Title

A freestream-preserving fourth-order finite-volume method in mapped coordinates with adaptive-mesh refinement

Authors

Guzik, Stephen M
Gao, Xinfeng
Owen, Landon D
et al.

Publication Date

2015-12-01

DOI

10.1016/j.compfluid.2015.10.001

Peer reviewed

A Freestream-Preserving Fourth-Order Finite-Volume Method in Mapped Coordinates with Adaptive-Mesh Refinement

Stephen M. Guzik, Xinfeng Gao*, Landon D. Owen

Computational Fluid Dynamics and Propulsion Laboratory
Colorado State University, Fort Collins, CO 80525, USA

Peter McCorquodale, Phillip Colella

Applied Numerical Algorithms Group
Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA

Abstract

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 are combined with detailed mechanisms for accommodating the adapting grids. These considerations ensure that conservation is maintained and that the divergence of a constant vector field is always zero (freestream-preservation property). The solution in time is advanced with a fourth-order Runge-Kutta method. A series of tests verifies that the expected accuracy is achieved in smooth flows and the solution of a Mach reflection problem demonstrates the effectiveness of the algorithm in resolving strong discontinuities.

Keywords: High-Order Finite-Volume Method, Freestream-preserving, Mapped Grids, Adaptive-Mesh Refinement, Finite-Volume Method, Hyperbolic Conservation Laws

*Corresponding author
Email address: Xinfeng.Gao@colostate.edu (Xinfeng Gao)
URL: http://www.engr.colostate.edu/~gao (Xinfeng Gao)

1. Introduction

The algorithm described herein for solving time-dependent hyperbolic systems of conservation laws features three main elements: mapped grids, a fourth-order finite-volume method, and an adaptive-mesh-refinement (AMR) scheme. Challenges arise from the integration of the three elements. Specifically, some care is required to ensure the solution maintains three properties: freestream-preservation, conservation, and fourth-order accuracy in smooth regions. Accordingly, novel strategies are designed, implemented, and verified. In this section, the rationale for the core elements is summarized along with an explanation for the complexity of the algorithm.

The addition of the metric terms can introduce subtle difficulties. For example, a constant solution in physical space may appear with gradients when mapped to computational space. Therefore, 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. Colella et al. [1] have developed a high-order method 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 Colella [2] for Cartesian grids with the freestream-preserving technology on mapped grids of Colella et al. [1] to achieve a fourth-order, freestream-preserving, finite-volume method on mapped grids with AMR. 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. [3]. In that procedure, special adjustments 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, discussed in Section 5, to ensure the freestream-preservation property.

The resulting algorithm is somewhat complex and results from a desire to support non-analytic mappings and maintain high performance. Logistics related to maintaining conservation and freestream-preservation in regions of

AMR can be avoided if the mapping is analytic or if metric terms are calculated at the finest resolution everywhere in the domain. If metric terms can be evaluated exactly for an analytic mapping, the need for the considerations we employ to maintain conservation and freestream preservation is obviated. If performance is not of concern, one can evaluate all the metrics on the finest grid everywhere in the domain and average the result, in a consistent manner, to any coarser grid. Again, the need for special treatment is obviated. However, doing so undermines the whole point of AMR, that we cannot tolerate the memory or expense of the fine-grid calculations everywhere in the domain. This work applies to cases where these shortcuts cannot be taken: the metrics may only be known to some degree of accuracy (ideally at least fourth-order) and fine-grid calculations are only applied where strictly necessary. Versus a second-order accurate algorithm, accommodating the differences in grid metrics, due to changing grid resolutions, imposes additional difficulty. At second-order accuracy, the geometry can be treated as piecewise linear elements; changes in geometry induced by changes in grid resolution are therefore more easily understood. At fourth-order accuracy, maintaining consistency between grid levels requires the more rigorous algorithm defined herein and changes in geometry are instead understood by a volume-flux.

The methods devised for accommodating the mapped AMR grid follow the philosophy of finite-volume methods—especially the notion of a volume flux—and the same is used for solving the flow equations. Finite-volume methods are well 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. High-order finite-volume methods (fourth-order and beyond) are advantageous in that they can produce solutions to smooth flows much faster than low-order schemes, to the same level of accuracy. This can be especially important in applications, such as all-speed combustion, where the simulation time is extremely long. Moreover, high-order methods increase the computation per unit memory which makes better use of modern and upcoming computer architectures.

The approach for accommodating the mapping is orthogonal to the particular details of the finite-volume algorithm used for discretizing the partial differential equations to obtain a high-order solution with limiters. Any finite-volume method, applied to Cartesian grids, can be adapted to use the methodologies outlined here for AMR on mapped grids. For example, the approach is equally applicable to the essentially non-oscillatory (ENO) [4] and weighted (WENO) [5] family of methods or the piecewise parabolic method (PPM) [6]. The mapped grids algorithm can be directly applied to many of the methods summarized by van Leer [7], with the exception of residual distribution which requires a dual mesh. A thorough comparison of underlying technologies from several reconstruction methods was performed by Liska and Wendroff [8] for several solutions of the Euler equations. Another comparison between many of these methods on unstructured grids is given by Wang [9]. The PPM method is used herein, only because it is an area of research in which the authors have devoted much previous effort [6]. For the high-order finite-volume scheme used in this work, our implementation follows the techniques described by McCorquodale and Colella [2] for achieving fourth-order accurate solutions of time-dependent hyperbolic conservation laws on adaptively-refined Cartesian grids.

Adaptive mesh refinement is an important aspect in our solver, allowing for the mesh resolution to change in response to the characteristics of the solution. The AMR technique is extremely advantageous for applications involving discontinuities, shock waves [10], space plasma physics [11], and combustion [12] or in any regions that require high resolution. AMR has been applied to high-order methods [13], and more recently to high-order ADER-WENO methods [14, 15]. AMR has been successfully applied to unstructured grids [16], overlapping grids [17], mapped grids [18], and body-fitted curvilinear grids [19]. Recent research in AMR extends to hp-refinement (grid and order refinement) [20, 21] and using adjoint error estimates to govern the refinement [22]. For this work, block-structured AMR is used with grids locally refined in space and time [10].

In addition to our approach of mapping [23] a structured grid in physical space to a Cartesian grid in computational space, other methods exist for apply-

ing finite-volume schemes to more complex geometry. One can resort to applying the finite-volume method directly to structured curvilinear meshes or unstructured meshes, or using embedded boundary (cut-cell) techniques [24, 25]. Our preference for mapped grids stems from the computational efficiency afforded by solving on a Cartesian grid in computation space and the ease at which boundary layers can be resolved. The other methods have their own advantages; a significant advantage of the embedded boundary approach is elimination of mesh generation from the standard workflow for generating a CFD solution.

The remainder of this paper is organized as follows. We discuss the fourth-order finite-volume method in Section 3, and the freestream-preserving mapping of coordinates in Section 4. Adaptive mesh refinement for mapped grids is described in Section 5. Results are presented in Section 6 where all solutions are for the Euler equations with the assumption of a perfect gas. We verify that both freestream preservation and conservation are not violated, that the expected accuracy is achieved, and finally demonstrate the algorithm on an unsteady problem featuring shock reflections.

2. Notation

The notation is generally explained as it is introduced. Bold type is reserved for real physical vectors, such as solution variables of a system of the governing equations. However, a bold symbol can also represent vectors in integer space and examples are given below. Matrices are named with capital letters, such as N, and the entries are named with subscripts, $\mathrm{N}_{i,j}$, where i and j are the indices for the row and the column, respectively. The vector symbol $\vec{}$ is used for vectors (or column matrices) depending on space dimensions. Some symbols are listed and defined here for convenience.

$\vec{\mathbf{F}}$	flux dyad, e.g., $[\mathbf{F}_x, \mathbf{F}_y, \mathbf{F}_z]$ for 3 dimensions space (e.g. in Cartesian coordinates)
$\mathbf{F}_d$	the d^{th} component of $\vec{\mathbf{F}}$
$\mathbf{F}$	flux vector, e.g., $[\mathrm{F}_1, \mathrm{F}_2, \cdots, \mathrm{F}_n]$ for n solution variables

F_d	the d^{th} component of $\mathbf{F}$
$\vec{x}$	physical space, e.g., (x, y, z) for 3 dimensions in space
x_d	the d^{th} component of $\vec{x}$
$\vec{\xi}$	computational space, e.g., (ξ, η, ζ) for 3 dimensions in space
ξ_d	the d^{th} component of $\vec{\xi}$
$\vec{\nabla}_x$	$\vec{\nabla}$ in physical space, a subscript with x representing the operator is in physical space
$\vec{\nabla}_\xi$	$\vec{\nabla}$ in computational space, a subscript with ξ representing the operator is in computational space
$\boldsymbol{i}$	grid indices, e.g., (i, j, k) for 3 dimensions in space
$\boldsymbol{e}^d$	unit vector in direction d
Δ	Laplacian operator
h	grid spacing
D	dimensions
τ	temporal interval or time step
$\hat{n}$	outward unit normal of cell face

3. Fourth-Order Finite-Volume Method

The fourth-order accurate finite-volume method developed by McCorquodale and Colella [2] for solving time-dependent hyperbolic systems of conservation laws on Cartesian grids with multiple levels of refinement serves as the base algorithm for our work on mapped grids. A brief review of the fundamental Cartesian method follows, since many details of the Cartesian algorithm, especially the limiting mechanisms and the methods for interpolating in time between two levels of refinement, remain unchanged when applied to mapped grids. The work by McCorquodale and Colella [2] should be consulted for further information.

We consider time-dependent solutions to hyperbolic systems of conservation laws having the general form

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

where $\mathbf{U}$ is the solution vector and $\vec{\mathbf{F}}$ is the flux dyad. From the general divergence form, the integral form of the governing equations can be derived as

$$\frac{\partial}{\partial t}\int_V \mathbf{U}\,\mathrm{d}V + \int_V \vec{\nabla}\cdot\vec{\mathbf{F}}\mathrm{d}V = 0\,, \tag{2}$$

where V is the volume of a finite region of space. The governing equations must be supplemented by an equation of state. The ideal gas law is assumed in this work.

3.1. Fourth-Order Finite-Volume Scheme

In the Cartesian-grid finite volume approach, the grid is defined by the integer points $(i_0, ..., i_{D-1}) = \boldsymbol{i} \in \mathbb{Z}^D$, which mark the cell centers of the control volumes. The control volumes take the form

$$V_{\boldsymbol{i}} = \Big(\boldsymbol{i}h, (\boldsymbol{i}+\boldsymbol{I})h\Big)\ ,\ \boldsymbol{i} \in \mathbb{Z}^D\ ,$$

where h is the grid spacing, $\mathbb{Z}^D$ is the integer space, and $\boldsymbol{I}$ is defined as the vector whose components are all equal to one. The solution is computed in the problem domain, $\Gamma \subset \mathbb{Z}^D$, a bounded subset of the integer points. A finite volume method satisfies the integral form of the conservation law, (2), to some degree of approximation for each of many contiguous control volumes, $V_{\boldsymbol{i}}$, in the problem domain. The cell-average value of $\mathbf{U}$ in a control volume is defined as

$$\langle \mathbf{U} \rangle_{\boldsymbol{i}} \quad \equiv \quad \frac{1}{V_{\boldsymbol{i}}}\int_{V_{\boldsymbol{i}}} \mathbf{U}\,dV\,.$$

In this work, the control volumes are time-invariant; accordingly the first term in (2) can be simplified to

$$\frac{\partial}{\partial t}\int_{V_{\boldsymbol{i}}} \mathbf{U}\,\mathrm{d}V = h^D \frac{\mathrm{d}}{\mathrm{d}t}\langle \mathbf{U} \rangle_{\boldsymbol{i}}\,.$$

Application of the divergence theorem of Gauss to the second term in (2) results in

$$\begin{aligned}\int_V \vec{\nabla}\cdot\vec{\mathbf{F}}dV &= \int_A \vec{\mathbf{F}}\cdot\hat{n}\,dA \\ &= \sum_{\pm=[+,-]}\sum_{d=1}^{D} \pm \int_{A_d^{\pm}} \mathbf{F}_d\,dA\,,\end{aligned} \tag{3}$$

where $A_d^{\pm}$ are the high and low faces bounding V with outward unit normal, $\hat{n}$, in the d^{th} coordinate direction.

Aside from grid structure, properties of different finite-volume methods generally arise from the specific algorithm used to evaluate the flux integrals in (3). For example, second-order algorithms may use the midpoint rule to approximate the flux integrals. Higher-order algorithms may use some form of quadrature to evaluate the flux integrals. Furthermore, various interpolations may be devised to interpolate properties on the face, including solution-adaptive variants such as ENO [4] and WENO [5]. Flux evaluations may also include temporal gradients as in the ADER-WENO approach [14], in which case the method-of-lines approach for temporal evolution of the solution is avoided.

We will be developing fourth-order methods using the approach in [26]. The starting point for this approach is to replace the integrand in (3) by a Taylor expansion about the center of the face:

$$\int_{A_d} \mathbf{F}_d dA = \sum_{0 \le |\boldsymbol{r}| < R} \frac{1}{\boldsymbol{r}!} \vec{\nabla}^{\boldsymbol{r}} \mathbf{F}_d|_{\vec{x}=\vec{x}_{\boldsymbol{i}}} \int_{A_d} (\vec{x} - \vec{x}_{\boldsymbol{i}})^{\boldsymbol{r}} dA_x + O(h^{R+D-1}), \tag{4}$$

$$\boldsymbol{r}! = r_1! \dots r_D! \,. \tag{5}$$

For example, if we take $R = 4$, we obtain

$$\frac{1}{h^{D-1}} \int_{A_d} \mathbf{F}_d dA = \mathbf{F}_d(\vec{x}_{\boldsymbol{i}}) + \frac{h^2}{24} \sum_{d' \neq d} \frac{\partial^2 \mathbf{F}_d}{\partial x_{d'}^2} + O(h^4). \tag{6}$$

If we replace the derivatives by finite-difference approximations of a suitable order (in this case second-order is sufficient) that are smooth functions of their inputs, the resulting approximation of the average of the flux divergence over a cell is $O(h^4)$. Readers are referred to see Appendix A.1 for details.

This approach can now be applied to solve hyperbolic conservation laws. The integrals over cells and faces are represented as averages yielding

$$\frac{\mathrm{d}}{\mathrm{d}t} \langle \mathbf{U} \rangle_{\boldsymbol{i}} + \frac{1}{h} \sum_{d=1}^{D} \left(\langle \mathbf{F}_d \rangle_{\boldsymbol{i} + \frac{1}{2}\boldsymbol{e}^d} - \langle \mathbf{F}_d \rangle_{\boldsymbol{i} - \frac{1}{2}\boldsymbol{e}^d} \right) = 0 \,. \tag{7}$$

A given implementation of the finite-volume method is defined by the techniques used to determine the average flux on the faces, $\langle \mathbf{F}_d \rangle_{\boldsymbol{i} \pm \frac{1}{2}\boldsymbol{e}^d}$, and the

time-marching scheme used to resolve the derivative with respect to time.

