High-order finite-volume methods for hyperbolic conservation laws on mapped multiblock grids

P. McCorquodale[1], M.R. Dorr[2], J.A.F. Hittinger[2], and P. Colella[1]

[1]Computational Research Division, Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Mail Stop 50A1148, Berkeley, CA 94720 USA
[2]Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, P.O. Box 808, L-561, Livermore, CA 94551-0808 USA

June 8, 2015

Disclaimer: This document was prepared as an account of work sponsored by the United States Government. While this document is believed to contain correct information, neither the United States Government nor any agency thereof, nor the Regents of the University of California, nor any of their employees, makes any warranty, express or implied, or assumes any legal responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by its trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof, or the Regents of the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof or the Regents of the University of California.

Copyright notice: This manuscript has been authored by an author at Lawrence Berkeley National Laboratory under Contract No. DE-AC02-05CH11231 with the U.S. Department of Energy. The U.S. Government retains, and the publisher, by accepting the article for publication, acknowledges, that the U.S. Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this manuscript, or allow others to do so, for U.S. Government purposes.

High-order finite-volume methods for hyperbolic conservation laws on mapped multiblock grids

P. McCorquodale[a,*], M.R. Dorr[b], J.A.F. Hittinger[b], P. Colella[a]

[a] *Computational Research Division, Lawrence Berkeley National Laboratory, 1 Cyclotron Road, Mail Stop 50A1148, Berkeley, CA 94720 USA*
[b] *Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, P.O. Box 808, L-561, Livermore, CA 94551-0808 USA*

Abstract

We present an approach to solving hyperbolic conservation laws by finite-volume methods on mapped multiblock grids, extending the approach of Colella, Dorr, Hittinger, and Martin (2011) for grids with a single mapping. We consider mapped multiblock domains for mappings that are conforming at inter-block boundaries. By using a smooth continuation of the mapping into ghost cells surrounding a block, we reduce the inter-block communication problem to finding an accurate, robust interpolation into these ghost cells from neighboring blocks. We demonstrate fourth-order accuracy for the advection equation for multiblock coordinate systems in two and three dimensions.

Keywords: finite-volume method, high-order discretization, mapped grids, multiblock, hyperbolic partial differential equations

1. Introduction

The solution of partial differential equations using structured-grid-based discretizations can be challenging when the solution domain has significant geometric structure or is more easily expressed in non-Cartesian coordinates. For instance, in the simulation of the plasma near the edge of a tokamak fusion reactor, coordinates defined by the magnetic field are advantageous.

*Corresponding author. Tel. +1-510-495-2458; fax +1-510-495-2505.
Email addresses: PWMcCorquodale@lbl.gov (P. McCorquodale), dorr1@llnl.gov (M.R. Dorr), hittinger1@llnl.gov (J.A.F. Hittinger), PColella@lbl.gov (P. Colella)

As shown in Figure 1(a), the *single-null* topology of the magnetic field in the edge region [42, 33] (shown in a poloidal cross-section) possesses both open and closed field lines separated by a *separatrix* – a flux surface that is self-intersecting. There is no simple mapping of a single rectangular domain to this edge geometry.

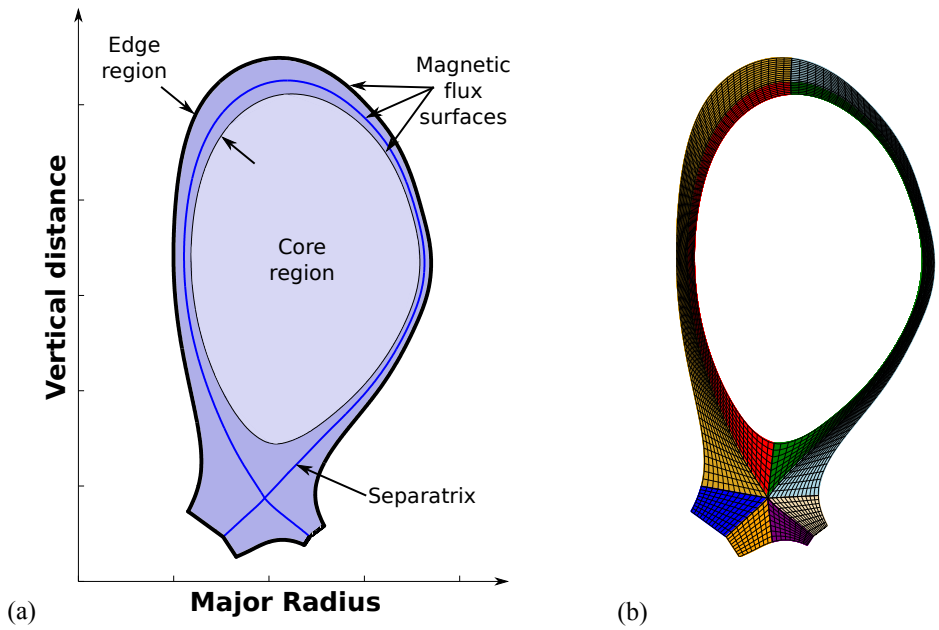

Figure 1: Poloidal cross section of the single-null magnetic field geometry in a tokamak fusion reactor showing (a) the edge and core regions and the separatrix that separates open and closed magnetic field lines and (b) the decomposition of the single-null domain into eight blocks.

Other examples of solution domains that are more easily expressed in non-Cartesian coordinates include the interior of a star or planet and the atmosphere, which is effectively a thin shell over a spherical surface. Although spherical coordinates can be used for both of these cases, they pose difficulties because of the singularities at the center and at the poles.

While mapped-grid approaches based on a single, rectangular Cartesian mesh have the advantage of simplicity and regular access patterns due to the mesh structure, these approaches are extremely limited in the types of domains they can represent well. In contrast, fully unstructured approaches can more easily represent complex geometry, but these require additional storage of mesh associativity data. A popular alternative is to use multiblock

meshes (also known as composite patches or zonal grids), where the domain is decomposed into multiple sub-domains that each map to a rectangular block. The complicated tokamak edge geometry, for instance, can be mapped to eight rectangular subdomains that connect at the intersection point of the separatrix, the *X-Point*, as shown in Figure 1(b).

Mapped multiblock grids and, more generally, composite grids (including overset and patch-based refinement) have been used in the solution of partial differential equations (PDEs) since the 1970s [22, 34]. A substantial amount of development was done in the computational aerodynamics community for external flows around complex bodies. The Cubed Sphere [31] is a type of multiblock grid that has also been developed for solving PDEs on a spherical surface; in [39], this grid is used with a high-order finite-volume method to solve the shallow-water equations. There is a rich literature on the subjects of mapped and multiblock grids that is too extensive to summarize here; we refer the interested reader to several review articles [35, 37, 2].

The starting point for the present work is the high-order finite-volume method in Colella et al. [10]. The advantage of this approach is that it is strongly conservative in the sense of [41, 40], high-order accurate, and freestream-preserving. It also has the advantage of using a smoothly-varying structured grid for its underlying discretization of space. Discretizations on such grids preserve many of the desirable properties of discretizations on Cartesian grids, such as cancellation of error in centered differences, and relatively simple quadrature rules for computing averages over cells and faces. We extend this method to the case of mapped multiblock grids, in which the computational domain in physical space is represented as the disjoint union of images of mappings that are *conforming*, meaning that they are aligned at common boundaries in such a way that when the maps are discretized, the individual faces of control volumes at those boundaries coincide. To maintain the mapped-grid formalism constraint that mappings are sufficiently differentiable, we define local mappings for each block that, beyond being conforming, need not coincide in any other way. By using a smooth continuation of each mapping beyond its block boundary, we reduce the problem of inter-block communication to that of the accurate interpolation of solution values from neighboring blocks into the halo regions.

Interpolation between neighboring grids is a common problem in multiblock, overset, and patch- and block-based adaptive mesh refinement (AMR) methods. A variety of polynomial interpolation techniques on both solution values and interface fluxes have been developed [30, 32, 9, 29]. A major con-

cern has been interpolation procedures that ensure conservation [6, 30, 9, 29] and stability [5, 28, 29]. Here, since the blocks share only a lower-dimensional interface (the PDEs are not solved in the halo regions), conservation is easily ensured by using consistent interface fluxes on the block boundaries. The main challenge, instead, is identifying a suitable stencil over which to interpolate. As in overset or AMR techniques, the halo extensions beyond a block may overlap multiple blocks, particularly in the vicinity of mesh singularities. Identifying a suitable collection of cells from the original block and its neighbors is therefore not trivial. In the fully unstructured and "mesh-free" computational-fluid-dynamics literature, one technique for reconstruction is least-squares interpolation [3, 4, 24, 15, 27, 23, 19, 8], which does not presume any underlying spatial relationship between the values used in the interpolation. This is the approach we take here. The K-exact reconstruction of Barth [4] uses averages on a selected number of neighboring cells to reconstruct a polynomial that reproduces exactly polynomials of degree up to K and preserves the average value within the computational cell, but our procedure, which is used to find that average value within the computational cell, is not required to be K-exact.

Although less common for structured grids, least-squares reconstruction is intrinsic to the genuinely multi-dimensional, high-order, Central Essentially Non-Oscillatory (CENO) finite-volume schemes [18, 16, 38, 36, 17] that have been successfully applied in 2D and 3D to inviscid and viscous compressible flow, reacting turbulent flow, and ideal magnetohydrodynamics on body-fitted, multiblock grids with block-based adaptive mesh refinement. In particular, the CENO approach has been applied to a block-adaptive cubed-sphere grid [18, 17], where the least-squares reconstruction in the flux calculation produces a uniformly high-order solution, even at points of reduced connectivity. In contrast, our use of least-squares interpolation is not intrinsic to our multidimensional finite-volume algorithm. While we require mappings on blocks to be conforming at block boundaries, these mappings are not required to be conforming beyond the boundaries (in ghost regions). We therefore use least-squares interpolation to transfer cell-average information between the overlapping, curvilinear grids. Least-squares interpolation has also been used at block boundaries for embedded-boundary methods in [20, 21] and for high-order coarse-fine mesh interpolation in AMR [25]. The methods of the present paper are applied in [26] to the solution of the shallow-water equations on the surface of a sphere, using AMR.

In the next section, we define the necessary constraints on mappings

and block arrangements within the context of the high-order finite-volume mapped-grid formalism [10]. The main challenge is the preservation of sufficient accuracy between adjacent blocks (relative to the that of the interior scheme), and the proposed algorithm for this process is described in Section 3. Several examples of advection in two and three dimensions are provided and discussed in Section 4, and these demonstrate the rate of convergence for the procedure. As will be seen, the applicability of the proposed approach presumes the existence of smooth block mappings with smooth extensions beyond block boundaries. The construction of such mappings may be clear in some applications, but may require additional effort in others. The motivating edge plasma application is one such example, which we briefly discuss in Section 5 along with directions for future research.

2. Approach

It is necessary first to introduce several definitions for and assumptions placed on the types of mappings and block arrangements that we will consider. We briefly review the high-order finite-volume formalism on mapped grids [10] and the necessary extensions to accommodate multiple blocks.

