CosmosDG: An hp-adaptive Discontinuous Galerkin Code for Hyper-resolved Relativistic MHD

Peter Anninos1, Colton Bryant2, P. Chris Fragile3, A. Miguel Holgado4, Cheuk Lau1, and Daniel Nemergut5
1 Lawrence Livermore National Laboratory, P.O. Box 808, Livermore, CA 94550, USA
2 Department of Engineering Sciences & Applied Mathematics, Northwestern University, 2145 Sheridan Road, Evanston, Illinois, 60208, USA
3 Department of Physics & Astronomy, College of Charleston, 66 George Street, Charleston, SC 29424, USA
4 Department of Astronomy & National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign, Urbana, Illinois, 61801, USA
5 Operations & Engineering Division, Space Telescope Science Institute, 3700 San Martin Drive, Baltimore, MD 21218, USA

Received 2017 April 15; revised 2017 July 13; accepted 2017 July 14; published 2017 August 9

Abstract

We have extended COSMOS++, a multidimensional unstructured adaptive mesh code for solving the covariant Newtonian and general relativistic radiation magnetohydrodynamic (MHD) equations, to accommodate both discrete finite volume and arbitrarily high-order finite element structures. The new finite element implementation, called COSMOSDG, is based on a discontinuous Galerkin (DG) formulation, using both entropy-based artificial viscosity and slope limiting procedures for the regularization of shocks. High-order multistage forward Euler and strong-stability preserving Runge–Kutta time integration options complement high-order spatial discretization. We have also added flexibility in the code infrastructure allowing for both adaptive mesh and adaptive basis order refinement to be performed separately or simultaneously in a local (cell-by-cell) manner. We discuss in this report the DG formulation and present tests demonstrating the robustness, accuracy, and convergence of our numerical methods applied to special and general relativistic MHD, although we note that an equivalent capability currently also exists in COSMOSDG for Newtonian systems.

Key words: hydrodynamics – magnetohydrodynamics (MHD) – methods: numerical – relativistic processes

1. Introduction

Discontinuous Galerkin (DG) finite element (FE) methods have raised great interest over the past few decades, particularly in the engineering communities and applied mathematics literature (e.g., Johnson et al. 1984; Cockburn et al. 1989; Kershaw et al. 1995; Cockburn & Shu 1998; Hartmann & Houston 2002; Kuzmin & Turek 2004; Guermond et al. 2011). These methods were originally introduced more than 40 years ago for neutron transport (Reed & Hill 1973), but they have since expanded in scope and have become popular for solving more general systems of conservation laws across a variety of physical disciplines, including computational fluid dynamics, acoustics, and electromagnetics. However, with the exception of a few groups, they have yet to be widely adopted in computational astrophysics, particularly relativistic astrophysics, where finite difference and finite volume (FV) methods dominate. One interesting attempt to use it was presented in Meier (1999). In that work, the Einstein field equations were discretized in all four spacetime dimensions, thus treating time entirely equivalent to space. We are aware of only three other applications of DG to relativistic magnetohydrodynamics (MHD): Radice & Rezzolla (2011), Zanotti et al. (2015), and Kidder et al. (2017). The first two have significant restrictions, with the first being limited to one-dimensional spherical symmetry, while the second is limited to special relativity. The third paper considers multidimensional relativistic MHD, although without adaptivity. The methods described in the current paper apply to both special and general relativistic MHD, to multidimensional spacetimes with no symmetry restrictions, and to dynamically adaptive mesh and polynomial representations.

Finite element methods possess a number of desirable properties, including their suitability for conservation equations, their cost competitiveness with finite volume methods, their compatibility with Riemann solvers, their potential for achieving spectral-like convergence rates, and their applicability to unstructured meshes and local (cell-by-cell) refinement. They are related to FV methods in the sense that basic cell-centered FV schemes correspond to the DG(1) method, i.e., to the discontinuous Galerkin method using piecewise linear polynomials ($p = 1$). Consequently, the DG($p$) method, with $p > 1$, can be regarded as a natural extension of the FV method to higher orders. For continuous FE methods, however, high order comes at a cost as it requires the storage and inversion of a very large, global matrix. Such global matrix inversions do not scale well to large numbers of processors and, therefore, limit the sizes and types of problems to which continuous FE can be applied. This was the shortcoming of the Meier (1999) implementation. In contrast, the discontinuous nature of DG methods allows high-order polynomial approximations to be made within a single element rather than across wide stencils, as in the case of high-order FV methods, or the entire grid, as in continuous FE methods. Thus, all matrix inversions are done locally, rather than globally. Furthermore, as elements only need to communicate with adjacent elements with a common face (von Neumann neighbors), regardless of the order of accuracy of the scheme, inter-element communications are minimal, making the method highly parallelizable. Furthermore, the parallelization can be efficiently accomplished through simple domain decomposition.

Another advantage of DG methods is that they are relatively straightforward to implement on unstructured meshes. Unstructured meshes, themselves, have the advantages that they can more accurately discretize complex geometries, easily adapt to surface boundaries, and enhance solution accuracy and efficiency through the use of local (i.e., cell-by-cell) adaptive mesh refinement (AMR; also commonly referred to as $h$-refinement). The discontinuous nature of DG methods
relaxes the strong continuity restrictions of continuous FE methods, leaving elements free to be refined or coarsened, without affecting solutions or data structures in other elements. DG methods also allow the easy implementation of \( p \)-refinement, or adaptive order refinement (AOR), where the polynomial degree of the basis is varied. Thus, the order of accuracy can be different from element to element. This type of refinement is potentially very powerful, as exponential convergence rates are possible when solutions are smooth (Babuška et al. 1986; Schwab 1999). Combined with \( h \)-refinement, these high rates of convergence are even possible when singularities are present (Schwab 1999).

For all of these reasons, DG methods are particularly appealing and a natural progression for COSMOS++ (Anninos & Fragile 2003; Anninos et al. 2003, 2005), an unstructured \( hr \)-adaptive mesh code we have developed for both Newtonian and general relativistic astrophysical applications. So, we have taken this opportunity to upgrade COSMOS++ for both FV and DG frameworks, and to implement \( p \)-refinement. COSMOS++ supports numerous Physics packages, including hydrodynamics, ideal magnetic fields, primordial chemistry, nuclear reaction networks, Newtonian self-gravity, dynamical general relativistic spacetimes, radiation transport, molecular viscosity, thermal conduction, etc., but we focus exclusively on MHD in this paper since only those packages have been generalized so far to work within COSMOSDG. The equations, methods, and tests of our code are described in the remaining sections, emphasizing the DG aspects. The reader is referred to our previous papers, most notably Anninos et al. (2005), for additional details not included in this paper, such as mesh hierarchy constructions, parallelism, and class inheritance designs. The code itself is available upon request. Unless otherwise noted, standard index notation is used for labeling spacetime coordinates: repeated indices represent summations, the raising and lowering of indices is done with the four-metric tensor, and Latin (Greek) indices run over spatial (four-space) dimensions.

2. Basic Equations

2.1. General Relativistic MHD

We begin by writing the contravariant stress energy density tensor for a viscous fluid with ideal MHD as a linear combination of the hydrodynamic, magnetic, and viscosity contributions:

\[
T^{\alpha\beta} = (\rho h + 2P_B/c^2 - Q_B/c^2)u^\alpha u^\beta + (P + P_B - Q_B)g^{\alpha\beta} - Q^{\alpha\beta}_S - b^\alpha b^\beta.
\]

(1)

Here, \( \rho \) is the fluid mass density, \( h = 1 + c^2/\epsilon + P/(\rho c^2) \) is the specific enthalpy, \( \epsilon \) is the speed of light, \( u^\alpha = u^\alpha V^\alpha \) is the contravariant velocity, \( V^\alpha \) is the transport velocity, \( P \) is the fluid pressure (for an ideal gas, \( P = (\Gamma - 1)\epsilon \), \( \epsilon = \rho c \) is the fluid internal energy density, \( \Gamma \) is the adiabatic index), \( b^\alpha \) is the magnetic field, \( P_B = g_{\alpha\beta}b^\alpha b^\beta/2 \) is the magnetic pressure, \( Q_B \) is the bulk viscosity, \( Q^{\alpha\beta}_S \) is the symmetric shear viscosity tensor representing artificial or molecular viscosity, and \( g_{\alpha\beta} \) is the curvature metric. Although COSMOS++ supports molecular viscosity, it is not currently incorporated into the DG framework, so we do not consider it further in this paper.