3.2. Spatial Discretization

Our approach to computing $\langle \mathbf{F}_d \rangle_{\boldsymbol{i} \pm \frac{1}{2}\boldsymbol{e}^d}$ as a function of $\langle \mathbf{U} \rangle_{\boldsymbol{i}}$ primarily follows the method described in McCorquodale and Colella [2] and is briefly summarized here. As it often is done for second-order methods, we introduce a nonlinear change of variables $\mathbf{W} = \mathbf{W}(\mathbf{U})$. In the case of Euler equations of gas dynamics, this is the conversion from the conserved quantities of mass, momentum, and energy, $\mathbf{U} = [\rho, \rho\boldsymbol{u}, \rho E]$, to primitive variables $\mathbf{W} = [\rho, \boldsymbol{u}, p]$, where ρ is the gas density, $\boldsymbol{u}$ is the velocity vector, E is the total energy per unit mass, and p is the pressure. Some care is required in transforming from conservative to primitive variables in order to preserve fourth-order accuracy. Specifically, the steps to compute $\langle \mathbf{F}_d \rangle_{\boldsymbol{i} \pm \frac{1}{2}\boldsymbol{e}^d}$ are given by:

1. **Compute average primitive state in cells:** Convert from cell-averaged conserved variables to cell-averaged primitive variables, through cell-centered values. First, calculate a fourth-order approximation to $\mathbf{U}$ at cell centers,

$$\mathbf{U}_{\boldsymbol{i}} = \langle \mathbf{U} \rangle_{\boldsymbol{i}} - \frac{h^2}{24} \Delta \langle \mathbf{U} \rangle_{\boldsymbol{i}} \,,$$

   where Δ is the Laplacian, accurate to at least second-order.

   Next convert to primitive variables:

$$\mathbf{W}_{\boldsymbol{i}} = \mathbf{W}(\mathbf{U}_{\boldsymbol{i}}) \,,$$
$$\overline{\mathbf{W}}_{\boldsymbol{i}} = \mathbf{W}(\langle \mathbf{U} \rangle_{\boldsymbol{i}}) \,.$$

   Then, calculate a fourth-order approximation to cell-averaged $\mathbf{W}$,

$$\langle \mathbf{W} \rangle_{\boldsymbol{i}} = \mathbf{W}_{\boldsymbol{i}} + \frac{h^2}{24} \Delta \overline{\mathbf{W}}_{\boldsymbol{i}} \,. \tag{8}$$

2. **Compute average primitive state on faces:** Interpolate from cell-averaged $\mathbf{W}$ to fourth-order face-averaged $\mathbf{W}$ over faces in dimension d, by

$$\langle \mathbf{W} \rangle_{\boldsymbol{i} + \frac{1}{2}\boldsymbol{e}^d} = \frac{7}{12} \left(\langle \mathbf{W} \rangle_{\boldsymbol{i}} + \langle \mathbf{W} \rangle_{\boldsymbol{i} + \boldsymbol{e}^d} \right) - \frac{1}{12} \left(\langle \mathbf{W} \rangle_{\boldsymbol{i} - \boldsymbol{e}^d} + \langle \mathbf{W} \rangle_{\boldsymbol{i} + 2\boldsymbol{e}^d} \right) \,, \tag{9}$$

(see Appendix A.2 for the derivation of (9)). Both slope flattening and a variant of the piecewise parabolic method (PPM) limiter [6] are used to adjust the face-averaged primitive state in order to suppress oscillations near discontinuities. A complete discussion is given by McCorquodale and Colella [2]. At domain boundaries, the face-averaged primitive state is adjusted as required to enforce the desired boundary conditions.

3. **Compute average flux on faces:** First obtain a face-centered approximation of $\mathbf{W}$:

$$\mathbf{W}_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = \langle \mathbf{W} \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} - \frac{h^2}{24} \Delta^{\perp,d} \langle \mathbf{W} \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}$$

where the $\Delta^{\perp,d}$ is a second-order accurate Laplacian computed only in directions orthogonal to d, i.e.,

$$\Delta^{\perp,d} = \Delta - \frac{\partial^2}{\partial x_d^2} \,.$$

Then calculate the face-averaged fluxes in each dimension d,

$$\langle \mathbf{F}_d \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = \mathbf{F}_d(\mathbf{W}_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}) + \frac{h^2}{24} \Delta^{\perp,d} \mathbf{F}_d(\langle \mathbf{W} \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}) \,, \tag{10}$$

for every d-face $\boldsymbol{i} + \frac{1}{2}\boldsymbol{e}^d$. Finally, the divergence is computed as in (7).

In Step 1 above, the Laplacian in (8) is applied to $\overline{\mathbf{W}}_{\boldsymbol{i}}$ instead of $\mathbf{W}_{\boldsymbol{i}}$ in order to minimize the size of stencil required. The stencil to compute $\mathbf{W}_{\boldsymbol{i}}$ has radius 3, but the $\overline{\mathbf{W}}_{\boldsymbol{i}}$ can be computed from the known $\langle \mathbf{U} \rangle_{\boldsymbol{i}}$ for each cell. This substitution makes a difference of $O(h^4)$ because the discrete Laplacian is multiplied by h^2. Similarly, in Step 3 above, $\Delta^{\perp,d}$ in (10) is applied to $\mathbf{F}_d(\langle \mathbf{W} \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d})$ instead of $\mathbf{F}_d(\mathbf{W}_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d})$ for the same reason. Also, the value of $\mathbf{W}$ differs from that of $\langle \mathbf{W} \rangle$ by a term of second order as shown in (A.6) from Appendix A.1. As discussed in [2], stencils are modified near physical boundaries to be one-sided.

3.3. Temporal Discretization

The semi-discrete system of ordinary differential equations, given by (7), is discretized in time using an explicit, four-stage, fourth-order classical Runge-Kutta scheme.

3.4. Comments on Discretization

It is worth asking whether the Taylor expansion used to achieve high order of accuracy may impose possible restrictions, for example, for long-time simulations. There are no reasons to expect drawbacks in long-time simulations. The method has been demonstrated to be stable, to the extent that any finite volume method has been shown to be stable—a combination of von Neumann analysis for advection and empirical observation for more complicated problems. Since the method is also consistent, we expect convergence. The only other issue with long-time solutions is the representation of steady-state or quasi-steady solutions. Since we are using the method of lines, we can converge to a discrete steady state solution that is independent of the time step. However, since we are primarily interested in transient phenomena, we have not investigated that issue.

4. Freestream-Preserving Mapping of Coordinates

The finite volume method described above for Cartesian grids is now applied to mapped grids by extending the method in Colella et al. [27] to our case of non-linear systems of conservation laws. The formulation begins again with the integral form of the conservation laws, but applied to the control volumes in physical space, $\vec{x}$,

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

Next, the switch to computation space, $\vec{\xi}$, is made using the metric terms such that

$$\frac{\partial}{\partial t}\int_{V_i} J\mathbf{U}\,\mathrm{d}\vec{\xi} + \int_{V_i} \vec{\nabla}_\xi \cdot (\mathrm{N}^T\vec{\mathbf{F}})\,\mathrm{d}\vec{\xi} = 0\,, \tag{11}$$

where the metric Jacobian is defined by $J \equiv \det(\vec{\nabla}_\xi \vec{x})$ and the transformation matrix, N^T, describes the grid metrics. Given that $\mathrm{N}^T = J\vec{\nabla}_x\vec{\xi}$, with $\sum_j \frac{\partial \mathrm{N}_{i,j}}{\partial \xi_j} = 0$ (note that the column of N^T is divergence free), the relation, $\vec{\nabla}_\xi \cdot (\mathrm{N}^T\vec{\mathbf{F}}) =$

$\sum_j \frac{\partial(\mathrm{N}_{i,j}\mathbf{F}_i)}{\partial \xi_j} = J\vec{\nabla}_x \cdot \vec{\mathbf{F}}$, exists, so that

$$\int_{\vec{x}(V_{\boldsymbol{i}})} \vec{\nabla}_x \cdot \vec{\mathbf{F}} \, \mathrm{d}\vec{x} = \int_{V_i} \vec{\nabla}_\xi \cdot (\mathrm{N}^T \vec{\mathbf{F}}) \, \mathrm{d}\vec{\xi} \, .$$

Then, application of the divergence theorem of Gauss to (11) results in

$$\frac{\partial}{\partial t} \int_{V_{\boldsymbol{i}}} J\mathbf{U} \, \mathrm{d}\vec{\xi} + \int_{\partial V_{\boldsymbol{i}}} (\mathrm{N}^T \vec{\mathbf{F}}) \cdot \hat{n} \, \mathrm{d}\mathcal{S} = 0 \, . \tag{12}$$

Following (7), the integrals can be represented as averages yielding

$$\frac{\mathrm{d}}{\mathrm{d}t} \langle J\mathbf{U} \rangle_{\boldsymbol{i}} + \frac{1}{h} \sum_{d=1}^{D} \left(\langle (\mathrm{N}_d^T \vec{\mathbf{F}}) \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} - \langle (\mathrm{N}_d^T \vec{\mathbf{F}}) \rangle_{\boldsymbol{i}-\frac{1}{2}\boldsymbol{e}^d} \right) = 0 \, , \tag{13}$$

where the subscript d denotes the d^{th} row of N^T.

Our approach to determining fourth-order accurate averages of $\langle (\mathrm{N}_d^T \vec{\mathbf{F}}) \rangle$ on the faces is to use a product rule of the form

$$\langle uv \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = \langle u \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} \langle v \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} + \frac{h^2}{12} \sum_{d' \neq d} \frac{\partial u}{\partial \xi_{d'}} \frac{\partial v}{\partial \xi_{d'}} + O(h^4) \, , \tag{14}$$

which is valid on Cartesian grids (see (A.7) from Appendix A.1) and where a fourth-order estimate of $\langle \vec{\mathbf{F}} \rangle$ is known.

For a single solution variable, the averages in equation (13) can be obtained from

$$\langle (\mathrm{N}_d^T \vec{\mathrm{F}}) \rangle_{\boldsymbol{i}\pm\frac{1}{2}\boldsymbol{e}^d} = \sum_{s=1}^{D} \langle \mathrm{N}_{d,s}^T \rangle_{\boldsymbol{i}\pm\frac{1}{2}\boldsymbol{e}^d} \langle \mathrm{F}_s \rangle_{\boldsymbol{i}\pm\frac{1}{2}\boldsymbol{e}^d} + \frac{h^2}{12} \sum_{s=1}^{D} \sum_{d' \neq d} \frac{\partial \mathrm{N}_{d,s}^T}{\partial \xi_{d'}} \frac{\partial \mathrm{F}_s}{\partial \xi_{d'}} + O(h^4) \, . \tag{15}$$

Some care is required to obtain freestream preservation, the property that the discrete divergence of a constant vector field is zero. If the flux, $\vec{\mathrm{F}}$, in (15) is constant, then the second term vanishes and we only need to derive quadrature formulas for $\langle \mathrm{N}^T \rangle$ so that the discrete divergence of a constant vector field given by (15) is zero. The existence of such quadratures is a consequence of Stokes' theorem and the Poincaré lemma. The columns of the matrix N^T (or rows of N) are divergence-free as can be seen by a direct calculation. The column vectors of N^T (the row vectors of N) are denoted by N_s.

The Poincaré lemma implies the existence of vectors $\vec{\mathcal{N}}_s$ such that, in three dimensions, $\mathrm{N}_s = \vec{\nabla}_\xi \times \vec{\mathcal{N}}_s$, and thus $\vec{\nabla}_\xi \cdot (\vec{\nabla}_\xi \times \vec{\mathcal{N}}_s) = 0$. Stokes' theorem then states that

$$\int_\Sigma (\vec{\nabla}_\xi \times \vec{\mathcal{N}}_s) \cdot \hat{n} \, \mathrm{d}\mathcal{S} = \oint_{\partial\Sigma} \vec{\mathcal{N}}_s \cdot \mathrm{d}\vec{r}, \tag{16}$$

with Σ being an oriented face and $\partial\Sigma$ being the boundary edge of Σ, oriented using the right-hand rule. Using Stokes' theorem, the integral of N_s can instead be evaluated by integrating $\vec{\mathcal{N}}_s$; for example, in three dimensions, $\vec{\mathcal{N}}_s$ is integrated on the edges of a face to find N_s on the face of a cell.

The extension to general dimensions is made by the observing that the three-dimensional cross product can be represented as

$$\vec{\nabla}_\xi \times \vec{\mathcal{N}}_s = \begin{bmatrix} 0 & \mathcal{N}_{s,(3)} & -\mathcal{N}_{s,(2)} \\ -\mathcal{N}_{s,(3)} & 0 & \mathcal{N}_{s,(1)} \\ \mathcal{N}_{s,(2)} & -\mathcal{N}_{s,(1)} & 0 \end{bmatrix} \cdot \begin{bmatrix} \frac{\partial}{\partial \xi_1} \\ \frac{\partial}{\partial \xi_2} \\ \frac{\partial}{\partial \xi_3} \end{bmatrix}, \tag{17}$$

where subscript () selects a component of $\vec{\mathcal{N}}_s$. Note the anti-symmetry of the matrix. The rows represent indices of d, the (hyper)face, and the columns represent indices of $d' \neq d$. As such, this matrix can be expressed by

$$\vec{\nabla}_\xi \times \vec{\mathcal{N}}_s = \begin{bmatrix} 0 & \mathcal{N}_{s,(1,2)} & \mathcal{N}_{s,(1,3)} \\ \mathcal{N}_{s,(2,1)} & 0 & \mathcal{N}_{s,(2,3)} \\ \mathcal{N}_{s,(3,1)} & \mathcal{N}_{s,(3,2)} & 0 \end{bmatrix} \cdot \begin{bmatrix} \frac{\partial}{\partial \xi_1} \\ \frac{\partial}{\partial \xi_2} \\ \frac{\partial}{\partial \xi_3} \end{bmatrix}, \tag{18}$$

where

$$\mathcal{N}_{s,(d,d')} = -\mathcal{N}_{s,(d',d)} \, . \tag{19}$$

The general formula for a row of the matrix in (18) is

$$\mathrm{N}^T_{d,s} = \sum_{d' \neq d} \frac{\partial \mathcal{N}_{s,(d,d')}}{\partial \xi_{d'}} \quad \text{with} \quad \mathcal{N}_{s,(d,d')} = -\mathcal{N}_{s,(d',d)} \, , \tag{20}$$

By Stokes' theorem, the integral of (20) over a cell face A_d is given by

$$h^{D-1} \langle \mathrm{N}^T_{d,s} \rangle = \int_{A_d} \mathrm{N}^T_{d,s} \, \mathrm{d}A_\xi = \int_{A_d} \sum_{d' \neq d} \frac{\partial \mathcal{N}_{s,(d,d')}}{\partial \xi_{d'}} \, \mathrm{d}A_\xi = \oint_{l_{d,d' \neq d}} \mathcal{N}_{s,(d,d')} \, \mathrm{d}r_\xi \, , \tag{21}$$

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

$$\langle \mathrm{N}^T_{d,s} \rangle = \frac{1}{h^{D-1}} \sum_{\pm=[+,-]} \sum_{d'\neq d} \pm \int_{l_{d,d',\pm}} \mathcal{N}_{s,(d,d')} \, \mathrm{d}r_\xi \,, \tag{22}$$

with quadratures of sufficient order replacing the integrals. Equations (21) and (22) are the extension of (16) to any number of dimensions and applied on a Cartesian grid with spacing h. Because (22) only involves terms normal to the face (consider the left hand side of (16)), only the normal components of $\langle \mathrm{N}^T \rangle$ can be determined using (22). However, this is all that is required for hyperbolic conservation laws. If ever required, transverse components of $\langle \mathrm{N}^T \rangle$ can be computed directly or averaged from the normal components of $\langle \mathrm{N}^T \rangle$ on nearby faces in orthogonal directions.

