shock location and strength along the surface, more pronounced at the root slice A. As for the skin friction distributions, stations away from the junction demonstrate similar differences as seen for the wing-only case, i.e., larger discrepancies near the leading edge of the wing likely due to the matching point used for the wall model lying beyond of the log-layer. At stations B–F, these differences are reduced as we approach the trailing edge, but this is not the case for slice A near the junction. In fact, the behavior at slice A is seemingly reversed, where we observe good agreement in skin friction at the leading edge, but large differences towards the trailing edge.

To once again better understand the source of these differences, we apply the interpolation and wall model approaches to obtain pressure and skin friction distributions from the body-fitted flow field. A comparison of the pressure and skin friction distributions obtained using this method are labeled as "BF Wall Model" in Fig. 16 and 17. These are plotted along with the results from the immersed boundary methods. From these results, we see that the discrepancy in the shock at the junction remains, indicating a difference in the flow field off of the surface due to the application of the

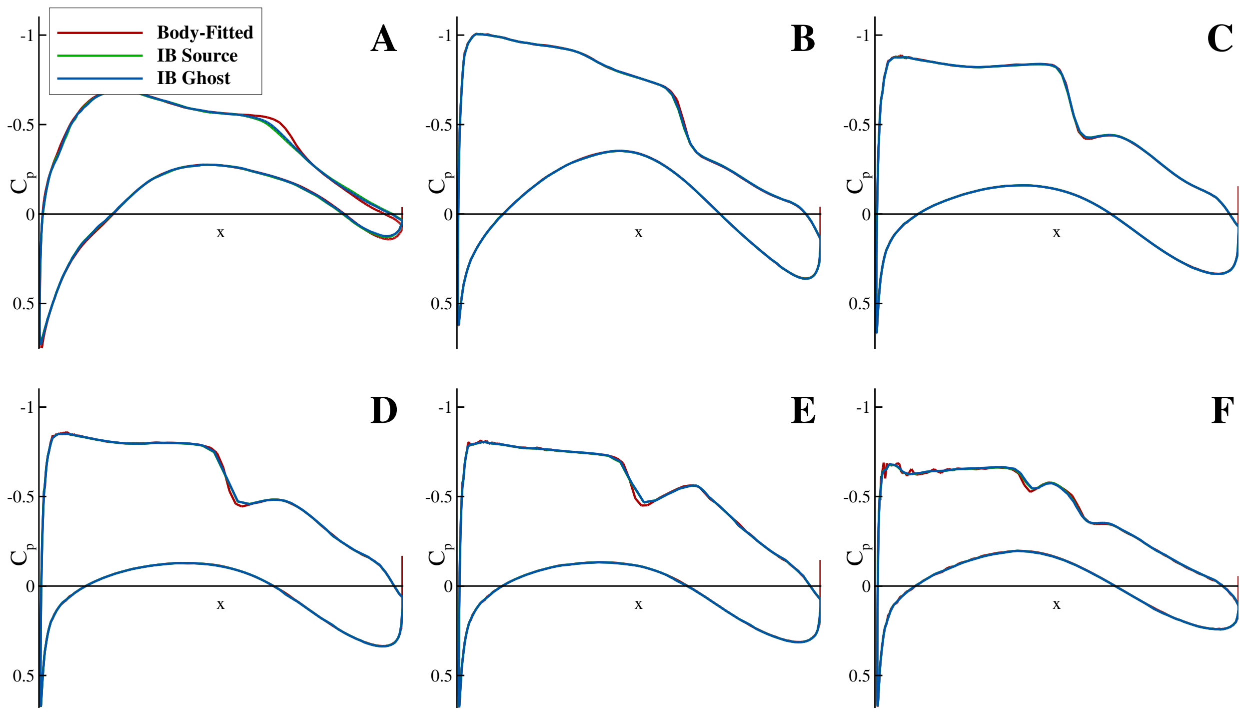


Fig. 14 Pressure coefficient C_p distributions for the wing-body-tail geometry obtained using the body-fitted and immersed boundary methods. Station locations are shown in Fig. 13.

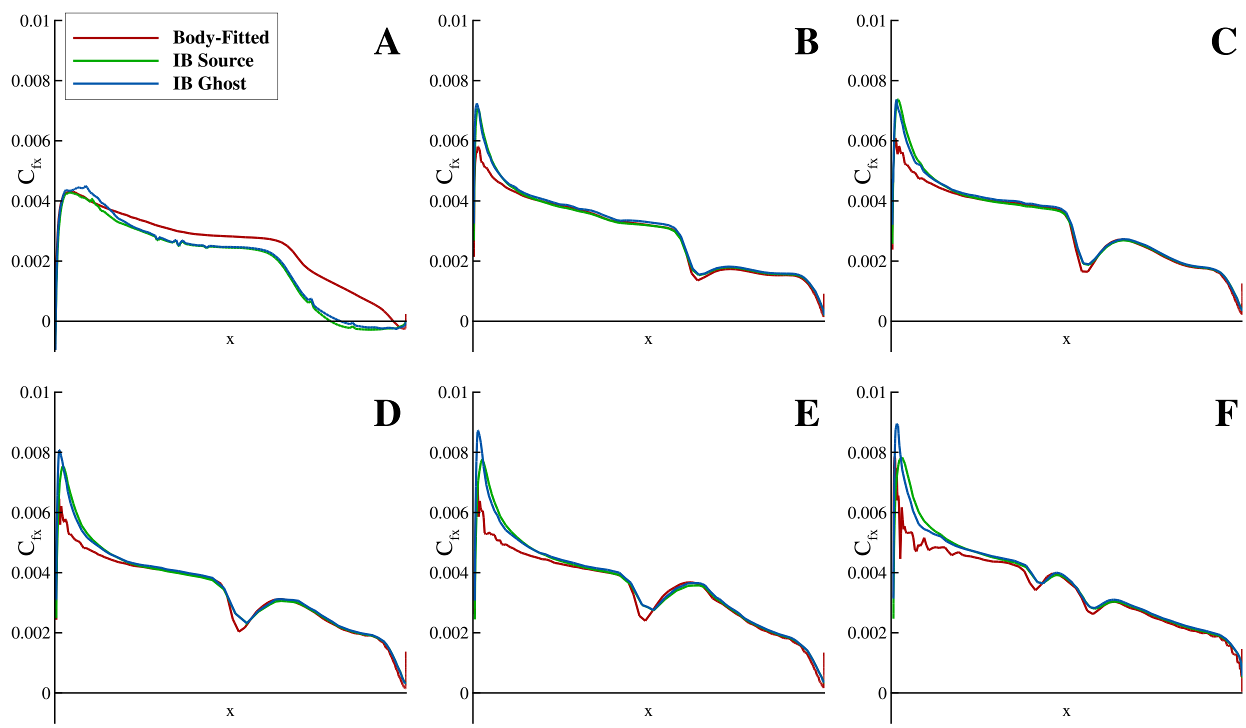


Fig. 15 Skin friction coefficient C_{fx} distributions (upper surface only) for the wing-body-tail geometry obtained using the body-fitted and immersed boundary methods. Station locations are shown in Fig. 13.

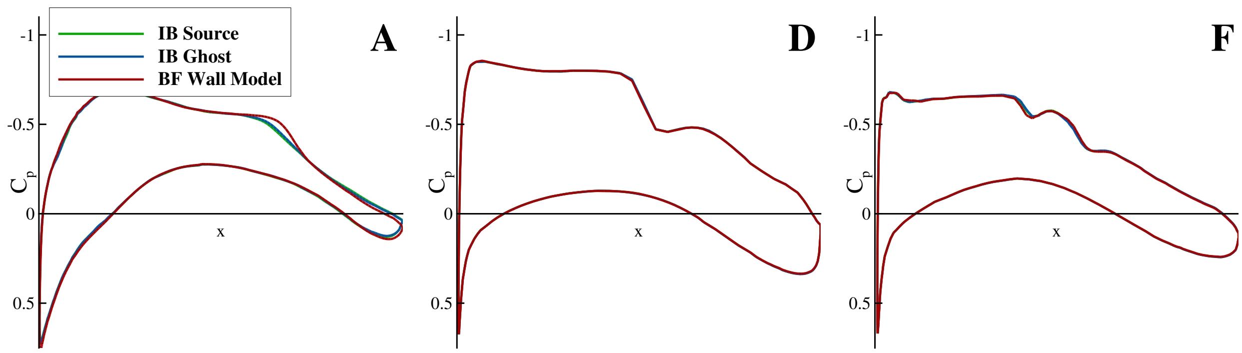


Fig. 16 Pressure coefficient C_p distributions for the wing-body-tail geometry obtained by interpolating flow field pressures onto the surface triangulation for the body-fitted and immersed boundary approaches. Station locations are shown in Fig. 13.

