

A Freestream-Preserving High-Order Finite-Volume Method for Mapped Grids with Adaptive-Mesh Refinement

Stephen M. Guzik^{*,a}

Lawrence Livermore National Laboratory, Livermore, CA, 94551, USA

Peter McCorquodale^{†,b} and Phillip Colella^{‡,b}

Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA

A fourth-order accurate finite-volume method is presented for solving time-dependent hyperbolic systems of conservation laws on mapped grids that are adaptively refined in space and time. Novel considerations for formulating the semi-discrete system of equations in computational space combined with detailed mechanisms for accommodating the adapting grids ensure that conservation is maintained and that the divergence of a constant vector field is always zero (freestream-preservation property). Advancement in time is achieved with a fourth-order Runge-Kutta method.

I. Introduction

FINITE-volume methods have become well established for discretizing partial differential equations, especially when conservation is paramount. They are suited for problems with discontinuities because the resulting discretization satisfies a discrete form of the divergence theorem, leading to a local conservation property for time-dependent problems.

Recently, there has been much interest in increasing the accuracy of the method to fourth-order and beyond. For smooth flows, the extra computation is more than compensated by the reduction in error. As well, increasing the computation per unit memory makes better use of modern and upcoming computer architectures. In this work, we follow the techniques described by McCorquodale and Colella¹ for achieving fourth-order solutions of time-dependent hyperbolic conservation laws on adaptively refined grids. Adaptive mesh refinement (AMR) allows one to change the resolution of the mesh in response to the characteristics of the solution. It is especially useful for sufficiently resolving large errors in some regions of a problem while avoiding the expense of the increased resolution in regions with lower errors. A method of block-structured AMR is used with grids locally refined in space and time.²

Finite-volume methods employed on Cartesian grids are computationally efficient and additionally benefit from having well-understood characteristics in terms of solution accuracy. To solve more complex geometries, one can resort to applying the finite-volume method directly to structured curvilinear meshes or unstructured meshes, using embedded boundary (cut-cell) techniques, or mapping³ a structured grid in physical space to a Cartesian grid in computational space. The latter approach effectively recovers Cartesian methods with the cost of some additional complexity in terms of grid metrics. The mapped grids approach is the focus of this work and has gained some favor in the aerospace community, no doubt because wings are often easily meshed by this approach.

^{*}Post-Doctoral Researcher, Center for Applied Scientific Computing, Email: guzik2@llnl.gov, Member AIAA

[†]Scientist, Applied Numerical Algorithms Group

[‡]Senior Scientist, Applied Numerical Algorithms Group, Senior Member AIAA

^aThis work was performed under the auspices of the U.S. Department of Energy (DOE) by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344 and by DOE contracts from the ASCR Applied Math Program.

^bResearch at the Lawrence Berkeley National Laboratory supported by the Office of Advanced Scientific Computing Research of the US Department of Energy under contract number DE-AC02-05CH11231. LLNL-CONF-520651

However, the addition of the metric terms can introduce subtle difficulties. A constant solution in physical space may appear with gradients when mapped to computational space. It is important to design methods that guarantee freestream preservation, a property that ensures a uniform flow is not affected by the choice of mapping and discretization. A high-order method has been developed by Colella et al.⁴ that retains the freestream preservation property at any order of accuracy on mapped grids.

In this work, we merge the fourth-order AMR work of McCorquodale and Colella¹ for Cartesian grids with the freestream-preserving technology on mapped grids of Colella et al.⁴ To maintain conserved quantities as the grid levels appear, disappear, and migrate within the computational domain following solution features, we adopt the procedure described by Bell et al.⁵ In that procedure, special adjustments, also described herein, precede a modification of the grid hierarchy to ensure that the transfer of solution information between grid levels is conservative. The combination of mapped grids with AMR requires additional treatment to ensure that the freestream-preservation property is retained.

In the following sections, we first present the high-order finite-volume method designed for mapped grids followed by the additional accommodations necessary for AMR. In the section on results, we verify that both freestream preservation and conservation are not violated and demonstrate the technology on an unsteady problem featuring shock reflections.

II. Fourth-Order Finite-Volume Method on Mapped Grids

Previous related efforts involved finding a method for ensuring freestream preservation on mapped grids⁴ and designing a fourth-order accurate finite-volume method for solving time-dependent hyperbolic systems of conservation laws on Cartesian grids with multiple levels of refinement.¹ Important details from these previous works are presented in this section. We consider time-dependent solutions to hyperbolic systems of conservation laws having the general form

$$\frac{\partial \mathbf{U}}{\partial t} + \nabla \cdot \vec{\mathbf{F}}(\mathbf{U}) = 0. \quad (1)$$

The results we show are all solutions of the Euler equations, supplemented with the ideal gas law where a perfect gas is assumed with a specific heat ratio of $\gamma = 1.4$.

II.A. Metrics Terms

We assume that we have a smooth mapping \mathbf{X} from some abstract coordinate space into physical space:

$$\mathbf{X} = \mathbf{X}(\boldsymbol{\xi}), \quad \mathbf{X} : [0, 1]^D \rightarrow \mathbb{R}^D.$$

Given this mapping, the divergence of a vector field in physical space can be written in terms of derivatives in the mapping space,

$$\nabla_{\mathbf{X}} \cdot \vec{f} = \frac{1}{J} \nabla_{\boldsymbol{\xi}} \cdot (\mathbf{N}^T \vec{f}), \quad (2)$$

$$J \equiv \det(\nabla_{\boldsymbol{\xi}} \mathbf{X}), \quad \mathbf{N}_{d,s}^T \equiv \det((\nabla_{\boldsymbol{\xi}} \mathbf{X})^T (d|e^s)), \quad (3)$$

where e^d is the unit vector in direction d and $\mathbf{A}(p|\mathbf{v})$ denotes a modification of matrix \mathbf{A} by replacing row p with vector \mathbf{v} . The relationship (2) is a consequence of the chain rule, equality of mixed partials, and Cramer's rule.⁴

II.B. Finite-Volume

The problem domain is discretized using a grid $\Gamma \subset \mathbb{Z}^D$ that is a bounded subset of the lattice defined by the points $(i_0, \dots, i_{D-1}) = \mathbf{i} \in \mathbb{Z}^D$ marking the cell centers of the control volumes. For Cartesian grid finite-volume methods, the control volumes take the form

$$V_{\mathbf{i}} = [\mathbf{x}_0 + (\mathbf{i} - \frac{1}{2}\mathbf{u})h, \mathbf{x}_0 + (\mathbf{i} + \frac{1}{2}\mathbf{u})h], \quad (4)$$

where $\mathbf{x}_0 \in \mathbb{R}^D$ is some fixed origin of coordinates, h is the mesh spacing, and $\mathbf{u} \in \mathbb{Z}^D$ is the vector whose components are all equal to one.