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

The family of functions, $\mathcal{N}_{s,(d,d')}, d' \neq d$, satisfying (20) is not unique. The form of $\mathcal{N}_{s,(d,d')}$ used herein is given explicitly by

$$\mathcal{N}_{s,(d,d')} = \frac{1}{D-1} \det((\vec{\nabla}_\xi \vec{x})^T (d|\boldsymbol{e}^s)(d'|\vec{x})) \,, \tag{23}$$

where $\mathrm{A}(p|\boldsymbol{v})$ denotes a modification of matrix A by replacing row p with vector $\boldsymbol{v}$. That (23) is anti-symmetric is trivial; simply swap rows d and d'. The proof that (23) satisfies (20) is obtained by applying Leibnitz' rule to determinants [1].

Note that the preceding discussion on freestream preservation concerns only the flux divergence. The treatment of source terms is straightforward on the mapped grid, as long as the control volumes are stationary, since freestream preservation concerns do not arise.

4.1. Discretization

The method for determining $\langle \vec{\mathbf{F}} \rangle$ and evolving the solution is the same as that described by McCorquodale and Colella [2] 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 Sekora [28]), slope flattening, and artificial viscosity are all applied to stabilize the method and minimize oscillations near large gradients. To evolve the solution in time, the standard fourth-order Runge-Kutta method is employed.

4.2. Artificial Viscosity

The dissipation produced by the limiters is insufficient to suppress oscillations at strong shocks (see [6] for a detailed discussion). The two mechanisms proposed by Colella and Woodward [6] for introducing additional dissipation, flattening of the interpolated profiles and artificial viscosity, were implemented by McCorquodale and Colella [2] for fourth-order solutions to the hyperbolic system in Cartesian coordinates. For mapped grids, the application of artificial viscosity needs to be modified. Using the relation, $\vec{\nabla}_x \cdot \vec{\mathbf{F}} = J^{-1} \vec{\nabla}_\xi \cdot (\mathrm{N}^T \vec{\mathbf{F}})$, the divergence of the velocity in physical space, λ, can be computed as

$$\begin{aligned} \lambda_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} &= \vec{\nabla}_x \cdot \boldsymbol{u}_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = J^{-1} \vec{\nabla}_\xi \cdot \left(\mathrm{N}^T \boldsymbol{u}\right)_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} \\ &= \frac{1}{J} \left[\sum_{s=1}^{D} \left(\langle \boldsymbol{u}_s \rangle \sum_{d'=1}^{D} \frac{\partial \langle \mathrm{N}^T_{d',s} \rangle}{\partial \xi_{d'}} \right) + \sum_{d'=1}^{D} \sum_{s=1}^{D} \langle \mathrm{N}^T_{d',s} \rangle \frac{\partial \langle \boldsymbol{u}_s \rangle}{\partial \xi_{d'}} \right]_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}. \end{aligned} \tag{24}$$

The velocity, $\boldsymbol{u}$, is known at the cell centers while row d of N^T is only known on the d-direction faces. The first term in the brackets of (24) drops out because the columns of matrix N^T are known to be divergence free. The second term in (24) is approximated by

$$\begin{aligned} \left(\sum_{d'=1}^{D} \sum_{s=1}^{D} \langle \mathrm{N}^T_{d',s} \rangle \frac{\partial \langle \boldsymbol{u}_s \rangle}{\partial \xi_{d'}} \right)_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} &= \sum_{s=1}^{D} \left\{ \left(\langle \mathrm{N}^T_{d'=d,s} \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} \frac{\langle \boldsymbol{u}_s \rangle_{\boldsymbol{i}+\boldsymbol{e}^d} - \langle \boldsymbol{u}_s \rangle_{\boldsymbol{i}}}{h} \right) \right. \\ + \sum_{d' \neq d} \left[\langle \mathrm{N}^T_{d',s} \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} \frac{1}{2} \left(\frac{\langle \boldsymbol{u}_s \rangle_{\boldsymbol{i}+\boldsymbol{e}^{d'}} - \langle \boldsymbol{u}_s \rangle_{\boldsymbol{i}-\boldsymbol{e}^{d'}}}{2h} + \frac{\langle \boldsymbol{u}_s \rangle_{\boldsymbol{i}+\boldsymbol{e}^d+\boldsymbol{e}^{d'}} - \langle \boldsymbol{u}_s \rangle_{\boldsymbol{i}+\boldsymbol{e}^d-\boldsymbol{e}^{d'}}}{2h} \right) \right] \left. \right\}. \end{aligned} \tag{25}$$

The transverse metrics components, $\langle \mathrm{N}^T_{d' \neq d,s} \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}$, can either be computed by numerical quadrature or averaged from nearby faces in direction d'.

We then compute the artificial viscosity coefficient $\nu_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}$ by

$$\nu_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = (\Delta x_d \lambda)_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} \min\left(\frac{((\Delta x_d \lambda)_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d})^2}{(c^{\min}_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d})^2 \beta}, 1\right) \tag{26}$$

at faces where $\lambda_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} < 0$; otherwise, $\nu_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}$ is set to zero. Here, $\beta = 0.42$ and $c^{\min}_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}$ is the minimum speed of sound given by

$$c^{\min}_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = \min(c_{\boldsymbol{i}+\boldsymbol{j}}, c_{\boldsymbol{i}+\boldsymbol{e}^d+\boldsymbol{j}}), \boldsymbol{j}_d = 0 \text{ and } \max_{d'=1\ldots D}(|\boldsymbol{j}_{d'}| \leq 1).$$

The physical mesh spacing on a face is computed by $(\Delta x_d)_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = h/\|J^{-1}\mathrm{N}_d^T\|$ with J averaged from adjacent cells. The artificial viscosity is then applied as follows

$$\vec{\nabla}_x \langle \mathbf{U} \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = J^{-1}\mathrm{N}_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} \cdot \vec{\nabla}_\xi \langle \mathbf{U} \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}, \tag{27}$$

$$\langle \mu_d \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = \alpha \nu_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} \frac{\partial \langle \mathbf{U} \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}}{\partial x_d} \Delta x_d, \text{with } \alpha = 0.3, \tag{28}$$

$$\langle J\mathbf{U} \rangle_{\boldsymbol{i}} := \langle J\mathbf{U} \rangle_{\boldsymbol{i}} - \frac{\Delta t}{h} \sum_d \left(\langle \mathrm{N}^T \mu_d \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} - \langle \mathrm{N}^T \mu_d \rangle_{\boldsymbol{i}-\frac{1}{2}\boldsymbol{e}^d} \right). \tag{29}$$

4.3. Time-Marching Constraint

For the Runge-Kutta time-marching scheme, the most severe constraint on the time step results from the first-order fluxes produced by the limiter [27]. An analysis, performed by Colella et al. [27], revealed that a stable time-step in a cell for first-order upwind fluxes is given by

$$\frac{\Delta t}{h} \sum_{d=1}^{\mathbf{D}} \left| \boldsymbol{v} \cdot \boldsymbol{e}^d \right| \lessapprox 1.3925. \tag{30}$$

This estimate is based on a constant-coefficient problem and ignores contributions from transverse gradients. For the Euler equations, the maximum acoustic wave speed, $\boldsymbol{v}$, is determined in computational space. The contravariant velocity, aligned with the grid in computational space, is obtained from $J^{-1}\mathrm{N}^T\boldsymbol{u}$ and the speed of sound, c, is scaled by $h/\Delta x = \|J^{-1}\mathrm{N}^T\|$. In practice, N^T is known on the faces so we first determine the face-averaged primitive state, $\langle \mathbf{W} \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}$, using second-order methods. For each direction, the maximum acoustic wave

speed in a cell is determined from the average of the opposing faces as

$$\left|\boldsymbol{v}\cdot\boldsymbol{e}^d\right| = \frac{1}{2J}\sum_{\pm=+,-}\left(\left|\mathrm{N}_d^T\boldsymbol{u}\right| + c\left\|\mathrm{N}_d^T\right\|\right)_{\boldsymbol{i}\pm\frac{1}{2}\boldsymbol{e}^d}\,. \tag{31}$$

A global minimum of the local time-step in each cell is used as the global time step. For multi-level AMR grids, the time-step constraint is also determined on finer levels and scaled to the coarsest grid before computing a global minimum. The global time step is then scaled by the grid refinement-ratio,

$$\Delta t^{\ell+1} = \frac{\Delta t^\ell}{n_{ref}^\ell}\,,$$

before being sent back to the finer levels.

5. Adaptive Mesh Refinement for Mapped Grids

To implement adaptive mesh refinement, we make use of the Chombo library for parallel AMR [29] and follow the strategies used therein. The concept of the discretized problem domain as a subset of an integer lattice, $\Gamma \subset \mathbb{Z}^D$, is extended to a nested hierarchy of grids, $\Gamma^0 \ldots \Gamma^{\ell_{\max}}$ with $\Gamma^\ell = \mathcal{C}_{n_{ref}^\ell}(\Gamma^{\ell+1})$. The integer n_{ref}^ℓ is the refinement ratio between level ℓ and $\ell+1$ so that the Cartesian mesh spacings $h^\ell = n_{ref}^\ell h^{\ell+1}$. The operator $\mathcal{C}$ is a coarsening of the grid. Adaptive mesh refinement calculations are performed on a hierarchy of nested meshes $\Omega^\ell \subset \Gamma^\ell$, with $\Omega^\ell \supset \mathcal{C}_{n_{ref}^\ell}(\Omega^{\ell+1})$. 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

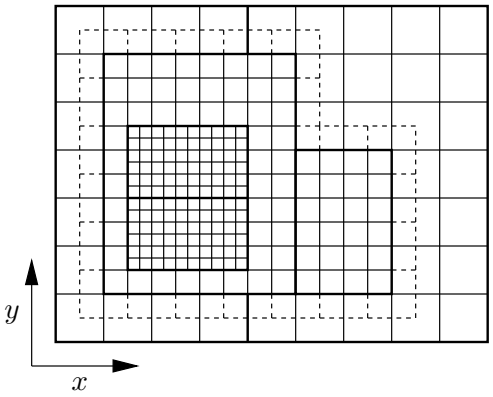


Figure 1: A three-level grid with $n_{ref} = 2$ and nesting sufficient for one cell to separate level $\ell+1$ from level $\ell-1$. A single layer of invalid ghost cells surrounding the middle grid level is shown by dashed lines.

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. Two boxes are shown on each grid level in Figure 1 and are outlined with thick lines. 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).

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 (a rearrangement of (14)),

$$\langle \mathbf{U} \rangle = \frac{1}{J}\left(\langle J\mathbf{U} \rangle - \frac{h^2}{12} \sum_d^D \frac{\partial \bar{\mathbf{U}}}{\partial \xi_d} \frac{\partial J}{\partial \xi_d} \right) \tag{32}$$

   before using the methods described in Section 4.1 to find $\langle \vec{\mathbf{F}} \rangle$, compute $\langle \mathrm{N}_d^T \vec{\mathbf{F}} \rangle$ using (15), and evolve the semi-discrete form (7) with a Runge-Kutta time-stepping method. In (32), only second-order accuracy is required for the gradients. Therefore, $\bar{\mathbf{U}}$ can be obtained from $\langle J\mathbf{U} \rangle / \langle J \rangle$ and gradients of J can be evaluated using $\langle J \rangle$.

3. Interpolate to the invalid ghost cells surrounding level $\ell+1$. Interpolation in time requires a careful tuning to the Runge-Kutta advancement on the

coarser level [2]. In space, a least-squares algorithm is used to compute the interpolating polynomial in each coarse cell. Our expectation was that 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. However, through verification, we found it necessary to impose a conservation constraint for this interpolation. The details are presented in Section 5.3.

4. Start level $\ell+1$ at step 1. Level $\ell+1$ is refined in time (sub-cycled) with a time step $\Delta t^{\ell+1} = \Delta t^{\ell}/n^{\ell}_{ref}$.
5. Average the solution from $\ell+1$ to underlying cells on level ℓ and correct fluxes at coarse-fine interfaces to ensure conservation [30, 10]. In the latter correction, the fluxes computed on the coarse grid are replaced by fluxes computed on the fine grid at the interface between the two grid levels.

5.1. Freestream Preservation at AMR Interfaces

Freestream preservation at AMR interfaces must also be maintained. The solution will be freestream-preserving as long as $\vec{\mathcal{N}}_s$ is made consistent between the (hyper)edges that are shared between the two levels, and $\langle \mathrm{N}^T \rangle$ is computed from that consistent value. This concept is shown in Figure 2 where $\langle \mathrm{N}^T_{d=1} \rangle^{\ell}$ and $\langle \mathrm{N}^T_{d=2} \rangle^{\ell}$ are computed using the same $\vec{\mathcal{N}}_s{}^{\ell} = \sum \vec{\mathcal{N}}_s{}^{\ell+1}$.

However, on faces in $\Omega^{\ell}_{\text{valid}}$, adjacent to and orthogonal to the exterior of $\Omega^{\ell+1}$ (e.g., consider the face where $\langle \mathrm{N}^T_{d=2} \rangle^{\ell}$ 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 $\vec{\mathcal{N}}_s$ are smoothly varying functions of space. The cancellation is lost because $\vec{\mathcal{N}}_s{}^{\ell}$ is overwritten with sums from the finer level on edges within the interface. In other words, when calculating $\langle \mathrm{N}^T_{d=2} \rangle^{\ell}$ using (22), the methods used to compute $\vec{\mathcal{N}}_s{}^{\ell}$ are not continuous across the face. For this reason, it is necessary to have $\vec{\mathcal{N}}_s$ computed to $O(h^{p+1})$ accuracy in order for the $\langle \mathrm{N}^T_d \rangle$ to be accurate to $O(h^p)$ everywhere.