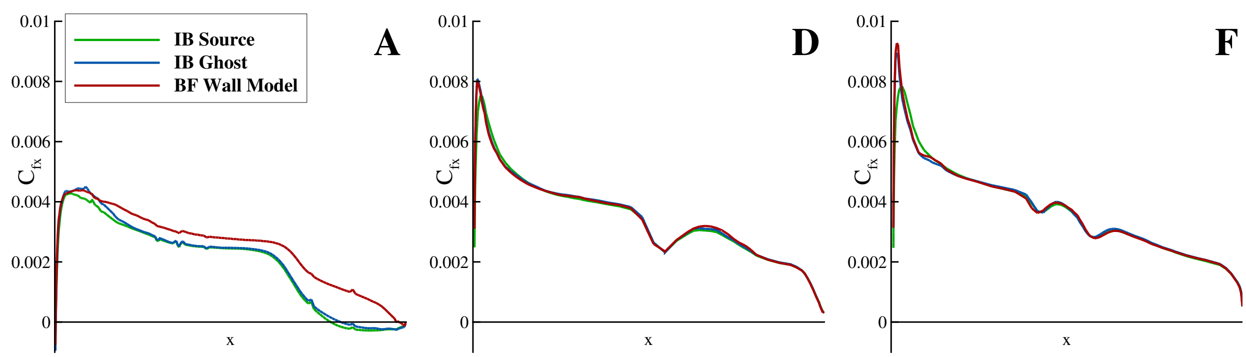
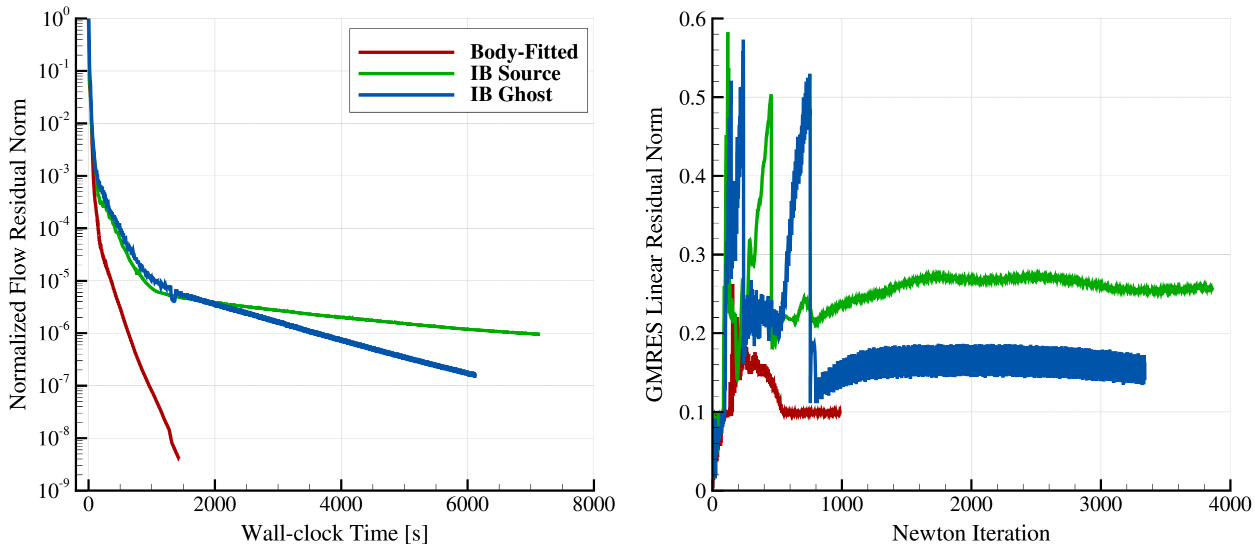


Fig. 17 Skin friction coefficient C_{fx} distributions (upper surface only) for the wing-body-tail geometry obtained using the wall model applied to the flow fields from the body-fitted and immersed boundary approaches. Station locations are shown in Fig. 13.

immersed boundary methods. Regarding the skin friction distributions, the discrepancies between the body-fitted and immersed boundary results have been reduced at slices D and F with the application of the wall model on the body-fitted results. In fact, the skin friction at the leading edge of these slices now closely agrees with results from the ghost-node immersed boundary approach. However, differences in the skin friction at the trailing edge of the root slice A remain relatively unchanged from Fig. 15. We note that results from the body-fitted grid at this location may have errors due to the highly skewed nature of the grid cells in the collar grid at this location, as seen in Fig. 5b. A more thorough study of the flow field obtained using the body-fitted and immersed boundary approaches at the junction is required.

We also present a computational cost comparison between the body-fitted and both immersed boundary methods. Shown in Fig. 18 is the residual norm convergence history for a flow solve using each approach, along with the residual norm of the linear system at each Newton iteration. For this comparison, flow solves using the immersed boundary approaches were performed on the same 38 million node grid. The flow solve for the body-fitted approach was conducted on the 23 million node grid. The comparison between these two grids is deemed fair as they both incorporate similar effective grid spacing. Results were obtained using 960 Intel E5-2680v3 processors capable of executing AVX2 instruction sets. All solver parameters—for example, preconditioner fill level, GMRES Krylov subspace size—were identical. We first see that the body-fitted approach performs the best. This is, in part, thanks to the reduced number of grid nodes on which the body-fitted approach solves the governing flow equations. Despite the same effective grid spacing used, the ghost-node immersed boundary approach solves the governing equations on 28.8% more grid nodes compared to the body-fitted method, while the source-term approach solves on 70.7% more nodes. The better performance of the body-fitted approach is likely also due to its better overall linear system convergence. We recall



(a) Flow residual convergence history **(b) Linear residual convergence**

Fig. 18 Flow residual convergence history against wall-clock time (a) and GMRES linear residual convergence for each Newton iteration (b) for the body-fitted and immersed boundary methods.

that at each Newton iteration, the linear system is solved using GMRES to either a tolerance of 0.1 or until a Krylov subspace size of 40 vectors has been reached, whichever is attained first. As seen in Fig. 18b, the body-fitted approach consistently reaches the 0.1 linear residual tolerance after 600 Newton iterations. This better performance from the linear system provides the body-fitted approach with more accurate flow field updates, thereby reducing the total number of Newton iterations required.

Now comparing the immersed boundary methods, we see that both approaches achieve similar residual convergence until approximately 2000 s (33 minutes), where the residual has dropped by over 5 orders of magnitude. After this point we observe that the ghost-node approach provides better convergence of the residual. As was the case for the wing-only geometry, the application of the source-term immersed boundary method requires that the governing flow equations be solved on 32.5% more nodes for this particular grid compared to the ghost-node approach. This, in part, may explain the slower convergence of the source-term approach plotted against wall-clock time. From Fig. 18b, we see that, though neither immersed boundary approach is capable of attaining the linear residual norm threshold of 0.1, we do observe that the ghost-node immersed boundary method consistently achieves a deeper convergence at each Newton iteration. The worse linear system convergence behavior of the source-term approach is likely owing to the stiffness added to the residual Jacobian from the penalty terms used to enforce the zero velocity condition within the immersed domain.

VI. Discrete Adjoint Results

In this section, we analyze the behavior of the discrete adjoint solver recently implemented into the LAVA framework for application with the source-term and ghost-node immersed boundary approaches. Results are presented for two cases: a NACA 0012 airfoil geometry and the CRM wing-only geometry.