The four fluid equations (energy and the three components of momentum) are derived from the conservation of stress energy:

\[
\nabla_\mu T^{\mu\nu} = \partial_\nu T^{\mu\nu} + \Gamma^{\nu}_{\mu\lambda} T^{\mu\lambda} - \Gamma^{\lambda}_{\mu\nu} T^{\mu\lambda} = S_\nu,
\]

where \( \Gamma^{\nu}_{\mu\lambda} \) are the Christoffel symbols and \( S_\nu \) represent arbitrary source terms. In addition to energy and momentum, we also require equations for the conservation of mass, \( \nabla_\mu (\rho u^\mu) = \partial_\nu (\sqrt{-g} u^\nu \rho) + \partial_\nu (\sqrt{-g} u^\nu V^\nu) = 0 \), and magnetic induction, \( \nabla_\mu (u^\mu b^\nu - b^\mu u^\nu) = 0 \).

Expanding out the space and time coordinates, the four-divergence (\( \nabla_\mu T^{\mu\nu} = S_\nu \)) of the mixed index stress tensor is written

\[
\partial_\nu (\sqrt{-g} T^{\nu\mu} + \sqrt{-g} T^{\nu\mu}_{\nu} = \sqrt{-g} T^{\nu}_{\nu} = \sqrt{-g} S_\nu.
\]

(2)

Further defining energy and momentum as \( \mathcal{E} = -\sqrt{-g} T^{0}_{\nu} \) and \( \mathcal{S}_\lambda = -\sqrt{-g} T^{\lambda}_{\nu} \), the equations take on a traditional transport formulation,

\[
\partial_\nu \mathcal{E} + (\partial_\lambda \sqrt{-g}) T^{\lambda\nu}_{\nu} = \partial_\nu (\sqrt{-g} (P + P_B) V^\nu) = -\sqrt{-g} T^{\nu}_{\nu} = \Gamma^{\nu}_{\nu},
\]

(3)

\[
\partial_\nu \mathcal{S}_\lambda + (\partial_\lambda \sqrt{-g}) T^{\lambda\nu}_{\nu} = \partial_\nu (\sqrt{-g} (P + P_B) g^{\nu}_{\lambda} V^\nu) = \sqrt{-g} T^{\nu}_{\nu} = \Gamma^{\nu}_{\nu},
\]

(4)

for energy, and

\[
\partial_\nu \mathcal{S}_\lambda + (\partial_\lambda \sqrt{-g}) T^{\lambda\nu}_{\nu} = \partial_\nu (\sqrt{-g} (P + P_B) g^{\nu}_{\lambda} V^\nu) = \sqrt{-g} T^{\nu}_{\nu} = \Gamma^{\nu}_{\nu},
\]

(5)

for momentum. Completing the system of equations, the energy and momentum conservation are supplemented with mass conservation,

\[
\partial_\nu D + (\partial_\nu (DV^\nu) - B^\nu V^\nu) = 0,
\]

(6)

where \( D = \sqrt{-g} u^\nu \rho = W_\rho \) is the boost density, and the magnetic induction

\[
\partial_\nu B^\nu + (\partial_\nu (B \nu V^\nu) - B^\nu V^\nu) = \eta g^{\nu\mu} \partial_\mu (\partial_\nu B^\nu),
\]

(7)

where \( B^\nu = W (b^\nu - b^\nu V^\nu) \) (with \( b^\nu = -B^\nu V_\nu/(WV^\nu V_\nu) = B^\nu V_\nu u^\nu/\sqrt{-g} \) is the evolved spatial \( B^\nu = 0 \), divergence-free \( \partial_\nu B^\nu = 0 \)) representation of the field, distinct from the rest-frame field \( b^\nu \). The additional source term on the right-hand side of Equation (8) is a form of divergence cleanser used to drive Equation (8) to satisfy \( \partial_\nu B^\nu = 0 \) on a scale defined by the choice of parameter \( \eta \), typically set proportional to the largest characteristic speed in the flow.

Mesh motion is easily accommodated by a straightforward replacement of generic advective terms,

\[
\partial_\nu (\sqrt{-g} T^{0}_{\nu}) + \sqrt{-g} T^{0}_{\nu} V^\nu = 0,
\]

(9)

with

\[
\partial_\nu (\sqrt{-g} T^{0}_{\nu}) + \sqrt{-g} (T^{0}_{\nu} (V^\nu - V^\nu_{\nu}) + \sqrt{-g} T^{0}_{\nu} \partial_\nu V^\nu_{\nu},
\]

(10)

where \( V^\nu_{\nu} \) is the grid velocity and \( T^{0}_{\nu} \) is used here to represent any evolved field, including \( E, S_\nu, D, \) and \( B^\nu \).

2.2. Newtonian MHD

For comparison and future reference, we add in this section the equivalent covariant form of the corresponding Newtonian MHD equations. Drawing an analogy with the relativistic equations presented above, we write the effective Newtonian
stress energy tensor as
\[ T^{\alpha\beta} = \rho \nu^{\alpha\nu^\beta} + (P + \Pi - Q_b) g^{\alpha\beta} - Q_b^{\alpha\beta} - b^{\alpha} b^{\beta}. \]  
(11)

In moving curvilinear coordinates, the Newtonian conservation equations take on a form similar to their relativistic counterparts:
\[ \partial_t (\sqrt{g} \rho) + \sqrt{g} \rho \partial_i V_i + \partial_t (\sqrt{g} \rho (v_i - V_i)) = 0, \]
\[ \partial_t (\sqrt{g} E) + \sqrt{g} E \partial_i V_i + \partial_t (\sqrt{g} E (v_i - V_i)) + \sqrt{g} F^{0i} = - \rho \nu^{0i} \partial_i \phi, \]
\[ \partial_t (\sqrt{g} b_j) + \sqrt{g} b_j \partial_i V_i + \partial_t (\sqrt{g} b_j (v_i - V_i)) + \sqrt{g} F^{ij} = \eta g^{ij} \partial_i (\sqrt{g} b_j), \]
(12) 
\[ \partial_t (\sqrt{g} b_j) + \sqrt{g} b_j \partial_i V_i + \partial_t (\sqrt{g} b_j (v_i - V_i)) + \sqrt{g} b_j = \frac{\eta}{\sqrt{g}} \partial_i (\sqrt{g} b_i). \]
(13)

with flux terms
\[ F^{0i} = (P + \Pi) v^i - b^i b_j v^j, \]
\[ F^{ij} = (P + \Pi) g^{ij} - b^i b^j. \]
(16)

Here, \( E \) is the total energy density including internal, magnetic, and kinetic energy contributions: \( E = e + b^i b_j / 2 + \nu^i v_j / 2 \). This definition does not include gravitational energy, which is treated as an add-on (nonconservative) source represented by the potential \( \phi \) in the right-hand sides of the energy and momentum equations. In this form, Newtonian and relativistic fluxes and source terms are easily interchangeable in the numerical solver frameworks.

2.3. Primitive Fields

At the beginning (or end) of each time cycle, a series of coupled nonlinear equations are solved to extract primitive fields (mass density, internal energy, velocity) from evolved conserved fields (boost density, total energy, momentum), after which the equation of state is applied to compute thermodynamic quantities like pressure, sound speed, and temperature. For Newtonian systems, this procedure is straightforward, but relativity complicates the interdependency of primitives, and their extraction from conserved fields requires special iterative treatment. We have implemented several procedures for doing this, solving one-, two-, or five-dimensional inversion schemes (Noble et al. 2006; Fragile et al. 2012), or a nine-dimensional fully implicit method (including coupling terms) when radiation fields are present (Fragile et al. 2014).

One of the more robust procedures reduces the number of equations from the number of evolved fields (five in the simplest case of hydrodynamics) to two, taking advantage of projected conserved constraints to facilitate the reduction. This 2D method solves two constraints, energy and momentum, derived from a projection of the stress energy tensor to the normal observer frame with four-velocity \( n_\nu = [-\alpha, 0, 0, 0] \) and lapse function \( \alpha \),
\[ \frac{\tau_\mu}{\alpha} = - \frac{n_\nu T^\nu_\mu}{\alpha} = T^0_\mu = (\rho + 2P_b) u^0 u_\mu + (P + \Pi) u^0 b_\mu. \]
(18)

Defining \( \vec{B}^\mu = \alpha B^\mu / \sqrt{-g} \), the energy \( E \) and momentum \( \vec{m}^2 \) constraints take the form
\[ E = \tau_\mu n^\mu \tau^0_\mu = - \frac{\vec{B}^2 (1 + v^2)}{2} + \frac{(\alpha T^0_\mu \vec{B}^\mu)^2}{2 w^2} - w + P, \]
\[ \vec{m}^2 = \tau^\nu \tau_\nu = \tau^\nu \tau_\nu + \alpha^2 (\tau^0_\nu)^2 = v^2 (w + \vec{B}^2)^2 \]
\[ - \frac{(2w + \vec{B}^2)}{w^2} (\alpha T^0_\nu \vec{B}^\nu)^2, \]
(19) 
(20)

where \( \tau^\nu = (g^\nu_\mu + n^\nu n_\mu) \tau_\mu \), \( v^2 = 1 - (1 / \gamma^2) \), \( \gamma = \alpha u^0 \) is the Lorentz boost, \( w = (\alpha u^0)^2 (\rho + 2P_b) \) is the scaled enthalpy, and the pressure and its gradients \( (\partial P / \partial w, \partial P / \partial v^2) \) are calculated from the ideal gas law
\[ P = \frac{\Gamma - 1}{\Gamma} (w(1 - v^2) - \alpha u^0 \rho \sqrt{1 - v^2}). \]
(21)

These constraints represent nonlinear equations for the two unknowns, \( w \) and \( v^2 \), and are solved by Newton iteration. All of the other terms in these equations are easily derived from evolved quantities.