2.1. Mappings and blocks

In our mapped multiblock framework, computations are performed on an abstract coordinate space in $\mathbb{R}^{\mathbf{D}}$, which we call *computational space*, with mappings to the real problem space, also $\mathbb{R}^{\mathbf{D}}$, which we call *physical space*. We denote a point in computational space by $\boldsymbol{\xi}$, and a point in physical space by $\boldsymbol{x}$. In computational space, computations are on a set of separate *blocks*, for each of which is specified:

- a rectangular region $\mathcal{D} \subset \mathbb{R}^{\mathbf{D}}$ in computational space called the *domain* of the block;
- a region $\mathcal{R} \subset \mathbb{R}^{\mathbf{D}}$ in physical space called the *range* of the block; and
- a smooth, one-to-one *mapping function* $\boldsymbol{X} : \mathcal{D} \to \mathcal{R}$, whose inverse is also smooth.

A simple example of a mapped, two-block configuration is shown in Figure 2. In this example, the two blocks share the same computational coordinate space, but each block has its own mapping to physical space. In general, mappings on different blocks will be different.

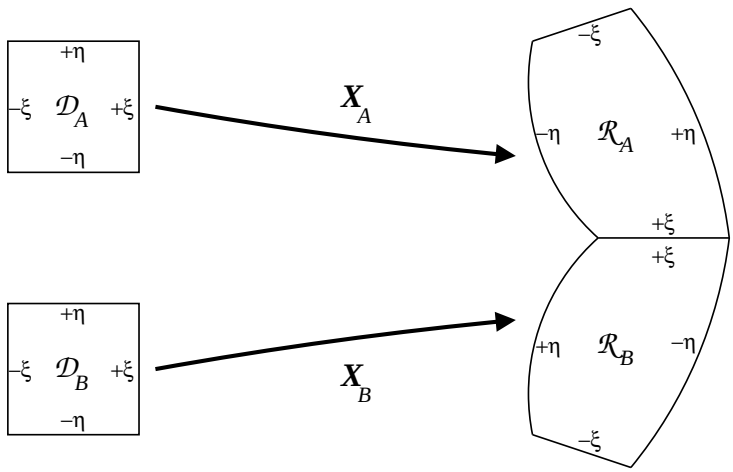

Figure 2: A multiblock coordinate system with two blocks, A and B. Left: computational space with disjoint block domains $\mathcal{D}_A$ and $\mathcal{D}_B$. Right: physical space with block ranges $\mathcal{R}_A$ and $\mathcal{R}_B$, intersecting on a shared boundary face. The faces of each block are labeled $+\xi$, $-\xi$, $+\eta$, and $-\eta$ to indicate the mappings of the block boundary faces. In this example, $\boldsymbol{X}_A^{-1}(\mathcal{R}_A \cap \mathcal{R}_B)$ is the right-hand $(+\xi)$ boundary face of $\mathcal{D}_A$, and the composite mapping $\boldsymbol{X}_B^{-1} \circ \boldsymbol{X}_A$ is an isometry from the right-hand $(+\xi)$ boundary face of $\mathcal{D}_A$ to the right-hand $(+\xi)$ boundary face of $\mathcal{D}_B$.

Because we are using a finite-volume method, we further restrict the mapped coordinate system to ensure that the images of the computational space blocks are disjoint in physical space in a specific way. First, we assume that the ranges of different blocks may intersect only at their boundaries, i.e., the ranges do not overlap. Furthermore, if the ranges of two blocks A and B do intersect (at their boundaries), then we impose an additional condition. Consider Figure 2, where A and B respectively have domains $\mathcal{D}_A$ and $\mathcal{D}_B$, mapping functions $\boldsymbol{X}_A$ and $\boldsymbol{X}_B$, and ranges $\mathcal{R}_A$ and $\mathcal{R}_B$. Then on $\boldsymbol{X}_A^{-1}(\mathcal{R}_A \cap \mathcal{R}_B)$, which is a subset of the boundary of $\mathcal{D}_A$, the composite mapping $\boldsymbol{X}_B^{-1} \circ \boldsymbol{X}_A$ to the boundary of $\mathcal{D}_B$ must be an isometry in computational space. This constraint ensures that the mappings are conforming, as will be discussed in the next section. Examples of 2D and 3D mapped multiblock meshes, which will be used later in example calculations, are provided in Figures 3 and 4.

2.2. Control volumes for finite-volume methods

In the finite-volume approach on mapped grids [10], the computational space in $\mathbb{R}^{\mathbf{D}}$ is discretized as a union of rectangular control volumes. Each

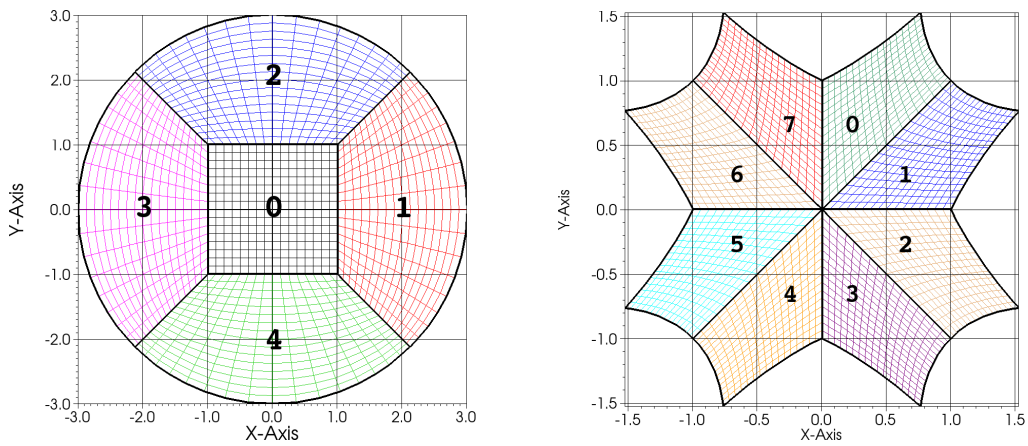

Figure 3: Left: Squared-Disk example of a disk made up of five blocks including a central square block. Right: X-Point example of a stylized version of the single-null edge-plasma geometry of Figure 1, made up of eight blocks that all intersect at the central "X" point.

control volume $V_{\boldsymbol{i}}$ takes the form

$$V_{\boldsymbol{i}} = [\boldsymbol{i}h, (\boldsymbol{i}+\boldsymbol{u})h], \quad \boldsymbol{u} = (1, 1, \ldots, 1), \tag{1}$$

where $\boldsymbol{i} = (i_1, i_2, \ldots, i_{\mathbf{D}}) \in \mathbb{Z}^{\mathbf{D}}$ is a multi-index identified with the control volume and h is the grid spacing. For each block, there is a rectangular subset of $\mathbb{Z}^{\mathbf{D}}$ that indexes all of the control volumes in the block; if some $\boldsymbol{i} \in \mathbb{Z}^{\mathbf{D}}$ is in this subset, then the entire control volume $V_{\boldsymbol{i}}$ is within the domain $\mathcal{D}$ of the block. This condition imposes a constraint on the choice of grid spacing h, such that all boundary faces of all block domains must lie along grid lines equally spaced by distance h. Associated with each control volume $V_{\boldsymbol{i}} \subset \mathcal{D}$ of a block in computational space is its image $\boldsymbol{X}(V_{\boldsymbol{i}}) \subset \mathcal{R}$ in physical space, where $\boldsymbol{X}$ is the mapping function of the block.

Because of the isometry condition on block-boundary interfaces (see Section 2.1), such interfaces must be *conforming*, meaning that whenever a control volume abuts a boundary with another block, the face of its image in physical space on that block boundary must coincide with the face of the image of some control volume of the other block on that same block-boundary face. More precisely, let A and B be blocks with, respectively, mapping functions $\boldsymbol{X}_A$ and $\boldsymbol{X}_B$, and ranges $\mathcal{R}_A$ and $\mathcal{R}_B$, such that $\mathcal{R}_A$ and $\mathcal{R}_B$ intersect (on their boundaries). Then for every control volume $V_{\boldsymbol{a}}$ of block A whose image $\boldsymbol{X}_A(V_{\boldsymbol{a}})$ intersects $\mathcal{R}_B$, there is a control volume $V_{\boldsymbol{b}}$ of block B such that $\boldsymbol{X}_B(V_{\boldsymbol{b}}) \cap \mathcal{R}_A = \boldsymbol{X}_A(V_{\boldsymbol{a}}) \cap \mathcal{R}_B$.

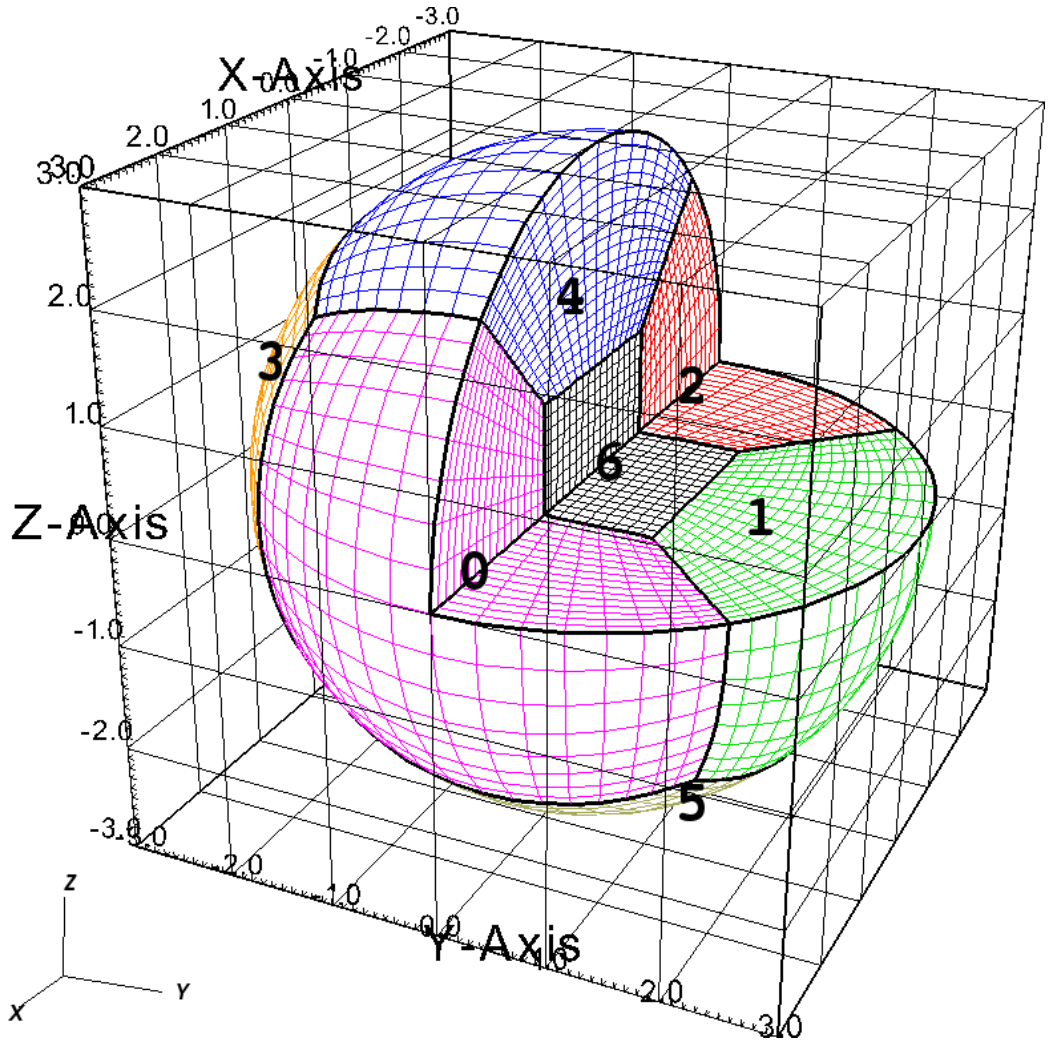

Figure 4: Cubed-Sphere example: a solid sphere, made up of seven blocks, shown here in cutaway form in order to reveal the central cubic block.