In the finite-volume method, the integral form of (1) is applied to the control volumes in physical space,

$$\frac{\partial}{\partial t} \int_{\mathbf{x}(V_i)} \mathbf{U} d\mathbf{x} + \int_{\mathbf{x}(V_i)} \nabla \cdot \vec{\mathbf{F}} d\mathbf{x} = 0. \quad (5)$$

The switch to computation space is made using the metric terms such that

$$\frac{\partial}{\partial t} \int_{V_i} J\mathbf{U} d\xi + \int_{V_i} \nabla \cdot \mathbf{N}^T \vec{\mathbf{F}} d\xi = 0. \quad (6)$$

Application of the divergence theorem of Gauss results in

$$\frac{\partial}{\partial t} \int_{V_i} J\mathbf{U} d\xi + \int_{\partial V_i} \mathbf{N}^T \vec{\mathbf{F}} \cdot \hat{n} dS = 0. \quad (7)$$

The integrals can be represented as averages yielding

$$h^D \frac{d}{dt} \langle J\mathbf{U} \rangle_i + h^{D-1} \sum_{d=1}^D \left(\langle \mathbf{N}_d^T \vec{\mathbf{F}} \rangle_{i+\frac{1}{2}\mathbf{e}^d} - \langle \mathbf{N}_d^T \vec{\mathbf{F}} \rangle_{i-\frac{1}{2}\mathbf{e}^d} \right) = 0, \quad (8)$$

$$\frac{d}{dt} \langle J\mathbf{U} \rangle_i + \frac{1}{h} \sum_{d=1}^D \left(\langle \mathbf{N}_d^T \vec{\mathbf{F}} \rangle_{i+\frac{1}{2}\mathbf{e}^d} - \langle \mathbf{N}_d^T \vec{\mathbf{F}} \rangle_{i-\frac{1}{2}\mathbf{e}^d} \right) = 0. \quad (9)$$

Assuming a fourth-order estimate of $\langle \vec{\mathbf{F}} \rangle$ is known (see section II.C), our approach to determining fourth-order accurate averages of $\langle \mathbf{N}_d^T \vec{\mathbf{F}} \rangle$ on the faces is to use a product rule of the form

$$\langle uv \rangle_{i+\frac{1}{2}\mathbf{e}^d} = \langle u \rangle_{i+\frac{1}{2}\mathbf{e}^d} \langle v \rangle_{i+\frac{1}{2}\mathbf{e}^d} + \frac{h^2}{12} \sum_{d' \neq d} \frac{\partial u}{\partial \xi_{d'}} \frac{\partial v}{\partial \xi_{d'}} + O(h^4), \quad (10)$$

which is valid on Cartesian grids. For a single solution variable, the averages in equation (9) can be obtained from

$$\langle \mathbf{N}_d^T \vec{\mathbf{F}} \rangle_{i\pm\frac{1}{2}\mathbf{e}^d} = \sum_{s=1}^D \langle \mathbf{N}_{d,s}^T \rangle_{i\pm\frac{1}{2}\mathbf{e}^d} \langle \vec{\mathbf{F}}_s \rangle_{i\pm\frac{1}{2}\mathbf{e}^d} + \frac{h^2}{12} \sum_{s=1}^D \sum_{d' \neq d} \frac{\partial \mathbf{N}_{d,s}^T}{\partial \xi_{d'}} \frac{\partial \vec{\mathbf{F}}_s}{\partial \xi_{d'}} + O(h^4). \quad (11)$$

Some care is required to obtain freestream preservation, the property that the discrete divergence of a constant vector field is zero. If the flux, $\vec{\mathbf{F}}$, in (11) is constant, then the second term vanishes and we only need to derive quadrature formulas for $\langle \mathbf{N}^T \rangle$ so that the discrete divergence of a constant vector field given by (11) is zero.

The existence of such quadratures is a consequence of Stokes' theorem and the Poincaré lemma. The columns of the matrix \mathbf{N}^T (or rows of \mathbf{N} , in the following denoted by \mathbf{N}_s) are divergence-free as can be seen by a direct calculation. Then, by the Poincaré lemma,⁶ there must exist functions, $\mathcal{N}_{d,d'}^s$, $d \neq d'$; in three dimensions, this is described by the vector identity

$$\nabla \cdot \left(\underbrace{\nabla \times \vec{\mathcal{N}}^s}_{\mathbf{N}_s} \right) = 0. \quad (12)$$

Next, Stokes' theorem,

$$\iint_{A_d} (\nabla \times \vec{\mathcal{N}}^s) \cdot \vec{n} dA = \oint_{E_{d,d' \neq d}} \vec{\mathcal{N}}^s \cdot \vec{r}'(E) dE, \quad (13)$$

with \vec{r}' being a right-hand unit tangent vector to edge E , is used to evaluate the integral of \mathbf{N}_s on the face by instead integrating $\vec{\mathcal{N}}^s$ over the edges. For a general number of dimensions, we find that the cross product in (12) can be represented as

$$\mathbf{N}_{d,s}^T = \sum_{d' \neq d} \frac{\partial \mathcal{N}_{d,d'}^s}{\partial \xi_{d'}} \quad \text{with} \quad \mathcal{N}_{d,d'}^s = -\mathcal{N}_{d',d}^s. \quad (14)$$