An alternative, though generally more costly, option utilizes Newton iteration to solve a full, unprojected matrix system of nonlinear equations constructed from the primitive field dependency of all of the conserved or evolved quantities. Thus, within each iteration, one constructs a \( (5 \times 5) \) for the case of hydrodynamics) Jacobian matrix \( A_{ij} = \partial U^i / \partial P^j \) evaluated at guess primitive solutions. Here, \( U^i \equiv [D, E, S_i] = \sqrt{-g} [u^i \rho, -T^0_\nu, T^i_\nu] \) is a vector list of conserved fields, and \( P^j \equiv [\rho, e, \vec{u}^\nu] \) is a vector list of corresponding primitive fields. We use \( u^k = u^0 - u^0 b^0 / \sqrt{g^{00}} \) with \( u^0 = \gamma / \alpha \) as the primitive velocity in place of \( v^2 \) in this procedure.

3. Numerical Methods
3.1. DG Framework

The DG framework reviewed here is presented in the context of generic conservation laws expressed in the following vector form:
\[ \partial_t u + \nabla \cdot F(u) = s, \]
(22)

where \( u \) is the conserved quantity of interest (density, momentum, energy), \( F \) is the flux, and \( s \) is an arbitrary source term. We switch to vector notation in this section in order not to confuse spacetime indices with basis function labels or indexing of matrix elements.

We begin by multiplying Equation (22) by a set of weight functions \( p(x) \), and integrating the resulting equations over the volume \( V_k \) of each cell \( k \):
\[ \int_{V_k} dV p_k(x) (\partial_t u + \nabla \cdot F(u)) = \int_{V_k} dV p_k(x) s. \]
(23)
Although we have written the DG framework in a modular way, anticipating adding more options for basis sets in the future, we have for this work adopted Lagrange interpolatory polynomials defined as

\[ p_i(x) = \prod_{k=1, k \neq i}^n \frac{x - x_k}{x_i - x_k}. \]  

(24)

The shape functions of this basis are unity at their respective nodes and zero at all other nodes. A multidimensional version is constructed through tensor products of one-dimensional polynomials on a unit reference element covered with \((p + 1)^n\) nodes, where \(p\) is the order and \(n\) is the number of dimensions.

The divergence theorem is then applied to Equation (23), which results in the so-called weak form of Equation (22),

\[
\int_{V_l} dV p_i \delta k u - \int_{\partial V_l} dA \mathbf{p}_i \cdot \mathbf{H}(u^+, u^-) = \int_{\partial V_l} dV \mathbf{p}_i s,
\]

(25)

where \(\delta k\) is the surface of cell \(k\), \(\mathbf{n}\) is the outward-pointing vector normal to the surface, and \(\mathbf{H}(u^+, u^-)\) is an appropriately calculated flux at the cell boundaries. \(\mathbf{H}(u^+, u^-)\) takes into account discontinuities across cell faces and depends on both interior and adjoining neighbor state solutions.

A simple method for determining surface fluxes is standard upwinding, which uses the value of \(u\) inside the cell for the exiting flux \((\mathbf{H}(u^-) = \mathbf{F}(u(x^- - \varepsilon \mathbf{n}))\) for \(\mathbf{n} \cdot \mathbf{v} \geq 0\) and the value outside for the incoming flux \((\mathbf{H}(u^+) = \mathbf{F}(u(x^+ + \varepsilon \mathbf{n}))\) for \(\mathbf{n} \cdot \mathbf{v} < 0\), where \(x^+\) is a location on the cell surface and \(\varepsilon\) is some arbitrarily small positive value. Alternative, less diffusive options for computing surface fluxes can be easily substituted for simple upwinding. Our implementation currently supports both Lax–Friedrichs (LF) and Harten–Lax–van Leer (HLL) approximate Riemann solvers (Harten et al. 1983):

\[ H_{LF}(u^+, u^-) = \frac{1}{2}(\mathbf{F}(u^+) + \mathbf{F}(u^-) - \alpha_+(u^+ - u^-)), \]

(26)

\[ H_{HLL}(u^+, u^-) = \frac{1}{\alpha_+ + \alpha_-}(\alpha_+ \mathbf{F}(u^-) + \alpha_- \mathbf{F}(u^+)) - (\alpha_+ \alpha_-)(u^+ - u^-)), \]

(27)

where \(\alpha_{\pm}\) are the minimum and maximum characteristic wave speeds. The relation of DG methods to Riemann solvers thus comes from the discontinuous representation of the solution at element interfaces, which requires a relaxation of the cross-element continuity condition. Instead of enforcing a single, continuous solution at element interfaces, like the continuous FE method, the DG method supports “left” and “right” states on either side of the interface similar to finite volume or finite difference methods. It then treats the element boundary by solving a local Riemann problem to calculate the appropriate flux, ensuring the method remains conservative while capturing the shock characteristics.

Time and space dependencies of each evolved quantity and source term are split into separable form and expanded using a set of spatial basis functions, which for Galerkin methods are equal to the weight functions \(p_j(x)\) introduced earlier,

\[ u = \sum_{j=1}^J u_j(t) p_j(x). \]

(28)

Substituting Equation (28) into Equation (25) and performing the integrals with quadratures produces the following linear system for the expansion (or support) coefficients \(u \equiv u_j(t)\):

\[ M \frac{\partial u}{\partial t} + Su + Ru = Ms. \]

(29)

\(M\) is the mass matrix associated with each cell,

\[ M_{ij} = \sum_{q=1}^{Q_v} p_{iq} p_{jq} w_q, \]

(30)

where \(Q_v\) is the number of quadrature points for the volumetric integral of cell \(k\), \(p_{iq}\) is the \(i\)th weight function evaluated at the \(q\)th quadrature point, and \(w_q\) is the weight of the \(q\)th quadrature point. \(S\) is the stiffness matrix defined as

\[ S_{ij} = \sum_{q=1}^{Q_v} p_{iq} v_q \cdot \nabla p_{jq} w_q, \]

(31)

where \(v_q\) is the velocity at the \(q\)th quadrature point within the cell volume. \(R\) is the surface matrix defined for each element as

\[ R_{ij} = \sum_{q=1}^{Q_s} p_{iq} v_q p_{jq} w_q, \]

(32)

where \(Q_s\) is the number of quadrature points for the integral over the surface (faces) of each cell. In general, the weight functions are defined over a reference cell with local coordinates \(\xi\), and mapped onto each physical cell with global coordinates \(\eta\) using a Jacobian matrix \(J\) defined for each element as

\[ J_{ij} = \sum_{m=1}^M (\partial \xi p_m) \eta_{jm}, \]

(33)

where \(M\) is the number of weight functions, \(p_m\) is the \(m\)th weight function defined over the reference element, and \(\eta_{jm}\) is the location of the \(m\)th support point with respect to the \(j\)th global coordinate. To apply the Jacobian mapping, we multiply Equations (30)–(32) by the determinant of \(J\), which is built separately for each zone and cell face elements.

The DG formulation is completed with a procedure for discretizing the remaining time derivative term in Equation (29). COSMOS+++ and COSMOSDG support numerous high-order time integration options, some of which are discussed in Section 3.4, but we write out an explicit expression for illustration here using a simple, single-step forward Euler solution,

\[ u^{n+1} = u^n + \Delta t^n (S^n u^n - R^n u^n + M s^n), \]

(34)

where \(n\) denotes the time level and \(\Delta t^n\) is the time-step size. Equation (34) can alternatively be written as

\[ u^{n+1} = u^n + \Delta t^n ((M)^{-1}B^n + s^n), \]

(35)

where we have absorbed the evolved and velocity fields into an inclusive flux term and merged the two source matrices into a
single source combining volume and surface contributions:

\[ B_{ij} = \sum_{q=1}^{Q_{ij}} p_{ijq} F_q(u) \cdot \nabla p_{ijw} - \sum_{q=1}^{Q_{ij}} p_{ijq} H_q(u^+, u^-) p_{ijwq}. \]  

(36)

Whereas the form (34) is useful for transport models when the velocity and conserved fields are easily disentangled, Equation (35) is applicable to more general flux constructs.