2.3. Finite-volume discretization with mapped grids

This work begins with the finite-volume discretization on single mapped grids in [10], in which coordinates $\boldsymbol{x}$ in physical space are functions of coordinates $\boldsymbol{\xi}$ in computational space, mapped by $\boldsymbol{x} = \boldsymbol{X}(\boldsymbol{\xi})$. We are solving for U in a scalar hyperbolic conservation law

$$\frac{\partial U}{\partial t} = -\nabla_{\boldsymbol{x}} \cdot \boldsymbol{F} \tag{2}$$

where $\boldsymbol{F}$ is a function of U, $\boldsymbol{x}$, and t. Equation (2) includes the divergence of the vector field $\boldsymbol{F} = (F^1, \ldots, F^{\mathbf{D}})$ in physical space, which can be written in terms of derivatives in the computational space:

$$\nabla_{\boldsymbol{x}} \cdot \boldsymbol{F} = \frac{1}{J} \nabla_{\boldsymbol{\xi}} \cdot (\boldsymbol{N}^T \boldsymbol{F}), \tag{3}$$

$$J = \det\left(\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}}\right), \quad (\boldsymbol{N}^T)_{p,q} = \det\left(R_p\left(\frac{\partial \boldsymbol{X}}{\partial \boldsymbol{\xi}}, \boldsymbol{e}^q\right)\right), \tag{4}$$

where $R_p(M, \boldsymbol{v})$ denotes the matrix obtained by replacing the pth row of the matrix M by the vector $\boldsymbol{v}$, and $\boldsymbol{e}^d$ denotes the unit vector in the dth coordinate direction.

We adopt a method-of-lines approach and integrate the PDE (2) over each control volume $V_{\boldsymbol{i}}$, obtaining an ordinary differential equation:

$$\frac{d}{dt} \int_{\boldsymbol{X}(V_{\boldsymbol{i}})} U d\boldsymbol{x} = -\int_{\boldsymbol{X}(V_{\boldsymbol{i}})} \nabla_{\boldsymbol{x}} \cdot \boldsymbol{F} d\boldsymbol{x}. \tag{5}$$

From (3)–(4) and the divergence theorem, we have the exact relationship

$$\int_{\boldsymbol{X}(V_{\boldsymbol{i}})} \nabla_{\boldsymbol{x}} \cdot \boldsymbol{F} d\boldsymbol{x} = \int_{V_{\boldsymbol{i}}} \nabla_{\boldsymbol{\xi}} \cdot (\boldsymbol{N}^T \boldsymbol{F}) d\boldsymbol{\xi} = \sum_{d=1}^{\mathbf{D}} \sum_{\pm=+,-} \pm \int_{A_d^{\pm}} (\boldsymbol{N}^T \boldsymbol{F})_d d\boldsymbol{A}_{\boldsymbol{\xi}}, \tag{6}$$

where $A_d^{\pm}$ are the high $(+)$ and low $(-)$ faces bounding $V_{\boldsymbol{i}}$ with normals pointing in the dth coordinate direction. For grid spacing h, in [10] the approximation is derived:

$$\int_{\boldsymbol{X}(V_{\boldsymbol{i}})} \nabla_{\boldsymbol{x}} \cdot \boldsymbol{F} d\boldsymbol{x} = \sum_{d=1}^{\mathbf{D}} (\mathcal{F}^d_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} - \mathcal{F}^d_{\boldsymbol{i}-\frac{1}{2}\boldsymbol{e}^d}) + O(h^4), \tag{7}$$

where $\mathcal{F}^d_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}$ is an approximation to $\langle(\boldsymbol{N}^T\boldsymbol{F})_d\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}$, set to

$$\mathcal{F}^d_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = \sum_{s=1}^{\mathbf{D}} \langle \boldsymbol{N}_{s,d}\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} \langle F^s\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} + \frac{h^2}{12}\sum_{s=1}^{\mathbf{D}} (\boldsymbol{G}^{\perp,d}(\langle \boldsymbol{N}_{s,d}\rangle))_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} \cdot (\boldsymbol{G}^{\perp,d}(\langle F^s\rangle))_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}, \tag{8}$$

where the operator $\langle\cdot\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}$ denotes a fourth-order-accurate average over the face centered at $\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d$, and where $\boldsymbol{G}^{\perp,d} \approx \nabla_{\boldsymbol{\xi}} - \boldsymbol{e}^d \frac{\partial}{\partial \xi_d}$ is a second-order accurate difference approximation to the components of the gradient operator orthogonal to the dth direction. Computing $\mathcal{F}^d_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}$ by the formula of (8) requires the integral of U over the set of control volumes $\{V_{\boldsymbol{j}} : \boldsymbol{j} = \boldsymbol{i} + \sum_{s=1}^{\mathbf{D}} c_s \boldsymbol{e}^s, c_d \in \{-1,0,1,2\},\ \text{and}\ c_s \in \{-1,0,1\}\ \text{for}\ s \neq d\}$. That is, to find the flux on a face, we need the integral of U on a rectangular stencil of control volumes, where the extent of the stencil is defined by two control volumes on each side of the face in the normal direction, and one control volume on each side of the face in every tangential direction.

2.4. Extension of mapped-grid approach to multiblock

A complication that arises with mapped multiblock grids but not with single-block mapped grids is that in physical space, a boundary face of a block may be an interface with another block. In Figure 2, for instance, the $+\xi$ boundary faces of the two blocks are interfaces with other blocks, while all of the other boundary faces are external. Coordinate lines transverse to block boundaries will be continuous if they cross those boundaries, but will generally not be smooth. The power of a mapped-multiblock approach is the ability to assign a mapping suitable to each separate block. However, the mapped finite-volume formalism outlined in Section 2.3 assumes that the mapping is sufficiently differentiable so as to preserve the order of accuracy of the fluxes. To overcome this apparent contradiction at block boundaries, we propose to extend the domain of each block $\mathcal{D}$ in computational space with *ghost cells*, which are control volumes of the form $V_{\boldsymbol{i}}$ of (1) that lie outside $\mathcal{D}$. Figures 5–6 show two layers of block-boundary ghost cells of block 2 outlined with dotted blue lines in the squared-disk geometry. We also extend the corresponding mapping function $\boldsymbol{X}$ with a smooth continuation so that the images of ghost cells lie within the ranges of neighboring blocks. In this

way, the mapped-grid finite-volume formalism remains valid not only on the interior of each block but also on the boundaries that map to interfaces with other blocks. In general, the images of ghost cells of a block will not align with images of control volumes in neighboring blocks (see Figures 5–6), so we interpolate to the ghost cells of $\mathcal{D}$ from control volumes in $\mathcal{D}$ and neighboring blocks. This interpolation is performed to high order using the method of least squares, as detailed in the next section.

When $\boldsymbol{i} + \frac{1}{2}\boldsymbol{e}^d$ is on the interface between two blocks, since we calculate $\boldsymbol{F}$ on each block separately, from cell-averaged data for the block and interpolated data on its ghost cells, there will be two numerically different results for $\mathcal{F}^d_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}$ in (8). In this case, we set $\mathcal{F}^d_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}$ to be the mean of the two values from (8) for the two different blocks.

3. Exchange operations on block boundaries

Values in block ghost cells are set by interpolation from values in *valid cells*, i.e., cells within the physical domain but perhaps within another block. These *exchange* operations are based on solving an overdetermined system of equations by the method of least squares, as in the spatial coarse-fine interpolation in [25]. For the purposes of presentation, this description is specialized to fourth-order interpolation but can be easily generalized to arbitrarily higher order.

3.1. Determination of the stencil

How the least-squares interpolation stencil is constructed can be critical for providing both accuracy and stability. For instance, an algorithm based solely on considering adjacent blocks across codimension-one boundaries can lead to poor stencil choices and high condition numbers. Such an algorithm results in poor stencil selection near the interior grid singularity in the idealized X-Point domain shown in Figure 3, where eight blocks connect at the X-Point.

The algorithm presented here for stencil selection for a ghost cell $\boldsymbol{g}$ begins with finding the valid cell $\boldsymbol{v_g}$ containing the center of $\boldsymbol{g}$, using geometry in physical space. This is Step 1 below; subsequent steps depend only on index connectivity. We include $\boldsymbol{v_g}$ and its immediate neighbors (or those of a shifted $\boldsymbol{v_g}$, in case $\boldsymbol{v_g}$ itself abuts an external boundary) in an *inner set* $\mathcal{N}^{\text{inner}}(\boldsymbol{g})$. In general, $\mathcal{N}^{\text{inner}}(\boldsymbol{g})$ will not have a sufficient number of stencil cells, and moreover, in the simplest case of the mapping being the identity

function, the stencil is required to include at least four cells in a row in each direction in order for interpolation to be fourth-order accurate. Hence, to ensure that there are enough cells and in the right positions for fourth-order interpolation, in Step 4 we include the *outer set* $\mathcal{N}^{\text{outer}}(\boldsymbol{g})$.

Following is the algorithm for finding the stencil of valid cells $\mathcal{N}(\boldsymbol{g}) = \mathcal{N}^{\text{inner}}(\boldsymbol{g}) \cup \mathcal{N}^{\text{outer}}(\boldsymbol{g})$ for fourth-order interpolation to the ghost cell $\boldsymbol{g}$. The stencil is independent of the data values being interpolated.

