



Applying High-Order, Adaptively-Refined, Finite-Volume Methods to Discrete Structured Representations of Arbitrary Geometry

Nathaniel Overton-Katz ^{*}, Xinfeng Gao [†], and Stephen Guzik [‡]
Computational Fluid Dynamics and Propulsion Laboratory
Colorado State University, Fort Collins, CO, USA

In this paper, we present a high-order finite-volume method for solving hyperbolic conservation laws with adaptive mesh refinement on discretely defined structured grids of complex geometries. The objective of this study is to extend conforming mapped multi-block methods to discrete structured representations of arbitrary geometry. Conforming mapped multi-block interfaces are treated by employing ghost cells which are interpolated from surrounding blocks. For each block, b-spline interpolation is performed on the discrete grid to allow high-order representation and adaptive refinement. The use of ghost cells, while allowing application of the interior high-order spacial discretizations everywhere, including block interfaces, introduces logistical challenges. The present study focuses on developing strategies to overcome these challenges efficiently. The resulting algorithm is verified and validated by comparing solutions to those of mapped single-block grids. Finally, the algorithm is applied to the compressible Navier-Stokes equations for solving flow over a bluff-body geometry, which shows promising results. Directions for future work that further optimizes the strategies are discussed.

Notation

$\vec{\mathbf{F}}$	flux dyad, e.g., $[\mathbf{F}_1, \dots, \mathbf{F}_D]$	$\langle \cdot \rangle$	cell-averaged or face-averaged quantity
\mathbf{F}	flux vector, e.g., $[\mathbf{F}_1, \dots, \mathbf{F}_N]$	\vec{x}	physical space, e.g., (x, y, z)
\mathbf{U}	solution vector, e.g., $[\mathbf{U}_1, \dots, \mathbf{U}_N]$	$\vec{\xi}$	computational space, e.g., (ξ, η, ζ)
F_j	the j^{th} component of vector \mathbf{F}	$\vec{\nabla}_x$	$\vec{\nabla}$ in physical space
\mathbf{N}^T	metric transformation matrix	$\vec{\nabla}_\xi$	$\vec{\nabla}$ in computational space
J	grid metric Jacobian	X	forward mapping function
i	grid indices, e.g., (i, j, k) in 3D	Ξ	inverse mapping function

I. Introduction

Finite-volume methods are valuable for solving fluid dynamics problems where strict conservation or shock capturing is important. Structured Cartesian grids are attractive because they have implicit data layouts that allow for high computational efficiency and parallelization. Additionally, they are well suited for adaptive mesh refinement (AMR) due to straightforward regridding. High-order CFD algorithms have been developed to provide improved solution accuracy and computational efficiency over low-order algorithms, especially on structured grids. By combining these elements, Chord [1–5], our in house CFD code, delivers fourth-order accurate finite-volume solutions for compressible Navier-Stokes flows with turbulence and combustion. Chord is built on the Chombo framework [6]. It features AMR in space and time, and has been shown to scale to at least 1×10^5 cores with flat MPI.

One of the challenges of using structured grids is complex geometry representation. Chord applies smooth analytic mapping functions to transform Cartesian structured grids into curvilinear ones. Analytic functions, while descriptive, are not easily created for arbitrary geometries. To use arbitrary discretely-defined grids, such as those from common mesh generation software, a mapping function needs to be generated by interpolation, as has been previously archived

^{*}PhD Student

[†]Associate Professor

[‡]Assistant Professor

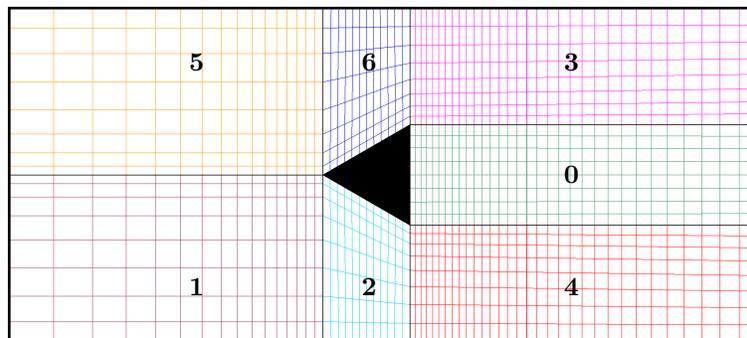
in Chord [7]. This grid mapping allows for many geometries of interest such as simple airfoils, but is inadequate for those with sharp corners or complex shapes. To represent complex geometry on structured grids, the present study investigates mapped multi-block techniques.

The mapped multi-block methodology enables further ability to represent complex geometries on structured grids by separating the global domain into a number of blocks, and applying a distinct mapping function to each block. This provides greater geometry flexibility, since the mapping function of each block needs only be chosen to represent local features. Nevertheless, special care must be given to block intersections in order to maintain solution properties such as conservation. Using the multi-block scheme developed by McCorquodale et al. [8], solutions with strict conservation, high-order accuracy, and free-stream preservation are possible, although restrictions to conforming multi-block grids must be made. Conforming means that blocks are aligned with each other at the connected boundaries. In the chosen mapped multi-block scheme, the challenge of handling block intersections is dealt with by a smooth continuation of each block's mapping function, and interpolation of values as needed. Although this method has previously been shown to work for nicely chosen analytic mapping functions, its application to generalized geometries is unknown. The aim of the present study is to investigate the mapped multi-block method for arbitrary geometries represented by discrete structured grids.

Beginning in section II, background is provided on the scheme used by Chord for working with mapped multi-block grids. The first part of this section details notation used for the scheme, and the second describes the conservation equations on mapped grids. The next section briefly discusses the application of b-spline interpolation for mapping discrete grids. Section IV describes how the mapped multi-block scheme is generalized for arbitrary grids, and addresses the challenges of doing so. Section V demonstrates applications of the generalized mapped multi-block method for discrete grids. Finally, conclusions are drawn and areas of further research are suggested.

II. Mapped Multi-Block Finite-Volume Scheme

A formal definition and background is needed to fully explain the details of the mapped multi-block scheme. A brief review of the high-order mapped finite-volume scheme and its extension to multi-block grids is given, so that the generalization of the method can be conveniently presented.



(a) A sample mapped multi-block grid in physical space.



(b) A sample mapped multi-block grid in computational space.

Fig. 1 A sample conforming mapped multi-block grid. Seven blocks are shown in this example, each with a mapping function from its Cartesian computational grid to the curvilinear one in physical space.

A. Mapping and Multi-Block Notation

The mapped multi-block framework operates in a Cartesian space which we denote as computational space, indexed by coordinates $\vec{\xi}$. This can be transformed into physical space, denoted by coordinates \vec{x} . The transformation between the two spaces is defined by the forward mapping $\vec{x} = X(\vec{\xi})$ and the inverse mapping $\vec{\xi} = \Xi(\vec{x})$, as illustrated in Fig. 2. The mapping function and its inverse must both be smooth and one-to-one. The forward mapping function $X(\vec{\xi})$

is always known and readily evaluated. On the other hand, the inverse mapping is generally not specified directly. Throughout this paper *the mapping function* is frequently referred to, and unless explicitly stated otherwise, it indicates the forward mapping $X(\vec{\xi})$.