We note that the inverse of the mass matrix appearing in Equations (34) and (35) depend only on the basis functions and Jacobian transformations mapping reference elements to actual physical cell geometries. It can thus be computed once at the start of the simulation and stored to save computational time. Of course, it would have to be recomputed and updated each time the mesh changes by AMR, AOR, or grid motion. But since the matrix elements are entirely local, this can be done on a cell-by-cell basis, as needed. It does not have to be recomputed globally every cycle across the entire grid.

### 3.2. Artificial Viscosity

We have implemented several variations of an artificial viscosity method for regularizing shock discontinuities. All versions are conservative in nature and based on previous works rooted in entropy-based shock detection models (Hartmann & Houston 2002; Guermond et al. 2011) but modified here to work with relativistic MHD. The advantage of artificial viscosity, compared to slope limiting discussed in the next section, is that it easily generalizes to unstructured grids, to multiple dimensions, to high-order finite elements, and to adaptive mesh and/or order refinement. Viscosity is evaluated on each node within a cell, so it is in effect applied subzonaly and respects high-order compositions of cell elements. Of course, its dissipative nature has to be taken into account when choosing parameters such as the shock detection threshold and strength of dissipation.

Artificial viscosity is introduced as a flux-conservative, covariant, Laplacian source term added to the right-hand side of each evolution equation of the form

\[ \partial_t(\epsilon_u \nu(R(U), \delta J(F^\pm)) \sqrt{-g} g^{ij} \partial_j u), \]  

(37)

where \( u \) is the evolved field, \( \epsilon_u \) is a constant that can differ for each evolved field based on, e.g., (magnetic) Prandtl number scaling, and \( \nu(R(U), \delta J(F^\pm)) \) is the viscosity coefficient that depends on shock jump detection algorithms across cell interfaces \( \delta J(F^\pm) \) and entropy residuals within zone elements \( R(U) \). The essential viscosity formulation singles out regions of high entropy production by employing three different detection algorithms to define reasonable quantitative measures of viscous heating and combines these measures into a viscosity coefficient that triggers locally over shocks. The three detection functions are based on (1) calculating zonal residuals from the transport of an effective entropy function \( R(U) = \partial_t U + \partial_i F^i(U) \); (2) computing flux discontinuities across cell interfaces \( \delta J(F^\pm) \); and (3) providing an upper bound determined by the maximum local wave speed in each element, \( V_{\text{max}} \). The viscosity is then chosen by

\[
\nu(R(U), \delta J(F^\pm)) = \min \left( C_l V_{\text{max}}, \max \left( C_q \frac{\delta J}{||J_0||}, C_q \ell^2 \frac{R}{||R_N||} \right) \right). 
\]

(38)

The parameter \( C_l \) and \( ||J_0|| \) and \( ||R_N|| \) are locally constructed normalization factors, \( \ell = \Delta x / p \) is the cell width reduced by the basis order (or equivalently the distance between nodal subgrid elements), and \( C_q \) and \( C_\ell \) are the linear and quadratic viscosity coefficients typically in the ranges \( C_q \in [0.1, 0.5] \) and \( C_\ell \in [0.2, 1.0] \). Numerous options for \( \ell \) provide reasonable zonal residuals, including entropy \( (\rho \mathbf{p})^{1/2} \), relativistic enthalpy, stress energy tensor \( T_0 \), and enthalpy scaled mass density. Interface jump detection is sensed by comparing flux discontinuities in the fields selected for residual evaluation across cell faces projected into cell face normals \( \delta (IVNI) \). These zonal residual and interface jump calculations are typically normalized by the residual fields (e.g., entropy, enthalpy) averaged over zone quadratures, but can also be normalized by the minimum or maximum quadrature values if the viscosity needs to be strengthened or weakened.

### 3.3. Slope Limiting

Another option we developed for suppressing spurious oscillations near sharp features is slope limiting. Our implementation uses a least-squares slope formulation in each cell and applies (optional) limiting to either primitive, conserved, or characteristic fields with a traditional minmod operator. Specifically, in each zone we set up and solve the following least-squares problem:

\[
\begin{bmatrix}
\delta u_x \\
\delta u_y \\
\delta u_z
\end{bmatrix} = 
\begin{bmatrix}
u_0 \\
u_1 \\
u_2
\end{bmatrix},
\]

(39)

where \( (x_i, y_j, z_k) \) are the coordinates of the ith support node in the zone, \( u_i \) are the values of the field to be limited at each node, and matrix inversion gives the least-squares solution for the slope vector \( \delta u \).

Limiting is applied to these slopes along each dimension by

\[
\delta u_i = \minmod(\delta u_i, \beta(\pi - \pi^-), \beta(\pi^+ - \pi)),
\]

(40)

with

\[
\minmod(a, b, c) = \begin{cases} 
\text{sign}(a) \min(|a|, |b|, |c|) & \text{if } \text{sign}(a) = \text{sign}(b) = \text{sign}(c) \\
0 & \text{otherwise}. 
\end{cases}
\]

(41)

Here, \( \pi \) denotes the integral of support fields over zone quadratures on the unit reference element, effectively a quadrature-weighted average, and \( \pi^\pm \) denotes the weighted average in neighboring zones along the positive and negative directions. The parameter \( \beta \in [0.5, 1] \) sets the amount of limiting to be used.

We also implement a bounded version of this limiter in the spirit of Cockburn & Shu (1989) and Schaal et al. (2015), where the central difference slope is first evaluated against a threshold parameter before applying the minmod operator,

\[
\minmod_b(a, b, c) = \begin{cases} 
a & \text{if } |a| \leq M \\
\minmod(a, b, c) & \text{otherwise}, 
\end{cases}
\]

(42)

allowing the user to set a threshold slope below which the limiter will not activate. The parameter \( M \) depends upon (and is
sensitive to) several factors, such as zone size and the maximum expected curvature near smooth extrema in the solution. Its optimal value is in general determined empirically through trial and error.

We note that the combination of the local nature of DG finite elements, the least-squares approach for calculating slopes, and quadrature folding of high-order solutions to low-order bases allow these limiting procedures to work easily for any basis order and with adaptive order refinement. Adaptive mesh refinement, too, is as easily accommodated with the additional caveat that if a neighboring zone is on a different refinement level, \( \pi \) is averaged over all of the children in that zone.

### 3.4. Time Integration

The preferred high-order (greater than second-order) time discretization method in COSMOS++ has been a low-storage version of the forward Euler method (Shu & Osher 1988). In this method, the solution for a generic differential equation, represented as \( \partial_t U = L(U) \), at any stage \( i \) can be expressed as

\[
U^{(i)} = \eta_{i-1} U^{(0)} + (1 - \eta_{i-1}) \{U^{(i-1)} + \Delta t \eta_n L(U^{(i-1)})\},
\]

where \( U^{(0)} = U^n \) is the solution at \( t = t^n \). Solutions at any stage \( i \) can thus be constructed from the initial solution \( U^{(0)} \) and the results of advancing the previous stage, \( i - 1 \). The coefficients, \( \eta_i \), for the three lowest orders are \( \eta_0 = 0 \) for the first order; \( \eta_1 = 0, \eta_1 = 1/2 \) for the second; and \( \eta_2 = 0, \eta_1 = 3/4, \eta_2 = 1/3 \) for the third. Unfortunately, coefficients have not been found to extend the low-storage Euler method to higher order. Spiteri & Ruuth (2002) speculate that no such coefficients exist. In addition, no four-stage, fourth-order method has been found (Gottlieb & Shu 1998).