A. Airfoil Geometry Adjoint Solution

This section presents and discusses discrete adjoint solutions for a NACA 0012 airfoil geometry. An overset body-fitted grid containing 350 000 nodes extruded into three spanwise stations is used as the baseline. An overset immersed boundary grid containing 390 000 nodes extruded into three spanwise stations is used for both the source-term and ghost-node immersed boundary approaches. Results are obtained for a Mach number of 0.72, a Reynolds number of 10 million, and an angle of attack of 2°. The functional $\mathcal{J}$ used for all adjoint solutions in this section is the lift coefficient C_L.

Let us first look at the convergence history of the adjoint residual for all approaches, shown in Fig. 19. We observe

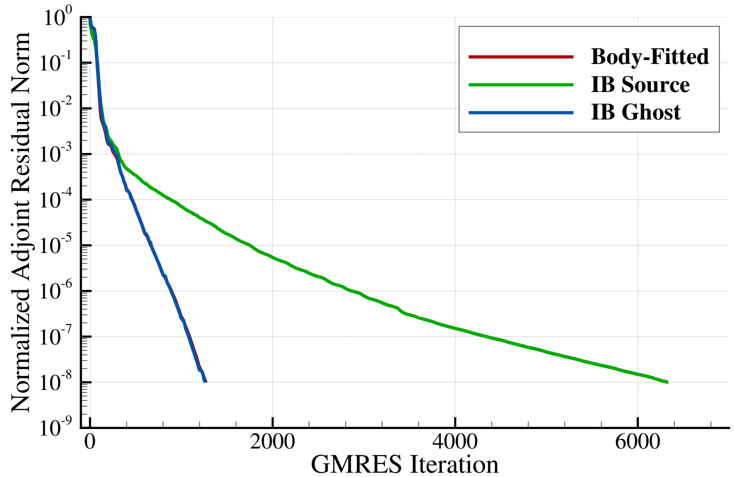
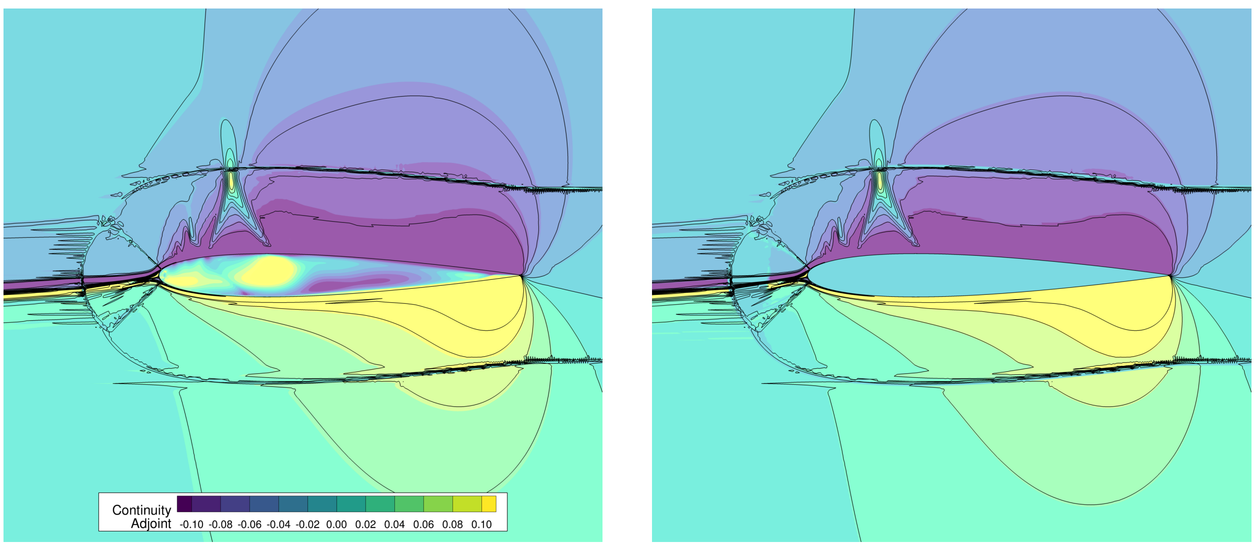


Fig. 19 Convergence history of the discrete adjoint problem for the NACA 0012 airfoil geometry for the lift coefficient functional C_L.



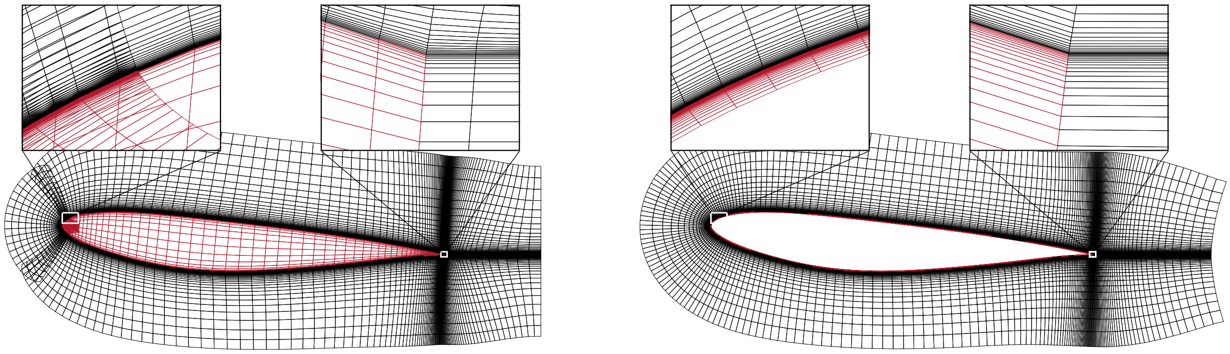
(a) Source-term immersed boundary

(b) Ghost-node immersed boundary

Fig. 20 Comparison of the adjoint solution of the discrete mass conservation equation for the lift functional C_L between the source-term (a) and ghost-node (b) immersed boundary methods. Black contour lines show the adjoint solution from the body-fitted approach.

that the body-fitted and ghost-node immersed boundary approaches demonstrate very similar convergence behavior. This is perhaps unsurprising given the similarities between the methods of enforcing the boundary conditions at the wall. More surprising is the slow convergence of the source-term approach. It is clear that the increased stiffness provided by the source terms inside the immersed domain negatively impact the convergence behavior of the linear solver GMRES. In an attempt to alleviate this poor convergence, the rows of the transposed residual Jacobian $(\partial \boldsymbol{R}_\mathrm{aero}/\partial \boldsymbol{Q})^T$ were scaled by the user-specified penalty time scale $\Delta\tau$ from (2) in order to better balance the norms of the residuals of the linear equations, but this had little impact on improving performance. Further investigation is required to alleviate the negative impact of the source-terms on the convergence of the adjoint solution.

It is also interesting to look at the adjoint solutions provided by the two immersed boundary methods. Shown in Fig. 20 are contour plots for the adjoint of the mass conservation equation. In this figure, we include black contour lines for the adjoint solution obtained using the standard body-fitted approach. Here, we see that the adjoint solution



(a) Immersed topology 1 **(b) Immersed topology 2**

Fig. 21 The two different near-field grid topologies on which the adjoint is solved for the CRM wing-only geometry. Topology 1 (a) is compatible with both the source-term and ghost-node immersed boundary methods; topology 2 (b) is only compatible with the ghost-node method.

qualitatively looks similar between all approaches. Again, unsurprisingly the ghost-node immersed boundary method in Fig. 20b closely resembles the adjoint solution of the body-fitted approach due to the similarities in boundary condition implementation. We note that some differences in the adjoint solution between the ghost-node and body-fitted approaches are observed forward of the leading edge in Fig. 20b. These differences are primarily attributed to differences in the implicit hole cutting of the grid around that same area. The source-term immersed boundary method results in Fig. 20a also show a similar adjoint solution in the fluid domain, though with slightly larger differences with the body-fitted adjoint. Due to the fact that the discrete flow equations are solved within the immersed domain, an adjoint solution is obtained here as well. This intuitively makes sense as this indicates that a change in the flow residual within the body will have an impact on the lift functional of the airfoil.