1. Let $\boldsymbol{x_g}$ be the point in physical space that is mapped from the center of the rectangular ghost cell $\boldsymbol{g}$ in computational space. Find the block $b_{\boldsymbol{g}}^{\text{valid}}$ that has $\boldsymbol{x_g}$ in its range, and let $\boldsymbol{v_g}$ be the valid cell of block $b_{\boldsymbol{g}}^{\text{valid}}$ that contains $\boldsymbol{x_g}$.
2. Let $\boldsymbol{c_g}$ be the valid cell in block $b_{\boldsymbol{g}}^{\text{valid}}$ that is closest to $\boldsymbol{v_g}$ in index space and is separated from the *external* boundary by at least one cell in every dimension. Then $\boldsymbol{c_g}$ will be the same as $\boldsymbol{v_g}$ except when $\boldsymbol{v_g}$ abuts the external boundary.
3. Let $\mathcal{N}^{\text{inner}}(\boldsymbol{g})$ be the set of all valid cells that have any vertices in common with cell $\boldsymbol{c_g}$ of block $b_{\boldsymbol{g}}^{\text{valid}}$ in physical space. If cell $\boldsymbol{c_g}$ abuts an interface of block $b_{\boldsymbol{g}}^{\text{valid}}$ with another block, then $\mathcal{N}^{\text{inner}}(\boldsymbol{g})$ will include valid cells of that other block.
4. Let $\mathcal{N}^{\text{outer}}(\boldsymbol{g})$ be the set of all valid cells $\boldsymbol{v_g} \pm 2\boldsymbol{e}^d$ for each dimension d, except for any such cells that are already in $\mathcal{N}^{\text{inner}}(\boldsymbol{g})$. If any particular $\boldsymbol{v_g} \pm 2\boldsymbol{e}^d$ is invalid because it goes beyond a boundary face in dimension d of block $b_{\boldsymbol{g}}^{\text{valid}}$, then:
   - If the boundary face is an interface with another block b' and $\boldsymbol{v_g}$ is separated from that interface by one cell, so that $\boldsymbol{v_g} \pm \boldsymbol{e}^d$ is a valid cell in block $b_{\boldsymbol{g}}^{\text{valid}}$ that abuts the interface, then include in $\mathcal{N}^{\text{outer}}(\boldsymbol{g})$ the valid cell in block b' that shares a face with $\boldsymbol{v_g} \pm \boldsymbol{e}^d$ in block $b_{\boldsymbol{g}}^{\text{valid}}$.
   - If the boundary face is an interface with another block b' and $\boldsymbol{v_g}$ abuts that interface, then include in $\mathcal{N}^{\text{outer}}(\boldsymbol{g})$ the valid cell in block b' that shares the opposite face of the valid cell in block b' that shares a face with $\boldsymbol{v_g}$ in block $b_{\boldsymbol{g}}^{\text{valid}}$.
   - If the boundary face is external and if $\boldsymbol{v_g}$ abuts that face, then include in $\mathcal{N}^{\text{outer}}(\boldsymbol{g})$ the valid cell $\boldsymbol{v_g} \mp 3\boldsymbol{e}^d$ (that is, three cells away from $\boldsymbol{v_g}$ in the *opposite* direction from the boundary) of block $b_{\boldsymbol{g}}^{\text{valid}}$. Otherwise, do nothing.

For the two geometries of Figure 3, in Figures 5 and 6 we show four examples of interpolation stencils for ghost cells in the disk geometry, and in Figure 7 we show an example of an interpolation stencil for a ghost cell in the X-Point geometry.

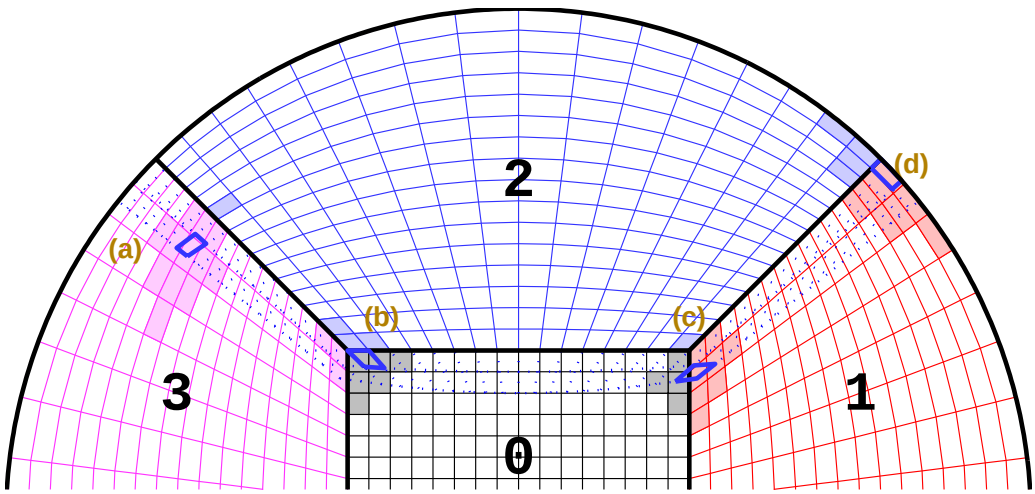

Figure 5: Interpolation stencils for four sample ghost cells in the 2D multiblock disk geometry of Figure 3. Two layers of ghost cells of block 2 are shown with dotted blue lines, and four of these cells, marked (a) through (d), are indicated with thicker blue outlines. The shaded cells around each of these ghost cells correspond to the interpolation stencil of the ghost cell. Further details on these four stencils are shown in Figure 6.

3.2. Least-squares interpolation

Once a sufficient number of valid cells for the interpolation stencil have been found (based on the order requirements), the least-squares interpolation proceeds as follows. Let $\mathcal{N}(\boldsymbol{g}) = \mathcal{N}^{\text{inner}}(\boldsymbol{g}) \cup \mathcal{N}^{\text{outer}}(\boldsymbol{g})$ be the set of valid cells to be used for the interpolation stencil of cell $\boldsymbol{g}$. In what follows, $R_{\boldsymbol{g}}$ is the mean distance from $\boldsymbol{x}_{\boldsymbol{g}}$, the center in physical space of $\boldsymbol{g}$, to the centers in physical space of the cells of $\mathcal{N}(\boldsymbol{g})$, where the center of a cell in physical space is the point mapped from the center of the rectangular cell in computational space.

We denote the average over control volume $V_{\boldsymbol{i}}$ by $\langle \cdot \rangle_{\boldsymbol{i}}$, defined by:

$$\langle f \rangle_{\boldsymbol{i}} = \frac{\int_{\boldsymbol{X}(V_{\boldsymbol{i}})} f(\boldsymbol{x}) d\boldsymbol{x}}{\int_{\boldsymbol{X}(V_{\boldsymbol{i}})} d\boldsymbol{x}}. \tag{9}$$

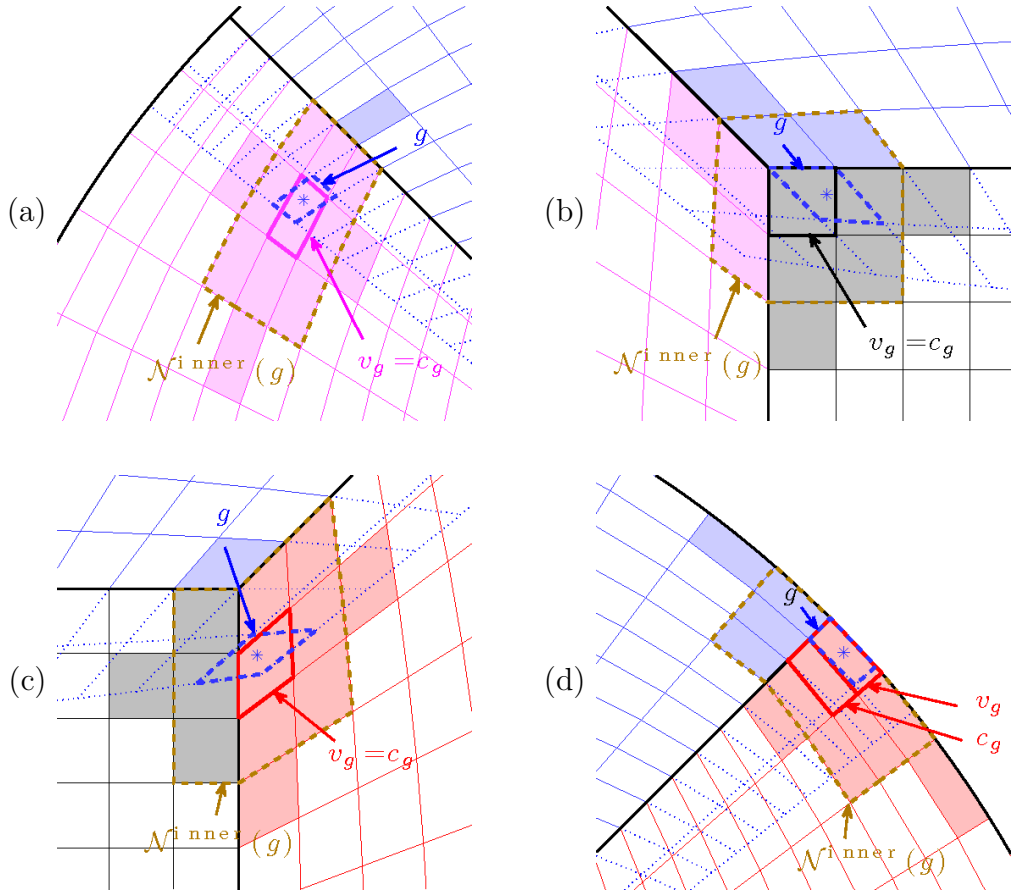

Figure 6: Close-up views of the stencils of the four ghost cells (a) through (d) shown in Figure 5. In each case above, the ghost cell is indicated by $\boldsymbol{g}$ and is shown with a dashed outline in blue, the color of block 2. The center of $\boldsymbol{g}$ is marked with a blue $*$, and the valid cell that contains this point is $\boldsymbol{v_g}$, defined in Step 1 of Section 3.1. Here $\boldsymbol{v_g}$ is indicated with a thicker outline, as is $\boldsymbol{c_g}$, which is defined in Step 2. Stencil cells are shaded the color of their respective blocks. The cells of the inner set $\mathcal{N}^{\text{inner}}(\boldsymbol{g})$, defined in Step 3, are marked with a dashed brown outline around the whole set, and the remaining stencil cells belong to the outer set $\mathcal{N}^{\text{outer}}(\boldsymbol{g})$ as defined in Step 4. Note that in cases (a), (b), and (c), $\boldsymbol{v_g}$ and $\boldsymbol{c_g}$ are the same cell, and $\mathcal{N}^{\text{outer}}(\boldsymbol{g})$ has four cells, but in case (d), $\boldsymbol{c_g}$ is a cell adjacent to $\boldsymbol{v_g}$ because $\boldsymbol{v_g}$ itself is on the external boundary, and $\mathcal{N}^{\text{outer}}(\boldsymbol{g})$ has only three cells for the same reason. Also note that the inner set $\mathcal{N}^{\text{inner}}(\boldsymbol{g})$ has nine cells in cases (a), (c), and (d), but only eight cells in case (b), because in case (b), the central cell $\boldsymbol{c_g}$ is at a corner of block 0 (colored black), where block 0 meets only two other blocks.