To extend COSMOSDG to fourth order, we therefore consider an alternative five-stage, strong-stability-preserving Runge-Kutta (SSPRK) method presented in Spiteri & Ruuth (2002). Generically, an \( s \)-stage, explicit Runge–Kutta method can be written as

\[
U^{(i)} = \sum_{k=0}^{i-1} (\alpha_{ik} U^{(k)}) + \Delta t \beta_{ik} L(U^{(k)}), \quad \text{for} \ i = 1, 2, \ldots, s,
\]

\[
U^{n+1} = U^{(s)}.
\]

We only consider cases where the constants \( \alpha_{ik}, \beta_{ik} \geq 0 \), and \( \alpha_{ik} = 0 \) if \( \beta_{ik} = 0 \). For the weighting coefficients to be consistent, the \( \alpha_{ik} \) must satisfy \( \sum_{k=0}^{i-1} \alpha_{ik} = 1 \). This Runge–Kutta method is strong-stability-preserving provided

\[
\Delta t \leq c \Delta t_{\text{FE}},
\]

where

\[
c = \min_{i,k} \frac{\alpha_{ik}}{\beta_{ik}},
\]

and \( \Delta t_{\text{FE}} \) comes from stability requirements on the forward Euler time step. For DG methods, this is calculated as \( \Delta t_{\text{FE}} = c_{\text{pl}}/(p + 1) \) times the minimum estimated stability time step over all physics packages, where \( p \) is the basis order. The Courant constant \( c_{\text{pl}} \) is typically set to 0.5 or less. The coefficients \( \alpha_{ik} \) and \( \beta_{ik} \) are displayed in Table 1, using standard (row, column) indexing, along with the corresponding time-step coefficients, \( c \), for the five-stage method at convergence orders 2, 3, and 4. It is possible to write coefficients for a first-order, five-stage scheme, but since it offers no efficiency advantage over a standard Euler scheme, it is not implemented.

As we mentioned, the five-stage method is required to achieve fourth-order convergence. However, Spiteri & Ruuth (2002) have shown that the five-stage scheme can have advantages at lower order, too. This is because the effective time step that results from Equation (47) can be considerably larger than \( \Delta t_{\text{FE}} \). So, although the five-stage method is undoubtedly more expensive per full update cycle, it can require far fewer total cycles. As an example, a simple two-stage, second-order scheme will be able to step forward \( 4 \Delta t_{\text{FE}} \) each cycle, whereas a five-stage, second-order scheme will be able to step forward \( 4 \Delta t_{\text{FE}} \). Thus, although the five-stage scheme requires \( 5/2 \) more work per cycle, it goes four times further each cycle, making it \([(4/1)/(5/2) - 1] \times 100\% = 60\% \) more efficient over the full evolution.

It is worth mentioning that in order for any SSPRK scheme to be implemented as a low-storage method, the constants \( \alpha_{ik} \) and \( \beta_{ik} \) must be such that no intermediate-stage solutions are required in the final stage. A low-storage scheme would require \( \alpha_{ik} = 0 \) for \( k < i - 1 \) whenever \( i < s \) and for \( k = 1, \ldots, s - 1 \) for \( i = s \). Similarly, it would require \( \beta_{ik} = 0 \) for \( k < i - 1 \) for any \( i \). We see that, of the five-stage options in Table 1, only the second-order one could be done using the low-storage approach.
Table 1
Coefficients for Five-stage, Strong-stability-preserving Runge–Kutta Integration Method

| Order | c | $\alpha_k$ | $\beta_k$ |
|-------|---|------------|-----------|
| 2     | 4 | 1          | 0         |
|       |    | 0          | 0         |
|       |    | 0          | 0         |
|       |    | 0          | 0         |
|       |    | 0          | 0         |
|       | 2  | 0          | 0         |
|       | 0.25 | 0   | 0.25     |
| 3     | 2.65062919294483 | 1 | 0
|       |    | 0          | 0         |
|       |    | 0          | 0         |
|       |    | 0          | 0         |
| 4     | 1.50818004975927 | 1 | 0
|       |    | 0          | 0         |
|       |    | 0          | 0         |

Table 2
Eigenmode Solutions of Linear Sonic and Magnetosonic Waves

| $\delta\rho$ | 1.52284 \times 10^{-8} + 0i |
|-------------|-----------------------------|
| $\delta\xi$ | 6.3 \times 10^{-7} + 0i     |
| $\delta\eta$ | 6.2 \times 10^{-7} + 0i    |
| $\delta\zeta$ | 6.1 \times 10^{-7} + 0i   |

Table 3
L1-norm Errors of Mass Density for Linear Wave Tests

| $N_s$ | 5   | 10  | 20  | 40  | 80  | 160 | 320 |
|-------|-----|-----|-----|-----|-----|-----|-----|
| FV-sonic | 6.3 \times 10^{-7} | 3.5 \times 10^{-7} | 1.0 \times 10^{-7} | 2.5 \times 10^{-8} | 6.2 \times 10^{-9} | 1.5 \times 10^{-9} | 3.8 \times 10^{-10} |
| FV-fast  | 6.4 \times 10^{-7} | 3.6 \times 10^{-7} | 1.0 \times 10^{-7} | 2.5 \times 10^{-8} | 6.2 \times 10^{-9} | 1.5 \times 10^{-9} | 3.5 \times 10^{-10} |
| FV-slow  | 6.5 \times 10^{-7} | 5.2 \times 10^{-7} | 1.4 \times 10^{-7} | 2.9 \times 10^{-8} | 6.4 \times 10^{-9} | 1.6 \times 10^{-9} | 3.8 \times 10^{-10} |
this gives an internal energy density of $e_0 = \rho_0 c_0 = 9.13705584 \times 10^{-3}$. When magnetic fields are present, the background field is evenly split between $x$ and $y$ components such that the Alfvén speed is $v_{\text{A},0} = 0.2$. The first-order perturbation constants $\delta q^*$ are provided in Table 2. The wave number is taken as $k = 2\pi/L$, where $L$ is the grid length set to unity for all tests. All calculations are run to as a function of $DG$ perturbation test in Table 3.

Figure 1. Plot of the $L_1$-norm errors in mass density for the slow magnetosonic wave perturbation test in Table 3.

We consider two cases: a stationary background ($V^x = 0$), where pulses travel in opposite directions with equal amplitudes (case ALF-1), and a moving background ($V^x = 0.1c$), where pulses split into asymmetrical waves (case ALF-2). These cases correspond to models ALF-1 and ALF-3 of De Villiers & Hawley (2003). The fluid is initialized with uniform unit density, zero transverse magnetic field components $B^y = B^z = 0$, specific energy $\epsilon = 10^{-2}$, and ideal gas constant $\Gamma = 5/3$. The longitudinal field component $B^x$ is set by the parameter $\beta = 0.001$ for ALF-1 and 0.01 for ALF-2. The transverse velocity function $f(x, t)$ is initialized as a square pulse; $V^y = f(x, 0) = 10^{-5}c$ for $1 < x < 1.5$, $V^y = f(x, 0) = -10^{-5}c$ for $1.5 < x < 2$, and zero everywhere else (the grid length runs from 0 to 3 units, with periodic boundary conditions). The Alfvén wave speeds are $|v^+_{\text{A}}| = 0.96c$ for ALF-1, and $v^+_{\text{A}} = 0.79c$ and $v^-_{\text{A}} = -0.71c$ for ALF-2.

Numerical results are plotted in Figure 2, where we compare analytic to DG(1) solutions. Two calculations are shown for both cases: one using entropy viscosity to capture the discontinuities, the second using the slope limiter. Solutions for the two discontinuity capturing approaches are very similar, and they both match the finite volume calculations. All solver permutations (DG, FV, viscosity, limiter) reproduce the plateau values to better than 0.002%, and converge globally to the analytic solution at rates close to unity.

Taking advantage of the semi-analytic nature of this solution, we additionally consider a smooth waveform for the function $f(x - \nu_{\text{A}} t) = f_0 \sin(2\pi x - \nu_{\text{A}} t)/L$, with small amplitude $f_0 = 10^{-8}$ to expand the perturbation regime. This allows us to perform convergence studies similar to those conducted in Section 4.1. Although this problem is not as rigorous a test for hydrodynamics as those presented above, it is nonetheless a useful diagnostic of magnetically dominated flows. For these series of tests, we use the fourth-order, five-stage Runge–Kutta integrator and extend our testing to include DG(4), fifth-order spatial discretization. L1-norm errors for $B^x$ are presented in Table 4 and plotted in Figure 3. As before, all calculations were run for three complete wave periods of the fast Alfvén mode. We find convergence rates generally consistent with the spatial order of each scheme: second order for the FV and DG(1) schemes, and order $p + 1$ for the DG($p$) methods.
Hydrodynamic Shocks

We consider two special relativistic hydrodynamic shock tube tests: a relatively mild boost case (H DST-1) with \( V = 0.7c \), and a second higher boost case (H DST-2) with \( V = 0.9c \). These tests set up two different fluid states separated by a membrane in the middle of the domain that is removed at \( t = 0 \). The fluid subsequently evolves to form a leftward propagating rarefaction wave, and rightward propagating contact discontinuity and shock wave. The initial data for H DST-1 is specified as \( \rho_L = 10, \ p_L = 10, \) and \( V^2_L = 0 \) to the left of the partition; \( \rho_R = 1, \ p_R = 10^{-2}, \) and \( V^2_R = 0 \) to the right; and a \( \Gamma = 4/3 \) ideal gas equation of state. H DST-2 is similar, but with a significantly greater pressure, to the left of the membrane, \( p_L = 170 \), which makes the leading Lorentz contracted density discontinuity much harder to resolve on meshes with limited cell resources.