By Stokes' theorem, the integral of (14) over face A_d is given by

$$h^{D-1} \langle \mathbf{N}_{d,s}^T \rangle = \int_{A_d} \mathbf{N}_{d,s}^T dA_{\boldsymbol{\xi}} = \int_{A_d} \sum_{d' \neq d} \frac{\partial \mathcal{N}_{d,d'}^s}{\partial \boldsymbol{\xi}_{d'}} dA_{\boldsymbol{\xi}} = \oint_{E_{d,d' \neq d}} \mathcal{N}_{d,d'}^s dE_{\boldsymbol{\xi}}, \quad (15)$$

where $E_{d' \neq d} = \partial A_d$ are the (hyper)edges of face A_d . This can be written as

$$\langle \mathbf{N}_{d,s}^T \rangle = \frac{1}{h^{D-1}} \sum_{\pm=+,-} \sum_{d' \neq d} \pm \int_{E_{d,d'}^{\pm}} \mathcal{N}_{d,d'}^s dE_{\boldsymbol{\xi}}, \quad (16)$$

with fourth-order quadratures replacing the integrals. Equations (15) and (16) are the extension of (13) to any number of dimensions and applied on a Cartesian grid with spacing h . Because (16) only involves terms normal to the face (consider the left hand side of (13)), only the normal components of $\langle \mathbf{N}^T \rangle$ can be determined using (16). However, this is all that is required for hyperbolic conservation laws. If ever required, transverse components of $\langle \mathbf{N}^T \rangle$ can be computed directly or averaged from the normal components of $\langle \mathbf{N}^T \rangle$ on nearby faces in orthogonal directions.

For each edge, the same integrals of \mathcal{N}^s over the edge appear for the integral over each face adjacent to that edge but with opposite signs. Therefore, the integration of $\langle \mathbf{N}^T \rangle$ over the complete cell volume is zero as long as the integrals of \mathcal{N}^s are approximated with the same quadrature formulas wherever they appear.

The family of functions, $\mathcal{N}_{d,d'}^s, d' \neq d$, satisfying (14) is not unique. The form of $\mathcal{N}_{d,d'}^s$ used herein is given explicitly by

$$\mathcal{N}_{d,d'}^s = \frac{1}{D-1} \det((\nabla_{\boldsymbol{\xi}} \mathbf{X})^T (d | \mathbf{e}^s) (d' | \mathbf{X})). \quad (17)$$

II.C. Discretization

The method for determining $\langle \vec{\mathbf{F}} \rangle$ and evolving the solution is the same as that described by McCorquodale and Colella¹ for solving time-dependent hyperbolic systems of conservation laws on Cartesian grids with multiple levels of refinement. A centered spatial interpolation is used to obtain $\langle \vec{\mathbf{F}} \rangle$ to fourth-order accuracy on the faces. Slope limiting (a variant of the limiter proposed by Colella and Sekora⁷), slope flattening, and artificial viscosity are all applied to stabilize the method and minimize oscillations near large gradients. To evolve the solution in time, a fourth-order Runge-Kutta method is employed.

III. Adaptive Mesh Refinement on Mapped Grids

To implement adaptive mesh refinement, we make use of the Chombo library for parallel AMR⁸ and follow the strategies used therein. Adaptive mesh refinement calculations are performed on a hierarchy of nested meshes $\Omega^\ell \subset \Gamma^\ell$, with $\Omega^\ell \supset \mathcal{C}_{n_{ref}^\ell}(\Omega^{\ell+1})$ where n_{ref}^ℓ denotes the refinement ratio between levels ℓ and $\ell+1$ and \mathcal{C} is a coarsening operator. The grid levels are considered as overlapping rather than embedded. An example grid hierarchy is shown in Figure 1. At level ℓ , we label all cells inside Ω^ℓ as being valid and all cells outside Ω^ℓ (such as ghost cells) as being invalid. We assume that there are a sufficient number of cells on level ℓ separating the level $\ell+1$ cells from the level $\ell-1$ cells such that interpolations to fill invalid ghost cells on finer levels can be independently performed. We will refer to grid hierarchies that meet this condition as being *properly nested*. Typically, Ω^ℓ is decomposed into a disjoint union of rectangles (boxes) in order to perform calculations in parallel. Any relationships between boxes on the same level, or between different levels, are known simply through the vectors describing the corner locations on the integer lattice. Consequently, there is no need for tracking connectivity between boxes (although data-motion patterns are cached when the grids change for better efficiency).

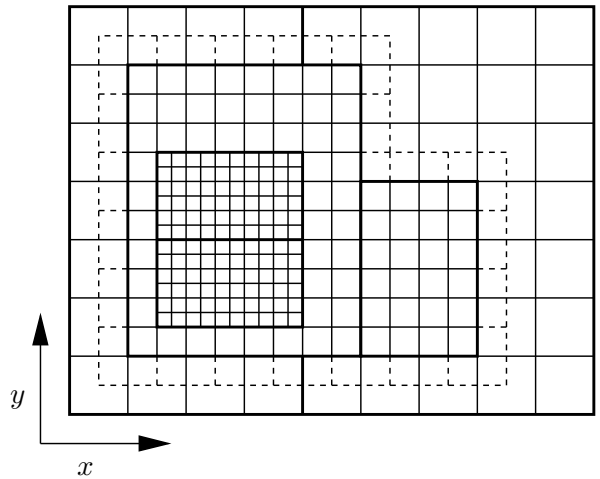


Figure 1. A three-level grid with $n_{ref} = 2$ and nesting sufficient for one cell to separate level $\ell+1$ from level $\ell-1$. A single layer of invalid ghost cells surrounding the middle grid level is shown by dashed lines.