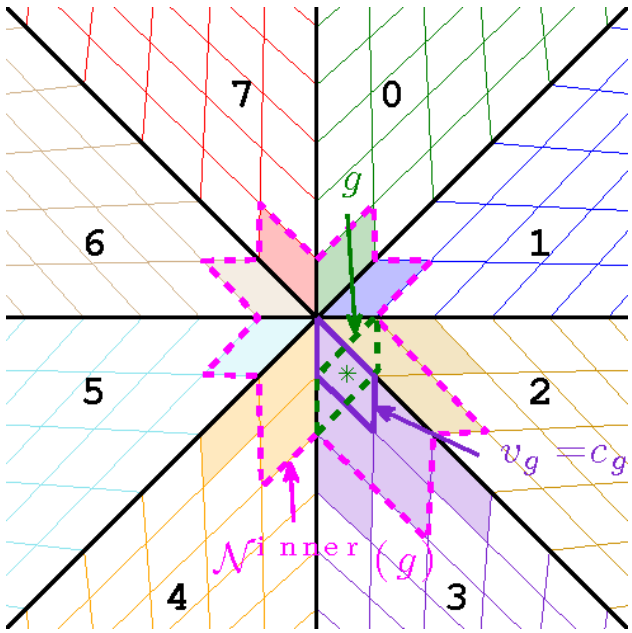

Figure 7: Interpolation stencil for a ghost cell near the center of the X-Point geometry of Figure 3. A ghost cell $\boldsymbol{g}$ of block 0 is shown with a green dashed outline, and its center, marked by a green $*$, lies in a valid cell $\boldsymbol{v_g}$ which is indicated here with a thicker outline of the violet color of its block $b_{\boldsymbol{g}}^{\text{valid}} = 3$. The seventeen cells in the stencil $\mathcal{N}(\boldsymbol{g})$ are shaded with the colors of their respective blocks. Of these, the thirteen cells of the inner set $\mathcal{N}^{\text{inner}}(\boldsymbol{g})$ are marked with a dashed magenta outline around the whole set, and include cells in all eight blocks because the central cell $\boldsymbol{c_g} = \boldsymbol{v_g}$ has a vertex at the "X" point where all eight blocks intersect. The remaining four stencil cells belong to the outer set $\mathcal{N}^{\text{outer}}(\boldsymbol{g})$, consisting of two cells in block 3, which is the block that contains $\boldsymbol{v_g}$, and one cell each in blocks 2 and 4, which are the two blocks that have codimension-one interfaces with block 3.

For clarity, we demonstrate the interpolation procedure in 2D; the extension to higher dimensions is natural. Letting $(x_{\boldsymbol{g}}, y_{\boldsymbol{g}}) = \boldsymbol{x}_{\boldsymbol{g}}$, the idea is to use values of f averaged over stencil cells $\mathcal{N}(\boldsymbol{g})$ to find coefficients a_{pq} (for $p, q \geq 0$ and $p + q \leq 3$) of a third-degree Taylor polynomial centered at $(x_{\boldsymbol{g}}, y_{\boldsymbol{g}})$,

$$f(x, y) = \sum_{p,q\geq 0; p+q\leq 3} a_{pq} \left(\frac{x - x_{\boldsymbol{g}}}{R_{\boldsymbol{g}}}\right)^p \left(\frac{y - y_{\boldsymbol{g}}}{R_{\boldsymbol{g}}}\right)^q + O(h^4), \tag{10}$$

and once we have the coefficients a_{pq}, we then evaluate (10) to find an approximation to the average of f over the ghost cell $\boldsymbol{g}$. In more detail:

1. Averaging both sides of (10) over each cell in the stencil $\mathcal{N}(\boldsymbol{g})$ suggests a method of finding values for coefficients a_{pq}. Since we know the average of f over each stencil cell, we can solve a least-squares problem to find a_{pq} satisfying as closely as possible the overdetermined system of equations

$$\sum_{p,q\geq 0; p+q\leq 3} a_{pq} \left\langle \left(\frac{x - x_{\boldsymbol{g}}}{R_{\boldsymbol{g}}}\right)^p \left(\frac{y - y_{\boldsymbol{g}}}{R_{\boldsymbol{g}}}\right)^q \right\rangle_{\boldsymbol{j}} = \langle f \rangle_{\boldsymbol{j}} \quad \text{for all } \boldsymbol{j} \in \mathcal{N}(\boldsymbol{g}). \tag{11}$$

2. Using the Taylor coefficients a_{pq} obtained in step 1, average the evaluation of the Taylor polynomial over ghost cell $\boldsymbol{g}$, in order to find a value for $\langle f \rangle_{\boldsymbol{g}}$:

$$\langle f \rangle_{\boldsymbol{g}} = \sum_{p,q\geq 0; p+q\leq 3} a_{pq} \left\langle \left(\frac{x - x_{\boldsymbol{g}}}{R_{\boldsymbol{g}}}\right)^p \left(\frac{y - y_{\boldsymbol{g}}}{R_{\boldsymbol{g}}}\right)^q \right\rangle_{\boldsymbol{g}}. \tag{12}$$

The cell averages of polynomials in the summands of (11) and (12) may be computed by fourth-order accurate quadrature rules. Starting with fourth-order accurate $\langle f \rangle_{\boldsymbol{j}}$ on the stencil cells $\boldsymbol{j} \in \mathcal{N}(\boldsymbol{g})$, since $R_{\boldsymbol{g}}$ associated with stencils scales linearly with the grid spacing, as do $x - x_{\boldsymbol{g}}$ and $y - y_{\boldsymbol{g}}$ for (x, y) in stencil cells, the coefficients a_{pq} calculated in Step 1 will be fourth-order accurate approximations to the Taylor coefficients in (10). Then $\langle f \rangle_{\boldsymbol{g}}$ as calculated in Step 2 will also be fourth-order accurate.

If f is constant, then this procedure sets interpolated ghost-cell values to the same constant: if $\langle f \rangle_{\boldsymbol{j}} = K$ for all stencil cells $\boldsymbol{j} \in \mathcal{N}(\boldsymbol{g})$, then the system (11) has the unique solution of $a_{00} = K$ and $a_{pq} = 0$ whenever $p > 0$ or $q > 0$, and hence in (12), $\langle f \rangle_{\boldsymbol{g}} = K$.

3.3. Constraints on stencils

In general, in a stencil for k^{th}-order interpolation, there must be at least k cells with different projections in each dimension. The construction of the outer set $\mathcal{N}^{\text{outer}}(\boldsymbol{g})$ in Step 4 of Section 3.1 for fourth-order interpolation ensures that there are at least four cells in a row in each dimension.

For the system (11) to be overdetermined, there must be at least as many equations as unknowns. Hence the number of cells in a kth-order interpolation stencil must be at least the number of coefficients of a Taylor polynomial of degree $k-1$, which is $\binom{\mathbf{D}+k-1}{\mathbf{D}}$. In the generic case, the inner set $\mathcal{N}^{\text{inner}}(\boldsymbol{g})$ constructed in Step 3 of Section 3.1 for fourth-order interpolation has $3^{\mathbf{D}}$ cells. However, if $\boldsymbol{g}$ is near a corner of a block, $\mathcal{N}^{\text{inner}}(\boldsymbol{g})$ may have fewer cells, as in the example in Figure 6(b). In any case, $\mathcal{N}^{\text{inner}}(\boldsymbol{g})$ will include at least $2^{\mathbf{D}}$ cells in block $b_{\boldsymbol{g}}^{\text{valid}}$ plus at least $2^{\mathbf{D}-1}$ additional cells for each of the $\mathbf{D}$ dimensions, which are extended from the faces of the $2^{\mathbf{D}}$ stencil cells in block $b_{\boldsymbol{g}}^{\text{valid}}$. Hence the size of $\mathcal{N}^{\text{inner}}(\boldsymbol{g})$ is at least $2^{\mathbf{D}-1}(2+\mathbf{D})$. By construction in Step 4, the size of the outer set $\mathcal{N}^{\text{outer}}(\boldsymbol{g})$ is at least $\mathbf{D}$, and is $2\mathbf{D}$ in the generic case. The total size of $\mathcal{N}(\boldsymbol{g})$ is therefore at least $2^{\mathbf{D}-1}(2+\mathbf{D})+\mathbf{D}$, and is $3^{\mathbf{D}}+2\mathbf{D}$ in the generic case. When $\mathbf{D}=2$, $\mathcal{N}(\boldsymbol{g})$ has at least 10 cells, and has 13 cells in the generic case. When $\mathbf{D}=3$, $\mathcal{N}(\boldsymbol{g})$ has at least 23 cells, and has 33 cells in the generic case. The number of unknown coefficients is $\binom{\mathbf{D}+3}{\mathbf{D}} = (\mathbf{D}+3)(\mathbf{D}+2)(\mathbf{D}+1)/6$, which is 10 when $\mathbf{D}=2$, and 20 when $\mathbf{D}=3$. In any dimension, the number of equations is always at least the number of coefficients, thus guaranteeing an over-constrained system for the fourth-order interpolation stencil of Section 3.1.

3.4. Numerical implementation

The system of equations (11) can be written in matrix form as $M(\boldsymbol{g})a = \langle f \rangle$, where $M(\boldsymbol{g})$ is the matrix with rows indexed by $\boldsymbol{j} \in \mathcal{N}(\boldsymbol{g})$ and columns indexed by p, q where $p, q \geq 0$ and $p+q \leq 3$. Let $m = |\mathcal{N}(\boldsymbol{g})|$ be the number of rows (stencil cells), and $n = \binom{\mathbf{D}+3}{\mathbf{D}}$ be the number of columns (coefficients).

In the Taylor polynomial (10), the reason we divide $x - x_{\boldsymbol{g}}$ and $y - y_{\boldsymbol{g}}$ by $R_{\boldsymbol{g}}$ is to improve the condition number of matrix $M(\boldsymbol{g})$. As in [19], we are scaling the columns of the matrix so that they have a similar order of magnitude, independent of the grid spacing. We find that it is not necessary to scale the rows of the matrix, by methods such as weighting stencil cells by distance, as is done in [24].