In Fig. 20, significant oscillations are observed in the discrete adjoint solutions near overset interfaces for all three approaches: body-fitted, source-term immersed boundary, and ghost-node immersed boundary. These oscillations, particularly noticeable at the overset interfaces connecting the near-body grid to the off-body grid extending to the far-field, are reminiscent of those depicted in Fig. 1a of Hicken and Zingg [50], which are characteristic of adjoint-inconsistent schemes. This similarity suggests that the oscillations are associated with adjoint-inconsistency, due to the intimate connection between adjoint consistency and the smoothness of discrete adjoint solutions [51]. Moreover, Nordström and Ghasemi [52] demonstrated that for linear conservation laws, adjoint consistency and conservation are equivalent concepts. Given that the primal discretization lacks discrete conservation at overset interfaces due to the application of trilinear interpolation of the solution, we strongly posit that the observed oscillations at overset interfaces can be attributed to adjoint-inconsistency.

B. Wing-Only Adjoint Solution

This section presents and discusses the convergence history of the discrete adjoint of the CRM wing-only geometry. The body-fitted approach uses a multi-block grid containing 28 million nodes. Two different grid topologies are used for the immersed boundary methods; the near-body grid for each is shown in Fig. 21. The first grid, shown in Fig. 21a, covers the entirety of the immersed domain, and hence is compatible with both the source-term and ghost-node immersed boundary methods. Here, an overset region is present near the leading edge of the wing to circumvent the grid singularity which would occur if the surface grid was simply marched inwards. The second grid, shown in Fig. 21b, only covers a small portion of the immersed domain near the interface. For this grid, no overset region is necessary as we do not march sufficiently into the immersed domain to reach the singularity. Due to the fact that we do not fully encompass the immersed domain, this latter grid is only compatible with the ghost-node approach.

The convergence history of the discrete adjoint solution is shown in Fig. 22. Overall, we observe that the body-fitted approach converges in the least amount of linear iterations, followed by the ghost-node approach using grid topology 2. We note that the source-term immersed boundary stalls after the adjoint residual norm reduces by approximately 2 orders of magnitude. The ghost-node approach converges slowly with the same grid topology but does not stall. Upon closer inspection, we found that the largest adjoint residuals for the source-term approach were found within the immersed

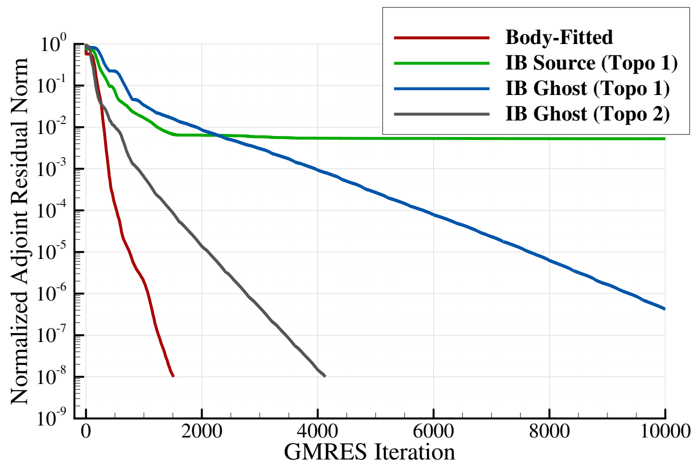


Fig. 22 Convergence history of the discrete adjoint problem for the CRM wing-only geometry for the lift coefficient functional C_{L}.

domain, though no particular region within this domain consistently showed the largest residual among the various grid topologies tested. Further investigation is required to determine the exact cause of the stalling adjoint for the source-term immersed boundary method. It is possible that improved scaling of the linear equations, an improved preconditioner, and/or utilizing a GMRES variant with subspace recycling may aid in improving the convergence behavior.

VII. Optimization Results

In this section we present two optimization results obtained using the ghost-node immersed boundary approach. The first is the inverse design of an RAE 2822 airfoil using a pressure distribution target. The second is the 4th test case from the ADODG [44], specifically the optimization of a wing-only geometry for drag reduction. Due to the poor convergence behavior of the adjoint solution for the source-term immersed boundary approach, no optimization results were obtained using this method.

A. Inverse Design of an RAE 2822 Airfoil

We demonstrate the performance of the ghost-node immersed boundary method for the inverse design of an airfoil at transonic flow conditions. The objective for this optimization problem is,

$$\mathcal{J} = \sum_{j=1}^{n_{\mathrm{tri}}} \left(C_{\mathrm{p},j} - C^{*}_{\mathrm{p},j} \right)^{2}, \tag{12}$$

where $C_{\mathrm{p},j}$ is the pressure coefficient at the j^{th} triangle's centroid on the triangulated surface representation, $C^{*}_{\mathrm{p},j}$ is the target pressure coefficient we wish to achieve at the triangle centroid, and n_{tri} is the number of triangles on the surface. The design variables used for this problem are the z-displacements of the control points for the FFD volume shown in Fig. 23. This FFD volume is composed of 11 control points in the chordwise direction for both the upper and lower surfaces. The design variables are constrained such that the z-displacements of the control points are constant along the span of the extruded airfoil, ensuring that any changes in geometry remain essentially two-dimensional. In total, this results in 22 design variables.

The selected inverse design problem was originally presented by Nemec and Zingg [53]. For this problem, the target pressure distributions corresponds to that of the best approximation of the RAE 2822 airfoil which lies within the design space of the FFD volume of Fig. 23. For this test case, the freestream Mach number is 0.7, the angle of attack is 3°, and the Reynolds number is 9 million. Results for the inverse design problem are shown in Fig. 24. We observe that the optimizer is capable of recovering the target pressure distribution of the approximate RAE 2822 airfoil, with the objective function (12) reaching as low as 10^{-20}. After 100 iterations, we observe that the optimality has dropped by over 10 orders of magnitude. Good convergence of the merit function and optimality are observed for this problem, suggesting sufficiently accurate gradients are being provided to the optimizer. The convergence history of this problem

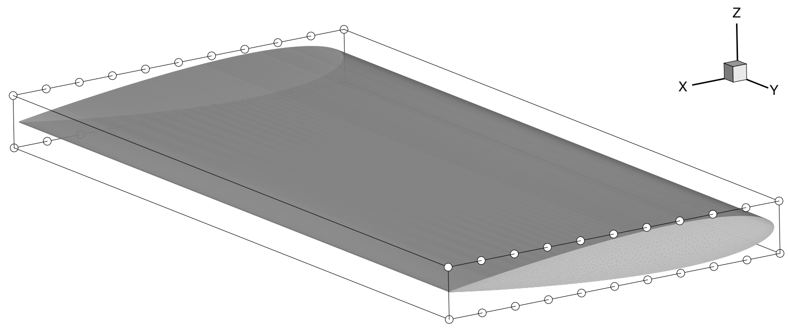
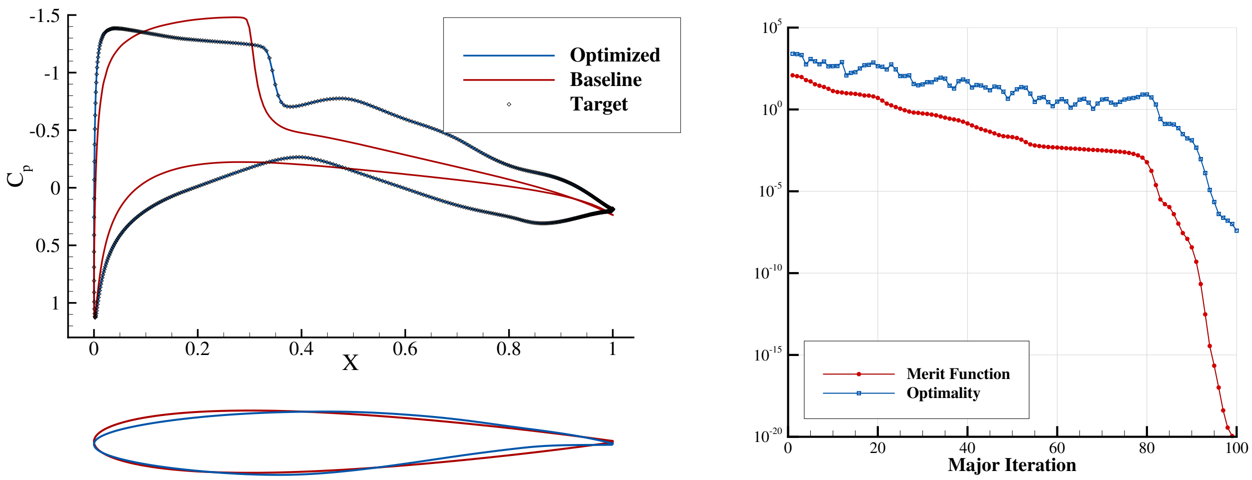


Fig. 23 Extruded NACA 0012 airfoil surface triangulation embedded into an FFD volume. Design variables are the z-displacements of the control points shown as circles.