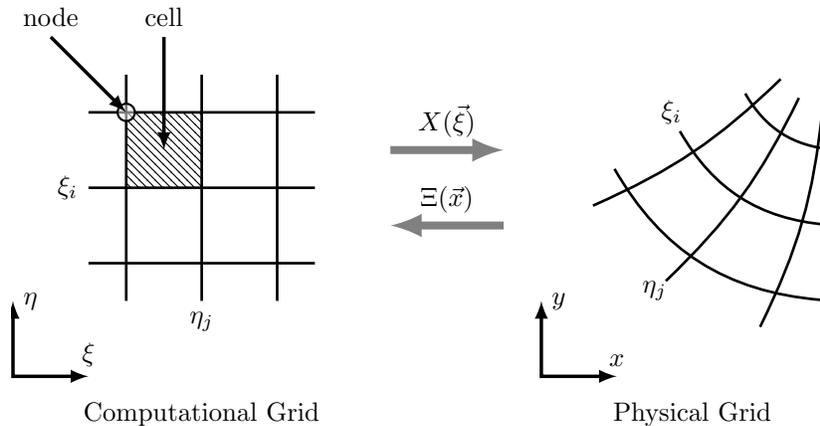


Fig. 2 A representation of the transformation between computational and physical space.

In the multi-block framework, the global domain is subdivided into a number of regions which we call blocks. For each block we define the block-domain in computational space as $\mathcal{D} \subset \mathbb{R}^D$, and the block-range in physical space as $\mathcal{R} \subset \mathbb{R}^D$, as illustrated in Fig. 3. The block-domain, which indicates a particular region in computational space, is not to be confused with global computational and physical domains, which identify the entire multi-block grid in the appropriate space. Each block has an individual mapping function, block-domain, and block-range. Restriction to conforming multi-block assumes the block-ranges may only intersect at the boundaries without overlapping. When two blocks are connected, i.e. their block-ranges intersect at a boundary, an additional restriction is enforced. That is, for connected blocks A and B it must be that $X_A^{-1}(\mathcal{R}_A \cup \mathcal{R}_B)$ and $X_B^{-1}(\mathcal{R}_A \cup \mathcal{R}_B)$, which are subsets of boundaries in \mathcal{D}_A and \mathcal{D}_B respectively, are isometries of one another. In effect, this enforces that every cell has a single neighbor per face, and thus a single flux is computed which maintains conservation.

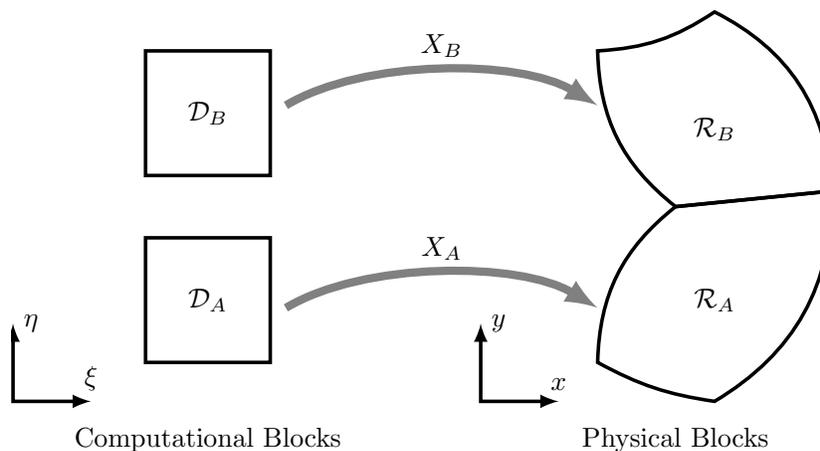


Fig. 3 A sample multi-block system with two blocks.

B. Conservation Laws

The starting point for this work is the high-order finite-volume method on single mapped grids. This is summarized for convenience, and explained in further detail by Guzik et al.[2, 9]. The finite-volume method for Cartesian grids is

formulated using the integral form of the conservation laws with control volumes V_i in physical space, \vec{x} ,

$$\frac{\partial}{\partial t} \int_{\vec{x}(V_i)} \mathbf{U} d\vec{x} + \int_{\vec{x}(V_i)} \vec{\nabla}_x \cdot \vec{\mathbf{F}} d\vec{x} = 0.$$

Transforming this into computation space, $\vec{\xi}$, is done by using grid metric terms such that

$$\frac{\partial}{\partial t} \int_{V_i} J\mathbf{U} d\vec{\xi} + \int_{V_i} \vec{\nabla}_\xi \cdot (N^T \vec{\mathbf{F}}) d\vec{\xi} = 0,$$

where the transformation matrix, $N^T = J\vec{\nabla}_x \vec{\xi}$ describes the grid metrics, and the metric Jacobian $J \equiv \det(\vec{\nabla}_\xi \vec{x})$. After applying the divergence theorem of Gauss, the integrals can be represented as cell averaged values, yielding

$$\frac{d}{dt} \langle J\mathbf{U} \rangle_i + \frac{1}{h} \sum_{d=1}^D \left(\langle N_d^T \vec{\mathbf{F}} \rangle_{i+\frac{1}{2}e^d} - \langle N_d^T \vec{\mathbf{F}} \rangle_{i-\frac{1}{2}e^d} \right) = 0, \quad (1)$$

where the subscript d denotes the d^{th} row of N^T . Note that this is an exact formulation, and the order of the solution accuracy depends upon the methods used to evaluate fluxes and time derivatives. The algorithm for arriving at a high-order, free-stream preserving $\langle N_d^T \vec{\mathbf{F}} \rangle$ from $N_d^T \vec{\mathbf{F}}$ is a rather long process, detailed in work by Guzik et al.[2], which provides the framework for the present study. Solving these quantities requires the mapping $\vec{x} = X(\vec{\xi})$ and its derivatives on codimension two elements of the grid (e.g., vertices in 2-D and edges in 3-D). Additionally, this framework is compatible with adaptive mesh refinement, due to the fact that the mapping function may be evaluated anywhere inside the domain as needed.

For mapped single-block grids, the fluxes are computed using regular stencils, such as a fourth-order centered stencil, which can be applied uniformly to the interior of the domain, since structured grids have uniform data layout. For boundaries, stencils must either be made one-sided or use ghost cells outside the domain to apply centered stencils. For mapped multi-block grids, special treatment of stencils next to block interfaces must be also made. Using one-sided stencils on either side of a block interface appears a straightforward solution. However, this method is likely to yield different constructions on either side of a interface, which may incur inappropriate upwinding and impact solution stability. The approach taken in this paper is to apply centered stencils everywhere by employing ghost cells, and is discussed in section IV .