Using (11) and (12), we can compute the interpolated value $\langle f \rangle_{\boldsymbol{g}}$ as simply a sum of weighted stencil-cell values

$$\langle f \rangle_{\boldsymbol{g}} = \sum_{\boldsymbol{j} \in \mathcal{N}(\boldsymbol{g})} w_{\boldsymbol{j}}(\boldsymbol{g}) \langle f \rangle_{\boldsymbol{j}}, \tag{13}$$

where the weights $w_{\boldsymbol{j}}(\boldsymbol{g})$ depend only on the grid, and hence can be computed once and stored. We use the thin QR factorization [13] $M(\boldsymbol{g}) = Q_1 R_1$ where Q_1 is the first n columns of an orthogonal $m \times m$ matrix, and R_1 is an upper-triangular $n \times n$ matrix. If we solve for the $n \times m$ matrix X in $R_1 X = Q_1^T$, then each column $\boldsymbol{j}$ of X contains coefficients a_{pq} approximating the function that is constant 1 on stencil cell $\boldsymbol{j}$ and 0 on all the other stencil cells. Therefore, the weights in (13) are

$$w_{\boldsymbol{j}}(\boldsymbol{g}) = \sum_{p,q \geq 0; p+q \leq 3} X_{pq,\boldsymbol{j}} \left\langle \left(\frac{x - x_{\boldsymbol{g}}}{R_{\boldsymbol{g}}} \right)^p \left(\frac{y - y_{\boldsymbol{g}}}{R_{\boldsymbol{g}}} \right)^q \right\rangle_{\boldsymbol{g}}. \tag{14}$$

4. Example problems

To demonstrate the least-squares approach to inter-block transfers, we consider solution of the linear advection equation for scalar $u(\boldsymbol{x}, t)$:

$$\begin{aligned} \frac{\partial u}{\partial t} + \nabla \cdot (u \boldsymbol{v}) = 0 \quad & \text{on } \boldsymbol{x} \in \Omega \subset \mathbb{R}^{\mathbf{D}}, t \geq 0; \\ u(\boldsymbol{x}, 0) = u_0(\boldsymbol{x}), & \end{aligned} \tag{15}$$

where $\boldsymbol{v}(\boldsymbol{x})$ is the prescribed time-independent velocity. We will solve this problem on three different mapped multiblock domains using similar initial conditions

$$u(\boldsymbol{x}, 0) = u_0(\boldsymbol{x}) = B_\rho(\boldsymbol{x} - \boldsymbol{c}_0), \tag{16}$$

where $\boldsymbol{c}_0$ is a domain-specific offset. The function $B_\rho(\boldsymbol{x})$ is a cosine bell with maximum value of one and radius ρ:

$$B_\rho(\boldsymbol{x}) = \begin{cases} (\frac{1}{2} + \frac{1}{2}\cos(\frac{\pi |\boldsymbol{x}|}{\rho}))^3 & \text{if } |\boldsymbol{x}| \leq \rho; \\ 0, & \text{otherwise.} \end{cases} \tag{17}$$

On the circle $|\boldsymbol{x}| = \rho$, the first five derivatives of B_ρ are all zero, so B_ρ is C^5 continuous everywhere.

With $\boldsymbol{F} = u\boldsymbol{v}$, (15) is of form (2), and hence the spatial integral of u over each control volume satisfies the ordinary differential equation (5). We evaluate the right-hand side of (5) as outlined in Sections 2.3–2.4, and integrate over time with the explicit, four-stage, fourth-order classical Runge–Kutta scheme as in [10], but without a limiter. For mapped grids in physical space, we apply the stability conditions in computational space using the computational-space velocity vector $\boldsymbol{w} = J^{-1}\boldsymbol{N}^T\boldsymbol{v}$. As shown in [10], the Runge–Kutta stability condition gives a global limit on the ratio of the time step Δt to the grid spacing h:

$$\frac{\Delta t}{h} \max_{\boldsymbol{i}}\{\sum_{d=1}^{\mathbf{D}} |\boldsymbol{w}_{\boldsymbol{i}} \cdot \boldsymbol{e}^d|\} \lesssim 2.06 \tag{18}$$

On all of these examples, we set homogeneous Dirichlet conditions on the external boundaries, so that u and hence the flux $u\boldsymbol{v}$ are zero there.

The examples to follow have known exact solutions. If $\epsilon_{\boldsymbol{i}}$ is the difference between the exact solution and the calculated solution over control volume $V_{\boldsymbol{i}} \subset \Omega$, we report the L^∞ norm of this error, which is $||\epsilon||_\infty = \max_{\boldsymbol{i}\in\mathcal{I}} |\epsilon_{\boldsymbol{i}}|$, where $\mathcal{I} \subset \mathbb{Z}^{\mathbf{D}}$ is the set of indices of valid cells of the domain. We also report the rate at which error diminishes with successive refinements of the grids by a factor of 2. Writing ϵ^h for the error with grid spacing h, this rate from $2h$ to h is $\log_2(||\epsilon^{2h}||_\infty/||\epsilon^h||_\infty)$. In the examples to follow, the coarsest grid spacing we use is such that each domain block is divided into 32 cells in each dimension, in order to be close to the asymptotic error regime as well as to avoid complications in some geometries where singularities arise when extending continuation mappings too far beyond the domain block. These examples were implemented in the Chombo framework [1].

4.1. Circular advection on the squared-disk

The first example is on the squared-disk multiblock coordinate system shown in Figure 3. The mappings here are adapted from the cubed-sphere mappings in Diener et al. [11]. (The full 3D cubed-sphere mapping will be used in the example in Section 4.3.) The outer boundary of the disk has radius r_1, and the central square has length $2r_0$, where we choose $r_0 = 1$ and $r_1 = 3$. Each block has domain $[-1, 1]^2$. We define the mappings for each block from computational-space coordinates $(a, b) \in [-1, 1]^2$ to physical-space coordinates (x, y) as follows. For the outer blocks (all but block 0), we

define the following:

$$r = \frac{1}{2}[r_0(1-b) + r_1(1+b)]; \tag{19}$$

$$s = r\sqrt{\frac{r_1 - r_0}{(r_1 - r) + (r - r_0)(1+a^2)}}. \tag{20}$$

Then the five blocks are:

- Block 0, the central square: $x = r_0 a, y = r_0 b$.
- Block 1, around positive x-axis: $x = s, y = as$.
- Block 2, around positive y-axis: $x = -as, y = s$.
- Block 3, around negative x-axis: $x = -s, y = -as$.
- Block 4, around negative y-axis: $x = as, y = -s$.

The purpose of this example is to demonstrate that fourth-order convergence is obtained on a 2D mapped multiblock geometry that is a surrogate of the 3D cubed-sphere geometry.

On this problem domain, the velocity vector for rotation about the origin is defined to be

$$\boldsymbol{v}(x, y) = 2\pi\omega(-y, x), \tag{21}$$

where we take $\omega = 1$, so that one full rotation takes unit time.

For the initial conditions defined in (17), we take the radius $\rho = 1$ and center the bell at $\boldsymbol{c}_0 = (1, 1)$. These initial conditions are plotted in Figure 8.

At time t, the center of the cosine bell will have rotated though an angle of $\theta(t) = 2\pi\omega t$ and will be located at

$$\boldsymbol{c}(t) = \begin{pmatrix} \cos\theta(t) & -\sin\theta(t) \\ \sin\theta(t) & \cos\theta(t) \end{pmatrix} \boldsymbol{c}_0. \tag{22}$$

The exact solution of (15) is then

$$u(\boldsymbol{x}, t) = B_\rho(\boldsymbol{x} - \boldsymbol{c}(t)). \tag{23}$$

Over the interval $t \in [0, 1]$, the center of the advected cosine bell begins at the point $\boldsymbol{c}_0 = (1, 1)$ at the intersection of the ranges of blocks 0, 1, and 2; it then passes through the ranges of blocks 2, 3, 4, and 1 (being at a corner of

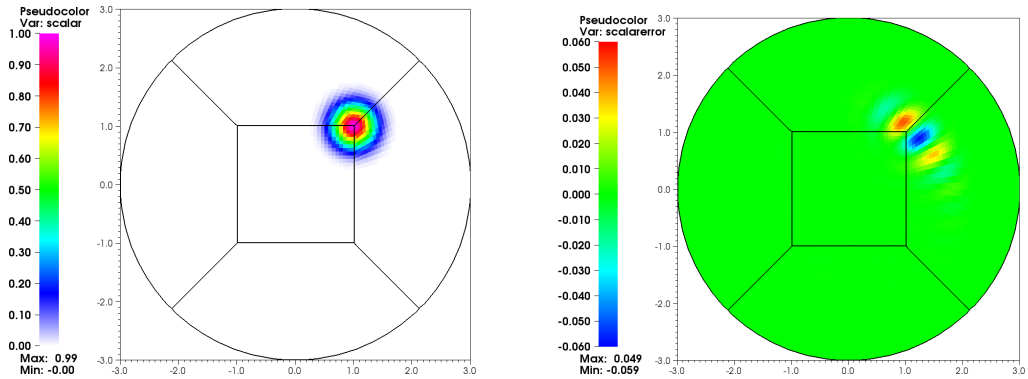

Figure 8: Left: profile of initial solution for circular advection problem of section 4.1, with each block having 32×32 cells. The final solution after one full counterclockwise rotation should look the same. Right: profile of error in final solution, after 200 time steps of length $\Delta t = 0.005$.

the range of block 0 at times $t = 1/4$, $1/2$, and $3/4$), before returning to its initial position at time $t = 1$.

For the grid convergence study, we take grid spacings of $h = 1/N$, where there are N cells along the length of each block and where N is a power of 2, from $N = 32$ to $N = 1024$. With the finest resolution of $N = 1024$, there are roughly 1000 cells across the nonzero portion of the solution. Since the computational-space velocity $\boldsymbol{w}$ corresponding to $\boldsymbol{v}$ in (21) satisfies $\max_{\boldsymbol{i}} \sum_{d=1}^{\mathbf{D}} |\boldsymbol{w_i} \cdot \boldsymbol{e}^d| \lesssim 10.49$, we choose the time step Δt such that $\Delta t/h = 0.16$, which corresponds to a CFL number of 1.68.

Table 1: L^∞ norms of solution error in circular advection problem of Section 4.1, with each block being divided into $N \times N$ cells.

N	32	64	128	256	512	1024
time steps	200	400	800	1600	3200	6400
error	5.89e-02	3.73e-03	2.36e-04	1.49e-05	9.33e-07	5.84e-08
rate		3.98	3.98	3.98	4.00	4.00

Results of a grid convergence study are presented in Table 1, which shows the L^∞ norm of the difference between calculated and exact solutions for different resolutions. The convergence rate, computed using two successive resolutions, is shown to achieve fourth order.

4.2. Uniform advection in X-Point geometry

The X-Point geometry multiblock coordinate system of Figure 3 serves as an idealized stand-in for the most challenging feature of the tokamak edge plasma geometry. The blocks are numbered 0 through 7. Each block has domain $[0,1]^2$. For each point $(a,b) \in [0,1]^2$ in computational space, the physical-space point (x,y) in the range of block B is determined by the transformation

$$\begin{pmatrix} x \\ y \end{pmatrix} = \frac{M_B}{4} \begin{pmatrix} a(3+(1-\alpha b)^2) \\ b(3+(1-\alpha a)^2) \end{pmatrix}, \tag{24}$$

where $\alpha = 3-\sqrt{5}$, and the matrices M_B are as follows:

$$M_0 = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix},\ M_1 = M_0^T,\ M_2 = \begin{pmatrix} 1 & 1 \\ -1 & 0 \end{pmatrix},\ M_3 = \begin{pmatrix} 0 & 1 \\ -1 & -1 \end{pmatrix},$$

$$M_4 = -M_0,\ M_5 = -M_1,\ M_6 = -M_2,\ \text{and } M_7 = -M_3.$$

In this example, we advect the cosine bell with constant velocity directly through the X-Point to demonstrate the ability of the approach to obtain fourth-order convergence even at this singular point in the grid.

Consider the uniform velocity to be

$$\boldsymbol{v}(\boldsymbol{x}) = (0.8, 0.4)$$

and initial conditions (16) of the cosine bell of radius $\rho = 0.4$ centered at

$$\boldsymbol{c}_0 = (-0.4, -0.2),$$

with B_ρ as defined in (17). Since $\boldsymbol{v}$ is uniform, the exact solution of (15) is

$$u(\boldsymbol{x}, t) = u_0(\boldsymbol{x} - \boldsymbol{v}t),$$

and hence, from (16),

$$u(\boldsymbol{x}, t) = B_\rho(\boldsymbol{x} - \boldsymbol{c}(t)), \tag{25}$$

where $\boldsymbol{c}(t) = \boldsymbol{c}_0 + \boldsymbol{v}t$.