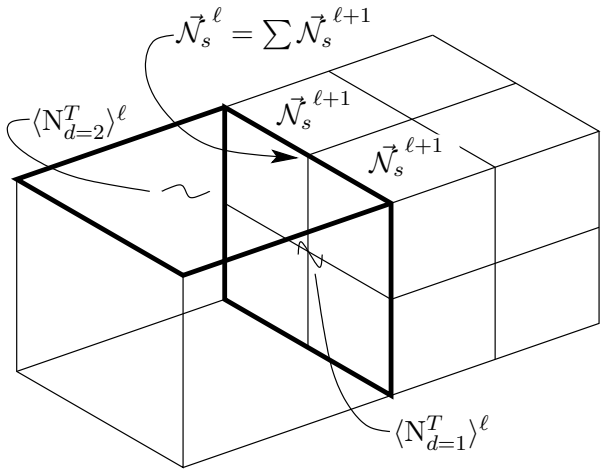


Figure 2: By overwriting $\vec{\mathcal{N}}_s$ with a sum of fine-grid values, we ensure that $\langle \mathrm{N}_{d=1}^T \rangle^\ell = (\sum \langle \mathrm{N}_{d=1}^T \rangle^{\ell+1})/(n_{ref}^\ell)^{\mathbf{D}-1}$ on the coarse face (shown in bold) that overlaps the fine faces. Additionally, since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

5.2. Regridding

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, $\{\Omega^{\ell,\mathrm{new}}\}_{\ell=\ell_{\mathrm{base}}+1,\ldots,\ell_{\mathrm{max}}}$ 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 cells on the mapped grid 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 $\vec{\mathcal{N}}_s$, using (23), on all codimension two (hyperedge) centerings of Ω^ℓ. We assume in this work that a single analytic expression of the mapping applies everywhere in the domain, allowing $\vec{\mathcal{N}}_s$ to be easily computed at any resolution. Next, the $\vec{\mathcal{N}}_s$ from a finer level, if it exists, are summed to this level, overwriting the $\vec{\mathcal{N}}_s$ just computed where there is overlap. The normal components of $\langle \mathrm{N}^T \rangle$ are computed using (22). We note that a fourth-order accurate approximation to the cell volumes can be determined from our usual divergence formula by instead letting $\vec{\mathrm{F}}(\vec{x}) = \vec{x}/D$. In that case, we have

$$\begin{aligned}\frac{1}{h^D}\int\limits_{x(V_{\boldsymbol{i}})} \mathrm{d}\vec{x} =& \langle J \rangle_{\boldsymbol{i}} \\ =& \frac{1}{h^D}\int\limits_{V_{\boldsymbol{i}}} \vec{\nabla}_\xi \cdot (\mathrm{N}^T \vec{\chi}) \mathrm{d}\vec{\xi} \\ =& \frac{1}{h}\sum_{d=1}^{D} \langle \mathrm{N}_d^T \vec{\chi} \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} - \langle \mathrm{N}_d^T \vec{\chi} \rangle_{\boldsymbol{i}-\frac{1}{2}\boldsymbol{e}^d} ,\end{aligned} \tag{33}$$

where $\vec{\chi}(\vec{\xi}) = \frac{\vec{x}(\vec{\xi})}{D}$. Continuing with the geometry calculation, $\langle \mathrm{N}_d^T \vec{\chi} \rangle^{\ell,\text{new}}_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}$, 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, $\langle J \rangle^\ell$ is computed using (33). Because $\vec{\mathcal{N}}_s$ and $\langle \mathrm{N}_d^T \vec{\chi} \rangle$ are overwritten from finer levels, the metrics $\langle \mathrm{N}^T \rangle$ and $\langle J \rangle$ are made consistent between all the levels of the grid. For example, in Figure 2, $\langle \mathrm{N}^T_{d=1} \rangle^\ell = (\sum \langle \mathrm{N}^T_{d=1} \rangle^{\ell+1})/(n^\ell_{ref})^{D-1}$ on the interface as a result of overwriting $\vec{\mathcal{N}}_s{}^\ell$ with $\sum \vec{\mathcal{N}}_s{}^{\ell+1}$ 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:

$$F_{\delta J,\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = \begin{cases} \langle \mathrm{N}^T \vec{\chi} \rangle^{\ell,\text{new}} - \langle \mathrm{N}^T \vec{\chi} \rangle^{\ell,\text{old}} & \text{on the faces of } \Omega^{\ell,\text{old}} \cap \Omega^{\ell,\text{new}}, \\ 0 & \text{otherwise.} \end{cases} \tag{34}$$

$$F^d_{\delta(J\mathbf{U}),\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = \begin{cases} -\mathbf{U}_{\boldsymbol{i}+\boldsymbol{e}^d}(F^d_{\delta J,\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}) & \text{if } F^d_{\delta J,\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} > 0, \\ -\mathbf{U}_{\boldsymbol{i}}(F^d_{\delta J,\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}) & \text{otherwise.} \end{cases} \tag{35}$$

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,

$$\langle J\mathbf{U} \rangle^{\ell,\text{old}} := \langle J\mathbf{U} \rangle^{\ell,\text{old}} - \frac{1}{h} \sum_{d=1}^{\mathbf{D}} F^d_{\delta(J\mathbf{U}),\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} - F^d_{\delta(J\mathbf{U}),\boldsymbol{i}-\frac{1}{2}\boldsymbol{e}^d} \,. \tag{36}$$

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 $\ell = \ell_{\text{base}}, \ldots, \ell_{\max} - 1$ (where $\ell_{\max}$ may be a new grid level), 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

$$\sum_{\|\boldsymbol{p}\|_1 \leq 3} a_{\boldsymbol{p}} \langle \vec{\xi}^{\boldsymbol{p}} \rangle_{\boldsymbol{j}^\ell} = \langle \mathbf{U} \rangle_{\boldsymbol{j}^\ell} \;,\; \boldsymbol{j}^\ell \in \mathcal{I}(\boldsymbol{i}^\ell) \,,$$

where $\mathcal{I}$ denotes the stencil around cell $\boldsymbol{i}^\ell$, subject to the constraint

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

The term $\langle \vec{\xi}^{\boldsymbol{p}} \rangle$ expands to $\langle \xi_1^{p_1} \xi_2^{p_2} \xi_3^{p_3} \rangle$ (or equivalently $\langle \prod_d \xi_d^{p_d} \rangle$) in three dimensions and $\|\boldsymbol{p}\|_1 \leq 3 \equiv p_1 + p_2 + p_3 \leq 3$ limits the degree of the polynomial. Related to the familiar form $\min_{\mathrm{B}x=d} \|\mathrm{A}x - b\|_2$, $\mathrm{A} = \langle \vec{\xi}^{\boldsymbol{p}} \rangle_{\boldsymbol{j}^\ell}$, $b = \langle \mathbf{U} \rangle_{\boldsymbol{j}^\ell}$, $\mathrm{B} = \langle J\vec{\xi}^{\boldsymbol{p}} \rangle_{\boldsymbol{i}^{\ell+1}}$, $d = \langle J\mathbf{U} \rangle_{\boldsymbol{i}^\ell}$, and $x = a_{\boldsymbol{p}}$. The moments $\langle \vec{\xi}^{\boldsymbol{p}} \rangle$ can be computed analytically, and the $\langle J\vec{\xi}^{\boldsymbol{p}} \rangle$ are computed using the product formula. Given this interpolant, we can construct $\langle J\mathbf{U} \rangle$ on the control volumes at level $\ell + 1$,

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

This interpolation step is both freestream-preserving and conservative. The solution from the old grid level is then copied over the interpolated solution on the new grid where there is overlap. The solution is therefore interpolated only where new grid levels appear.

5.3. Ghost Cells

As mentioned earlier, the grid Ω^ℓ on level ℓ is partitioned into *boxes* that are distributed among processors to perform calculations in parallel. In order to keep the calculations independent, ghost cells surround each box. Away from domain boundaries, two types of ghost cells are defined:

Valid ghost	A ghost cell within the valid region of Ω^ℓ. These result from partitioning and are filled by means of exchange.
Invalid ghost	A ghost cell outside the valid region of Ω^ℓ. These result from grid refinement and are use to couple a finer grid level to a coarser grid level. They are filled by interpolation from the coarser level.

In the Cartesian algorithm [2], 5 layers of ghost cells containing $\langle \mathbf{U} \rangle$ are required to ensure the calculations are independent. In the mapped algorithm, 5 ghosts cells of $\langle \mathbf{U} \rangle$ are also required to compute the flux in computational space. Fewer ghosts cells are needed for the metric terms and $\langle J\mathbf{U} \rangle$. In general, 1 layer of ghost cells is required for $\langle J \rangle$ and $\langle J\mathbf{U} \rangle$ in order to compute the gradients in (32) everywhere using a centered stencil. As $\langle J \rangle$ itself is computed using a product formula involving $\langle \mathrm{N}^T \rangle$, the latter is required on the faces of two layers of ghost cells. Note that $\vec{\mathcal{N}}_s$ is only required on the same ghosts as $\langle \mathrm{N}^T \rangle$ because it is one codimension higher. Although $\langle \mathrm{N}^T \rangle$ is also one codimension higher than $\langle J \rangle$, it is needed on an extra ghost in order to compute the volume flux, $\langle \mathrm{N}^T \vec{\chi} \rangle$, via a product rule, which is then used to compute $\langle J \rangle$.

The calculation of $\langle \mathbf{U} \rangle$ in valid cells is preceded by an exchange of $\langle J\mathbf{U} \rangle$ so that it is available in one layer of valid ghost cells. Equation (32) is applied to obtain $\langle \mathbf{U} \rangle$ in valid cells. A second exchange, this time of $\langle \mathbf{U} \rangle$, is used to fill all five layers of valid ghost cells.

The filling of invalid ghosts is achieved through interpolation from the coarser level. A summary of the ghost cells at an AMR interface is presented in Figure 3 The interpolator must fill $\langle \mathbf{U} \rangle$ in the five ghost cells and obtain $\langle J\mathbf{U} \rangle$ in one ghost cell adjacent to the interface. From the invalid ghost cells, $\langle \mathbf{U} \rangle$ is only used to help compute the flux in the valid cells, and $\langle J\mathbf{U} \rangle$ is only used to compute $\bar{\mathbf{U}}$, so that a centered difference of $\bar{\mathbf{U}}$ can be used in (32) to compute $\langle \mathbf{U} \rangle$ in the valid cells adjacent to the interface.

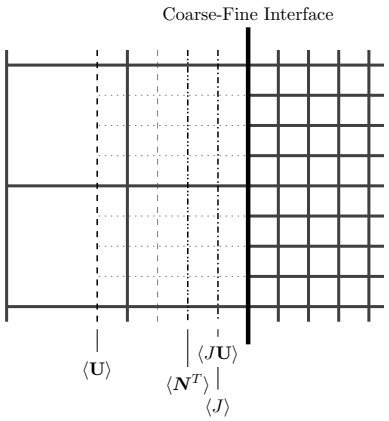


Figure 3: Extent of invalid ghost cells at an AMR interface for several solution variables and grid metrics.

Arguably, a fourth order interpolation should be sufficient to ensure a fourth-order accurate, conservative scheme. However, verification experiments revealed some cases where expected convergence was not observed. It appears necessary to constrain the interpolation so that the sum of $\langle \mathbf{U} \rangle$ in the interpolated fine cells equals $(n_{ref})^D \langle \mathbf{U} \rangle$ in the underlying coarse cells. A least-squares procedure is used to determine the coefficients of an interpolating polynomial from neighboring coarse cells. As the system is over-determined, there is sufficient freedom to constrain the solution. We solve a least-squares system for the coefficients of a polynomial interpolant

$$\sum_{\|\boldsymbol{p}\|_1 \leq 3} a_{\boldsymbol{p}} \langle \vec{\xi}^{\boldsymbol{p}} \rangle_{\boldsymbol{j}^\ell} = \langle \mathbf{U} \rangle_{\boldsymbol{j}^\ell} \ , \ \boldsymbol{j}^\ell \in \mathcal{I}(\boldsymbol{i}^\ell) \, , \tag{37}$$

where $\mathcal{I}$ denotes the stencil around cell $\boldsymbol{i}^\ell$ (see [2] for the interpolation stencils), subject to the constraint

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

The moments $\langle \vec{\xi}^{\boldsymbol{p}} \rangle$ can be computed analytically. To simply the constraint, we follow the procedure of McCorquodale and Colella [2] and modify the moments

to be

$$\vec{\xi}^{\,\boldsymbol{p}} = \prod_d \left(\xi_d^{p_d} - K(p_d)\right) , \tag{38}$$

where

$$K(q) = \begin{cases} \dfrac{1}{2^q(q+1)} & \text{if } q > 0 \text{ and } q \text{ is even,} \\ 0 & \text{otherwise.} \end{cases} \tag{39}$$

With this constant, the moments averaged over coarse cell $\boldsymbol{i}^\ell$ equal 1 if $\|\boldsymbol{p}\|_1 = 0$ and zero otherwise. The effect of this modification is to directly impose the conservation constraint while also yielding the solution for the first coefficient (when cast into form $\mathrm{A}x = b$, the first row of A is $[1, 0, \ldots, 0]$ so that $x_1 = b_1$). The remaining coefficients can be solved using a QR factorization and the method of normal equations, where Q is simply the identity matrix and $\mathrm{R} = 1$. The product of the remaining coefficients and moments for each fine cell equate to balanced deviations from the conserved value, depending on the displacement of the fine cell from the center of the coarse cell.

Although a formal conservation constraint could also be used as is done in the regrid procedure (based on $\langle J\mathbf{U}\rangle$ instead of $\langle\mathbf{U}\rangle$), doing so requires inverting a matrix in each cell to determine the coefficients of the polynomial. This results from $\langle J\rangle$ varying in the cells. By instead imposing a constraint on $\langle\mathbf{U}\rangle$, a matrix inversion must only be performed for each stencil, and can be cached in advance. The simplification imposed by (38) allows for the matrix to be multiplied by the fine-cell displacements in advance. Then, in terms of computational costs, filling a ghost cell only involves a dot product between a cached vector and the right-hand side of (37). Note also that gradients of $\langle\mathbf{U}\rangle$ can be determined from derivatives of the polynomial obtained from the least squares procedure. Using the product formula, the gradients can be used along with $\langle J\rangle$, $\langle\mathbf{U}\rangle$, and a one-sided, first-order approximation of $\vec{\nabla}_\xi\langle J\rangle$ to find $\langle J\mathbf{U}\rangle$ in one layer of invalid ghost cells. As stated earlier, conservation is not strictly required in ghost cells; and imposing a constraint on $\langle\mathbf{U}\rangle$ is sufficient to obtain the expected accuracy in the verification tests we performed.