III. Generalized High-Order Grid Mappings Using B-Spline Interpolation

The preceding algorithm for mapped single-block grids provides a framework for solutions on any geometry that has an analytic formulation of the mapping function and its derivatives. Furthermore, the mapping function is assumed to be smooth, one-to-one, and have an inverse. A number of analytic mappings which satisfy these constrains can be developed, but doing so is often time consuming and geometry specific. For arbitrary geometries, it can be extremely difficult, if feasible at all, to produce analytic mappings.

For general geometries, it is desirable to define mapping functions from discrete points, rather than using analytic functions. In particular, many applications are often provided with grids that are generated from meshing software. Therefore, given a discrete grid it is preferred to introduce an interpolation scheme which creates a mapping function representative of the grid. Using this approach, Chord is no longer restricted to mapped grids defined by analytic functions, while retaining the ability to apply AMR. When constructing a mapping function for a specified numerical scheme, the mapping function needs only appear smooth relative to the reconstruction order used by the scheme. Practically, this allows for the mapping function to be constructed using piecewise polynomials, so long as the construction is globally one order of accuracy higher than the underlying finite-volume scheme. Multi-dimensional b-spline interpolation achieves this, and we have demonstrated that it preserves the desired solution accuracy [7]. In essence, b-spline interpolation can be treated as multi-dimensional piecewise polynomials with order dependent continuity at connections, and are chosen because they have provable global order of accuracy [10]. For conciseness, the details of b-spline interpolation are not presented here, but are described by Overton et al. [7].

IV. High-Order Multi-Block Stencils

The major challenge of the mapped multi-block method is determining how to compute the fluxes on multi-block faces. For high-order methods, additional challenge is introduced due to the large stencils used. For example, fluxes at

interior cell-faces near a block boundary may have a stencil that extends beyond the block-domain. McCorquodale et al. [8] reduces the problem of solving stencils over multi-block boundaries to one performing high-order interpolations. Employing ghost cells, artificial cells which are extended outside the block-domains, allows the interior stencils to be applied everywhere. For mapped multi-block, ghost cells for each block are extended over block interfaces. Their values are then interpolated from the existing valid region. We further define ghost cells as either interior or exterior. Interior ghost cells are those whose cell center lies inside the global physical domain, while exterior ghost cells have a cell center that falls outside the global physical domain. For emphasis, non-ghost cells are referred to as valid cells, i.e. those cells being part of the global domain. This concept can be clearly seen in Fig. 4, where each block is shown in different colors and given black boundaries. The solid lines of each block show the valid cells, and dashed lines are for ghost cells. The enhanced blue ghost cell in the top left is an interior ghost cell, while the enhanced red ghost cell in the bottom right is an exterior ghost cell.

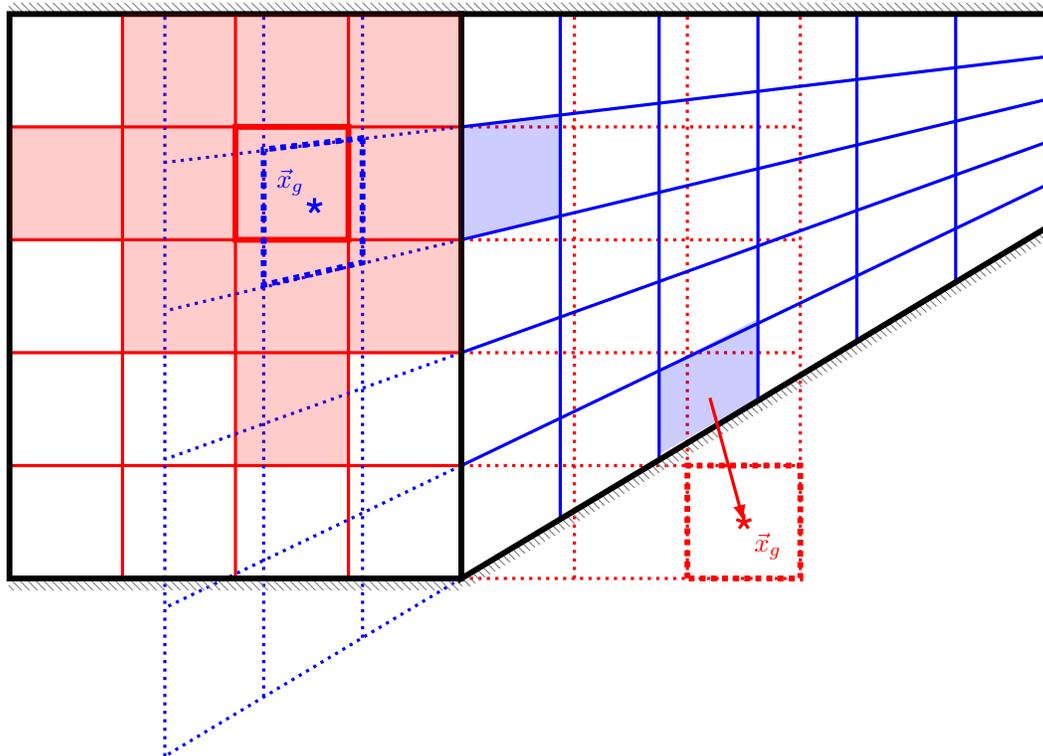


Fig. 4 A mapped multi-block grid with two blocks is shown in physical space. An example of both interior and exterior ghost cells are shown, with the regions used for filling each of them.

The approach for filling interior ghost cells in Chord is accomplished by interpolating from a least squares formulation of fourth-order multi-dimensional polynomials. The interpolation is centered about the physical location of the ghost cell center \vec{x}_g , as seen from the enhanced blue cell in Fig. 4. Given the point \vec{x}_g , a region of valid neighboring cells is established that is sufficiently sized to produce a fourth-order polynomial, as is illustrated by the highlighted cells surrounding the blue \vec{x}_g . In general, this neighborhood of cells may span several blocks, and be irregularly shaped. The neighborhood must contain at minimum $\frac{(P-1+D)!}{D!(P-1)!}$ components, where P is the desired order and D is the number of space dimensions. In Chord, where fourth-order reconstruction is used in either two or three dimensions, a neighborhood must contain at least 10 or 20 cells, respectively. The least squares method generates a specified order interpolation, while providing the flexibility of any arbitrarily shaped neighborhood, and any number of cells over the minimum required. The generation of the least squares stencil can be separated from the interpolation, allowing efficient interpolation at any future time.

Furthermore, we note that the mapping function must be of sufficient quality to interpolate ghost cells. For cases with exterior ghost cells, such as the red ghost cell seen in Fig. 4, interpolation is not possible so an extrapolation method must be devised instead. Finally, logistical concerns about how to identify where ghost cells should interpolate

values from arise when dealing with arbitrary grids. These logistical challenges must be addressed for the algorithm to operate with an acceptable computational cost.