All tests are performed separately with artificial viscosity or slope limiting, imposing flat (zero gradient) boundary conditions, and third- or fourth-order elements (DG(2), DG(3)) to demonstrate the robustness of the different shock regularization techniques and the application of high-order finite elements to shock problems. L1-norm errors of the mass density are shown in Table 5 for both cases, both regularizations, and across a range of grid resolutions to compute convergence rates. Errors are evaluated at a final time of \( t = 0.06 \). For the viscosity runs, common values of 0.2 and 0.6 are used for the linear and quadratic coefficients, respectively. The slope limiter calculations use a steepness parameter of unity. Notice that the errors quoted in Table 5 converge to the analytic solutions at roughly first order. This is what is expected for simulations that include strong discontinuities, such as shocks. The corresponding solutions are shown in Figure 4 where we plot mass densities at \( t = 0.15 \) for H DST-1 and \( t = 0.08 \) for H DST-2. Solid lines represent the analytic solutions derived by solving the exact Riemann problem, and dashed lines are the numerical solutions.
on grids resolving a domain from 0 to 0.5 with 1280 zones. The numerical solutions are calculated using slope limiting in the case of HDST-1 and artificial viscosity in the case of HDST-2. In the more difficult HDST-2 test, the analytic and calculated shock jump states agree to about 10% at the 1280 zone resolution used in producing Figure 4, but converge linearly with resolution.

4.4. Boosted Shock Collision

Anninos et al. (2005) derived an exact solution for the collision of two boosted fluids that tests the Lorentz invariance of the code under rigorous non-symmetric conditions, multiple jump discontinuities, and highly relativistic shocks. In the center-of-momentum frame, this problem consists of two colliding fluids, one flowing from the left, one from the right. The pre- and post-shock states of the two fluids are defined in the center-of-mass (primed) frame by zero post-shock velocities and pressure equilibrium assuming an infinite strength (cold fluid) approximation:

\[ P_{\text{post}} = \rho_{\text{post}} (\Gamma - 1)(W'_{\text{pre}} - 1), \]

\[ \rho_{\text{post}} = \frac{\rho_{\text{pre}}}{1 + \Gamma W'_{\text{pre}}}. \]

The observer is then boosted to the right at a specified velocity so that the shocked region appears to move to the left at very high velocities, even as they move apart (in opposite directions) in the center-of-momentum frame. The velocity of the center-of-mass frame and contact discontinuity is calculated by solving the nonlinear boost transformation equations. We do not repeat the derivation here, but refer the reader to Section 4.1.3 of Anninos et al. (2005) for a detailed discussion of the initialization, and to their Table 2, where the solutions of three specific cases are recorded.

These tests produce highly relativistic shocks that require extremely fine zoning to properly capture the jump conditions. We achieve this with adaptive mesh refinement, using up to seven levels of refinement on top of a base grid of length 0.06 covered by 80 zones. In addition, we have run these problems with both AMR and AOR in combination to test the simultaneous use of both refinement techniques, although in practice high-order polynomials are ineffective in these shock-dominated cases, producing results essentially identical to low-order solutions. We have reproduced solutions of comparable quality for all of the cases derived in Anninos et al. (2005), but show representative results in Figure 5 for the case where the colliding fluids have different densities. The two fluids have initial proper densities \( \rho_1 = 2 \) (left) and \( \rho_2 = 1 \) (right), pressures \( P = 10^{-6} \), and adiabatic index \( \Gamma = 5/3 \). In the center-of-mass frame, the fluids move in opposite directions, each with \( W' = 5 \). The observer is boosted to the right at \( W = 3 \), so the fluid moves at speeds up to 0.999c. The two curves in Figure 5 are solutions for mass densities calculated with artificial viscosity shock-capturing and flat boundary conditions at two different times: \( t = 0.01 \) (dashed) and \( t = 0.02 \) (solid), showing the shocked fluid moving to the left. The corresponding solutions with slope limiting appear very similar. For the artificial viscosity solution, we find fractional density errors of \( \approx 5 \times 10^{-2} \) and \( 2 \times 10^{-2} \) in the higher and lower density fluids, respectively.

Figure 4. Mass density \( \rho \) for the hydrodynamic shock tube test HDST-1 (left) at time \( t = 0.15 \), and HDST-2 (right) at time \( t = 0.08 \). Solid lines are the analytic Riemann solutions, and dashed lines are the numerical solutions using slope limiting for shock regularization in the case of HDST-1 and artificial viscosity in the case of HDST-2. Numerical solutions are calculated on a domain from 0 to 0.5 with 1280 zones.

Figure 5. Mass density \( \rho \) for the ultra-relativistic colliding shock test at two different times \( t = 0.01 \) (solid) and 0.02 (dashed), showing the fluids moving to the left at velocity 0.999c.
lower density post-shock plateaus, respectively, and energy density errors at the contact discontinuity of about $\delta e/e \approx 7 \times 10^{-2}$. Errors are slightly better for the slope limiter calculations: $\delta P/P \approx 3 \times 10^{-2}$ ($2 \times 10^{-3}$) for the high (low) density plateaus, and $\delta e/e \approx 10^{-3}$ for the post-shock energy density.

### 4.5. MHD Shocks

Next, we consider three magnetohydrodynamic shock tube tests: the first two are taken from Komissarov (1999), and the third is a relativistic version of the Brio-Wu shock tube (van Putten 1993). All are initialized with zero velocities and discontinuities separating the left and right states partitioned at the center of the grid. Additionally, all three are run with the same $\Gamma = 4/3$ ideal gas equation of state. Using the subscripts “L” and “R” to denote left and right states, the initial data are $\rho_L = 1$, $P_L = 1000$, $B_{L}^x = 1$, $B_{L}^y = 0$, $\rho_R = 0.1$, $P_R = 1$, $B_{R}^x = 1$, and $B_{R}^y = 0$ for case MHDST-1; $\rho_L = 1$, $P_L = 30$, $B_{L}^x = 0$, $B_{L}^y = 20$, $\rho_R = 0.1$, $P_R = 1$, $B_{R}^x = 0$, and $B_{R}^y = 0$ for case MHDST-2; and $\rho_L = 1$, $P_L = 1$, $B_{L}^x = 0$, $B_{L}^y = 1$, $\rho_R = 0.125$, $P_R = 0.1$, $B_{R}^x = 0$, and $B_{R}^y = -1$ for the Brio-Wu case MHDST-3.

In Table 6, we display the initial and final calculated values for each state of the shock tubes: “Left” is the left initial state, “FL” is the value at the foot of the left fast rarefaction wave, “SC” is the value at the slow compound wave, “CDL” is the left contact discontinuity, “CDR” is the right discontinuity, “FR” is the value at the foot of the right fast rarefaction fan, and “Right” is the initial right state. The numbers presented in Table 6 correspond to calculations run with artificial viscosity, but we note that results with slope limiting are similar and generally match the viscosity results to within a few percent. Like the hydrodynamic shock tube tests, these problems were run with flat boundary conditions, third-order finite elements to demonstrate the robustness of high-order DG on magnetized shock problems. Representative solutions of the mass density calculated on a 1024 zone grid are plotted in Figure 6 showing results from all three tests.

### 4.6. Orszag–Tang

The Orszag–Tang vortex problem (Orszag & Tang 1979) has become a standard test of magnetic fields and divergence conservation. Numerous solutions exist in the literature to which we can compare our results. In particular, we follow and adopt initial data from Sadowski et al. (2014) and Mocz et al. (2014): uniform density $\rho = 1^2/(4\pi)$, pressure $P = \Gamma/2(4\pi)/C^2$, velocity $V^i = [-\sin(2\pi y), \sin(2\pi x), 0]/C$, magnetic field $B^i = [-\sin(2\pi y), \sin(4\pi x), 0]/4\pi/C$, adiabatic index $\Gamma = 5/3$, and a scale factor $C = 100$. The problem is evolved out to a time of $t = 50$ on a $256 \times 256$ unit two-dimensional grid $0 \leq (x, y) \leq 1$ with periodic boundary conditions applied in both directions. Figure 7 shows the mass density and divergence error at the final time using third-order DG(2) finite elements. Also shown in Figure 8 is a horizontal line out of the density multiplied by $4\pi$ along $y = 0.75$, $4\pi \rho(x, y = 0.75)$. Both figures can be compared to the corresponding results from Figure 4 of Sadowski et al. (2014) and Figure 16 of Mocz et al. (2014). Agreement is excellent, considering that the solutions in Sadowski et al. (2014) and Mocz et al. (2014) were calculated on greater $640 \times 640$ and $512 \times 512$ resolution grids, respectively. In addition, we find global normalized divergence errors $\delta B/\Delta \ell/\sqrt{P_R}$ comparable to those reported by Mocz et al. (2014): roughly a few $10^{-3}$ that plateau early and remain constant throughout most of the simulation.