6. Results and Discussions

The results of several numerical experiments on a variety of mappings are presented to verify that we have indeed retained the properties we desire of our scheme, namely fourth-order accuracy, conservation, and freestream preservation. In the first experiment, the constant solution is not advanced. Refined grid patches are added and removed to verify the freestream preservation property. The second experiment is the advection of a Gaussian density profile to illustrate solution accuracy with AMR. The objective of this example is to show that the regions with large errors in the problem can be resolved with AMR to the same accuracy as using a single grid of the same resolution as the finest AMR level. The third case is a Gaussian acoustic pulse problem, which has a more complex wave structure than the advection of the Gaussian density profile; its analytical solution is unknown. We performed Richardson extrapolation for a 5^{th}-order solution as the "exact" solution in order to verify the error convergence rates both with and without AMR. Additionally, a shear problem is tested. While the fine patches logically follow the physics in the first two problems, in the shear problem, the refined region is fixed in the center of the problem domain. Although not logical, this test exposes the AMR interfaces to strong gradients and errors; a situation that the algorithm must gracefully accommodate. Next, the ability of the scheme to correctly recover wave dynamics is tested by solving Sod's shock tube on a mapped grid. Finally, we demonstrate the effectiveness of AMR with mapped grids in resolving large discontinuities on a shock-ramp problem. Note that the algorithm is multidimensional, but most of our examples are two-dimensional for the convergence studies. Two dimensional cases are adequate for verification of solution accuracy.

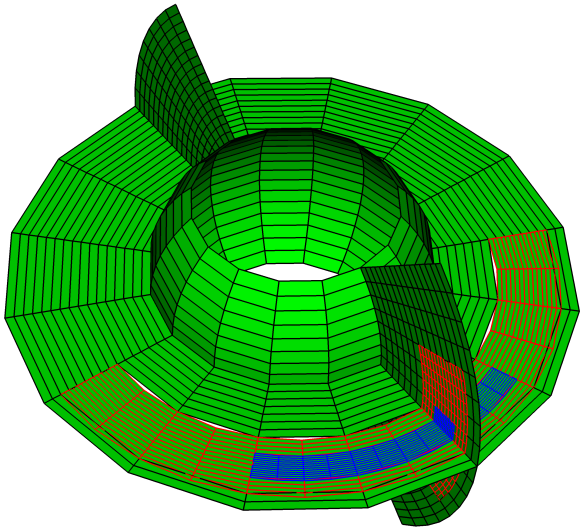


Figure 4: Freestream-preservation test.

6.1. Freestream Preservation

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

$$\begin{aligned} x &= r \sin\varphi \cos\theta \\ y &= r \sin\varphi \sin\theta \\ z &= r \cos\varphi \end{aligned} \tag{40}$$

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 4. 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.

6.2. Advection of the Gaussian Density Profile

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

$$\rho = \rho_0 + s(r)\,\Delta\rho\,\mathrm{e}^{-(100r^2)} \tag{41}$$

where

$$s(r) = \begin{cases} 0 & : \quad |2r^2| >= 1 \\ \cos^6(\pi r^2) & : \quad |2r^2| < 1 \end{cases}, \tag{42}$$

$\rho_0 = 1.4$, $\Delta\rho = 0.14$, and r specifies the distance of a point from the center of the periodic domain, $[0,1]^D$. Smoothing with continuous 5^{th} 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

$$x_d = \xi_d + \prod_{p=1}^{D} \sin(2\pi\xi_p) \qquad d = 1, 2\,. \tag{43}$$

The initial solution in both computational and physical space is shown in Figures 5 and 6, respectively. 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 relevant or meaningful 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 error norms are plotted in Figure 7 showing that the convergence rates on the single-level are 4.00 and those on the AMR grids are between 3.75 and 3.99.

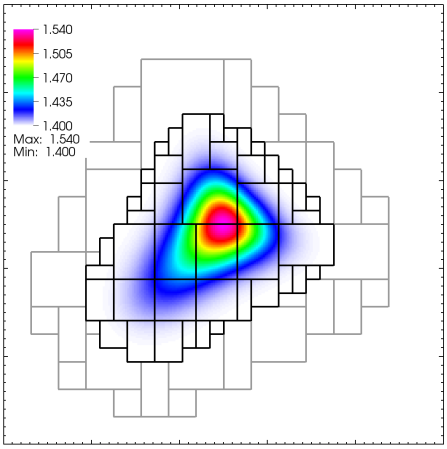


Figure 5: Initialized Gaussian density profile in computational space.

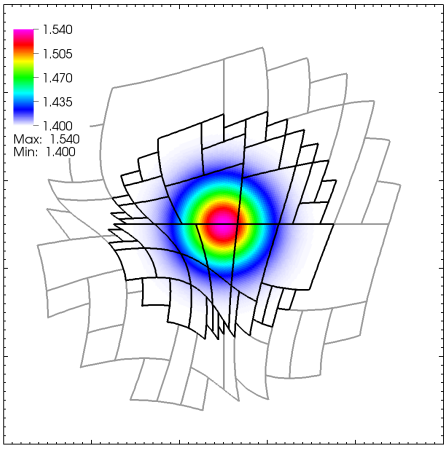


Figure 6: Initialized Gaussian density profile in physical space.

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 and Figure 7 shows the L_∞ curves overlap. 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 $\vec{\mathcal{N}}_s$ to be computed to an order of accuracy greater than that of the scheme as discussed in section 5). 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 $\langle J\mathbf{U}\rangle$ is performed at the start and end of the run. The double precision

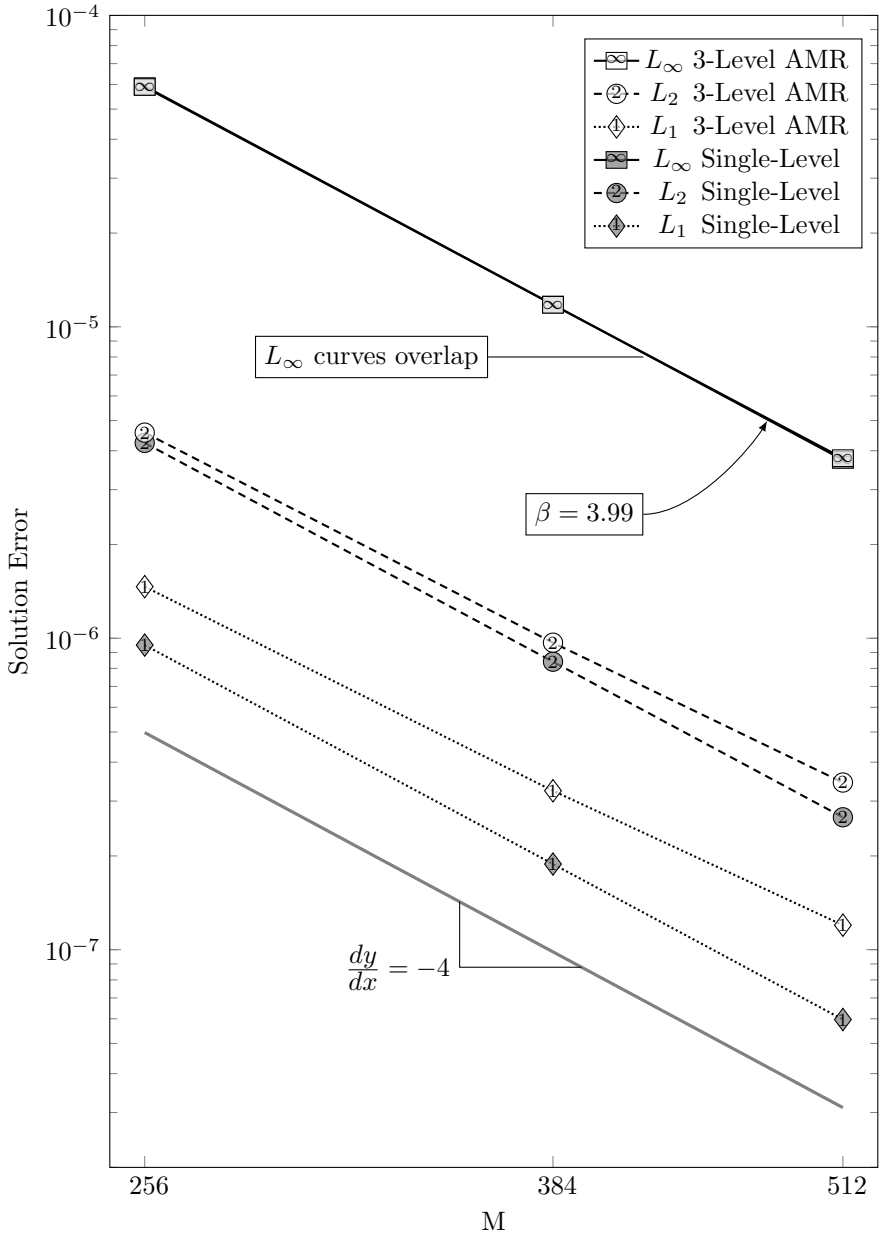


Figure 7: Solution accuracy for advection of a Gaussian profile using both single-level grids and AMR grids. The single level grids are 256×256, 384×384, and 512×512. For the AMR grids, each AMR grid has 3 levels. The resolution of the finest level of each AMR grid matches the resolution of the corresponding single-level grid.

Table 1: Initial and final conserved quantities for advection problem

$J\mathbf{U}$	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

results from the 512×512 AMR case are shown in Table 1.

6.3. Gaussian Acoustic Pulse

A Gaussian acoustic pulse in a polytropic gas is tested in a periodic domain. The initial conditions at a point in the domain are determined by the distance from the center. Initially the velocity is zero, and the density is

$$\rho(r) = \begin{cases} \rho_0 + (\Delta\rho_0)e^{-16r^2}\cos^6(\pi r) & \text{if} \quad \mathrm{r} \leq \frac{1}{2} \\ \rho_0 & \text{otherwise} \end{cases}, \tag{44}$$

where $\rho_0 = 1.4$, $\Delta\rho_0 = 0.14$, and r measures the distance between the point to the center. The smoothing factor $\cos^6(\pi r)$ ensures the condition of $\rho = \rho_0$ on the domain boundaries. For isentropic flows, the initial pressure is

$$p = (\frac{\rho}{\rho_0})^\gamma\,, \quad \text{where } \gamma = 1.4\,.$$

We tested this example in 2D on a single grid level; the size of the three cases are 128×128, 256×256, and 512×512, respectively. Throughout each run, the time step was fixed and set to $\Delta t = 0.192\Delta x$. We computed the convergence of density at time 0.12 when the solution waves have developed and interacted. We also ran this problem on two levels of AMR, with a refinement factor of 2 between the levels. Figures 8 and 9 show the density solution contours at time t = 0.12 in computational and physical space, respectively. The warped mapping from the previous case is reused here.

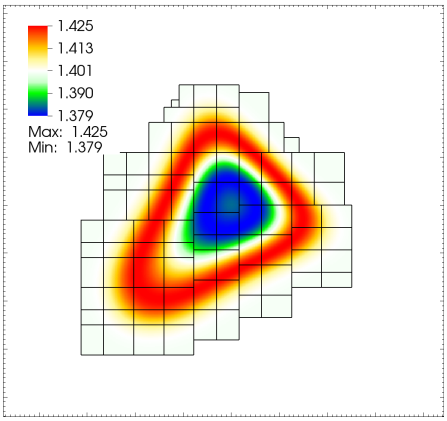


Figure 8: Gaussian pulse density solution contour at $t = 0.12$ in computational space.

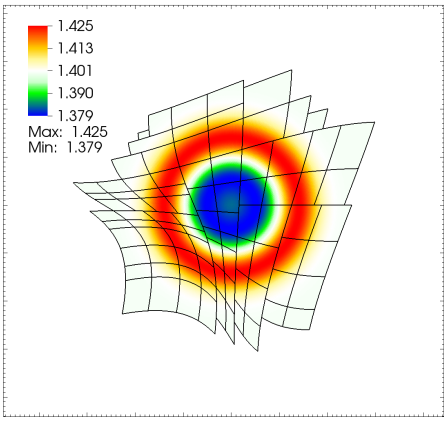


Figure 9: Gaussian pulse density solution contour at $t = 0.12$ in physical space.

Richardson extrapolation is performed on single-level grids to project a more accurate solution based on the numerical solutions obtained at two different resolutions. The more accurate solution is then be used as an "exact" solution to evaluate the convergence rate of the solver. The mapped grids solver makes Richardson extrapolation easy to apply due to the uniform and Cartesian coordinates in the computational domain.

We briefly describe the procedure of using Richardson extrapolation for the convergence study. We run simulations on four single-level grids: 128×128, 256×256, 512×512, and 1024×1024. For each pair, e.g., 128×128 and 256×256, we extrapolate an "exact" solution for the coarser of the two grids. Let Δx be the grid spacing on the grid of 128×128, and, correspondingly, the finer grid spacing is $\Delta x/2$. The "exact" solution, $\mathrm{U}(x)$ is evaluated by

$$\mathrm{U}(x) = \frac{1}{(2^4 - 1)} \left(2^4 \; u\big(x; \frac{\Delta x}{2}\big) - u\big(x; \Delta x\big) \right)$$

where $u(x; \Delta x)$ is the numerical solution on a grid with grid spacing of Δx and $u(x; \Delta x/2)$ represents the solution from the finer grid, averaged to the coarser grid. Note that the "exact" solution has an error of order 5. As the simulation produces averaged values in each cell, this averaging process is exact and introduces no additional error. With an estimate of the "exact" solution,

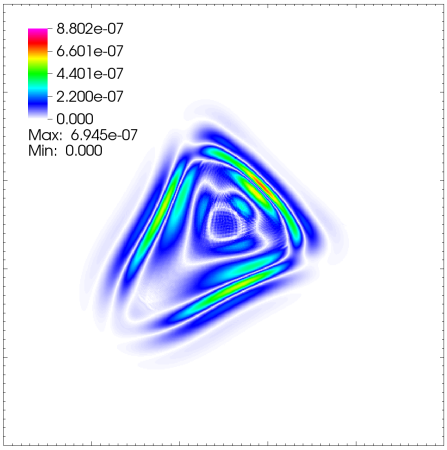


Figure 10: Solution error contour on a single grid $\Delta x = 1/512$ at $t = 0.12$.