A. Mapping Restrictions Imposed by Ghost Cell

Ghost cells are created by extending each block-domain by the required number of cells that enables interior stencils to be applied for calculations of block boundary fluxes. The mapping function is applied to these extended block-domains, and assumed to still hold its prescribed properties, which are one-to-one, smooth, and invertible. However, this assumption can be problematic for a number of mappings, when extending the original block-domains results in regions where grids degenerate. This is particularly problematic for grids with a high amount of stretching, as illustrated in Fig. 5, since continuation of the mapping may result in points where grid lines cross and the mapping is no longer one-to-one. Although unfortunate, it is left to the user to ensure that grids do not have degenerate ghost cells. Guidance for producing adequate grids is provided by methods which detect problematic regions. The user can often adjust the amount of stretching near block boundaries to improve the ghost cell quality. Alternatively, refining the grid near boundaries also improves the ghost cell quality, As cell size shrinks, the distance of ghost cells beyond the block-domains also reduces.

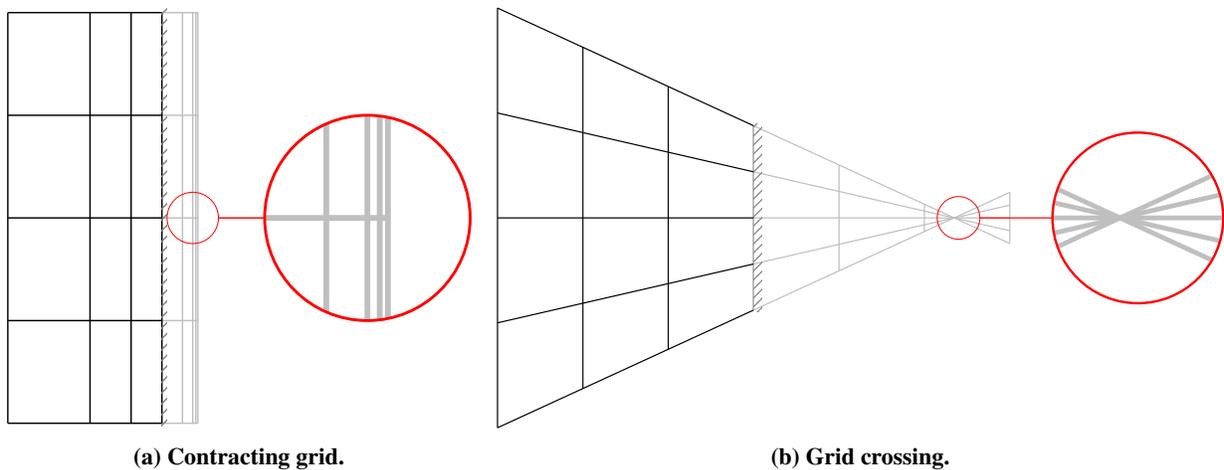


Fig. 5 Visualization of examples where ghost cells degenerate.

For discrete grids, the mapping function is represented using b-splines defined within each block-domain. This introduces another challenge. Extrapolation, as required of the mapping function to create ghost cells, is known to be error prone for polynomials. Fortunately, the extrapolation region is always known to be small, and so long as the mapping remains one-to-one the error is inconsequential. However, in cases where the mapping degenerates due to extrapolation errors from the b-splines, it is again left to the user to alter the grid.

B. Exterior Ghost Cell Extrapolation

For general mapped multi-block grids, there are cases where ghost cells extend outside the global physical domain and interpolation cannot exist. The framework developed thus far requires an extrapolation to be made which yields ghost cells of reasonable values. These values are then used to reconstruct fluxes at the solution interior.

The first task for dealing with exterior ghost cells is detection. The process of determining whether a ghost cell is exterior is not immediately obvious. Multi-block boundaries are potentially complex, and computational geometry algorithms with high-order b-splines are expensive. As part of the multi-block setup, the physical cell center of each ghost cell must solve for its inverse mapping to find a corresponding valid cell. Since the mapping and inverse must be one-to-one, exterior ghost cells can be detected based on the lack of an inverse inside the global domain.

Once detected, external ghost cell extrapolation is performed. Ideally, the extrapolation should be smoothly extended from the interior. The simplest solution is to perform a constant extrapolation along the grid lines to the exterior ghost cells. This is not an ideal method, but is sufficient to produce reasonable results at the cost of accuracy at boundaries. The extrapolation can be improved by taking values from the physically nearest cells, which intuitively is more reasonable. This process is illustrated in Fig. 4, where the enhanced red ghost cell in the bottom right is an

exterior ghost cell. The exterior ghost cell is extrapolated as a constant from indicated nearest valid cell. This approach has been observed to yield reasonable results, as shown in this study.

C. Efficient Mapping Inverse

One computationally expensive piece of this algorithm is finding the nearest valid cell to a given location in physical space \vec{x}_g . This is required for creating the least squares regions to interpolate each ghost cell quantity from, as shown in Fig. 4. With a high-order scheme that uses many ghost cells, the number of total ghost cells can frequently rival or even exceed the number of valid cells in the global domain. Although this search for the nearest valid cell must happen for every ghost cell, it needs only happen once per grid. However, if implemented poorly, this one-time setup can still be prohibitively expensive.

This challenge is approached by formulating a root finding problem, where $\vec{\xi}$ is desired to satisfy $f(\vec{\xi}) = 0$. For a given point \vec{x}_g , the distance function to satisfy is

$$f(\vec{\xi}) = \|X(\vec{\xi}) - \vec{x}_g\|.$$

Since solving the function $X^{-1}(\vec{x}_g)$ directly is not feasible, this problem must be solved iteratively. Newton's method may be used to solve the root finding problem efficiently, since the mapping function is represented by b-splines which have readily computed derivatives. However, Newton's method requires a sufficient initial guess to converge. Identifying a sufficient initial guess is challenging in general, but it is found empirically to be on the scale of a cell size.

To reach a region where open root solvers may be used, of which Newton's method is one of, a global root solver must first be used to arrive at a coarse approximation. However, global root solvers are expensive to evaluate. In this case, it is especially problematic to use a global root solver, because the global domain may be quite large, and a root solve must be performed for every ghost cell. An optimized approach for the global root solver is developed specifically for this scenario.