(a) Pressure distribution and airfoil shape

(b) Optimization convergence history

Fig. 24 Results for the transonic inverse design problem. Baseline and optimized geometries and pressure distributions, along with target pressure distribution (a), and convergence history for the merit function and optimality (b).

is remarkably similar to that achieved by Nemec and Zingg [53], with a significant drop in the merit function seen after 80 iterations.

B. Optimization of a Wing Geometry

In this section, we present preliminary aerodynamic shape optimization results obtained for the CRM wing using the body-fitted grid and ghost-node immersed boundary approaches. The optimization problem is based on the 4th ADODG test case [44], with flow conditions set to a Mach number of 0.85 and a Reynolds number of 5 million. The wing geometry is parameterized using the FFD volume shown in Fig. 25, which is composed of 11 control points in the chordwise direction for each of the upper and lower surfaces, and 15 spanwise stations. Design changes are limited to variations of the section shape of the wing, thus the FFD control points may only move in the vertical z-direction. The trailing edge of the wing is fixed, while the leading edge is free to move vertically except at the root. The angle of attack is also a design variable. The lift coefficient for this optimization case is constrained to be $C_{\mathrm{L}} = 0.5$, while the pitching moment constraint is $C_{\mathrm{My}} \leq -0.17$. The volume of the wing is constrained to be greater than or equal to its initial volume. The thickness of the wing at any given location may be no smaller than 25% the initial thickness of the wing.

Shape optimization using the ghost-node immersed boundary approach was performed on an overset grid containing 10 million nodes. Results for the body-fitted approach were obtained using a multi-block grid containing 3 million nodes. Figure 26 shows a comparison of the baseline geometry and the optimized geometry obtained using the ghost-node

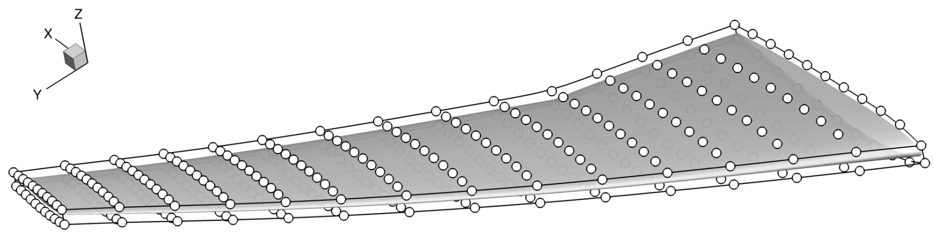


Fig. 25 FFD volume with 330 control points used for aerodynamic shape optimization of the CRM wing.

immersed boundary method. Additionally, Fig. 27 shows a comparison of the optimized geometries obtained using the body-fitted approach (left) and the ghost-node immersed boundary approach (right).

We note that the results obtained using the ghost-node immersed boundary method are preliminary for two reasons. First, some difficulty was encountered when determining proper interpolation points for the surface as the geometry of the wing was deformed throughout the optimization. This behavior led to regions on the triangulated surface geometry lacking pressure and shear stress information, which in turn would lead to an under-prediction of aerodynamic functionals. As a temporary remedy, the interpolation weights $\boldsymbol{W}$ from (8) are held frozen such that they did not change throughout the optimization. This leads to an approximation in the value of the surface quantities. To assess the error in the final value of C_D obtained using the ghost-node immersed boundary method, we can apply the optimized FFD control point deflections to the body-fitted grid to obtain a comparative drag value. After adjusting the angle of attack to achieve of lift coefficient of $C_\mathrm{L} = 0.5$, the deformed body-fitted grid provides a drag coefficient of $C_\mathrm{D} = 0.0196$. In comparison, the ghost-node approach with frozen interpolation weights predicted a drag coefficient of $C_\mathrm{D} = 0.0179$. This presents a 17 drag count difference between the immersed boundary method with frozen interpolation weights and the equivalently deformed body-fitted grid. As such, we cannot claim that freezing the surface interpolation weights provides a good approximation of the flow functionals after deformation of the grid. However, we will optimistically note that the drag coefficient $C_\mathrm{D} = 0.0196$ is lower than the baseline wing's drag of $C_\mathrm{D} = 0.0201$ on the undeformed body-fitted grid. We thus see that shape optimization with the immersed boundary method has reaped some benefit when applying its optimized design variables to the body-fitted grid.

The second reason for which the ghost-node immersed boundary optimization results are preliminary is because of the numerous grid movement failures observed during the optimization process. After several grid movement failures, the optimization concluded it could no longer proceed into what was deemed an undefined region. These failures mostly manifested in the form of negative metric Jacobians at the trailing edge of the wing obtained when the optimizer attempted to reduce its thickness. This behavior originates from the difference in resolution between the surface triangulation and the volume grid near the surface. The movement of each volume node is dictated by a weighted sum of the deformation of the vertices which compose the surface triangulation. The weights are based on the inverse distance between the volume node and the vertices. The nature of the computed weights used in the algorithm is such that the volume nodes near the surface would essentially rigidly translate with their nearest vertex. Such behavior is desirable for body-fitted grids as we want volume nodes within the boundary layer to closely follow the deformation of the corresponding surface node along the same grid line. However, in the case of non-matching volume and surface grids, if two or more volume nodes are located within a single surface triangle, each node may rigidly follow a different vertex, and possibly cross each other, leading to negative Jacobians. One strategy to combat such crossover is to construct a discrete representation of the surface for grid deformation by placing new surface nodes where the surface triangulation intersects with cell edges in the volume grid. In this way, a new discrete surface can be constructed which has the same resolution as the volume grid.

Despite the challenges surrounding the application of the ghost-node immersed boundary method, we still observe improved aerodynamic performance compared to the baseline design. When comparing results between the baseline and optimized designs in Fig. 26, we see that the optimizer has essentially eliminated the shock across the upper surface of the wing. Additionally, the lift distribution across the span of the optimized wing is more elliptical. In Fig. 27, we observe some significant differences between the body-fitted and the ghost-node immersed boundary optimized geometries. Notably, the body-fitted results show that the bulk of the wing volume is shifted inboard. This behavior is consistent with optimized results obtained by Lyu et al. [45] and Lee et al. [46]. Certain features seen in the optimized immersed boundary results, such as the wavy pressure distributions along the pressure side of the airfoil, are likely due to