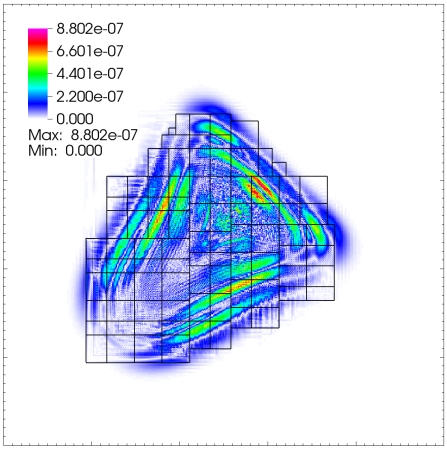


Figure 11: Solution error contour on an AMR grid with fine $\Delta x = 1/512$ at $t = 0.12$.

the error can be found on the coarser grid. This procedure is repeated for the next finer sequence of grids, e.g, 256×256 and 512×512. When the error converges at an asymptotic rate, i.e., the slope becomes independent of the grid size, we know that both the solution error and the extrapolation method is asymptotic.

For the AMR grids, the resolution of the fine level of each AMR grid matches the resolution of the corresponding single-level grid. For example, for the first AMR grid, its fine resolution is equal to that of the 256×256 grid and its coarse resolution is the same as that of the 128×128 grid. The exact solution of each patch on the AMR grid was computed by averaging down the exact solution of the single 512×512 grid, our best estimate of the "exact" solution.

Figures 10 and 11 show solution error distribution on the single grid with $\Delta x = 1/512$ and the AMR grid with the same resolution on the finest grid, respectively. The additional error introduced by AMR is apparent. The error norms are plotted in Figure 12 showing that the convergence rates on the single-level are 4.00 and the convergence rate on AMR grids is approximately on the order of $\mathcal{O}(\Delta x^{15/4})$ in max norm, which is consistent with our previous analysis [2].

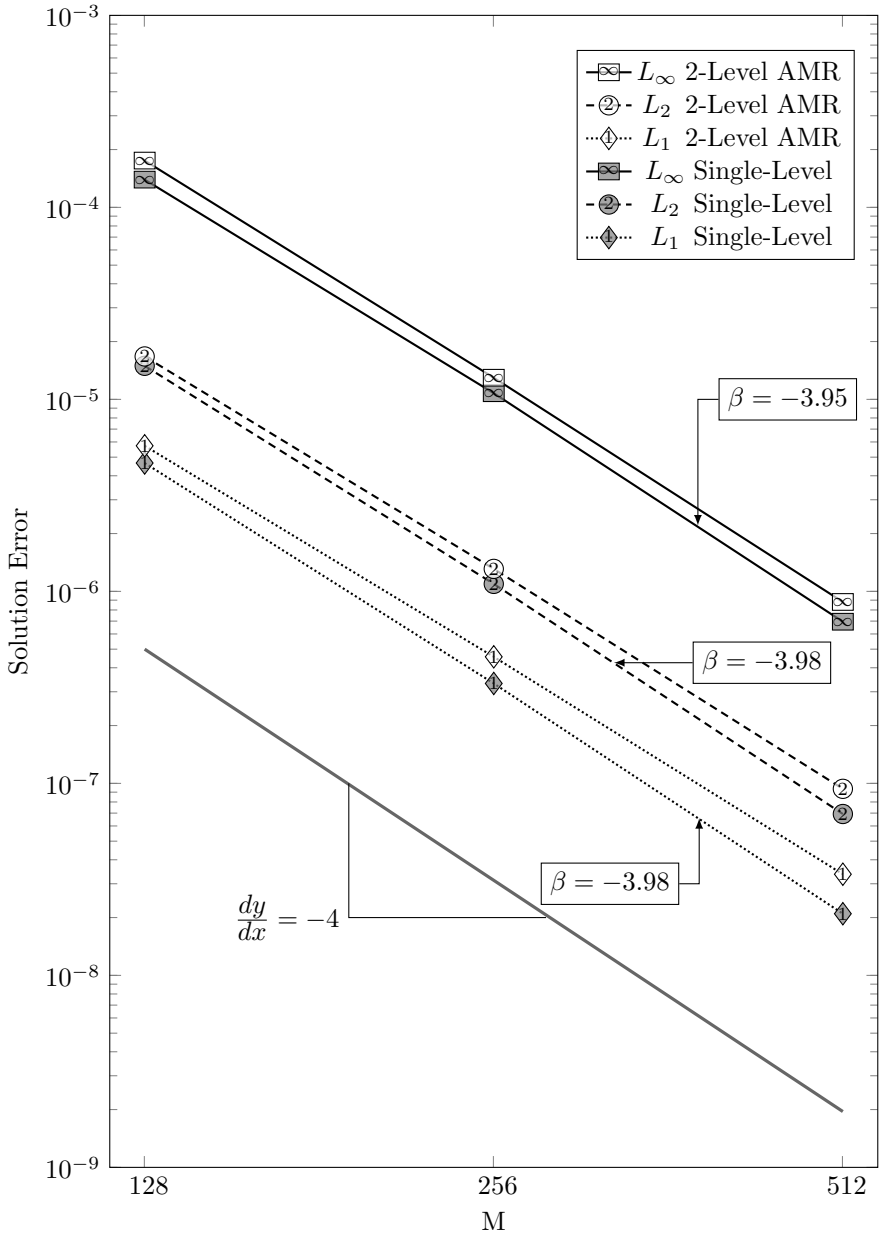


Figure 12: Solution accuracy for acoustic propagation of a Gaussian profile using both single-level grids and AMR grids. The single level grids are 128×128, 256×256, and 512×512. For the AMR grids, each AMR grid has 2 levels. The resolution of the fine level of each AMR grid matches the resolution of the corresponding single-level grid.

6.4. Shear Problem

We consider a 2D polytropic gas shear problem. The initial condition on the unit square $[0,1]^2$ starts with constant density $\rho = 1.4$, pressure $p = 7.0$, and the velocity profile of

$$u(x,y) = \cos(2\pi y)\,, \quad v(x,y) = \cos(2\pi x)\,.$$

The warped mapping described in the advection test was also used here. For this problem, the grid on the finer level was fixed in space from $(0.25, 0.25)$ to $(0.75, 0.75)$. Whereas in the previous cases, the significant physics were well inside the finest level, in this case large gradients are seen at the AMR interface. The solution accuracy is measured at t = 0.12 and plotted in Figure 13, where the slopes for the solution accuracy on the AMR grids are labeled. The error on the AMR grid is slightly larger than that on the single grid with matching resolution. This is not surprising as the interface is exposed to complex flows rather than being situated far away. However, a fourth-order error reduction rate is still achieved.

6.5. Sod's Shock Tube

The ability of the scheme to accurately resolve shock and rarefaction waves was validated by solving Sod's classical shock tube problem [31]. In this problem a diaphragm is imagined in a tube. On the left side, the pressure and density are both given values of 1.0. On the right side, the pressure is 0.1 and the density is 0.125. The diaphragm is instantly removed resulting in a Riemann problem. Rarefaction waves propagate into the high-pressure region on the left. A shock wave propagates into the low-pressure region on the right. The shock wave is followed by a contact surface moving at the flow velocity. The pressure remains constant across the contact surface while the there is a discontinuous change in density.

The problem was solved in a two-dimensional domain (with no flow in the y-direction), again using the warped mapping given by (43). Periodic boundary conditions are used in the y-direction and the solution is terminated before any

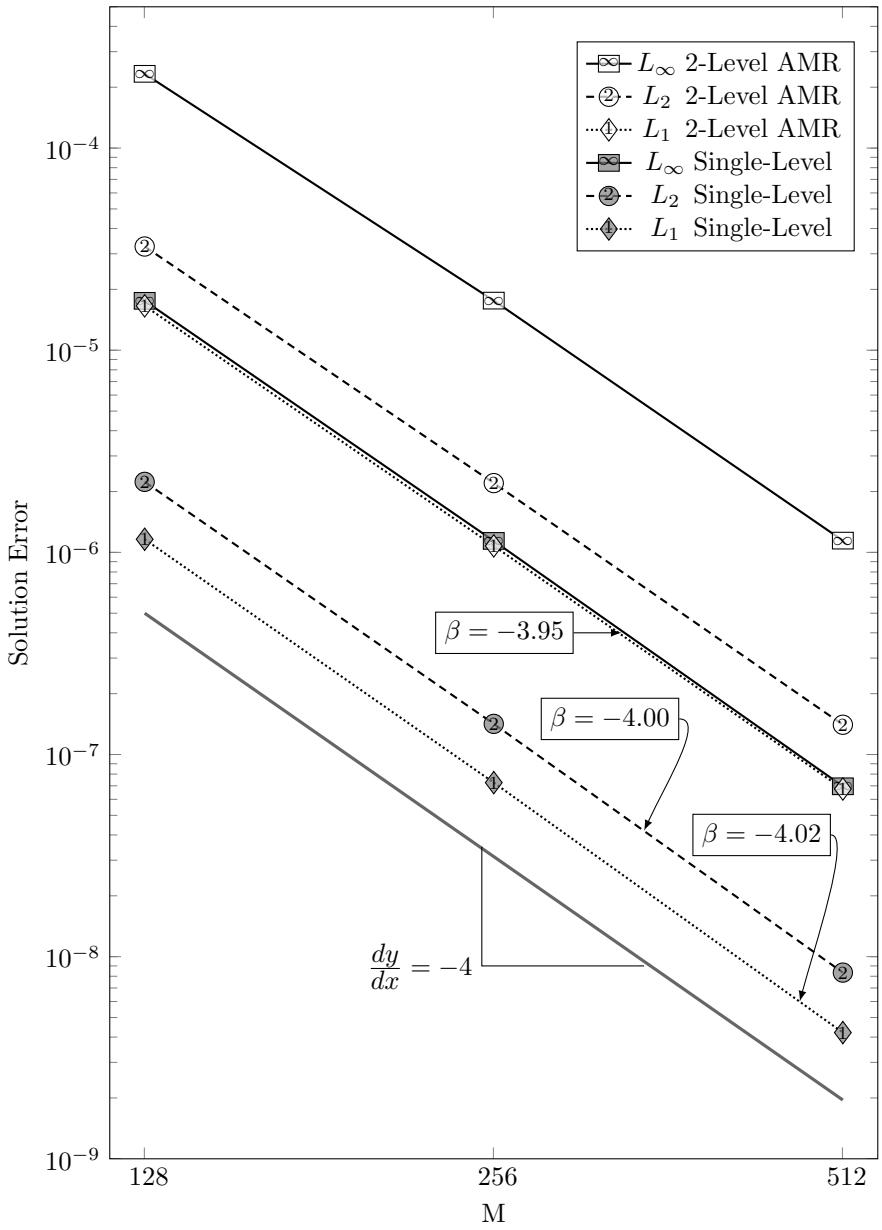


Figure 13: Solution accuracy for the trigonometric shear problem using both single-level grids and AMR grids. The single level grids are 128×128, 256×256, and 512×512. For the AMR grids, each AMR grid has 2 levels. The resolution of the finest level of each AMR grid matches the resolution of the corresponding single-level grid.

waves interact with boundaries in the x-direction. At time $t = 0.2$, the mesh is shown in Fig. 14b. A trace of the density along the line $y = 0.25$ is plotted in Fig. 14a. Along with an analytical solution, results are plotted for a single grid level of size 64×64 and a three-level AMR grid where two finer levels are added on top of the 64×64 base grid. Each new grid level refines the mesh by a factor of two. The computed solution closely follows the analytical solution and the speeds of all waves are properly captured. AMR is shown to be quite effective at reducing the error near the shock and contact discontinuities.

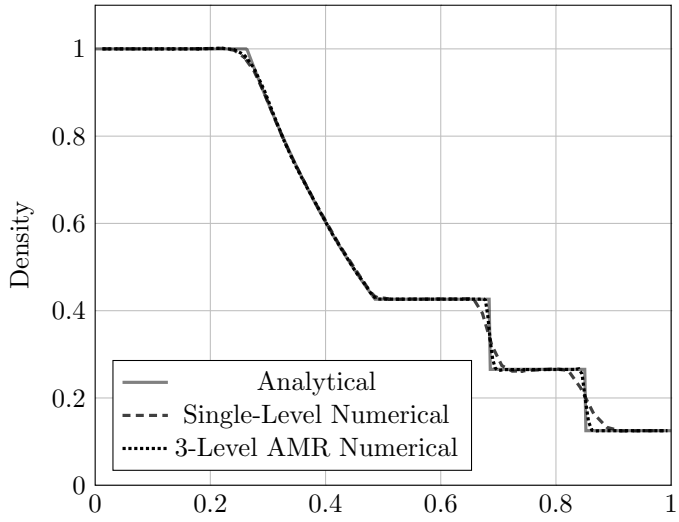
In Fig. 15, the mesh and contours of density are plotted at the earlier time $t = 0.06$. In this figure, AMR is used not only to resolve the discontinuities, but also to add mesh resolution in smooth flow where the warping otherwise creates cells that are too large. Cells are tagged for refinement when the magnitude of the relative gradient of density exceeds some threshold (in this case, the threshold is set to 0.15),

$$\text{refine cell } \boldsymbol{i} \text{ if } \left(\sum_{d=1}^{D} \left(\frac{\rho_{\boldsymbol{i}+\boldsymbol{e}^d} - \rho_{\boldsymbol{i}-\boldsymbol{e}^d}}{\rho_{\boldsymbol{i}+\boldsymbol{e}^d} + \rho_{\boldsymbol{i}-\boldsymbol{e}^d}} \right)^2 \right)^{\frac{1}{2}} \geq 0.15\,. \tag{45}$$

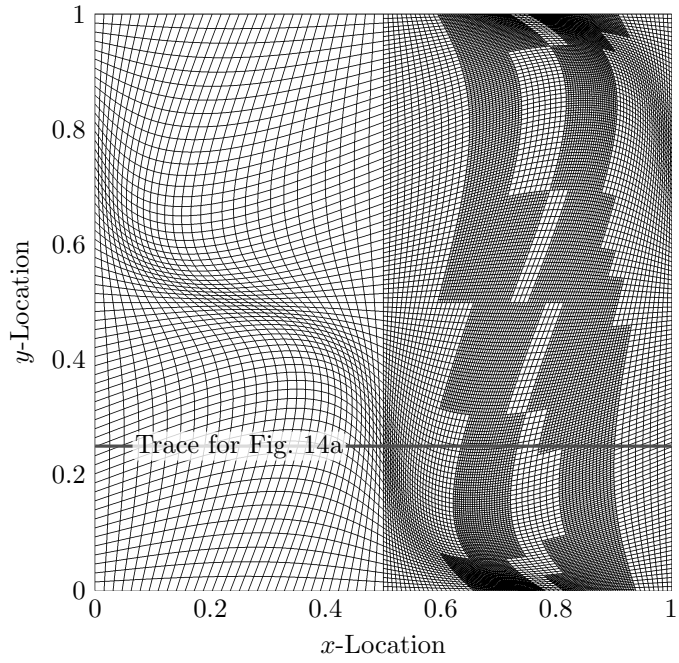
This operation is applied to the primitive variables on the grid in computational space.