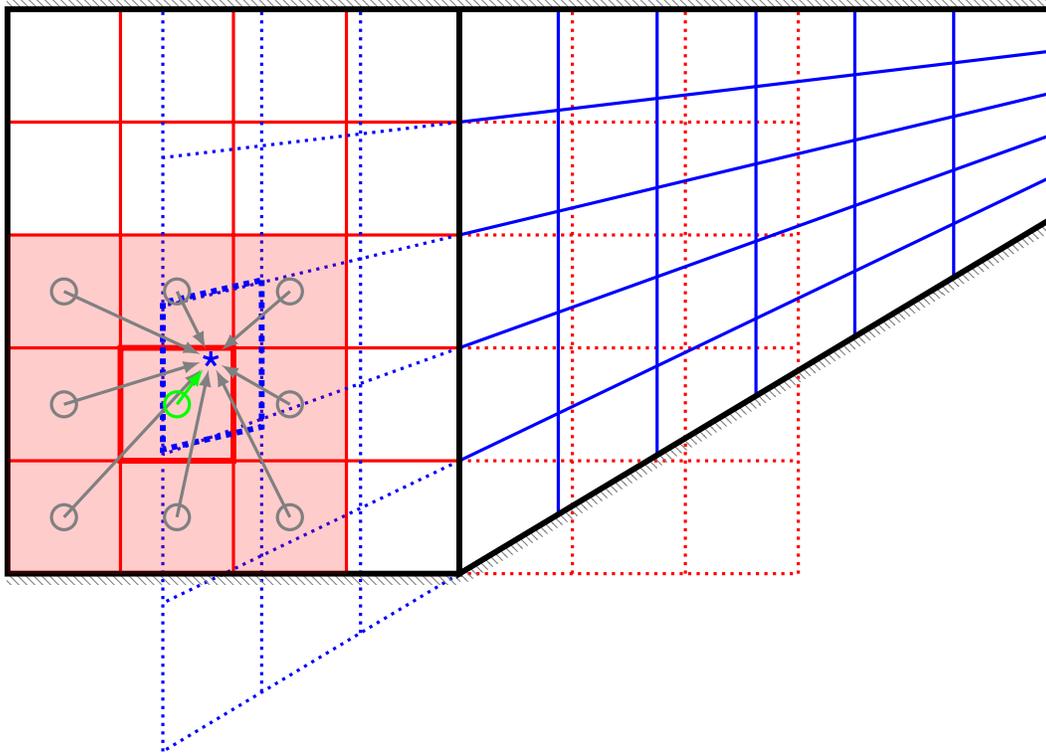


Fig. 6 Illustration of the mapping inverse criteria.

In this scenario, the global root solve can be improved by using knowledge of the distance function $f(\vec{\xi})$. The fact that $X(\vec{\xi})$ is one-to-one and may possess only a single root, allows for a method of certifying whether a given region is

eligible to contain the root. Knowledge that all points \vec{x}_g come from ghost cells with a number of steps off of a block boundary provides an expectation of the root location, and an area to begin searching for where a root may exist.

A region can be determined to contain a root by creating a bound for the region and testing a single point inside the region. For a region Ω with boundary $\partial\Omega$ that contains the point $\vec{\xi}_i$, a root is contained in the region when $f(\vec{\xi}_i) < \min\{f(\vec{\xi}_b) \forall \vec{\xi}_b \in \partial\Omega\}$. In application, $\vec{\xi}_i$ is chosen as a cell index $\vec{\xi}_i$ and the bound $\partial\Omega$ is chosen as the discrete collection of directly neighboring cells $\{\vec{\xi}_{i \pm e^d} \forall d \in 0 \dots D\}$. Once a bound has been formed that contains a root, Newton's method can be used to more efficiently converge to the inverse with desired precision. This process is illustrated in Fig. 6 where an interior ghost cell center is given by the blue point. The enhanced red cell is tested to determine if eligible to contain an inverse. Since the green line, which has the magnitude of the distance function f , is the shortest of any of its surrounding neighbors, the highlighted red region must contain an inverse for the blue point.

The region to search can be narrowed with the knowledge that the point \vec{x}_g is always offset from a multi-block interface. Using the assumption that the inverse follows grid lines often gives a reasonable initial point to begin searching. Furthermore, the search direction can be informed of the direction where the distance function is decreasing. Computational expense for the method can be quantified by letting M be the number of ghost cells, and N the number of valid cells. In the best case when each multi-block is a Cartesian grid in physical space, the inverse is found immediately. To find the inverse for all ghost cells in the domain, this best case scenario has computational expense proportional to $O(M)$. In more challenging cases where the inverse can not be immediately guessed, a search region is expanded and the root detection method applied. In the worst possible case, this method performs no better than naively searching the entire global domain for a root. Finding the inverse of all ghost cells has a worst case computational expense proportional to $O(M \cdot N)$, which is quite poor but provides an upper bound. However, this worst case still guarantees solutions. Additionally, the worst case is rarely encountered, since it requires a truly pathological grid.

V. Results

To demonstrate the validity and application of the resulting algorithm, a verification case is first examined to ensure the mapped multi-block scheme operates as intended. Next a comparison of a more complex problem is made between a single analytically mapped grid, and a discretely defined multi-block one. Finally, a case with an engineering geometry is examined, and demonstrates the abilities of the mapped multi-block algorithm.

A. Periodic Advection Cube

As a verification case, a Gaussian density profile is created and advected across the domain. This case verifies that the mapped multi-block algorithm maintains conservation, and achieves the desired order of accuracy over multi-block boundaries. This case on single mapped grids, both analytically and discretely defined, has previously been verified in Chord [2, 7]. Results of the multi-block method is compared to those previously obtained.

Using the Euler equations on a periodic domain, a constant uniform flow velocity is specified. The density profile is initialized as

$$\rho = \rho_0 + s(r) \Delta\rho e^{-(100r^2)},$$

where

$$s(r) = \begin{cases} 0 & : |2r^2| \geq 1 \\ \cos^6(\pi r^2) & : |2r^2| < 1 \end{cases},$$

$\rho_0 = 1.4$, and $\Delta\rho = 0.14$. The value r specifies the distance from the center of the periodic domain, $[0, 1]^D$. Pressure is initialized as a constant of 1 and the velocity is set to $(1.0, 0.5)$. With this choice of velocity, the density profile will return to the initial location after 2 time periods, and solution error is calculated using the exact solution. For verification, a Cartesian analytically mapped single-block and multi-block grid are generated to be identical in physical space. The multi-block grid is shown in Fig. 7, and although not shown, the single-block grid is indistinguishable but without block distinctions. For both of these methods, a convergence study is performed. Both cases exhibit the expected fourth-order convergence, as shown in Fig. 8. The solution errors between the single and multi-block method are seen to be similar, with the multi-block errors only marginally higher. Another comparison can be made between grids using discretely defined geometries, and analytic ones. For this simple geometry, results from discrete grids are identical to their analytic counterparts within machine precision.

Additionally, it is possible to test conservation of the periodic advection case. A finite-volume scheme with no source or sink terms, should experience no change in its net conservative solution quantities. This is tested by comparing

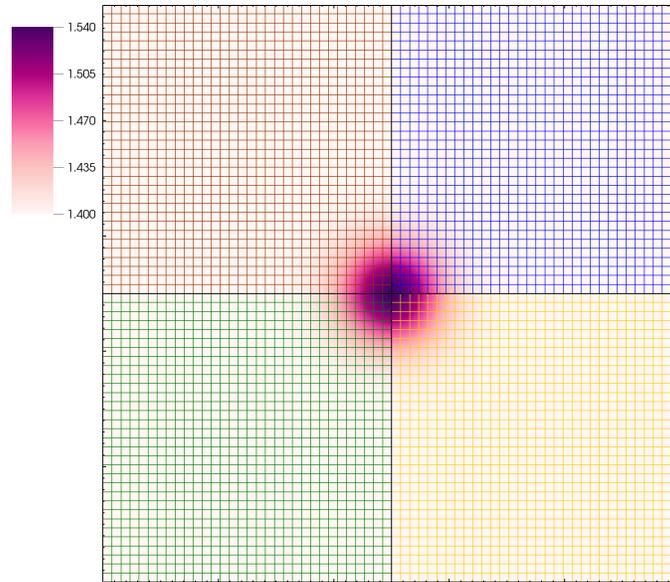


Fig. 7 The density profile of the periodic advection case is shown on the Cartesian multi-block grid. The grid of each block is shown in a different color.