The typical work-flow for advancing level ℓ is:

1. Regrid levels finer than ℓ if required. This involves tagging all cells which should compose the finer levels, often based on the magnitudes of solution gradients, and constructing a new, properly nested mesh hierarchy. In regions where new fine cells appear, the solution is interpolated from the coarser level. Additional considerations for mapped grids are described in the next section.
2. Advance level ℓ . The solution state is first computed using a cell-centered product rule,

$$\langle \mathbf{U} \rangle = \frac{1}{J} \left(\langle J \mathbf{U} \rangle - \frac{h^2}{12} \sum_d^D \frac{\partial \mathbf{U}}{\partial \xi_d} \frac{\partial J}{\partial \xi_d} \right) \quad (18)$$

before using the methods described in Section II.C to find $\langle \vec{\mathbf{F}} \rangle$, compute $\langle \mathbf{N}_d^T \vec{\mathbf{F}} \rangle$ using (11), and evolve the semi-discrete form (9) with a Runge-Kutta time-stepping method.

3. Interpolate to the invalid ghost cells surrounding level $\ell + 1$. A least-squares algorithm is used to compute the interpolating polynomial in each coarse cell. The interpolation need not be conservative because the resulting values in the ghost cells are only used to reconstruct the flux on the faces of the valid cells.
4. Start level $\ell + 1$ at step 1. Level $\ell + 1$ is refined in time (sub-cycled) with a time step $\Delta t^{\ell+1} = \Delta t^\ell / n_{ref}^\ell$.
5. Average the solution from $\ell + 1$ and correct fluxes at coarse-fine interfaces to ensure conservation.^{2,9}

Even with refined levels, the solution will be freestream-preserving as long as \mathcal{N}^s is made consistent between the (hyper)edges that are shared between the two levels, and $\langle \mathbf{N}^T \rangle$ is computed from that consistent value. This concept is shown in Figure 2 where $\langle \mathbf{N}_{d=1}^T \rangle^\ell$ and $\langle \mathbf{N}_{d=2}^T \rangle^\ell$ are computed using the same $\mathcal{N}^{s,\ell}$. However, on faces in Ω_{valid}^ℓ , adjacent to and orthogonal to the exterior of $\Omega^{\ell+1}$ (e.g., the face where $\langle \mathbf{N}_{d=2}^T \rangle^\ell$ is indicated in Figure 2), there is a one-order loss of accuracy due to a lack of error cancellation that otherwise occurs when the error in the quadratures approximating \mathcal{N}^s are smoothly varying functions of space. The cancellation is lost because \mathcal{N}^s is overwritten with sums from the finer level on all of the edges within the interface. For this reason, it is necessary to have \mathcal{N}^s computed to $O(h^{p+1})$ accuracy in order for the $\langle \mathbf{N}^T \rangle$ to be accurate to $O(h^p)$ everywhere.

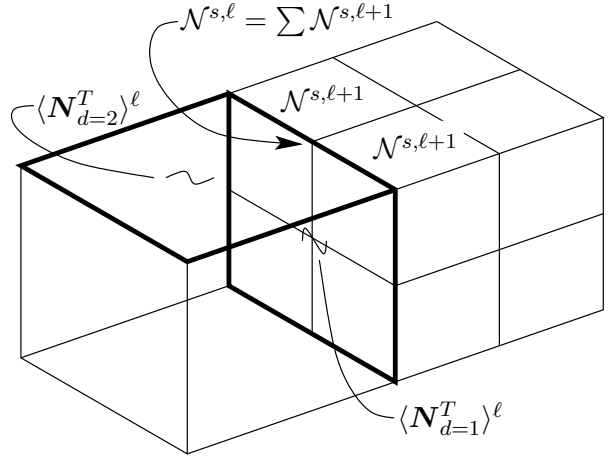


Figure 2. By overwriting \mathcal{N}^s with a sum of fine-grid values, we ensure that $\langle \mathbf{N}_{d=1}^T \rangle^\ell = (\sum \langle \mathbf{N}_{d=1}^T \rangle^{\ell+1}) / (n_{ref}^\ell)^{D-1}$ on the coarse face (shown in bold) that overlaps the fine faces. Additionally, since this single value of \mathcal{N}^s is used to compute both $\langle \mathbf{N}_{d=1}^T \rangle^\ell$ and $\langle \mathbf{N}_{d=2}^T \rangle^\ell$, the scheme is still freestream-preserving.

III.A. Regriding

Periodically, it is necessary to change the grid hierarchy in response to changes in the solution. During a regrid, we generate a new grid hierarchy, $\{\Omega^{\ell,new}\}_{\ell=\ell_{base}+1,\dots,\ell_{max}}$ leaving the mesh at ℓ_{base} and all coarser levels unchanged. We also will use the old grid information, which we will denote by the superscript *old*. This process is significantly more complicated than in the Cartesian grid case, since the geometry of the mapped grid cells changes depending on the extent to which they are covered by finer grids. In particular, the ratio between the volumes of a cell in physical and computational space may change according to the finest resolution at which the geometry is prescribed (the finer specifications are averaged down to coarser levels and can change the coarser cells). We organize the regrid process into three steps:

1. Generate the geometric information for the new grid hierarchy.
2. Adjust the values of the solution on the old hierarchy to be consistent with the new grid geometry. This step is critical for ensuring conservation of the solution.
3. Construct the solution on the new grid hierarchy.

Calculation of the new geometry proceeds from the finest level to ℓ_{base} and begins with the integration of \mathcal{N}^s , using (17), on all codimension two (hyper-edge) centerings of Ω^ℓ . We assume in this work that a single

analytic expression of the mapping applies everywhere in the domain, allowing \mathcal{N}^s to be easily computed at any resolution. Next, the \mathcal{N}^s from a finer level, if it exists, are summed to this level. The normal components of $\langle \mathbf{N}^T \rangle$ are computed using (16). We note that a fourth-order accurate approximation to the cell volumes can be determined from our usual divergence formula by instead letting $\vec{\mathbf{F}}(\mathbf{x}) = \mathbf{x}/D$. In that case,