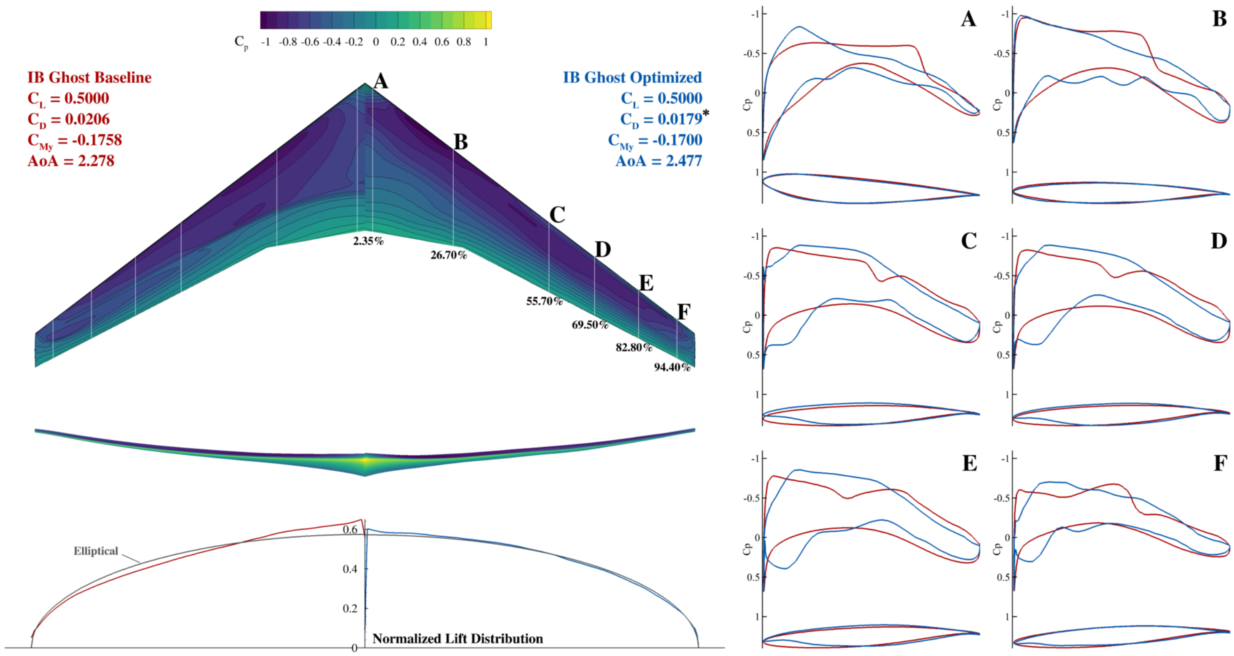


Fig. 26 Comparison of the initial and optimized results for the CRM wing geometry using the ghost-node immersed boundary method.

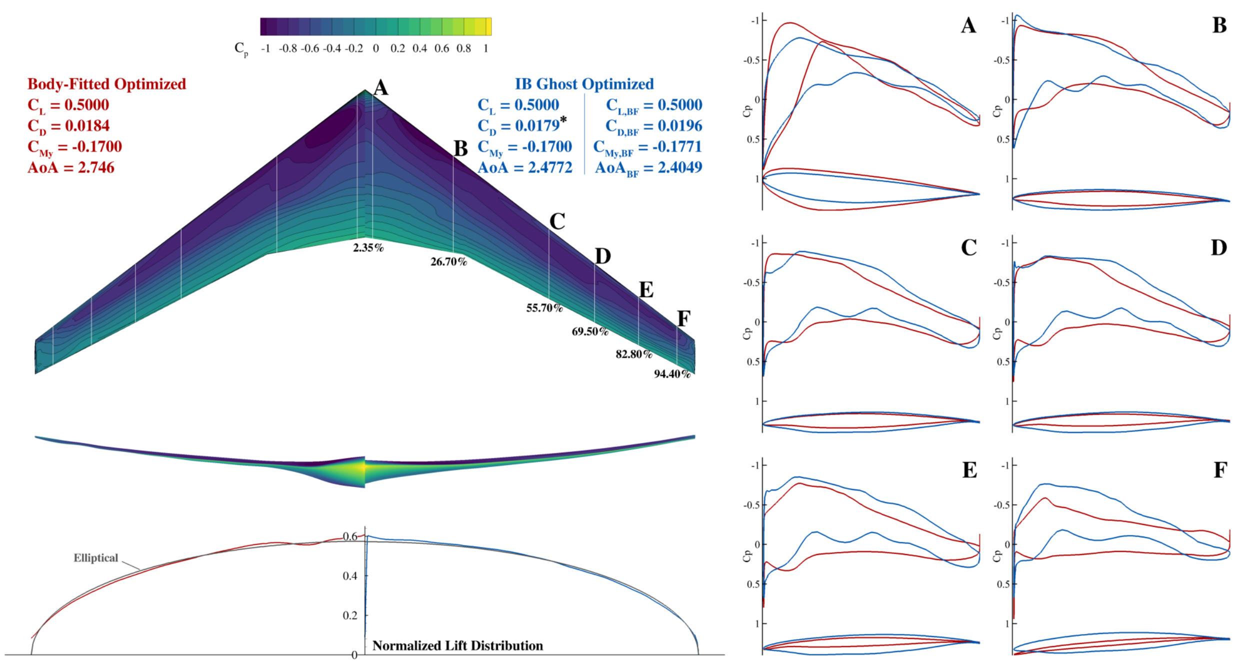


Fig. 27 Comparison of the optimized results for the CRM wing geometry using the body-fitted and ghost-node immersed boundary methods. Functionals and angle of attack with a subscript "BF" correspond to results on the body-fitted mesh deformed using the optimized design variables from the ghost-node approach and trimmed for the lift constraint.

*Optimized functionals C_L, C_D, C_{My} obtained for the ghost-node immersed boundary method are approximations due to frozen interpolation weights.

the premature termination of the optimization procedure due to grid movement failures. Additionally, the lack of volume shift towards the root of the wing is likely due to the same reason. With further improvement of the immersed boundary method for shape optimization, we expect to achieve a similar result to that obtained for the body-fitted approach.

VIII. Conclusion

In this paper, we presented two immersed boundary methods for flow analysis and aerodynamic shape optimization on overset curvilinear grids. The current iterations of the methods serve as a stepping stone towards replacing the need for collar grids at component intersections for overset grids. Forsaking collar grids will aid in both reducing the time required to generate curvilinear grids, as well as reduce the failure-prone nature of grid deformation at component intersections for shape optimization.

Flow analysis results indicate that similar surface quantities are obtained using the immersed boundary methods when compared to the body-fitted approach. Notably, reasonable agreement was seen for pressure distributions throughout the surface and skin friction distributions near the trailing edge. However, large discrepancies are observed for skin friction distributions at the leading edge and the junction of the geometries. The former is likely due to the matching point used for the wall model lying outside of the log-layer; further investigation is required to determine the source of the latter. Additionally, continuum drag values were over-predicted using the ghost-node approach, and larger errors in drag were seen at coarser grid levels for both immersed boundary methods. This may be due to non-zero velocities at the surface of the body stemming from the current implementations of the immersed boundary methods. More sophisticated implementations of the methods, such as extrapolating velocity targets [33], may rectify the differences, particularly at coarse grid levels.

Discrete adjoint solutions obtained around a NACA 0012 airfoil geometry were remarkably similar between the body-fitted and immersed boundary methods. However, on the three-dimensional wing-only geometry, numerical difficulties led to a lack of convergence for the adjoint solution of the source-term immersed boundary method. This numerical difficulty likely stems from the added stiffness in residual Jacobian originating from the penalty source terms. Further investigation into improved scaling of the linear equations, an improved preconditioner, and/or utilizing a GMRES variant with subspace recycling may aide in improving the convergence behavior. Due to the numerical difficulty in solving the adjoint problem with the source-term immersed boundary method, shape optimization was not conducted with this approach.

Aerodynamic shape optimization results were obtained using the ghost-node immersed boundary and body-fitted approaches. The inverse design of an approximate RAE 2822 airfoil using the ghost-node immersed boundary method showed excellent optimization convergence behavior, suggesting that sufficiently accurate gradients are obtained using the discrete adjoint method. Applying the ghost-node approach for the optimization of a wing-only geometry revealed numerous difficulties faced by the method. These include a lack of robustness in properly interpolating the flow field onto the surface triangulation for functional computations, as well as numerous grid movement failures due to non-matching resolutions near the surface between the volume grid and surface triangulation. Despite these difficulties, the optimizer was able to provide improved aerodynamic performance, though the final optimized geometry did not match that obtained using the traditional body-fitted approach.

Much work remains in order to increase the accuracy and reliability of the proposed immersed boundary methods for aerodynamic shape optimization. The ghost-node immersed boundary approach is computationally the most attractive thanks to its reduced computational cost and better convergence behavior for the adjoint solution. Continued work with this approach will require that the current implementation be updated to provide more accurate flow field results and functionals. Additionally, the grid movement and surface integration problems observed during shape optimization must be rectified. In all, the ghost-node immersed boundary method seems to provide a potential path forward to eliminating the need for collar grids at component intersections for curvilinear overset grids.