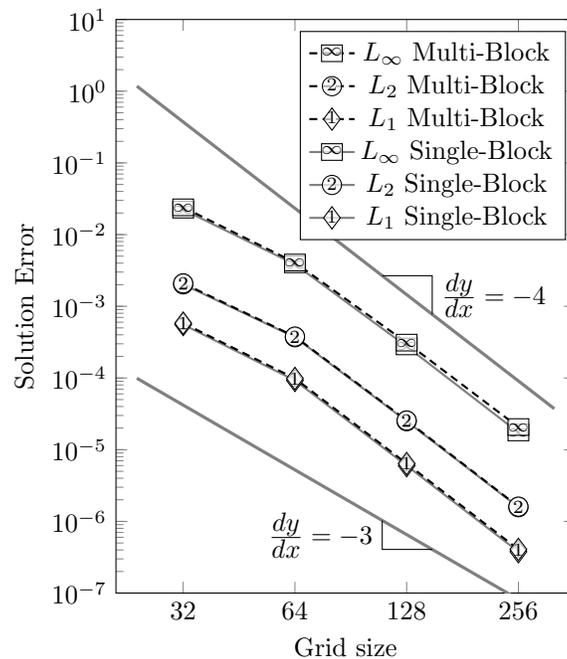


Fig. 8 Convergence rates for the periodic advection cube, using both a single mapped grid and a multi-block one. Solution errors from both methods lie very near one another, and both show fourth-order convergence.

integration of the solution variables \mathbf{U} in physical space at the initial and final solution times. Results from a global grid of size 256×256 are tabulated for the single-block method in Table 1, and multi-block method in Table 2. It is observed that in both cases, solutions are conserved to near machine precision. Importantly for this study, the multi-block scheme is shown to maintain conservation equally as well as the single-block scheme.

Table 1 Initial and final conservative values for the advection case, using the single-block grid

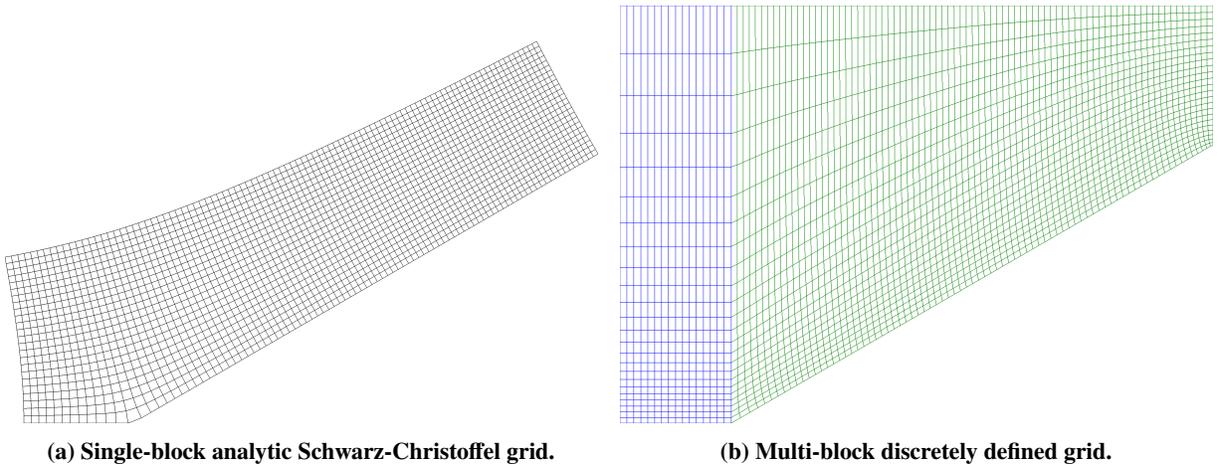
U	Initial	Final	Difference
ρ	9.197144478240506e+04	9.197144478240158e+04	0.000000000000348e+04
ρu	9.197144478240506e+04	9.197144478240144e+04	0.000000000000362e+04
ρv	4.598572239120253e+04	4.598572239120067e+04	0.000000000000186e+04
ρE	2.213221529890319e+05	2.213221529890331e+05	0.000000000000012e+04

Table 2 Initial and final conservative values for the advection case, using the multi-block grid

U	Initial	Final	Difference
ρ	9.197144478240506e+04	9.197144478240123e+04	0.000000000000383+04
ρu	9.197144478240506e+04	9.197144478240107e+04	0.000000000000405e+04
ρv	4.598572239120253e+04	4.598572239120053e+04	0.000000000000200e+04
ρE	2.213221529890319e+05	2.213221529890325e+05	0.000000000000006e+05

B. Mach Reflection

Previously, the Woodward-Colella mach reflection case has been studied in detail using Chord [2]. There are a number of interesting physical phenomenon in this flow, such as shocks and induced flow instabilities. In this study, the case is of interest because it is one of the most geometrically complex cases using a single mapped grid. The single-block solution uses a Schwarz-Christoffel mapping to generate the ramp. Although the geometry of the bottom boundary with a sharp feature is possible, it comes at loss of geometry specification in the remaining domain. Due to the geometry, applying boundary conditions requires special treatment, and the grid can no longer be aligned with the shock. However, the multi-block methodology can create a grid with full geometric flexibility, allowing for simpler boundaries and grid alignment to flow features. A comparison study is made between the previously validated single mapped grid solution, and the new multi-block one.

**Fig. 9 Coarsened representations of the grid used to capture the mach reflection geometry.**

A ramp of angle of 30° is created, and an incident shock of Mach 10 reflects off the ramp. The front of the shock has specified flow conditions $p = 1$, $\rho = \gamma$, and $\vec{u} = 0$. Conditions behind the shock are set from the shock relations. The bottom boundary is treated as a slip wall, while all others have specified Dirichlet conditions. The top boundary is specially adjusted based on the analytic position of the shock.