$$\begin{aligned} \frac{1}{h^D} \int_{\mathbf{X}(V_i)} d\mathbf{x} &= \langle J \rangle_i \\ &= \frac{1}{h^D} \int_{V_i} \nabla_{\boldsymbol{\xi}} \cdot (\mathbf{N}^T \tilde{\mathbf{X}}) d\boldsymbol{\xi} \\ &= \frac{1}{h} \sum_{d=1}^D \langle \mathbf{N}_d^T \tilde{\mathbf{X}} \rangle_{i+\frac{1}{2}\mathbf{e}^d} - \langle \mathbf{N}_d^T \tilde{\mathbf{X}} \rangle_{i-\frac{1}{2}\mathbf{e}^d} \end{aligned} \quad (19)$$

where $\tilde{\mathbf{X}}(\boldsymbol{\xi}) = \mathbf{X}(\boldsymbol{\xi})/D$. Continuing with the geometry calculation, $\langle \mathbf{N}_d^T \tilde{\mathbf{X}} \rangle_{i+\frac{1}{2}\mathbf{e}^d}^{\ell, \text{new}}$, which we label as a volume flux, is computed using a product rule and replaced by averages from finer levels where there are overlapping grids. Finally, $\langle J \rangle$ is computed using (19). Because \mathcal{N}^s and $\langle \mathbf{N}_d^T \tilde{\mathbf{X}} \rangle$ are overwritten from finer levels, the metrics $\langle \mathbf{N}^T \rangle$ and $\langle J \rangle$ are made consistent between all the levels of the grid. For example, in Figure 2, $\langle \mathbf{N}_1^T \rangle^\ell = (\sum \langle \mathbf{N}_1^T \rangle^{\ell+1}) / (n_{ref}^\ell)^{\mathbf{D}-1}$ on the interface as a result of overwriting $\mathcal{N}^{s,\ell}$ with $\sum \mathcal{N}^{s,\ell+1}$ on the edges of that face.

The second step is to modify the cell volumes and the conserved quantities on the old grid hierarchy so that they are consistent with the geometry of the new grid hierarchy, while still maintaining conservation. At level ℓ , this is done by computing on the faces of Ω^ℓ the changes in the contributions to the volume fluxes corresponding to the changes in the grid geometry, and fluxing the conserved quantities in response to those changes in the volumes. These fluxes are labeled *snapback* fluxes and express changes due to regridding at finer levels. On $\Omega^{\ell, \text{old}}$, define the snapback volume and solution flux as:

$$F_{\delta J, i+\frac{1}{2}\mathbf{e}^d} = \begin{cases} \langle \mathbf{N}^T \tilde{\mathbf{X}} \rangle^{\ell, \text{new}} - \langle \mathbf{N}^T \tilde{\mathbf{X}} \rangle^{\ell, \text{old}} & \text{on the faces of } \Omega^{\ell, \text{old}} \cap \Omega^{\ell, \text{new}}, \\ 0 & \text{otherwise.} \end{cases} \quad (20)$$

$$F_{\delta(J\mathbf{U}), i+\frac{1}{2}\mathbf{e}^d}^d = \begin{cases} -\mathbf{U}_{i+\mathbf{e}^d} F_{\delta J, i+\frac{1}{2}\mathbf{e}^d}^d & \text{if } F_{\delta J, i+\frac{1}{2}\mathbf{e}^d}^d > 0, \\ -\mathbf{U}_i F_{\delta J, i+\frac{1}{2}\mathbf{e}^d}^d & \text{otherwise.} \end{cases} \quad (21)$$

The snapback solution flux is replaced by averages from finer levels. Now the conserved quantities $\langle J\mathbf{U} \rangle^{\ell, \text{old}}$ are corrected to the new mesh as follows,

$$\langle J\mathbf{U} \rangle^{\ell, \text{old}} := \langle J\mathbf{U} \rangle^{\ell, \text{old}} - \frac{1}{h} \sum_{d=1}^D F_{\delta(J\mathbf{U}), i+\frac{1}{2}\mathbf{e}^d}^d - F_{\delta(J\mathbf{U}), i-\frac{1}{2}\mathbf{e}^d}^d. \quad (22)$$

Once we have realigned the values on all of the levels to be consistent with the new geometry, we then can proceed to step 3 and interpolate solution values. For $\ell = \ell_{base}, \dots, \ell_{max}^{\text{new}} - 1$, we use a least-squares algorithm to interpolate solution values on new overlapping fine cells. We solve a least-squares system for the coefficients of a polynomial interpolant

$$\sum_{\mathbf{p}} a_{\mathbf{p}} \langle \boldsymbol{\xi}^{\mathbf{p}} \rangle_{\mathbf{j}^\ell} = \langle U \rangle_{\mathbf{j}^\ell}, \quad \mathbf{j}^\ell \in \mathcal{I}(\mathbf{i}^\ell),$$

where \mathcal{I} denotes the stencil around cell \mathbf{i}^ℓ , subject to the constraint

$$\sum_{\mathbf{i}^{\ell+1} \in \mathcal{C}^{-1}\{\mathbf{i}^\ell\}} \sum_{\mathbf{p}} a_{\mathbf{p}} \langle J\boldsymbol{\xi}^{\mathbf{p}} \rangle_{\mathbf{i}^{\ell+1}} = \langle J\mathbf{U} \rangle_{\mathbf{i}^\ell}.$$

The moments $\langle \boldsymbol{\xi}^{\mathbf{p}} \rangle$ can be computed analytically, and the $\langle J\boldsymbol{\xi}^{\mathbf{p}} \rangle$ are computed using the product formula. Given this interpolant, we can construct $\langle J\mathbf{U} \rangle$ on the control volumes at level $\ell + 1$,

$$\langle J\mathbf{U} \rangle_{\mathbf{i}^{\ell+1}} = \sum_{\mathbf{p}} a_{\mathbf{p}} \langle J\boldsymbol{\xi}^{\mathbf{p}} \rangle_{\mathbf{i}^{\ell+1}}, \quad \mathbf{i}^{\ell+1} \in \mathcal{C}^{-1}(\mathbf{i}^\ell).$$

This interpolation step is both freestream-preserving and conservative.

IV. Results

The results of several numerical experiments on a variety of mappings are presented to demonstrate that we have indeed retained the properties we desire of our scheme, namely fourth-order accuracy, conservation, and freestream preservation.