The computed solution profiles at the initial, mid-point, and final times are shown in Figure 9. Over the interval $t \in [0,1]$, the center of the advected cosine bell in physical space begins at the point $\boldsymbol{c}_0 = (-0.4, -0.2)$ in the

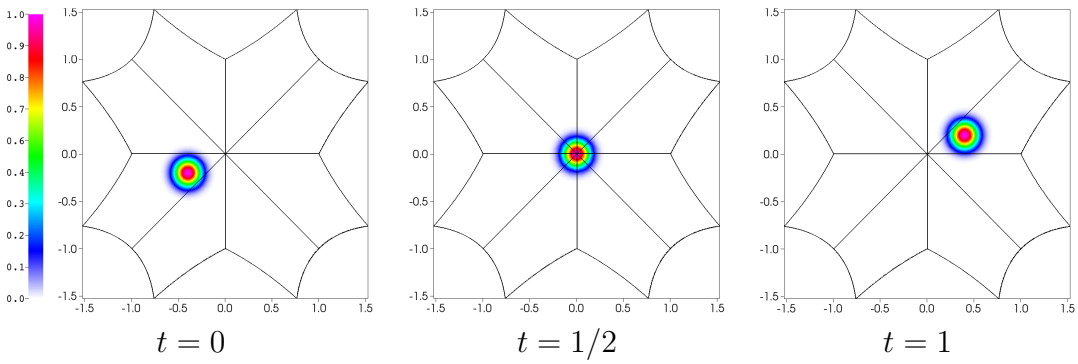

Figure 9: Profiles of advected quantity in uniform-advection example in X-Point geometry at initial time (left), mid-point (center), and final time (right) as calculated for blocks of length 128.

range of block 5. At the mid-point in time, the bell is at the X-Point $(0, 0)$, and it ends at $-\boldsymbol{c}_0 = (0.4, 0.2)$ in the range of block 1.

For the grid convergence study, we take a sequence of grid spacings of size $h = 1/N$, where there are N cells along the length of each block and where N is a power of 2, from $N = 32$ to $N = 1024$. With the finest resolution of $N = 1024$, there are roughly 600 cells across the nonzero portion of the solution. Since the computational-space velocity $\boldsymbol{w}$ corresponding to $\boldsymbol{v} = (0.8, 0.4)$ satisfies $\max_{\boldsymbol{i}} \sum_{d=1}^{\mathbf{D}} |\boldsymbol{w}_{\boldsymbol{i}} \cdot \boldsymbol{e}^d| \lesssim 2.34$, we choose the time step Δt such that $\Delta t/h = 32/45$, which corresponds to a CFL number of 1.67.

Table 2: L^∞ norms of solution error in uniform-advection problem of section 4.2 in X-Point geometry with each block being divided into $N \times N$ cells.

N	32	64	128	256	512	1024
time steps	45	90	180	360	720	1440
error	2.53e-02	1.80e-03	1.16e-04	7.23e-06	4.51e-07	2.82e-08
rate	3.81	3.96	4.00	4.00	4.00	

Results of a grid convergence study are presented in Table 2, which shows the L^∞ norm of the difference between calculated and exact solutions for different resolutions. The convergence rate, computed using two successive resolutions, is shown to achieve fourth order.

4.3. Uniform advection in solid cubed-sphere geometry

The final example is on the cubed-sphere multiblock coordinate system shown in Figure 4, which is based on the mappings from Diener et al. [11]. The outer boundary of the sphere has radius r_1, and the central cube has length $2r_0$, where we choose $r_0 = 1$ and $r_1 = 3$. Each block has domain $[-1, 1]^3$. We define the mappings for each block from computational-space coordinates $(a, b, c) \in [-1, 1]^3$ to physical-space coordinates (x, y, z) as follows. For the outer blocks (all but block 6), we define the following:

$$r = \frac{1}{2}[r_0(1 - c) + r_1(1 + c)]; \tag{26}$$

$$s = r\sqrt{\frac{r_1 - r_0}{(r_1 - r) + (r - r_0)(1 + a^2 + b^2)}}. \tag{27}$$

Then the seven blocks are

- Block 0, around positive x-axis: $x = s, y = bs, z = as$.
- Block 1, around positive y-axis: $x = -bs, y = s, z = as$.
- Block 2, around negative x-axis: $x = -s, y = -bs, z = as$.
- Block 3, around negative y-axis: $x = bs, y = -s, z = as$.
- Block 4, around positive z-axis: $x = -as, y = bs, z = s$.
- Block 5, around negative z-axis: $x = as, y = bs, z = -s$.
- Block 6, the central cube: $x = r_0 a, y = r_0 b, z = r_0 c$.

The purpose of this example is to demonstrate that fourth-order convergence is obtained on a 3D mapped multiblock geometry.

Consider a uniform velocity

$$\boldsymbol{v}(\boldsymbol{x}) = (2.8, 2.8, 2.8)$$

and initial conditions (16) of a cosine bell of radius $\rho = 0.4$ centered at

$$\boldsymbol{c}_0 = (-1.4, -1.4, -1.4),$$

with B_ρ as defined in (17). Since $\boldsymbol{v}$ is uniform, as in the example of Section 4.2, the exact solution of (15) is again given by (25).

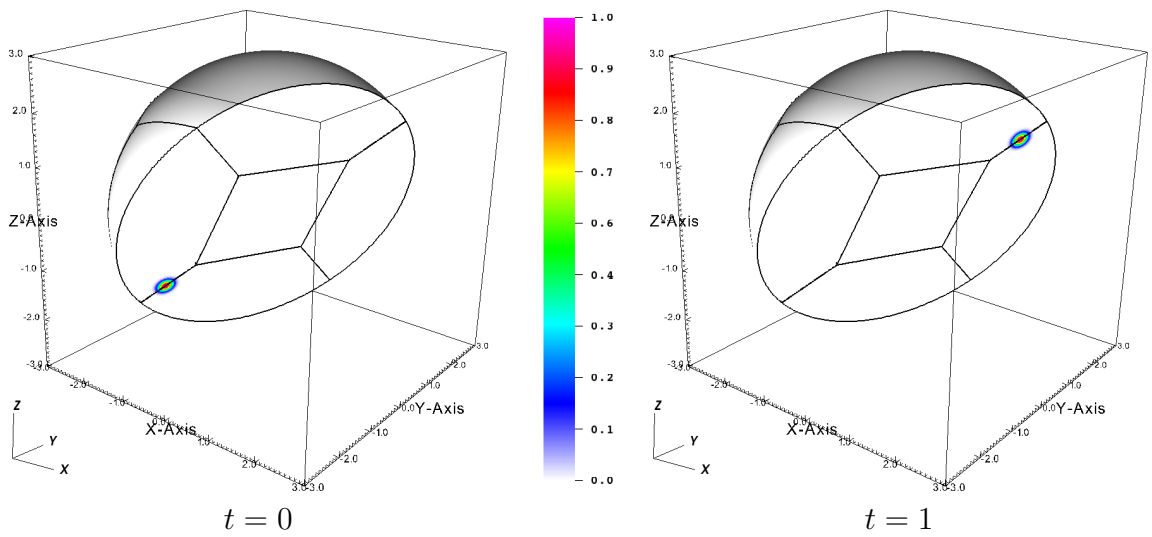

Figure 10: Cross-sectional profiles of advected quantity in uniform-advection example in 3D cubed-sphere geometry at initial time (left) and final time (right) as computed with blocks of length $N = 128$.

The computed solution at the initial ($t = 0$) and final ($t = 1$) times is shown in Figure 10. Over this time interval, the center of the advected cosine bell in physical space begins at $\boldsymbol{c}_0 = (-1.4, -1.4, -1.4)$ on the intersection line of the ranges of blocks 2, 3, and 5. At the mid-point in time, the bell is at $(0, 0, 0)$ in the range of the central block 6. Finally, it ends at $-\boldsymbol{c}_0 = (1.4, 1.4, 1.4)$ on the intersection line of the ranges of blocks 0, 1, and 4.

For the grid convergence study, we take a sequence of grid spacings of size $h = 1/N$, where there are N cells along the length of each block and where N is a power of 2, from $N = 32$ to $N = 256$. Note that the maximum number of cells per dimension is one quarter of that used in the previous 2D examples since the problem size grows as N^3. With the finest resolution of $N = 256$, there are roughly 160 cells across the nonzero portion of the solution. Since the computational-space velocity $\boldsymbol{w}$ corresponding to $\boldsymbol{v} = (2.8, 2.8, 2.8)$ satisfies $\max_{\boldsymbol{i}} \sum_{d=1}^{\mathbf{D}} |\boldsymbol{w_i} \cdot \boldsymbol{e}^d| \lesssim 8.36$, we choose the time step Δt such that $\Delta t/h = 0.2$, which corresponds to a CFL number of 1.67.

Results of a grid convergence study are presented in Table 3, which shows the L^∞ norm of the difference between calculated and exact solutions for different resolutions. The convergence rate, computed using two successive

Table 3: L^∞ norms of solution error in uniform-advection problem of section 4.3 in 3D cubed-sphere geometry with each block being divided into $N \times N \times N$ cells.

N	32	64	128	256
time steps	160	320	640	1280
error	3.58e-01	5.85e-02	3.78e-03	2.41e-04
rate	2.62	3.95	3.97	

resolutions, is clearly trending to fourth order.

5. Conclusions and further research

Many problems of practical interest are defined on domains that require meshes more complex than can be achieved with a single, mapped rectangular grid. Mapped multiblock techniques allow the generalization of single mapped rectangular grids while retaining the efficiency of the regular grid indexing. This work extends the high-order mapped finite-volume formulation of [10] by providing an accurate algorithm for communicating between blocks while preserving the requirement of differentiable mappings on a block. Each block maintains its own mapping that extends smoothly off the block into a halo of ghost cells. These ghost cells are filled through a least-squares interpolation from cells on the interior of neighboring blocks. A nearest-neighbors algorithm is used to select the cells from which to construct the interpolation. Grid convergence studies of advection on three different 2D and 3D mapped multiblock domains demonstrate that the expected fourth-order convergence rate is achieved. In cases of characteristic curves lying along block boundaries, we expect the loss of one order of accuracy, but we do not find that to occur in our tests.