This case is of particular interest when using the multi-block scheme, since it introduces a scenario that exhibits exterior ghost cells. Additionally, this case is produced using a discrete grid. Both solutions clearly capture the same physics, with minor differences, as shown in Fig. 10. This is particularly encouraging near the corner of the ramp,

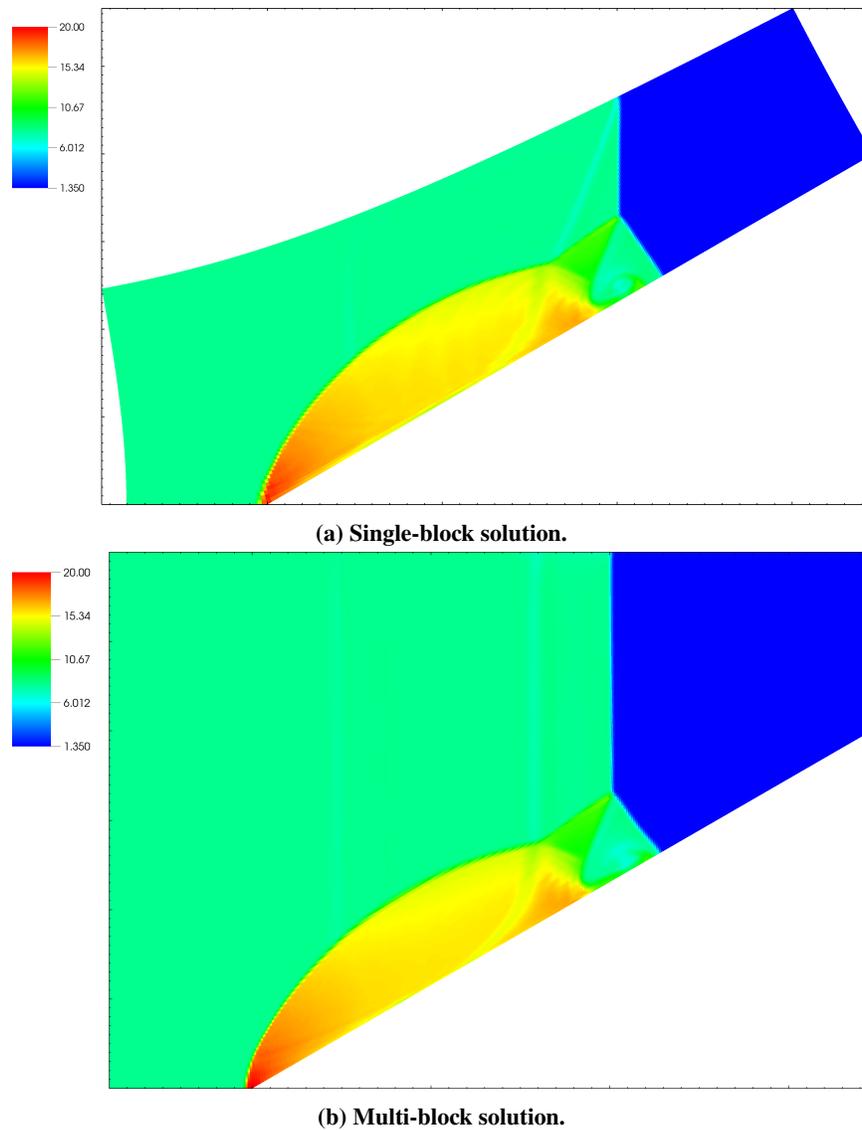


Fig. 10 The mach reflection case, comparing mapped single-block and multi-block solutions.

where the exterior ghost cell extrapolation method is active. The multi-block case, with the grid represented in Fig. 9, appears better able to capture the off body vertical shock along the top boundary, since it is grid aligned. However the grid is no longer entirely aligned with the shock orthogonal to the wall, and a minor distortion is observed. From the results of this comparison, we are able to validate the mapped multi-block methodology on arbitrary discrete grids.

C. Flow Over a Bluff Body

Finally, a case is demonstrated which showcases the new mapped multi-block capability enabled for Chord. The bluff body combustor case is of particular engineering interest, and has been featured in model validation and for propulsion workshops [11]. Some common interests with this case include turbulence modeling, and reacting flows in the presence of a realistic geometry. Although turbulence and combustion modeling are available in Chord, this preliminary study is mealy focused on slow cold flow to examine the mapped multi-block algorithm.

The flow entering the channel is specified with uniform horizontal velocity of $u = 15.7 \text{ m s}^{-1}$, and density $\rho = 1.24 \text{ kg m}^{-3}$. The outlet pressure is specified as $p = 100\,000 \text{ Pa}$. The boundary conditions and grid used for this case are shown in Fig. 11. The vortex structure as shown in Fig. 12 is reasonable. It is worth mentioning that the bluff

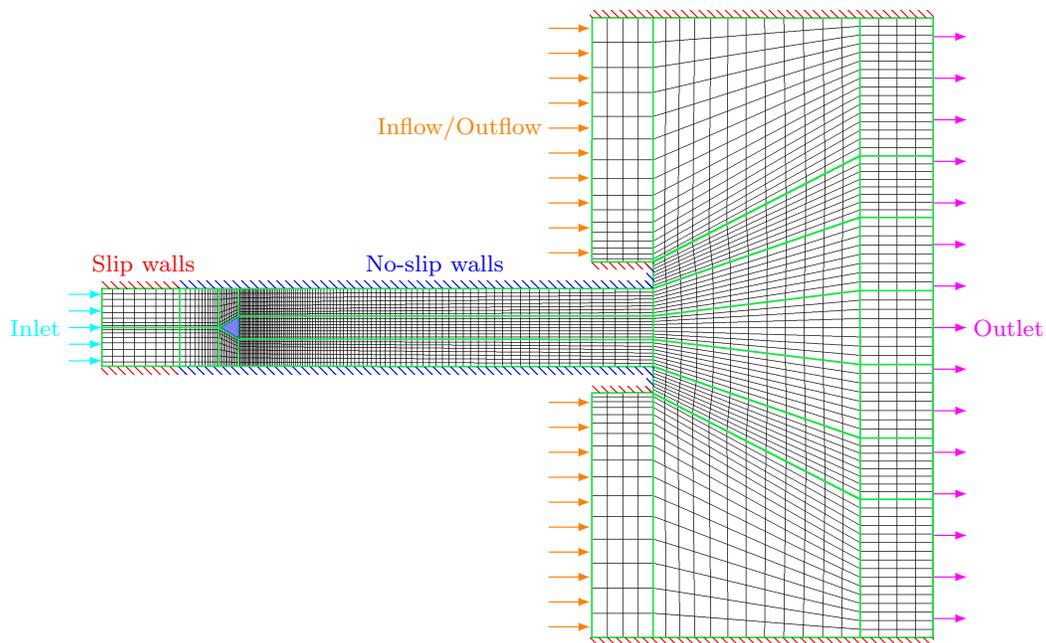


Fig. 11 Coarsened grid for the bluff body case. Boundary conditions are labeled and color coded. Block boundaries are highlighted in green.