6.6. Woodward-Colella 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 [32]. In front of the shock, the flow conditions are $p = 1$, $\rho = \gamma$, and $\vec{u} = 0$. Conditions behind the shock are set using shock relations for a wave speed of Mach 10. The left side of the domain experiences supersonic inflow while the right side is quiescent; at both extents, Dirichlet boundary conditions are used. At the top side of the domain, Dirichlet conditions are also used, but the conditions are adjusted based on the analytical position of the shock wave. In addition, a Riemann problem is solved between the imposed Dirichlet state and the interior solution at the top



(a) Density at $y = 0.25$. The single level results were obtained on a mesh sized 64×64. The AMR results add two levels of refinement, each with a 2 times increase in resolution.



(b) AMR mesh with three levels. The base grid is 64×64.

Figure 14: Sod's shock tube at $t = 0.2$.

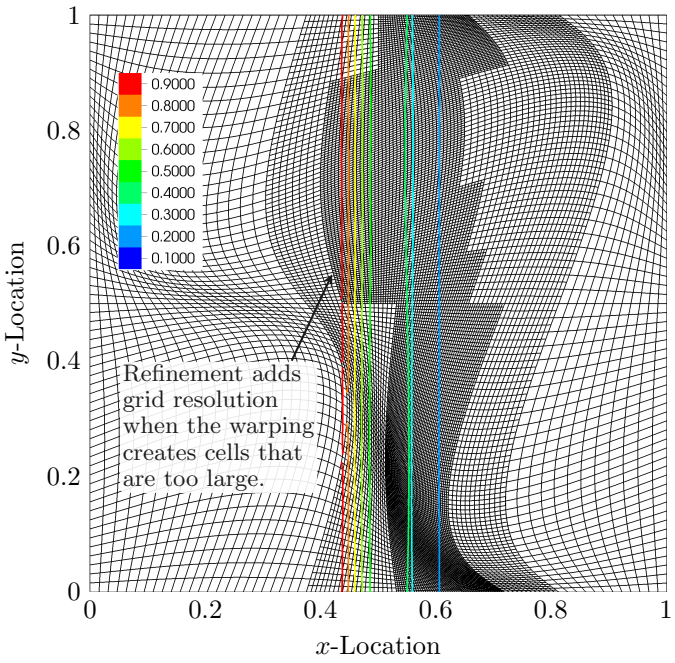


Figure 15: Mesh at time $t = 0.06$. In the expansion fan, the AMR adds mesh resolution where the mapping otherwise causes the cells to be too large. From left to right, the contours of density mark the locations of the rarefaction wave, contact surface, and shock wave.

boundary. Otherwise, there is a potential to trap small quantities of conserved state in between the advancing shock wave and the imposed Dirichlet boundary conditions. Adjustments along the slip wall at the bottom of the domain include setting the wall-normal velocity to zero (a change in velocity given by adding Δu) and then applying the acoustic correction

$$\Delta p = \rho c \Delta u \tag{46}$$

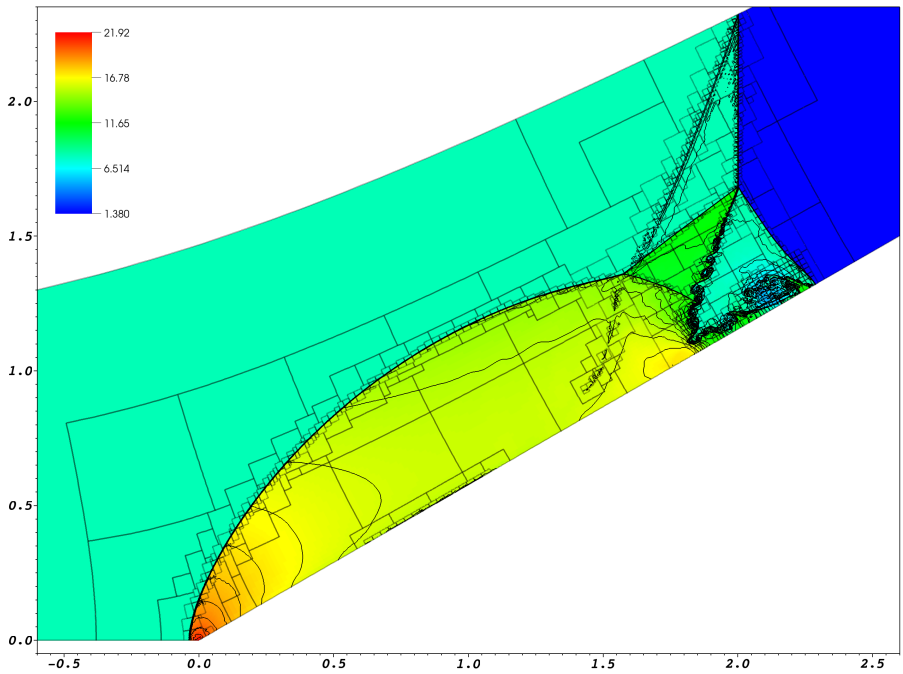


Figure 16: Contours of density are shown in physical space along with the shaded boundaries of boxes belonging to various levels of grid refinement.

Three levels of refinement above a base grid of size 96×24 are used with a four-times increase in resolution at each level. A Schwarz-Christoffel mapping [33] is used to generate the ramp. The solution is shown in Figure 16 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

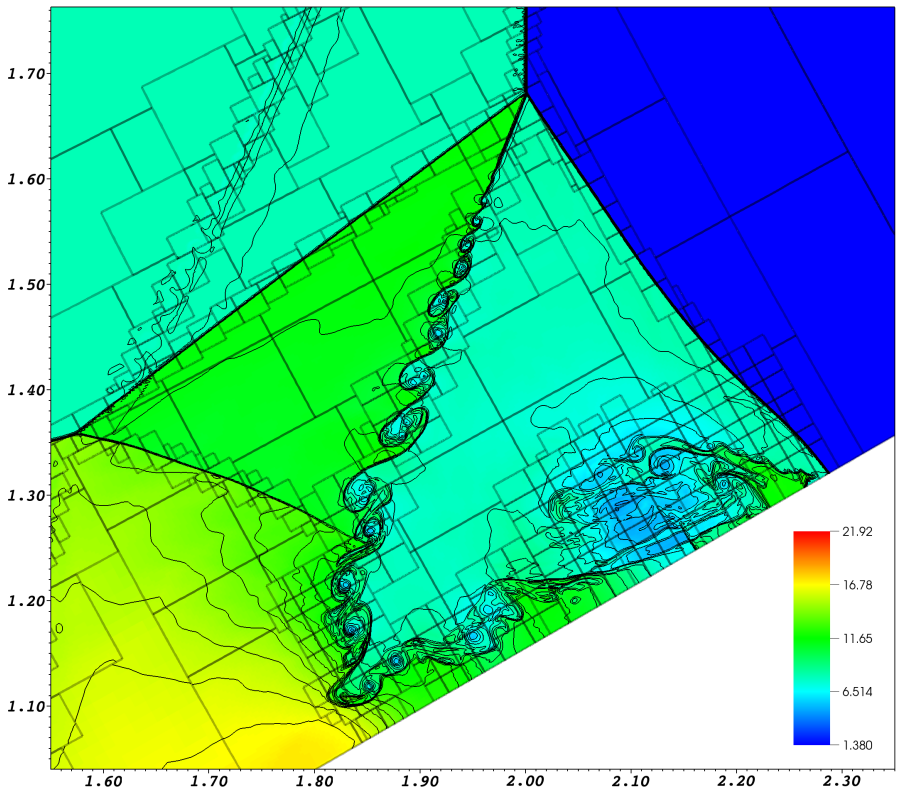


Figure 17: A closeup to show details of fluid instabilities along the slip line as shown in Figure 16.

Rayleigh-Taylor instability. These features are shown in greater detail in Figure 17. Some artifacts are also present, resulting from specifying an infinitely thin shock for initial conditions and at the upper boundary.

The effect of AMR on the time required to obtain a solution is quite dramatic for this example. In experiments performed using MPI across 40 cores of an Intel E5-2670 v2 architecture (2 $\times$ CPU), a solution on a single-grid at the same resolution as the finest AMR level took 42 times longer than the AMR solution!

7. Conclusion

We have presented a fourth-order finite-volume method for obtaining solutions to hyperbolic conservation laws on mapped grids with adaptive mesh refinement. The originality of this work lies in the numerical treatment to ensure that the freestream-preservation property is retained in the case of combining the high-order finite-volume method for mapped grids with the adaptive mesh refinement scheme. For refined levels, we achieve a solution with the freestream-preserving property by ensuring that $\vec{\mathcal{N}}_s$ is consistent between the (hyper)edges that are shared between the two levels, and $\langle \mathrm{N}^T \rangle$ is computed from this consistent value. Finally, we verified the scheme is fourth-order accurate, freestream-preserving, conservative, and demonstrated the technology on an unsteady Mach-reflection problem featuring strong discontinuities.

8. Acknowledgment

This work was performed under the auspices of the U.S. Department of Energy (DOE) by Lawrence Livermore National Laboratory under contract number DE-AC52-07NA27344 and by DOE contracts from the ASCR Applied Math Program. Research at the Lawrence Berkeley National Laboratory was supported by the Office of Advanced Scientific Computing Research of the US Department of Energy under contract number DE-AC02-05CH11231. Research at the Colorado State University was supported by DOE under contract number DE-EE0006086. The authors would also like to thank Louis Howell and Milo Dorr for providing comments on notation and notes that were included into the appendix.

Appendix A. Derivations for Mapped Grids

Appendix A.1. Relationship Between Center and Average Values

Averages $\langle\cdot\rangle$ of a quantity, w, on a face in direction d of a D-dimensional cell on a Cartesian grid are given by

$$\langle w\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = \frac{1}{h^{D-1}}\int_{A_x} w(\vec{x})\mathrm{d}A_x\,. \tag{A.1}$$

Replacing $w(\vec{x})$ using a Taylor-series expansion about the center of the face, up to $O(h^4)$, and integrating over the face of the cell results in (for $D=3$, $\vec{x}=(x,y,z)$, $d=z$)

$$\langle w\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^z} = \frac{1}{h^2}\int_{x_c-\frac{h}{2}}^{x_c+\frac{h}{2}}\int_{y_c-\frac{h}{2}}^{y_c+\frac{h}{2}}\left(w_c + \frac{1}{2}\frac{\partial^2 w}{\partial x^2}(x-x_c)^2 + \frac{1}{2}\frac{\partial^2 w}{\partial y^2}(y-y_c)^2\right)\mathrm{d}y\,\mathrm{d}x\,. \tag{A.2}$$

All coordinate terms with odd power from the Taylor-series expansion are not included because they cancel out when integrated over the face. For a face in coordinate direction d, this can be written generally as

$$\boxed{\langle w\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = w_c + \frac{h^2}{24}\sum_{d'\neq d}\frac{\partial^2 w}{\partial x_{d'}^2} + O(h^4)}\,, \tag{A.3}$$

for any number of dimensions. By the same arguments, the average at the center of a cell can be written as

$$\boxed{\langle w\rangle_{\boldsymbol{i}} = w_c + \frac{h^2}{24}\sum_{d}\frac{\partial^2 w}{\partial x_{d}^2} + O(h^4)}\,, \tag{A.4}$$

where subscript c is now the value at the center of the cell. Equations (A.3) and (A.4) can be used to interchange between average and center values at $O(h^4)$ accuracy.

Replacing $\langle w\rangle$ with $\langle uv\rangle$, one has on a face

$$\langle uv\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = u_c v_c + \frac{h^2}{24}\sum_{d'\neq d}\left(u\frac{\partial^2 v}{\partial x_{d'}^2} + 2\frac{\partial u}{\partial x_{d'}}\frac{\partial v}{\partial x_{d'}} + v\frac{\partial^2 u}{\partial x_{d'}^2}\right) + O(h^4)\,. \tag{A.5}$$

Using (A.3) to replace u_c and v_c results in

$$\langle uv\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = \langle u\rangle\langle v\rangle + \frac{h^2}{24}\sum_{d'\neq d}\left[(u-\langle u\rangle)\frac{\partial^2 v}{\partial x_{d'}^2} + 2\frac{\partial u}{\partial x_{d'}}\frac{\partial v}{\partial x_{d'}} + (v-\langle v\rangle)\frac{\partial^2 u}{\partial x_{d'}^2}\right] + O(h^4)\,. \tag{A.6}$$

From (A.3), it can be seen that the difference between u and $\langle u \rangle$ is $O(h^2)$ (assuming $u = u_c$). Therefore, these terms do not affect the accuracy up to fourth order such that

$$\boxed{\langle uv \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = \langle u \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} \langle v \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} + \frac{h^2}{12} \sum_{d' \neq d} \frac{\partial u}{\partial x_{d'}} \frac{\partial v}{\partial x_{d'}} + O(h^4)} \,. \tag{A.7}$$

Similarly, at cell centers,

$$\boxed{\langle uv \rangle_{\boldsymbol{i}} = \langle u \rangle_{\boldsymbol{i}} \langle v \rangle_{\boldsymbol{i}} + \frac{h^2}{12} \sum_{d} \frac{\partial u}{\partial x_d} \frac{\partial v}{\partial x_d} + O(h^4)} \,. \tag{A.8}$$

Appendix A.2. Face Average Interpolation

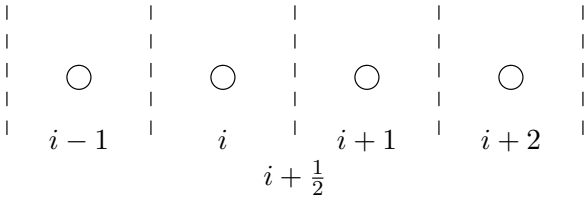


Figure A.18: Centered four-cell stencil for computing face averages.