Whereas the existence of smooth block mappings with smooth extensions through block boundaries is clear in many cases, such as the disk and sphere examples in Section 4, the specification of suitable mappings in other instances may require additional effort due to application-specific features and constraints. Our use of the methodology described herein to solve the edge-plasma problem in the geometry of Figure 1 (extending the single-block mapped finite-volume discretization of [12] and motivating Example 4.2) provides, in fact, an interesting case study demonstrating both the challenge of a general solution of the mapping generation problem as well as one possible

approach. To accommodate strong anisotropy in the edge-plasma application, the block mappings (for which the analytic mappings (24) serve as simplified proxies in Example 4.2) are aligned with magnetic field lines except in a vicinity of the X-Point, where field-aligned mappings become singular but also where field-induced anisotropy is of less concern. Using the magnetic field as input to a strategy similar to that employed in [7], we obtain block mappings by solving an optimization problem involving a functional that penalizes large first derivatives (non-smoothness) and deviation from field alignment away from the X-Point. The solution of the resulting discrete Euler equation system in each block is then smoothly interpolated to obtain continuous mappings with the desired degree of smoothness. For example, sixth-order B-splines can be employed to obtain the four continuous derivatives needed for a fourth-order discretization, or a spectral interpolation may be considered for additional smoothness. The specific choice of interpolant is based on its suitability for subsequent smooth extensions beyond block boundaries. A more detailed description of the implementation of this approach for the edge-plasma application is planned for a forthcoming paper.

Further work in [26] applies the methods of this paper to the surface of a sphere, which is a 2D manifold in a 3D space, and hence calculations must be consistent with its metric structure. The work in [26] also incorporates adaptive mesh refinement. We are also extending the fourth-order single mapped grid algorithm for elliptic problems in [10] to the mapped multiblock case. The extension to a fourth-order method on multiblock grids would require a fifth-order accurate interpolation method for the ghost cells. While the derivation of such a method using the least-squares approach described here is routine, it is nontrivial to verify that the resulting discrete operator has no eigenvalues in the right half-plane.

We have not addressed what, if any, extensions are required to handle the extensions to the mapped-multiblock case to the case of systems of equations with discontinuities. These issues have been addressed for the single mapped-coordinate case in [14]. We also note that many of the methods for interpolation of ghost cells at refinement boundaries in adaptive mesh refinement do not use limiting for that process (e.g. [1, 25, 14]) while still leading to robust simulations of shocks. The limiters and other dissipation mechanisms used in computing the fluxes from the ghost cell data are sufficient to obtain robust calculations of discontinuities, and it is likely that the same will be the case for interpolating ghost cells at block boundaries.

6. Acknowledgements

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Berkeley National Laboratory under Contract DE-AC02-05CH11231 and by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

References

[1] M. Adams, P. Colella, D. T. Graves, J. N. Johnson, H. S. Johansen, N. D. Keen, T. J. Ligocki, D. F. Martin, P. W. McCorquodale, D. Modiano, P. O. Schwartz, T. D. Sternberg, and B. Van Straalen. Chombo Software Package for AMR Applications - Design Document. Technical Report LBNL-6616E, Lawrence Berkeley National Laboratory, 2014.

[2] T. J. Baker. Developments and trends in three-dimensional mesh generation. In *Transonic Symposium: Theory, Application, and Experiment*, volume 1, pages 341–376. NASA Langley Research Center, 1989.

[3] T. J. Barth. A 3-D upwind Euler solver for unstructured meshes. In *10th Computational Fluid Dynamics Conference*, Fluid Dynamics and Co-located Conferences. American Institute of Aeronautics and Astronautics, June 1991. AIAA Paper 1991-1548.

[4] T. J. Barth. Recent developments in high order k-exact reconstruction on unstructured meshes. In *31st AIAA Aerospace Sciences Meeting and Exhibit*. American Institute of Aeronautics and Astronautics, January 1993. AIAA Paper 1993-0668.

[5] M. J. Berger. Stability of interfaces with mesh refinement. *Math. Comp.*, 45(172):301–318, 1985.

[6] M. J. Berger. On conservation at grid interfaces. *SIAM J. Numer. Anal.*, 24(5):967–984, 1987.

[7] J. U. Brackbill and J. S. Saltzman. Adaptive zoning for singular problems in two dimensions. *J. Comput. Phys.*, 46:342–368, 1982.

[8] M. R. J. Charest and C. P. T. Groth. A high-order central ENO finite-volume scheme for three-dimensional turbulent reactive flows on unstructured mesh. In *21st AIAA Computational Fluid Dynamics Conference*. American Institute of Aeronautics and Astronautics, June 2013. AIAA Paper 2013-2567.

[9] G. Chesshire and W. Henshaw. A scheme for conservative interpolation on overlapping grids. *SIAM J. Sci. Comput.*, 15(4):819–845, 1994.

[10] P. Colella, M. R. Dorr, J. A. F. Hittinger, and D. F. Martin. High-order, finite-volume methods in mapped coordinates. *J. Comput. Phys.*, 230:2952–2976, 2011.

[11] P. Diener, E. N. Dorband, E. Schnetter, and M. Tiglio. Optimized high-order derivative and dissipation operators satisfying summation by parts, and applications in three-dimensional multi-block evolutions. *J. Comput. Phys.*, 32(1):109–145, July 2007.

[12] M. R. Dorr, R. H. Cohen, P. Colella, M. A. Dorf, J. A. F. Hittinger, and D. F. Martin. Numerical simulation of phase space advection in gyrokinetic models of fusion plasmas. In *Proceedings of the 2010 Scientific Discovery through Advanced Computing (SciDAC) Conference*, pages 42–52, Chattanooga, Tennessee, July 2010. Oak Ridge National Laboratory. `http://computing.ornl.gov/workshops/scidac2010/`.

[13] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, fourth edition, 2013.

[14] S. M. Guzik, P. McCorquodale, and P. Colella. A freestream-preserving high-order finite-volume method for mapped grids with adaptive mesh refinement. In *50th AIAA Aerospace Sciences Meeting*. American Institute of Aeronautics and Astronautics, Nashville, TN, January 2012. AIAA Paper 2012-0574.

[15] A. Haselbacher. On constrained reconstruction operators. In *44th AIAA Aerospace Sciences Meeting and Exhibit*. American Institute of Aeronautics and Astronautics, January 2006. AIAA Paper 2006-1274.

[16] L. Ivan and C. P. T. Groth. High-order solution adaptive central essentially non-oscillatory (CENO) method for viscous flows. *J. Comput. Phys.*, 257:830–862, 2013.

[17] L. Ivan, H. De Sterck, S. A. Northrup, and C.P.T.Groth. Multi-dimensional finite-volume scheme for hyperbolic conservation laws on three-dimensional solution-adaptive cubed-sphere grids. *J. Comput. Phys.*, 255:205–227, 2013.

[18] L. Ivan, A. Susanto, H. De Sterck, and C. P. T. Groth. High-order central ENO finite-volume scheme for MHD on three-dimensional cubed-sphere grids. In *Proceedings of the 7th International Conference on Computational Fluid Dynamics*, Hawaii, USA, July 2012. paper ICCFD7-3605.

[19] A. Jalali and C. Ollivier-Gooch. Higher-order finite volume solution reconstruction on highly anisotropic meshes. In *21st AIAA Computational Fluid Dynamics Conference*. American Institute of Aeronautics and Astronautics, June 2013. AIAA Paper 2013-2565.

[20] E. P. C. Koh, H.-M. Tsai, and F. Liu. Euler solution using Cartesian grid with a gridless least-squares boundary treatment. *AIAA J.*, 43(2):246–255, February 2005.

[21] B. Lombard and J. Piraux. Numerical treatment of two-dimensional interfaces for acoustic and elastic waves. *J. Comput. Phys.*, 195(1):90–116, 2004.

[22] R. Magnus and H. Yoshihara. Inviscid transonic flow over airfoils. *AIAA Journal*, 8(12):2157–2162, 1970.

[23] J.C. Mandal and S. P. Rao. High resolution finite volume computations on unstructured grids using solution dependent weighted least squares gradients. *Comput. Fluids*, 44(1):23–31, 2011.

[24] D. Mavriplis. Revisiting the least-squares procedure for gradient reconstruction on unstructured meshes. In *16th AIAA Computational Fluid Dynamics Conference*, Fluid Dynamics and Co-located Conferences. American Institute of Aeronautics and Astronautics, June 2003. AIAA Paper 2003-3986.

[25] P. McCorquodale and P. Colella. A high-order finite-volume method for conservation laws on locally refined grids. *Comm. App. Math. and Comp. Sci*, 6:1–26, 2011.

[26] P. McCorquodale, P. A. Ullrich, H. Johansen, and P. Colella. An adaptive multiblock high-order finite-volume method for solving the shallow-water equations on the sphere. submitted to *Comm. App. Math. and Comp. Sci*, 2014.

[27] A. Nejat and C. F. Ollivier-Gooch. A high-order accurate unstructured finite volume Newton–Krylov algorithm for inviscid compressible flows. *J. Comput. Phys.*, 227(4):2582–2609, 2008.

[28] F. Olsson and N.A. Petersson. Stability of interpolation on overlapping grids. *Comput. Fluids*, 25(6):583–605, 1996.

[29] E. Pärt-Enander and B. Sjögreen. Conservative and non-conservative interpolation between overlapping grids for finite volume solutions of hyperbolic problems. *Comput. Fluids*, 23(3):551–574, 1994.

[30] M. M. Rai. A conservative treatment of zonal boundaries for Euler equation calculations. *J. Comput. Phys.*, 62(2):472–503, 1986.

[31] C. Ronchi, R. Iacono, and P. S. Paolucci. The "cubed sphere": A new method for the solution of partial differential equations in spherical geometry. *J. Comput. Phys.*, 124(1):93–114, 1996.

[32] S. E. Sherer and J. N. Scott. High-order compact finite-difference methods on general overset grids. *J. Comput. Phys.*, 210(2):459–496, 2005.

[33] W. M. Stacey. *Fusion Plasma Physics*. John Wiley & Sons, Weinheim, 2012. ISBN: 978-3-527-41134-4.

[34] G. Starius. Composite mesh difference methods for elliptic and boundary value problems. *Numer. Math.*, 28:243–258, 1977.

[35] J. L. Steger and J. A. Benek. On the use of composite grid schemes in computational aerodynamics. *Comput. Meth. App. Mech. Engin.*, 64(13):301–320, 1987.

[36] A. Susanto, L. Ivan, H. De Sterck, and C. P. T. Groth. High-order central ENO finite-volume scheme for ideal MHD. *J. Comput. Phys.*, 250:141–164, 2013.

[37] J. F. Thompson, Z. U. A. Warsi, and C. W. Mastin. Boundary-fitted coordinate systems for numerical solution of partial differential equations — A review. *J. Comput. Phys.*, 47(1):1–108, 1982.

[38] L. Tobaldini Neto and C. P. T. Groth. A high-order finite-volume scheme for large-eddy simulation of turbulent premixed flames. In *52nd AIAA Aerospace Sciences Meeting*. American Institute of Aeronautics and Astronautics, January 2014. AIAA Paper 2014-1024.

[39] P. A. Ullrich, C. Jablonowski, and B. L. van Leer. High-order finite-volume models for the shallow-water equations on the sphere. *J. Comput. Phys.*, 229:6104–6134, 2010.

[40] M. Vinokur. An analysis of finite-difference and finite-volume formulations of conservation laws. *J. Comput. Phys.*, 81(1):1–52, March 1989.

[41] H. Viviand. Formes conservatives des équations de la dynamique des gaz. *La Recherche Aérospatiale*, 158(1):65–66, 1974.

[42] J. Wesson. *Tokamaks*. International Series of Monographs on Physics 149. Oxford University Press, New York, fourth edition, 2011.