Acknowledgments

This work was sponsored by the Convergent Aeronautics Solutions (CAS) project and the Transformational Tools & Technologies (TTT) project, which are part of the Transformational Aeronautics Concepts Program (TACP) under the NASA Aeronautics Mission Directorate (ARMD). Special thanks are due to Marian Nemec from NASA Ames Research Center for his expert technical guidance and insightful discussions. The authors would like to gratefully acknowledge Timothy Chau for his helpful review, and the LAVA team for valuable advice and feedback. Computational resources were provided by the NASA High-End Computing (HEC) Program through the NASA Advanced Supercomputing

(NAS) Division at Ames Research Center.

References

[1] Steger, J. L., Dougherty, F. C., and Benek, J. A., "A Chimera Grid Scheme," *Advances in Grid Generation*, Applied Mechanics, Bioengineering, and Fluids Engineering Conference, Vol. 5, edited by K. N. Ghia and U. Ghia, The Fluids Engineering Division, ASME, 1983, pp. 59–70.

[2] Steger, J. L., and Benek, J. A., "On the Use of Composite Grid Schemes in Computational Aerodynamics," *Computer Methods in Applied Mechanics and Engineering*, Vol. 64, No. 1, 1987, pp. 301–320. https://doi.org/10.1016/0045-7825(87)90045-4.

[3] Chan, W., Gomez, R., Rogers, S., and Buning, P., "Best Practices in Overset Grid Generation," *32nd AIAA Fluid Dynamics Conference and Exhibit*, AIAA, St. Louis, Missouri, 2002. https://doi.org/10.2514/6.2002-3191.

[4] Osusky, L., Buckley, H., Reist, T., and Zingg, D. W., "Drag Minimization Based on the Navier–Stokes Equations Using a Newton–Krylov Approach," *AIAA Journal*, Vol. 53, No. 6, 2015, pp. 1555–1577. https://doi.org/10.2514/1.j053457.

[5] Reist, T. A., Koo, D., Zingg, D. W., Bochud, P., Castonguay, P., and Leblond, D., "Cross Validation of Aerodynamic Shape Optimization Methodologies for Aircraft Wing-Body Optimization," *AIAA Journal*, Vol. 58, No. 6, 2020, pp. 2581–2595. https://doi.org/10.2514/1.j059091.

[6] Secco, N. R., and Martins, J. R. R. A., "RANS-Based Aerodynamic Shape Optimization of a Strut-Braced Wing with Overset Meshes," *Journal of Aircraft*, Vol. 56, No. 1, 2019, pp. 217–227. https://doi.org/10.2514/1.c034934.

[7] Mader, C. A., Kenway, G. K. W., Yildirim, A., and Martins, J. R. R. A., "ADflow: An Open-Source Computational Fluid Dynamics Solver for Aerodynamic and Multidisciplinary Optimization," *Journal of Aerospace Information Systems*, Vol. 17, No. 9, 2020, pp. 508–527. https://doi.org/10.2514/1.i010796.

[8] Seraj, S., and Martins, J. R., "Aerodynamic Shape Optimization of a Supersonic Transport Considering Low-Speed Stability," *AIAA SCITECH 2022 Forum*, AIAA, San Diego, California, 2022. https://doi.org/10.2514/6.2022-2177.

[9] Elliott, J., and Herling, W., "A Chimera Approach to Aerodynamic Shape Optimization for the Compressible, High-Re Navier-Stokes Equations," *8th Symposium on Multidisciplinary Analysis and Optimization*, AIAA, Long Beach, California, 2000. https://doi.org/10.2514/6.2000-4729.

[10] Liao, W., and Tsai, H. M., "Aerodynamic Shape Optimization on Overset Grids Using the Adjoint Method," *International Journal for Numerical Methods in Fluids*, Vol. 62, No. 12, 2009, pp. 1332–1356. https://doi.org/10.1002/fld.2070.

[11] Lee, B. J., and Kim, C., "Aerodynamic Redesign Using Discrete Adjoint Approach on Overset Mesh System," *Journal of Aircraft*, Vol. 45, No. 5, 2008, pp. 1643–1653. https://doi.org/10.2514/1.34112.

[12] Lee, B. J., Liou, M.-S., and Kim, C., "Optimizing a Boundary-Layer-Ingestion Offset Inlet by Discrete Adjoint Approach," *AIAA Journal*, Vol. 48, No. 9, 2010, pp. 2008–2016. https://doi.org/10.2514/1.j050222.

[13] Kenway, G. K., Mishra, A., Secco, N. R., Duraisamy, K., and Martins, J. R. R. A., "An Efficient Parallel Overset Method for Aerodynamic Shape Optimization," *58th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, AIAA, Grapevine, Texas, 2017. https://doi.org/10.2514/6.2017-0357.

[14] Secco, N. R., Jasa, J. P., Kenway, G. K. W., and Martins, J. R. R. A., "Component-Based Geometry Manipulation for Aerodynamic Shape Optimization with Overset Meshes," *AIAA Journal*, Vol. 56, No. 9, 2018, pp. 3667–3679. https://doi.org/10.2514/1.j056550.

[15] Yildirim, A., Mader, C. A., and Martins, J. R. R. A., "A Surface Mesh Deformation Method Near Component Intersections for High-Fidelity Design Optimization," *Engineering with Computers*, Vol. 38, 2021, pp. 1393–1425. https://doi.org/10.1007/s00366-020-01247-w.

[16] Peskin, C. S., "Flow Patterns Around Heart Valves: A Numerical Method," *Journal of Computational Physics*, Vol. 10, No. 2, 1972, pp. 252–271. https://doi.org/10.1016/0021-9991(72)90065-4.

[17] Mohd-Yusof, J., "Combined Immersed-Boundary/B-Spline Methods for Simulations of Flows in Complex Geometries," *Annual Research Briefs*, Center for Turbulence Research, 1997, pp. 317–328.

[18] Peskin, C. S., "The Immersed Boundary Method," *Acta Numerica*, Vol. 11, 2002, pp. 479–517. https://doi.org/10.1017/s0962492902000077.

[19] Iaccarino, G., and Verzicco, R., "Immersed boundary technique for turbulent flow simulations ," *Applied Mechanics Reviews*, Vol. 56, No. 3, 2003, pp. 331–347. https://doi.org/10.1115/1.1563627.

[20] Mittal, R., and Iaccarino, G., "Immersed Boundary Method," *Annual Review of Fluid Mechanics*, Vol. 37, 2005, pp. 239–261. https://doi.org/10.1146/annurev.fluid.37.061903.175743.

[21] Mittal, R., and Seo, J. H., "Origin and Evolution of Immersed Boundary Methods in Computational Fluid Dynamics," *Physical Review Fluids*, Vol. 8, No. 10, 2023, 100501. https://doi.org/10.1103/physrevfluids.8.100501.

[22] Nemec, M., Aftosmis, M., and Pulliam, T., "CAD-Based Aerodynamic Design of Complex Configurations Using a Cartesian Method," *42nd AIAA Aerospace Sciences Meeting and Exhibit*, AIAA, Reno, Nevada, 2004. https://doi.org/10.2514/6.2004-113.

[23] Gobal, K., and Grandhi, R. V., "High-Fidelity Aerodynamic Shape Optimization using Immersed Boundary Method," *35th AIAA Applied Aerodynamics Conference*, AIAA, Denver, Colorado, 2017. https://doi.org/10.2514/6.2017-4364.

[24] Mochel, L., Weiss, P.-É., and Deck, S., "Zonal Immersed Boundary Conditions: Application to a High-Reynolds-Number Afterbody Flow," *AIAA Journal*, Vol. 52, No. 12, 2014, pp. 2782–2794. https://doi.org/10.2514/1.j052970.

[25] Weiss, P.-É., and Deck, S., "On the Coupling of a Zonal Body-Fitted/Immersed Boundary Method With Zdes: Application to the Interactions on a Realistic Space Launcher Afterbody Flow," *Computers & Fluids*, Vol. 176, 2018, pp. 338–352. https://doi.org/10.1016/j.compfluid.2017.06.015.

[26] Wong, M. L., Ghate, A. S., Stich, G.-D., Kenway, G. K., and Kiris, C. C., "Numerical Study on the Aerodynamics of an Iced Airfoil with Scale-Resolving Simulations," *AIAA SCITECH 2023 Forum*, AIAA, National Harbor, Maryland, 2023. https://doi.org/10.2514/6.2023-0252.