We start with the assumption that the face average can be interpolated from the indefinite integrals of w in each cell $W = \int_\xi w(\xi) d\xi$ such that

$$W(\xi_{i+\frac{1}{2}}) = W_{i+\frac{1}{2}} = \sum_{k>i} \langle w \rangle_k \Delta\xi_k \,, \tag{A.9}$$

where index $i+\frac{1}{2}$ is used to represent any face in Fig. A.18 ($i = \{i-1, \ldots, i+2\}$). The interpolating function $W(\xi) = a + b\xi + c\xi^2 + d\xi^3 + e\xi^4$ is defined from the five faces, starting at the location of $i - \frac{3}{2}$ in the stencil:

$$\begin{cases} W_{i-\frac{3}{2}} = 0 \\ W_{i-\frac{1}{2}} = \langle w \rangle_{i-1} h \\ W_{i+\frac{1}{2}} = \big(\langle w \rangle_{i-1} + \langle w \rangle_i\big) h \\ W_{i+\frac{3}{2}} = \big(\langle w \rangle_{i-1} + \langle w \rangle_i + \langle w \rangle_{i+1}\big) h \\ W_{i+\frac{5}{2}} = \big(\langle w \rangle_{i-1} + \langle w \rangle_i + \langle w \rangle_{i+1} + \langle w \rangle_{i+2}\big) h \end{cases} . \tag{A.10}$$

Replacing the left-hand side of above equation with appropriate values of interpolating function and rearranging the equations, we can solve a linear system, $Ax = b$, for the coefficients in the interpolating function. Differentiating the resulting interpolant yields the formula for computing face-averaged values, $\langle w \rangle_f = b + 2c\xi + 3d\xi^2 + 4e\xi^3$. Evaluating this result at $\xi = 2h$ yields

$$\boxed{\langle w \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = \frac{7}{12}\left(\langle w \rangle_{\boldsymbol{i}} + \langle w \rangle_{\boldsymbol{i}+\boldsymbol{e}^d}\right) - \frac{1}{12}\left(\langle w \rangle_{\boldsymbol{i}-\boldsymbol{e}^d} + \langle w \rangle_{\boldsymbol{i}+2\boldsymbol{e}^d}\right) + O(h^4)}\,. \quad \text{(A.11)}$$

Appendix A.3. Derivation of the Metric Factors in 3-D

Consider a smooth 3-D mapping $(\xi_1, \xi_2, \xi_3) \rightarrow (x_1, x_2, x_3)$ and its inverse. We define the matrices

$$\vec{\nabla}_\xi \vec{x} \equiv \frac{\partial(x_1, x_2, x_3)}{\partial(\xi_1, \xi_2, \xi_3)} = \begin{bmatrix} \dfrac{\partial x_1}{\partial \xi_1} & \dfrac{\partial x_1}{\partial \xi_2} & \dfrac{\partial x_1}{\partial \xi_3} \\ \dfrac{\partial x_2}{\partial \xi_1} & \dfrac{\partial x_2}{\partial \xi_2} & \dfrac{\partial x_2}{\partial \xi_3} \\ \dfrac{\partial x_3}{\partial \xi_1} & \dfrac{\partial x_3}{\partial \xi_2} & \dfrac{\partial x_3}{\partial \xi_3} \end{bmatrix} \quad \text{(A.12)}$$

$$\vec{\nabla}_x \vec{\xi} \equiv \frac{\partial(\xi_1, \xi_2, \xi_3)}{\partial(x_1, x_2, x_3)} = \begin{bmatrix} \dfrac{\partial \xi_1}{\partial x_1} & \dfrac{\partial \xi_1}{\partial x_2} & \dfrac{\partial \xi_1}{\partial x_3} \\ \dfrac{\partial \xi_2}{\partial x_1} & \dfrac{\partial \xi_2}{\partial x_2} & \dfrac{\partial \xi_2}{\partial x_3} \\ \dfrac{\partial \xi_3}{\partial x_1} & \dfrac{\partial \xi_3}{\partial x_2} & \dfrac{\partial \xi_3}{\partial x_3} \end{bmatrix}, \quad \text{(A.13)}$$

and must have

$$\vec{\nabla}_\xi \vec{x} \vec{\nabla}_x \vec{\xi} = \mathbf{I}\,. \quad \text{(A.14)}$$

Let

$$\begin{aligned} J &\equiv \det(\vec{\nabla}_\xi \vec{x}) \quad \text{(A.15)} \\ &= \frac{\partial x_1}{\partial \xi_1}\frac{\partial x_2}{\partial \xi_2}\frac{\partial x_3}{\partial \xi_3} - \frac{\partial x_1}{\partial \xi_1}\frac{\partial x_2}{\partial \xi_3}\frac{\partial x_3}{\partial \xi_2} - \frac{\partial x_1}{\partial \xi_2}\frac{\partial x_2}{\partial \xi_1}\frac{\partial x_3}{\partial \xi_3} \\ &\quad + \frac{\partial x_1}{\partial \xi_2}\frac{\partial x_2}{\partial \xi_3}\frac{\partial x_3}{\partial \xi_1} + \frac{\partial x_1}{\partial \xi_3}\frac{\partial x_2}{\partial \xi_1}\frac{\partial x_3}{\partial \xi_2} - \frac{\partial x_1}{\partial \xi_3}\frac{\partial x_2}{\partial \xi_2}\frac{\partial x_3}{\partial \xi_1}\,. \end{aligned}$$

For a smooth 2-D mapping, the metric Jacobian reduces to

$$J = \frac{\partial x_1}{\partial \xi_1}\,\frac{\partial x_2}{\partial \xi_2} - \frac{\partial x_1}{\partial \xi_2}\,\frac{\partial x_2}{\partial \xi_1}\,.$$

The matrix term, N^T, in (12), can be derived easily through the generalized curvilinear coordinate transformation. Further, the use of the invariants of the transformation helps arrive at a strong conservation-law form. Assuming the coordinates are time invariant, we obtain

$$\mathrm{N}^T = J\vec{\nabla}_x\vec{\xi} \tag{A.16}$$

Accordingly, $\mathrm{N} = J(\vec{\nabla}_x\vec{\xi})^T$. Readers are referred to [34] for details. Specifically, each term in N^T can be expressed as

$$\mathrm{N}^T_{p,q} = \det\left((\vec{\nabla}_\xi\vec{x})^T(p|\boldsymbol{e}^q)\right), \tag{A.17}$$

where $A(p|\mathbf{v})$ denotes a modification of matrix A by replacing row p with vector $\mathbf{v}$, $\boldsymbol{e}^q$ is a unit vector in the q^{th} coordinate direction.

Let's illustrate the relations in 2D. Writing (A.14) out in detail

$$\begin{bmatrix} \frac{\partial x_1}{\partial \xi_1} & \frac{\partial x_1}{\partial \xi_2} \\ \frac{\partial x_2}{\partial \xi_1} & \frac{\partial x_2}{\partial \xi_2} \end{bmatrix} \begin{bmatrix} \frac{\partial \xi_1}{\partial x_1} & \frac{\partial \xi_1}{\partial x_2} \\ \frac{\partial \xi_2}{\partial x_1} & \frac{\partial \xi_2}{\partial x_2} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \tag{A.18}$$

we verify using Cramer's rule that

$$\frac{\partial \xi_1}{\partial x_1} = J^{-1}\det\begin{bmatrix} 1 & \frac{\partial x_1}{\partial \xi_2} \\ 0 & \frac{\partial x_2}{\partial \xi_2} \end{bmatrix} = J^{-1}\det\left((\vec{\nabla}_\xi\vec{x})^T(1|\boldsymbol{e}^1)\right) = J^{-1}\mathrm{N}^T_{1,1}, \tag{A.19}$$

$$\frac{\partial \xi_2}{\partial x_1} = J^{-1}\det\begin{bmatrix} \frac{\partial x_1}{\partial \xi_1} & 1 \\ \frac{\partial x_2}{\partial \xi_1} & 0 \end{bmatrix} = J^{-1}\det\left((\vec{\nabla}_\xi\vec{x})^T(2|\boldsymbol{e}^1)\right) = J^{-1}\mathrm{N}^T_{2,1}, \tag{A.20}$$

$$\frac{\partial \xi_1}{\partial x_2} = J^{-1}\det\begin{bmatrix} 0 & \frac{\partial x_1}{\partial \xi_2} \\ 1 & \frac{\partial x_2}{\partial \xi_2} \end{bmatrix} = J^{-1}\det\left((\vec{\nabla}_\xi\vec{x})^T(1|\boldsymbol{e}^2)\right) = J^{-1}\mathrm{N}^T_{1,2}, \tag{A.21}$$

$$\frac{\partial \xi_2}{\partial x_2} = J^{-1} \det \begin{bmatrix} \frac{\partial x_1}{\partial \xi_1} & 0 \\ \frac{\partial x_2}{\partial \xi_1} & 1 \end{bmatrix} = J^{-1} \det \left((\vec{\nabla}_\xi \vec{x})^T (2|\boldsymbol{e}^2) \right) = J^{-1} \mathrm{N}^T_{2,2}. \qquad \text{(A.22)}$$

Therefore,

$$\mathrm{N}^T = \begin{bmatrix} \mathrm{N}^T_{1,1} & \mathrm{N}^T_{1,2} \\ \mathrm{N}^T_{2,1} & \mathrm{N}^T_{2,2} \end{bmatrix} = J \begin{bmatrix} \frac{\partial \xi_1}{\partial x_1} & \frac{\partial \xi_1}{\partial x_2} \\ \frac{\partial \xi_2}{\partial x_1} & \frac{\partial \xi_2}{\partial x_2} \end{bmatrix} = J(\vec{\nabla}_x \vec{\xi}), \qquad \text{(A.23)}$$

and hence

$$\mathrm{N} = J(\vec{\nabla}_x \vec{\xi})^T. \qquad \text{(A.24)}$$

References

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

[2] P. McCorquodale, P. Colella, A high-order finite-volume method for hyperbolic conservation laws on locally-refined grid, Communications in Applied Mathematics and Computational Science 6 (1) (2011) 1–25.

[3] J. Bell, P. Colella, J. Trangenstein, M. Welcome, Adaptive mesh refinement on moving quadrilateral grids, in: Proceedings of the AIAA 9th Computational Fluid Dynamics Conference, Buffalo, N.Y., 1989, pp. 471–479.

[4] A. Harten, B. Engquist, S. Osher, S. R. Chakravarthy, Uniformly high order accurate essentially non-oscillatory schemes, III, J. Comput. Phys. 131 (1997) 3–47.

[5] X. D. Liu, S. Osher, T. Chan, Weighted essentially non-oscillatory schemes, J. Comput. Phys. 115 (1994) 200–212.

[6] P. Colella, P. R. Woodward, The piecewise parabolic method (PPM) for gas-dynamical simulations, J. Comput. Phys. 54 (1984) 174–201.

[7] B. van Leer, Upwind and high-resolution methods for compressible flow: from donor cell to residual-distribution schemes, Communications in Computational Physics 1 (2) (2006) 192–206.

[8] R. Liska, B. Wendroff, Comparison of several difference schemes on 1D and 2D test problems for the Euler equations, SIAM J. Sci. Comput. 25 (3) (2003) 995–1017.

[9] Z. J. Wang, High-order methods for the Euler and Navier-Stokes equations on unstructured grids, Progress in Aerospace Sciences 43 (2007) 1–41.

[10] M. J. Berger, P. Colella, Local adaptive mesh refinement for shock hydrodynamics, J. Comput. Phys. 82 (1) (1989) 64–84.

[11] C. P. T. Groth, D. L. D. Zeeuw, K. G. Powell, T. I. Gombosi, Q. F. Stout, A parallel solution-adaptive scheme for ideal magnetohydrodynamics, AIAA 99-3273, 14th AIAA Computational Fluid Dynamics Conference (1999).

[12] M. S. Day, J. B. Bell, Numerical simulation of laminar reacting flows with complex chemistry, Combust. Theory Modelling 4 (4) (2000) 535–556.

[13] A. Baeza, P. Mulet, Adaptive mesh refinement techniques for high-order shock capturing schemes for multi-dimensional hydrodynamic simulations, Int. J. Numer. Meth. Fluids 52 (2006) 455–471.

[14] M. Dumbser, O. Zanotti, A. Hidalgo, D. S. Balsara, ADER-WENO finite volume schemes with space-time adaptive mesh refinement, J. Comput. Phys. 248 (2013) 257–286.

[15] M. Dumbser, A. Hidalgo, O. Zanotti, High order space-time adaptive ADER-WENO finite volume schemes for non-conservative hyperbolic systems, Comput. Methods Appl. Mech. Engrg. 268 (2014) 359–387.

[16] D. J. Mavriplis, Accurate multigrid solution of the Euler equations on unstructured and adaptive meshes., AIAA Journal 28 (2) (1990) 213–221.

[17] W. D. Henshaw, A fourth-order accurate method for the incompessible Navier-Stokes equations on overlapping grids, J. Comput. Phys. 113 (1994) 13–25.

[18] D. Calhoun, R. J. LeVeque, An accuracy study of mesh refinement on mapped grids, in: Adaptive Mesh Refinement - Theory and Applications Lecture Notes in Computational Science and Engineering, Vol. 41, 2005, pp. 91–101.

[19] X. Gao, C. P. T. Groth, A parallel solution-adaptive method for three-dimensional turbulent non-premixed combusting flows, J. Comput. Phys. 229 (9) (2010) 3250–3275.

[20] J.-F. Remacle, J. E. Flaherty, M. S. Shephard, An adaptive discontinuous Galerkin technique with an orthogonal basis applied to compressible flow problems, SIAM Review 45 (1) (2003) 53–72.

[21] R. E. Harris, Z. J. Wang, High-order adaptive quadrature-free spectral volume method on unstructured grids, Computers & Fluids 38 (2009) 2006–2025.

[22] M. Nemec, M. J. Aftosmis, M. Wintzer, Adjoint-based adaptive mesh refinement for complex geometries, AIAA 2008-0725, 46th AIAA Aerospace Sciences Meeting (2008).

[23] J. L. Steger, Implicit finite-difference simulation of flow about arbitrary two-dimensional geometries, AIAA Journal 16 (7) (1978) 679–686.

[24] C. Helzel, M. J. Berger, R. J. Leveque, A high-resolution rotated grid method for conservation laws with embedded geometries, SIAM J. Sci. Comput. 26 (3) (2005) 785–809.

[25] P. Colella, D. T. Graves, B. J. Keen, D. Modiano, A Cartesian grid embedded boundary method for hyperbolic conservation laws, J. Comput. Phys. 211 (2006) 347–366.

[26] M. Barad, P. Colella, A fourth-order accurate local refinement method for Poisson's equation, J. Comput. Phys. 209 (2005) 1–18.

[27] P. Colella, M. Dorr, J. Hittinger, D. Martin, P. McCorquodale, High-order finite-volume adaptive methods on locally rectangular grids, J. Phys.: Conf. Ser. 180 (2009) 012010.

[28] P. Colella, M. D. Sekora, A limiter for PPM that preserves accuracy at smooth extrema, J. Comput. Phys. 227 (2008) 7069–7076.

[29] 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, B. Van Straalen, Chombo Software Package for AMR Applications - Design Document, Tech. Rep. LBNL-6616E, Lawrence Berkeley National Laboratory (2014).

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

[31] G. A. Sod, A survey of several finite difference methods for systems of nonlinear hyberpolic conservation laws, J. Comput. Phys. 27 (1) (1978) 1–31.

[32] P. R. Woodward, P. Colella, The numerical simulation of two-dimensional fluid flow with strong shocks, J. Comput. Phys. 54 (1984) 115–173.

[33] R. L. Panton, Incompressible Flow, John Wiley & Sons, 1996.

[34] T. H. Pulliam, D. W. Zingg, Fundamental Algorithms in Computational Fluid Dynamics, Scientific Computation, Springer, 2014.