### 4.7. Kelvin–Helmholtz Instability

In this section, we present convergence studies of the linear growth phase of the two-dimensional magnetized Kelvin–Helmholtz instability (KHI). Following interesting observations by Mignone et al. (2009) on the performance of various Riemann solvers in this class of problems, Beckwith & Stone (2011) published a brief study of KHI turbulence that provides a useful test of high-order adaptive numerical methods, so we attempt to duplicate some of their findings here. The problem consists initially of two oppositely traveling $\Gamma = 4/3$ fluids in pressure equilibrium $P = 1$, densities $\rho = 1$ and $10^{-2}$, and the following velocity profiles:

\[ V_x = -V_{\text{shear}} \tan h \left( \frac{y - y_0}{a} \right), \]  

\[ V_y = -A_0 V_{\text{shear}} \sin(2\pi x) \exp \left( -\left( \frac{y - y_0}{\sigma} \right)^2 \right), \]  

with shear velocity $V_{\text{shear}} = 0.5$, perturbation amplitude $A_0 = 0.1$, shear layer thickness $\alpha = 0.01$, characteristic length scale $\sigma = 0.1$, and interface position $y_0 = L_z/2$, where $L_z$ is the grid length along the y-axis. The density is linearly interpolated similar to the shear velocity $V^i$ profile along the y-axis so that $\rho = 1$ in regions with $V^i = 0.5$ and smoothly extended to $\rho = 10^{-2}$ in regions with $V^i = -0.5$. A single-component magnetic field aligned along the x-direction with $B^x = 10^{-3}$ is introduced. In addition, 1% Gaussian perturbations are applied to both $x$ and $y$ components of the velocity, modulated by the same exponential damping function used in Equation (58). The computation domain covers $0 \leq (x, y) \leq 1$, and for the AMR calculations is resolved with a base grid of $64 \times 64$ cells. Periodic (reflection) boundary conditions are enforced along.

Table 6

| Variable | Left | FL | SC | CDL | CDR | FR | Right |
|----------|------|----|----|-----|-----|----|-------|
| MHDST-1: | $\rho$ | 1.0 | 0.07 | ... | 0.69 | ... | ... | 0.1 |
| MHDST-1: | $P$ | 1000 | 28.5 | ... | ... | ... | ... | 1.0 |
| MHDST-2: | $\rho$ | 1.0 | 0.24 | ... | 0.63 | ... | ... | 0.1 |
| MHDST-2: | $P$ | 30. | 4.6 | ... | 15.5 | ... | ... | 1.0 |
| MHDST-3: | $\rho$ | 1.0 | 0.51 | 0.67 | 0.55 | 0.35 | 0.11 | 0.125 |
| MHDST-3: | $P$ | 1.0 | 0.41 | 0.59 | 0.45 | 0.45 | 0.08 | 0.1 |
threshold refinement (derefinement) criteria for all cases is set to \( s_r = 0.05 \) (0.001), where \( s_r = |\hat{\ell} \cdot \partial \rho / \rho| \), \( \hat{\ell} \) is a vector of cell widths in each spatial dimension, and \( \bar{\rho} \) is the local (nearest neighbor) average of the mass density.

A good diagnostic of the linear growth stage of the KHI is the temporal history of the square of the transverse four-velocity weighted by cell volume and averaged over the entire grid, \( \langle |u|^2 \rangle \). This quantity is plotted in Figure 9 for four cases: KHFV representing the converged second-order finite volume solution, KHDG1-3L using second-order DG(1) with three AMR levels, KHDG1-4L using DG(1) with four AMR levels, and KHDG2-512 using third-order DG(2) (both adaptive and fixed polynomial order) on a uniform \( 512 \times 512 \) mesh. The finite volume calculation (KHFV; solid line) reproduces converged results from Beckwith & Stone (2011), matching the slope, magnitude, and peak position in time. These four calculations collectively demonstrate convergence toward the resolved solution with both mesh and basis order refinement. Notice in particular that run KHDG1-4L (second order with four AMR levels, or effectively \( 1024 \times 1024 \) resolution across the interface) is nearly identical to the result of KHDG2-512 (third order with \( 512 \times 512 \) resolution), and both significantly improve compared to the lowest-resolution result KHDG1-3L (second order with three AMR levels).

The density distribution is shown in Figure 10 at time \( t \approx 3.3 \) using the DG(1) method on a single \( 1024 \times 1024 \) grid. Interestingly, this calculation exhibits signs of a developing secondary vortex at \( x \approx 0.25 \) that is not present in the second-order calculations with resolutions less than \( 1024 \times 1024 \). Similar features, however, are observed at lower resolutions, provided high-order (greater than second) finite element representations are utilized. The third-order KHDG2-512 calculation, for example, produces a similar feature at \( 512 \times 512 \) resolution. Sensitivities associated with the development of a secondary vortex have been observed by Mignone et al. (2009) and Beckwith & Stone (2011), who attributed this behavior to the accuracy of the Riemann solver and its ability to capture the contact discontinuity. That DG evolves this feature with a two-speed HLL Riemann solver with no contact steepening is encouraging and represents yet another potential benefit of the DG methodology.

### 4.8. Bondi Accretion

A popular test with general relativistic spacetime curvature source terms is radial accretion onto a compact Schwarzschild black hole. The analytic solution is characterized by a critical point \( r_c \) in the flow (Michel 1972),

\[
(u_r')^2 = \frac{M}{2r_c},
\]

\[
v_c^2 = \frac{(u_r')^2}{1 - 3(u_r')^2} = \frac{(1 + n)T_c}{n(1 + 1 + n)T_c},
\]

where \( u_c' \) and \( v_c \) are the radial four-velocity and sound speed at the critical point, respectively, \( M \) is the black hole mass, \( n = 1/(\Gamma - 1) \) is the polytropic index, and \( T = P / \rho \) is the fluid temperature. The solution is completed with the following parametrization:

\[
T^n u' r^2 = C_1.
\]
The constants $C_1$ and $C_2$ are fixed by choosing the critical radius $r_c = 8GM/c^2$, setting $\Gamma = 4/3$, and defining the mass density at the critical radius ($\rho_c$) by setting the mass accretion rate to $M = 4\pi r_c^2 \rho_c u_c^r = -1$.

We choose spherical Kerr–Schild coordinates for this test to cover a two-dimensional computational domain bounded in radius from $r = 0.98 r_{\text{BH}}$ to $r = 20GM/c^2$, where $r_{\text{BH}} = 2GM/c^2$ is the radius of the black hole horizon. The angular extent is a thin wedge centered along the equatorial symmetry axis of width $\Delta \theta = \pi/20$. The Bondi solution is initialized from the outset at $t = 0$ then evolved over a time interval of $5GM/c^2$. Constant boundary conditions consisting of the analytic solution are imposed at the inner and outer radial boundaries, and symmetric boundaries are enforced in the angular coordinate. Accuracy and convergence are evaluated by calculating the L1-norm errors in density between the initial and final times along the equatorial plane over the entire radial extent of the grid. Although COSMOSDG supplies analytic metric gradients for many black hole metric representations, for this test we instead evaluate gradients numerically in order to test our implementation of finite element gradient operators. A series of nine calculations were performed with three DG orders (second, third, fourth) and three grid resolutions: $N_x \times N_y = 16 \times 2$, $32 \times 4$, and $64 \times 8$, where $N_x$ and $N_y$ are the number of zones along the radial and angular directions. All calculations were run with fourth-order time integration using the five-stage Runge–Kutta method. Like previous smooth field tests, we find DG methods produce significantly smaller evolution errors than FV, an order of magnitude or more depending on the basis order, and converge to the analytic solution at the appropriate rate. For example, FV
produces L1-norm errors of $3.8 \times 10^{-4}$ in mass density computed on a $64 \times 8$ grid. Equivalent errors from the DG (1), DG(2), and DG(3) methods come to $1.5 \times 10^{-4}$, $1.3 \times 10^{-5}$, and $1.2 \times 10^{-7}$, respectively. In addition, we find with each doubling of zones that errors are reduced by factors of about $2^{n+1}$ for methods DG($\rho$) as expected and as demonstrated in Table 7 and Figure 11.