IV.A. Freestream Preservation

This test is designed to illustrate the freestream preservation properties of the scheme. A three-dimensional mapping,

$$\begin{aligned} x &= r \sin \varphi \cos \theta \\ y &= r \sin \varphi \sin \theta \\ z &= r \cos \varphi \end{aligned} \quad (23)$$

is used to map spherical coordinates, $\boldsymbol{\xi}$, to physical space, \boldsymbol{X} . The ranges of both r and φ are limited to avoid singularities. The solution is not evolved; instead, a region to be refined, 0.1 units in radius, is analytically tagged and rotated about the axis $(1, -1, 0)$ within the unit cube in computational space, starting at position $(\frac{2\sqrt{2}}{5\sqrt{5}}, \frac{2\sqrt{2}}{5\sqrt{5}}, \frac{2\sqrt{2}}{5\sqrt{5}})$. Once the tagged region has completed one revolution, we verify that the constant solution is unchanged. Two levels of refinement are used as shown in Figure 3. In this figure, gaps can be observed between the various levels, illustrating how refinement can affect the volumes of the cells and the necessity of correcting the solution with the *snapback* flux during the regrid process to ensure conservation.

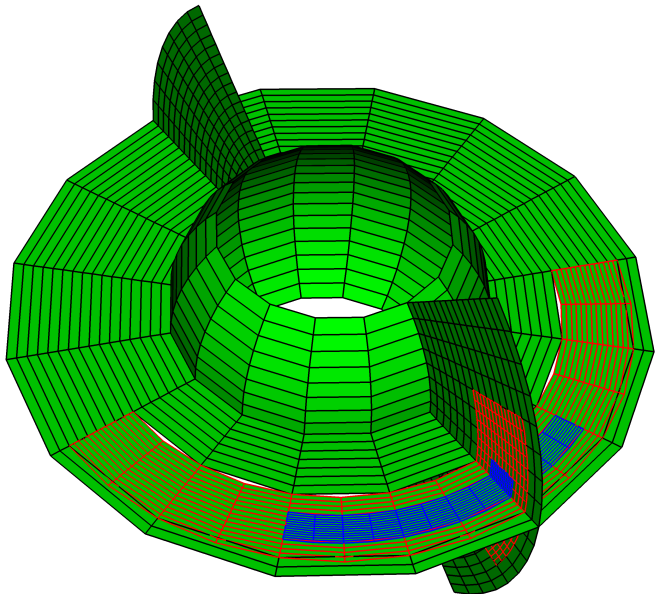


Figure 3. Freestream-preservation test.

IV.B. Accuracy of Mapped Grids with AMR

To test the accuracy of the scheme, we initialize the density with a Gaussian profile, and advect the solution on a two-dimensional periodic domain. The density is initialized by

$$\rho = \rho_0 + s(r) \Delta\rho e^{-(100r^2)} \quad (24)$$

where

$$s(r) = \begin{cases} 0 & : |2r^2| \geq 1 \\ \cos^6(\pi r^2) & : |2r^2| < 1 \end{cases}, \quad (25)$$

$\rho_0 = 1.4$, $\Delta\rho = 0.14$, and r specifies the distance of a point from the center of the periodic domain, $[0, 1]^D$. Smoothing with continuous 5th derivatives is applied with $s(r)$ so that the profile can be forced to zero at the domain boundaries. The pressure is initialized to a constant of 1 and the velocity is set to $(1.0, 0.5)$. After advecting for two time units, the profile recovers its original location and we measure the error against the exact solution.

For this problem, the mesh is deformed according to the mapping

$$\mathbf{x}_d = \boldsymbol{\xi}_d + \prod_{p=1}^D \sin(2\pi \boldsymbol{\xi}_p) \quad d = 1, 2. \quad (26)$$

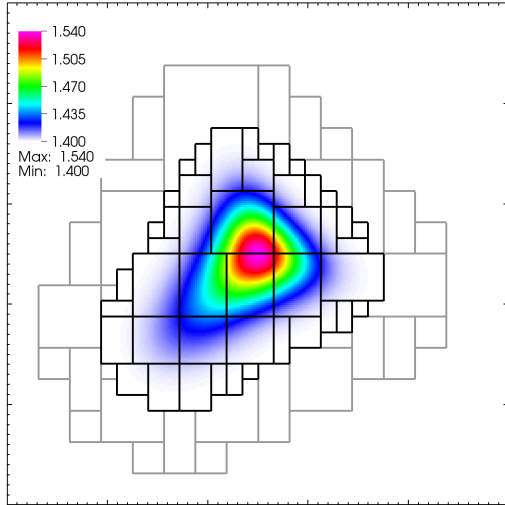


Figure 4. Initialized Gaussian profile in computational space.

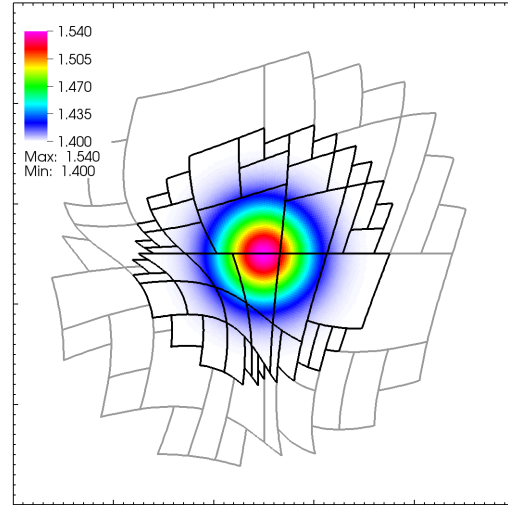


Figure 5. Initialized Gaussian profile in physical space.

The initial solution in both computational and physical space is shown in Figures 4 and 5. In both figures, the level 1 boxes (each containing a number of cells) are shown by gray lines and the level 2 boxes are shown by black lines. In order to obtain accurate convergence rates, the regions of refinement are analytically specified to be a distance of 0.35 from the center of the Gaussian for level 1 and 0.225 from the center of the Gaussian for level 2. Otherwise, fixed-size buffers of cells, associated with a solution-based identification of refinement regions, tend to shrink with successively finer base grids, thereby reducing the actual area covered by the finer grids. Solution error was predicted for single-level grids and AMR grids, with the finest level on an AMR grid matching the resolution of the corresponding single-level grid. The results are shown in Figure 6.