[27] Craig Penner, D. A., Housman, J. A., Stich, G.-D., Sousa, V. C. B., Koch, J. R. L., and Duensing, J. C., "Wall-Modeled Large-Eddy Simulations of a Swept Wing with Leading-Edge Ice," *AIAA AVIATION 2024 Forum*, AIAA, Las Vegas, NV, 2024.

[28] Kiris, C. C., Housman, J. A., Barad, M. F., Brehm, C., Sozer, E., and Moini-Yekta, S., "Computational Framework for Launch, Ascent, and Vehicle Aerodynamics (LAVA)," *Aerospace Science and Technology*, Vol. 55, 2016, pp. 189–219. https://doi.org/10.1016/j.ast.2016.05.008.

[29] Housman, J. A., Kiris, C. C., and Hafez, M. M., "Time-Derivative Preconditioning Methods for Multicomponent Flows—Part I: Riemann Problems," *Journal of Applied Mechanics*, Vol. 76, No. 2, 2009, 021210. https://doi.org/10.1115/1.3072905.

[30] Housman, J. A., Kiris, C. C., and Hafez, M. M., "Time-Derivative Preconditioning Methods for Multicomponent Flows—Part II: Two-Dimensional Applications," *Journal of Applied Mechanics*, Vol. 76, No. 3, 2009, 031013. https://doi.org/10.1115/1.3086592.

[31] Saad, Y., and Schultz, M. H., "GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems," *SIAM Journal on Scientific and Statistical Computing*, Vol. 7, No. 3, 1986, pp. 856–869. https://doi.org/10.1137/0907058.

[32] Angot, P., Bruneau, C.-H., and Fabrie, P., "A Penalization Method to Take Into Account Obstacles in Incompressible Viscous Flows," *Numerische Mathematik*, Vol. 81, 1999, pp. 497–520. https://doi.org/10.1007/s002110050401.

[33] Fadlun, E., Verzicco, R., Orlandi, P., and Mohd-Yusof, J., "Combined Immersed-Boundary Finite-Difference Methods for Three-Dimensional Complex Flow Simulations," *Journal of Computational Physics*, Vol. 161, No. 1, 2000, pp. 35–60. https://doi.org/10.1006/jcph.2000.6484.

[34] LeVeque, R. J., *Finite Volume Methods for Hyperbolic Problems*, Cambridge University Press, 2002. https://doi.org/10.1017/cbo9780511791253.

[35] Tseng, Y.-H., and Ferziger, J. H., "A Ghost-Cell Immersed Boundary Method for Flow in Complex Geometry," *Journal of Computational Physics*, Vol. 192, No. 2, 2003, pp. 593–623. https://doi.org/10.1016/j.jcp.2003.07.024.

[36] Berger, M., and Aftosmis, M., "Progress Towards a Cartesian Cut-Cell Method for Viscous Compressible Flow," *50th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*, AIAA, Nashville, Tennessee, 2012. https://doi.org/10.2514/6.2012-1301.

[37] Lee, Y., and Baeder, J., "Implicit Hole Cutting - A New Approach to Overset Grid Connectivity," *16th AIAA Computational Fluid Dynamics Conference*, American Institute of Aeronautics and Astronautics, Orlando, Florida, 2003. https://doi.org/10.2514/6.2003-4128.

[38] Gill, P. E., Murray, W., and Saunders, M. A., "SNOPT: An SQP Algorithm for Large-Scale Constrained Optimization," *SIAM Review*, Vol. 47, No. 1, 2005, pp. 99–131. https://doi.org/10.1137/s0036144504446096.

[39] Wu, N., Kenway, G., Mader, C., Jasa, J., and Martins, J., "pyOptSparse: Aerodynamic Shape Optimization on Overset Grids Using the Adjoint Method," *Journal of Open Source Software*, Vol. 5, No. 54, 2020, 2564. https://doi.org/10.21105/joss.02564.

[40] Hascoet, L., and Pascual, V., "The Tapenade Automatic Differentiation Tool: Principles, Model, and Specification," *ACM Transactions on Mathematical Software*, Vol. 39, No. 3, 2013, pp. 1–43. https://doi.org/10.1145/2450153.2450158.

[41] Sederberg, T. W., and Parry, S. R., "Free-Form Deformation of Solid Geometric Models," *ACM SIGGRAPH Computer Graphics*, Vol. 20, No. 4, 1986, pp. 151–160. https://doi.org/10.1145/15886.15903.

[42] Hajdik, H. M., Yildirim, A., Wu, N., Brelje, B. J., Seraj, S., Mangano, M., Anibal, J. L., Jonsson, E., Adler, E. J., Mader, C. A., Kenway, G. K. W., and Martins, J. R. R. A., "pyGeo: A Geometry Package for Multidisciplinary Design Optimization," *Journal of Open Source Software*, Vol. 8, No. 87, 2023, 5319. https://doi.org/10.21105/joss.05319.

[43] Secco, N. R., Kenway, G. K. W., He, P., Mader, C., and Martins, J. R. R. A., "Efficient Mesh Generation and Deformation for Aerodynamic Shape Optimization," *AIAA Journal*, Vol. 59, No. 4, 2021, pp. 1151–1168. https://doi.org/10.2514/1.j059491.

[44] Nadarajah, S., "AIAA Aerodynamic Design Optimization Discussion Group," https://sites.google.com/view/mcgill-computational-aerogroup/adodg, 2022. [Online; accessed 17-July-2024].

[45] Lyu, Z., Kenway, G. K. W., and Martins, J. R. R. A., "Aerodynamic Shape Optimization Investigations of the Common Research Model Wing Benchmark," *AIAA Journal*, Vol. 53, No. 4, 2015, pp. 968–985. https://doi.org/10.2514/1.j053318.

[46] Lee, C., Koo, D., and Zingg, D. W., "Comparison of B-Spline Surface and Free-Form Deformation Geometry Control for Aerodynamic Optimization," *AIAA Journal*, Vol. 55, No. 1, 2017, pp. 228–240. https://doi.org/10.2514/1.j055102.

[47] Berger, M. J., and Aftosmis, M. J., "An ODE-Based Wall Model for Turbulent Flow Simulations," *AIAA Journal*, Vol. 56, No. 2, 2018, pp. 700–714. https://doi.org/10.2514/1.j056151.

[48] Morrison, J. H., "4th AIAA CFD Drag Prediction Workshop," https://aiaa-dpw.larc.nasa.gov/Workshop4/workshop4.html, 2011. [Online; accessed 14-July-2024].

[49] Vassberg, J., Tinoco, E., Mani, M., Rider, B., Zickuhr, T., Levy, D., Brodersen, O., Eisfeld, B., Crippa, S., Wahls, R., Morrison, J., Mavriplis, D., and Murayama, M., "Summary of the Fourth AIAA CFD Drag Prediction Workshop," *28th AIAA Applied Aerodynamics Conference*, American Institute of Aeronautics and Astronautics, Chicago, Illinois, 2010. https://doi.org/10.2514/6.2010-4547.

[50] Hicken, J., and Zingg, D., "Dual Consistency and Functional Accuracy: A Finite-Difference Perspective," *Journal of Computational Physics*, Vol. 256, 2014, pp. 161–182. https://doi.org/10.1016/j.jcp.2013.08.014.

[51] Hartmann, R., "Adjoint Consistency Analysis of Discontinuous Galerkin Discretizations," *SIAM Journal on Numerical Analysis*, Vol. 45, No. 6, 2007, pp. 2671–2696. https://doi.org/10.1137/060665117.

[52] Nordström, J., and Ghasemi, F., "On the Relation Between Conservation and Dual Consistency for Summation-by-Parts Schemes," *Journal of Computational Physics*, Vol. 344, 2017, pp. 437–439. https://doi.org/10.1016/j.jcp.2017.04.072.

[53] Nemec, M., and Zingg, D. W., "Newton-Krylov Algorithm for Aerodynamic Design Using the Navier-Stokes Equations," *AIAA Journal*, Vol. 40, No. 6, 2002, pp. 1146–1154. https://doi.org/10.2514/2.1764.