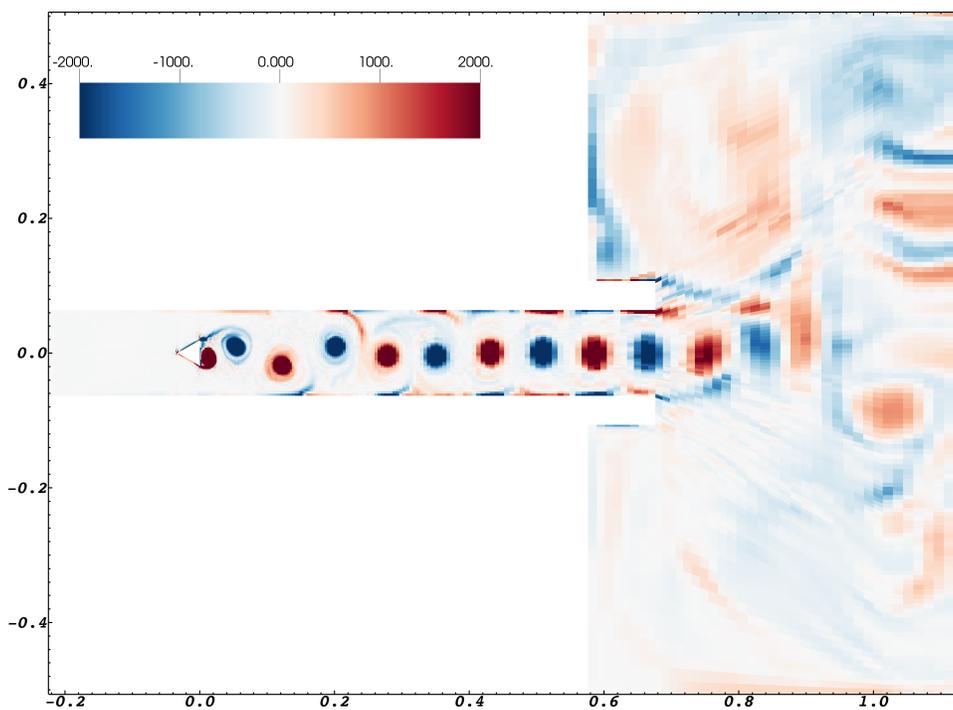


Fig. 12 Vorticity profile of the bluff body case.

body featured in the front of the domain has a number of exterior ghost cells generated around it, and no erroneous solution features appear in that region. Additionally, the solution for this case has many complex flow features that regularly cross block boundaries. Despite this, there appears to be no distortion due to multi-block interfaces.

VI. Concluding Remarks and Future Work

We have presented an approach that generalizes the high-order, adaptively-refined, finite-volume method using the mapped multi-block scheme on discretely defined arbitrary geometries. Mapping functions are recreated from discrete grids while maintaining high-order by using b-splines. Further, the multi-block method is used to provide flexibility in grids. A number of strategies are proposed, implemented, and tested to overcome the challenges to allow the high-order multi-block methods to handle arbitrary discrete geometries. There are some limitations to this approach concerning ghost cells, which need further investigation. Nevertheless, the algorithm has been verified and validated, and results for an initially promising application of these strategies demonstrated.

Future work will further address the challenges with ghost cell extrapolation over multi-block boundaries so the method can become less restrictive. This can possibly be done by creating mapping functions that explicitly account for ghost cells. The mapping function can then be altered to prevent the ghost degeneration, and thus operate with fewer restrictions. Other changes in the algorithm to minimize the number of required ghost cells will also be explored.

Another direction for future investigation is improving the method for ghost cell filling. Rather than an interpolation to fill each ghost cell when interior, and extrapolation for exterior, a reconstruction function could be performed at the block interface of each cell. Ghost cells could be filled to satisfy the reconstructed function at each cell-face to the desired order of accuracy. This removes the need for the mapping inverse and separate treatment of interior and exterior ghost cells.

VII. Acknowledgments

This research was supported by the Department of Defense United States Air Force (DOD-USAF-Air Force) under the award number FA9550-18-1-0057.

References

- [1] Gao, X., Guzik, S. M. J., and Colella, P., "A Fourth-Order Boundary Treatment for Viscous Fluxes on Cartesian Grid Finite-Volume Methods," AIAA 2014-1277, 52nd AIAA Aerospace Sciences Meeting, 2014.
- [2] Guzik, S. M., Gao, X., Owen, L. D., McCorquodale, P., and Colella, P., "A Freestream-Preserving Fourth-Order Finite-Volume Method in Mapped Coordinates with Adaptive-Mesh Refinement," *Comput. Fluids*, Vol. 123, 2015, pp. 202–217.
- [3] Guzik, S. M., Gao, X., and Olschanowsky, C., "A high-performance finite-volume algorithm for solving partial differential equations governing compressible viscous flows on structured grids," *Comput. Math Appl.*, Vol. 72, 2016, pp. 2098–2118.
- [4] Gao, X., Owen, L. D., and Guzik, S. M. J., "A Parallel Adaptive Numerical Method with Generalized Curvilinear Coordinate Transformation for Compressible Navier-Stokes Equations," *Int. J. Numer. Meth. Fluids*, Vol. 82, 2016, pp. 664–688.
- [5] Owen, L. D., Guzik, S. M., and Gao, X., "A high-order adaptive algorithm for multispecies gaseous flows on mapped domains," *Comput. Fluids*, Vol. 170, 2018, pp. 249–260.
- [6] Adams, M., Colella, P., Graves, D. T., Johnson, J. N., Johansen, H. S., Keen, N. D., Ligoeki, T. J., Martin, D. F., McCorquodale, P. W., Modiano, D., Schwartz, P. O., Sternberg, T. D., and Van Straalen, B., "Chombo Software Package for AMR Applications—Design Document," Tech. Rep. LBNL-6616E, Lawrence Berkeley National Laboratory, 2015.
- [7] Overton, N., Gao, X., and Guzik, S., "Spline Interpolation for A Fourth-Order Adaptive Mesh Refinement Algorithm on Arbitrary Mapped Grids," , No. 3107, 2017. <https://doi.org/10.2514/6.2017-3107>.
- [8] McCorquodale, P., Dorr, M. R., Hittinger, J. A. F., and Colella, P., "High-order finite-volume methods for hyperbolic conservation laws on mapped multiblock grids," *Journal of Computational Physics*, , No. 288, 2015, pp. 181–195. doi: 10.1016/j.jcp.2015.01.006.
- [9] Guzik, S. M., McCorquodale, P., and Colella, P., "A Freestream-Preserving High-Order Finite-Volume Method for Mapped Grids with Adaptive-Mesh Refinement," *50th AIAA Aerospace Sciences Meeting*, AIAA, 2012. <https://doi.org/10.2514/6.2012-574>.
- [10] De Boor, C., *A practical guide to splines; rev. ed.*, Applied mathematical sciences, Springer, Berlin, 2001.
- [11] Comer, A., Ihme, M., Li, C., Menon, S., Oefeline, J., Rankin, B., Sankaran, V., and Sankaran, V., "Model Validation for Propulsion Workshop," Accessed online, January 2018. <https://community.apan.org/wg/afrlcg/mvpps/p/proceedings>.