This example illustrates the objectives and compromises associated with AMR. The objective is to resolve the large errors in the problem with the same accuracy while reducing the expense of computing in areas with smaller errors. For this problem, the largest solution error, indicated by the L_∞ error norm, are near the large gradients in the density profile, well within the finest level of refinement. Consequently, the AMR mesh recovers nearly the same L_∞ error norm as the single level mesh. The compromise is that additional errors are introduced at the interfaces between the coarse and fine levels. These errors are a consequence of maintaining conservation and can be explained by modified equation arguments which suggest that up to one-order of accuracy can be lost in coarse cells adjacent to the interfaces (this is the same argument that requires \mathcal{N}^s to be computed to an order of accuracy greater than that of the scheme as discussed in section III). These small errors are further diminished when squared in the L_2 error norm but are quite apparent in the L_1 norm.

With periodic boundaries, this example also provides a good opportunity to test conservation. An integration of all conserved quantities in computational space $\langle JU \rangle$ is performed at the start and end of the run. The double precision results from the 512×512 AMR case are shown in Table 1.

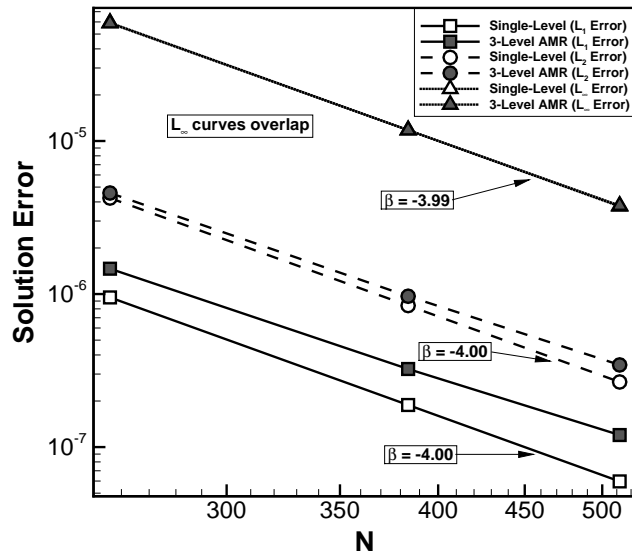


Figure 6. Solution accuracy for advection of a Gaussian profile using both single-level grids and AMR grids

Table 1. Initial and final conserved quantities for advection problem

Component	Initial	Final	Difference
$J\rho$	2.299286119591301e+04	2.299286119591318e+04	0.000000000000017e+04
$J\rho u$	2.299286119591301e+04	2.299286119591317e+04	0.000000000000016e+04
$J\rho v$	1.149643059795651e+04	1.149643059795661e+04	0.000000000000010e+04
$J\rho E$	5.533053824744564e+04	5.533053824744603e+04	0.000000000000039e+04

IV.C. Mach Reflection

The final test case illustrates the effectiveness of AMR in resolving large discontinuities. In this case, an incident shock at Mach 10 reflects off a ramp at 30° producing double Mach stems.¹⁰ A Schwarz-Christoffel mapping¹¹ is used to generate the ramp and the solution is shown in Figure 7 after 0.25 time units. In addition to the shocks, Kelvin-Helmholtz instabilities are quick to develop along the the slip line in the absence of viscosity and the “jet” of fluid behind the first stem forms a mushroom characteristic of a classic Rayleigh-Taylor instability. These features are shown in greater detail in Figure 8. Some artifacts are also present, resulting from specifying an infinitely thin shock for initial conditions and at the upper boundary.

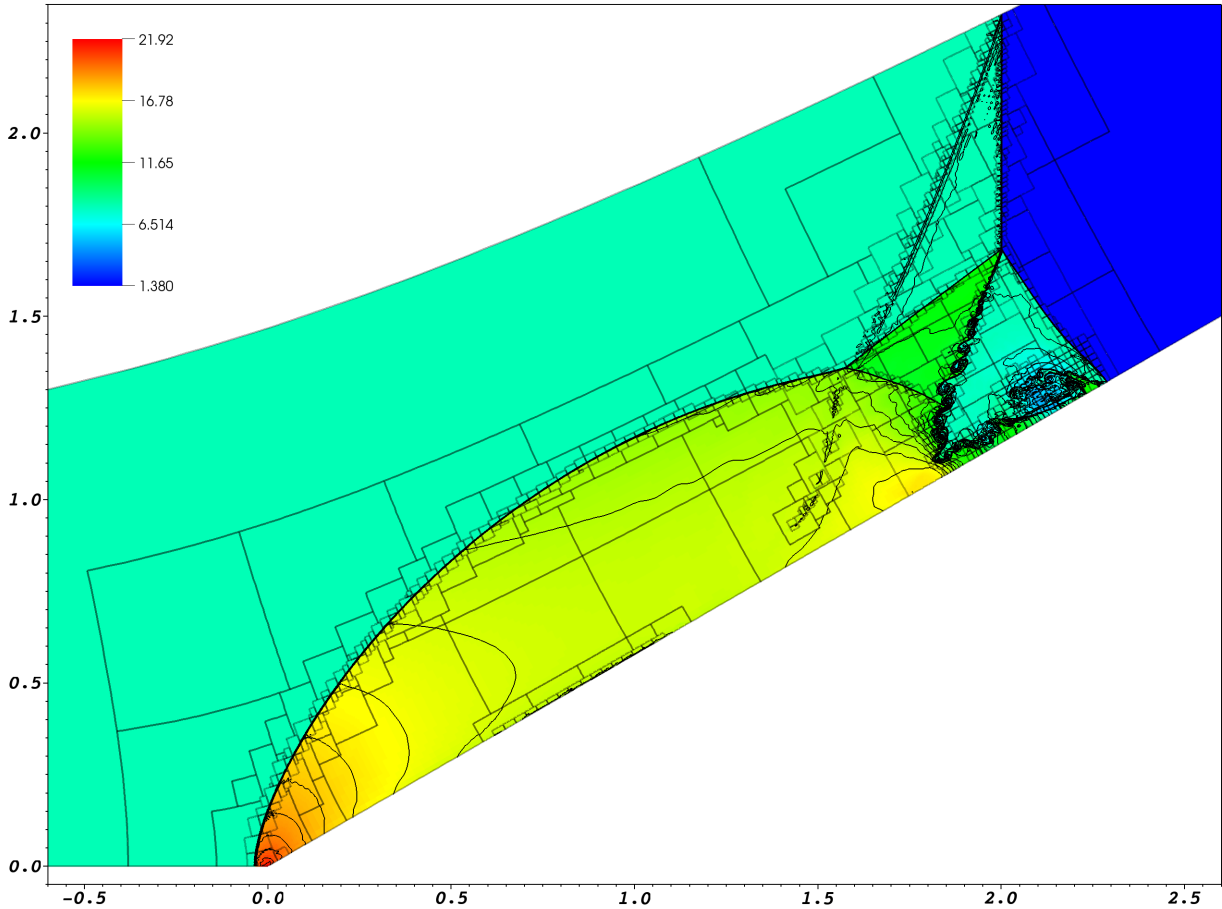


Figure 7. Two-dimensional Woodward-Colella Mach-reflection problem. Contours of density are shown along with the shaded boundaries of boxes belonging to various levels of refinement. Three levels of refinement (above a base grid of size 96×24) are used with a four-times increase in resolution at each level.

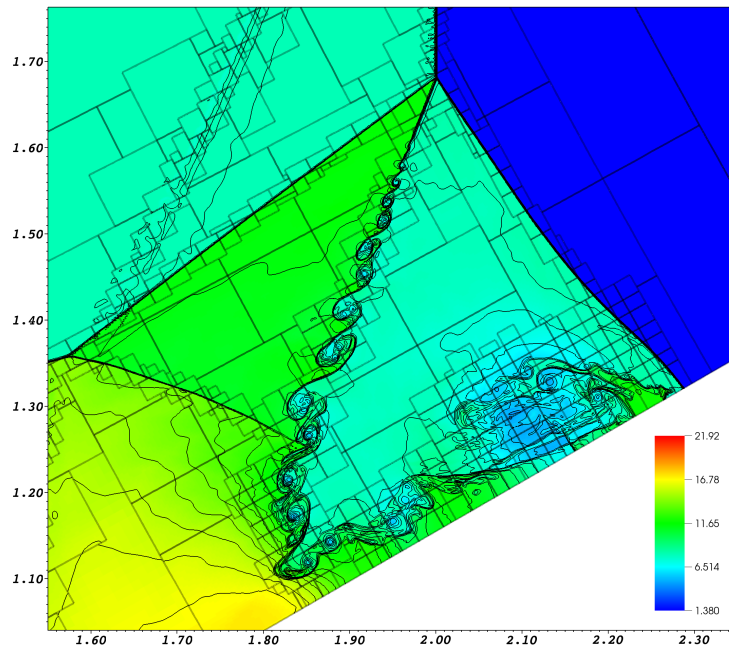


Figure 8. Details of fluid instabilities along the slip line in the Woodward-Colella Mach-reflection problem.

V. Conclusion

We have presented a fourth-order finite-volume method for obtaining solutions to hyperbolic conservation laws on mapped grids with adaptive mesh refinement. A careful implementation ensures that both a freestream preservation property is retained and that conservation is not violated.

References

- ¹McCorquodale, P. and Colella, P., “A high-order finite-volume method for hyperbolic conservation laws on locally-refined grid,” *Communications in Applied Mathematics and Computational Science*, Vol. 6, No. 1, 2011, pp. 1–25.
- ²Berger, M. J. and Colella, P., “Local Adaptive Mesh Refinement for Shock Hydrodynamics,” *J. Comput. Phys.*, Vol. 82, No. 1, May 1989, pp. 64–84.
- ³Steger, J. L., “Implicit Finite Difference Simulation of Flow about Arbitrary Two-Dimensional Geometries,” AIAA 77-665, 1978.
- ⁴Colella, P., Dorr, M. R., Hittinger, J. A. F., and Martin, D. F., “High-order finite-volume methods in mapped coordinates,” *J. Comput. Phys.*, Vol. 230, 2011, pp. 2952–2976.
- ⁵Bell, J., Colella, P., Trangenstein, J., and Welcome, M., “Adaptive Mesh Refinement on Moving Quadrilateral Grids,” *Proceedings of the AIAA 9th Computational Fluid Dynamics Conference*, Buffalo, N.Y., June 1989, pp. 471–479.
- ⁶Spivak, M., *Calculus on Manifolds*, Westview Press, 1965.
- ⁷Colella, P. and Sekora, M., “A limiter for PPM that preserves accuracy at smooth extrema,” *J. Comput. Phys.*, Vol. 227, 2008, pp. 7069–7076.
- ⁸Colella, P., Graves, D. T., Keen, N., Ligocki, T. J., Martin, D. F., McCorquodale, P., Modiano, D., Schwartz, P., Sternberg, T., and Straalen, B. V., *Chombo Software Package for AMR Applications - Design Document*, Lawrence Berkeley National Laboratory, 2009, <https://seesar.lbl.gov/anag/chombo/ChomboDesign-3.0.pdf>.
- ⁹Berger, M. J., “On Conservation at Grid Interfaces,” *SIAM J. Numer. Anal.*, Vol. 24, No. 5, 1987, pp. 967–984.
- ¹⁰Woodward, P. R. and Colella, P., “The numerical simulation of two-dimensional fluid flow with strong shocks,” *J. Comput. Phys.*, Vol. 54, 1984, pp. 115–173.
- ¹¹Panton, R. L., *Incompressible Flow*, John Wiley & Sons, 1996.