This hydrodynamic black hole accretion test can be adapted to include a radial magnetic field satisfying $\partial_r B^r = 0$ and which does not alter the analytic solution for any of the primitive fields ($\rho$, $p$, or $u^i$). Although this treatment does not satisfy the full Maxwell equations (Antón et al. 2006), it is a useful nontrivial test of magnetic fields in the code. We set the magnitude of the magnetic field by $|b|^2/\rho = 10.56$ at the critical point, effectively equating hydrodynamic and magnetic pressures at $r = r_\text{c}$. Performing equivalent calculations (identical grids, resolutions, fluid and black hole parameters) to the hydrodynamic version, we find errors similar to those presented in Table 7.

**4.9. Magnetized Black Hole Torus**

For a final test, we expand on the Bondi accretion problem and consider a magnetized torus of gas orbiting around a rotating black hole. There is no analytic solution for this problem so in its place we instead compare a DG rotating black hole. There is no analytic solution for this and consider a magnetized torus of gas orbiting around a

background gas) obeys an ideal gas equation of state with adiabatic index $\Gamma = 5/3$.

In order to seed the magnetorotational instability (MRI), the torus is threaded initially with a weak poloidal magnetic field derived from the following vector potential:

$$A_\phi = \begin{cases} k(\rho - \rho_{\text{cut}}) & \text{if } \rho > \rho_{\text{cut}}, \\ 0 & \text{otherwise,} \end{cases}$$

where $\rho_{\text{cut}} = 0.5\rho_{\max}$ effectively keeps the magnetic field inside the torus surface. The field is normalized by defining the constant $k$ so that $\beta = P/\mathcal{B} \geq 2$ throughout the torus. Although initially confined to the torus, eventually the field evolves to fill most of the background with a magnetized corona and high-$\beta$ outflows, in addition to seeding the MRI and launching an accretion flow from the torus to the black hole. It is this accretion flow that we use as a diagnostic for comparing results from the different numerical methods.

This problem is run on a two-dimensional, azimuthally symmetric grid resolved effectively with $128 \times 128$ cells; the finite volume calculation representing the “known” solution is run on a single $128 \times 128$ grid, while the finite element calculations are run on a two-level nested grid with a $64 \times 64$ base resolution to demonstrate the simultaneous use of hierarchical mesh and basis order refinement. We typically simulate accretion disks in this mode (with nested grids) to achieve greater resolution along the equatorial plane while simultaneously avoiding refining along the pole axis to relax the Courant constraint. However, we have verified that replacing nested grids with fully adaptive mesh refinement produces similar results. We also use a logarithmic radial coordinate of the form $r = 1 + \ln(\eta/r_{\text{BH}})$ and a concentrated angular coordinate $\xi$ such that $\theta = x_3 + \sin(2\xi)/4$ to further increase the resolution near the black hole horizon.
and along the equator. The grid covers $0.1 \leq \theta \leq 0.9\pi$ and $0.98_{\text{H}} \leq r \leq 120M$, resulting in cell widths of $\Delta r \approx 0.05GM/c^2$ near the inner radial boundary and $\Delta r \approx 0.5GM/c^2$ near the initial pressure maximum of the torus. By comparison, the characteristic wavelength of the MRI is $\lambda \equiv 2\pi\sqrt{\nu/\Omega} \approx 2.5GM/c^2$ near the initial maximum. Symmetric (reflective) boundary conditions are imposed along the radial (angular) directions.

Our comparison is between a traditional finite volume with piecewise parabolic reconstruction (case BHT-FV) and an adaptive order refinement using second-order finite elements in the background gas and third order in the torus body (case BHT-DG2), refining and derefining the polynomial order on the gas density (refining when the density exceeds 0.0005, derefining when the density drops below 0.0001). Figures 12 and 13 show images of the logarithmic gas density, comparing the BHT-FV and BHT-DG2 solutions at two different times in the evolutions. The left images correspond to an early time when the torus stream first hits the horizon. The right images show a later snapshot when the flow has fully developed after a couple of orbits. Figure 14 compares mass accretion rates as a function of time for each of the cases. We expect differences due to the turbulent nature of these calculations, but the level of differences between the methods appears significant, paying particular attention to the temporal variability of the mass accretion, the duration of the flow period before it begins to taper off, and the total accreted mass (about 12% and 19% of the initial torus mass after five orbits for cases BHT-FV and BHT-DG2, respectively). These results are consistent with previous similar calculations (Anninos et al. 2005), although
we note the slightly greater total accreted mass for the finite element calculation. More exhaustive studies of black hole accretion will be conducted in a future work to understand better the accuracy and benefits of high-order methods for capturing subzonal effects. Here we have taken the first step toward this goal, validating the DG finite element methodology and demonstrating its utility to this class of problems.

5. Discussion

We have developed a new version of COSMOS++, called COSMOSDG, a code for both Newtonian and general relativistic radiation magnetohydrodynamics, based on the Discontinuous Galerkin (DG) finite element formulation. The code infrastructure in COSMOSDG was upgraded to accommodate the simultaneous use of cell-by-cell adaptive mesh refinement (AMR) together with cell-by-cell adaptive basis order refinement (AOR). Our current implementation utilizes Lagrange interpolatory functions to construct local finite element approximations of arbitrary spatial order, with multiple options for high-order time integration, including third-order forward Euler, and fourth-order strong-stability preserving Runge–Kutta. Generalization to two and three dimensions is accomplished through tensor products of one-dimensional basis functions. Although our DG implementation can construct basis polynomials of arbitrary order, in practice we find that third or fourth order is a practical upper limit set by computational speed and saturation of errors due to machine precision constraints.

Two options are provided for the regularization of shock discontinuities: artificial viscosity and/or slope limiters (they can be invoked separately or together). Artificial viscosity is applied in a conservative manner using covariant Laplacian smoothing operators with properly normalized entropy or relativistic enthalpy diagnostics to trigger the application of viscosity locally when cell interface jumps or excessive subzonal heating is detected. The slope limiter option is a modification of the traditional minmod limiter commonly used in finite volume and finite difference codes, generalized here to work in multiple dimensions and for arbitrary order finite elements by projecting high-order solutions to a low-order basis using a least-squares method to compute slopes. Both approaches were demonstrated to provide adequate stabilization of shocks, for all polynomial orders. We note that artificial viscosity is implemented as subzonal dissipation terms, preserving the high-order nature of DG(p) in each cell. Slope limiting, however, is currently applied across zone interfaces like traditional finite volume methods, effectively folding high-order subgrid data into second-order solutions before enforcing monotonicity. We are currently investigating a number of approaches for extending slope limiting to work directly on unfiltered subgrid data.

We demonstrated the ability of DG finite element methods to achieve arbitrarily high-order convergence on perturbation problems with smooth profiles (i.e., sonic and magnetosonic waves), limited only by analytic solution and machine precision limits. Our extensive testing of DG methods furthermore proved them equal to or, in most cases, better than finite volume methods, using high-resolution shock-capturing or central difference schemes, even for problems with highly relativistic shocks, regimes with strong discontinuities where high order can break down and lead to unwanted Gibbs effects. Perhaps more importantly, we have subjected this methodology to the rigors of multidimensional modeling of energetic astrophysical environments, complex hydrodynamic instabilities, and strong spacetime curvature, successfully demonstrating its application to Bondi–Hoyle and MRI-induced black hole accretion.

This work was performed in part under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National Science Foundation grant number ACI-105575. P.C. F. and D.N. acknowledge support from National Science Foundation grants AST-1211230 and AST-1616185. P.C.F. was supported in part for this work by the National Science Foundation under grant NSF PHY-1125915.

ORCID iDs

P. Chris Fragile https://orcid.org/0000-0002-5786-186X
A. Miguel Holgado https://orcid.org/0000-0003-4143-8132

References

Anninos, P., & Fragile, P. C. 2003, ApJS, 144, 243
Anninos, P., Fragile, P. C., & Murray, S. D. 2003, ApJS, 147, 177
Anninos, P., Fragile, P. C., & Salmonson, J. D. 2005, ApJ, 635, 723
Antón, L., Zanotti, O., Miralles, J. A., et al. 2006, ApJ, 637, 296
Babuška, I., Ženíšek, O., C., Gago, J., & de A. Oliveira, E. R. (ed.) 1986, Accuracy Estimates and Adaptive Refinements in Finite Element Computations (Chichester: Wiley)
Beckwith, K., & Stone, J. M. 2011, ApJS, 193, 6
Cockburn, B., Lin, S.-Y., & Shu, C.-W. 1989, JCoPh, 84, 90
Cockburn, B., & Shu, C.-W. 1989, MaCom, 52, 411
Cockburn, B., & Shu, C.-W. 1998, JCoPh, 141, 199
De Villiers, J.-P., & Hawley, J. F. 2003, ApJ, 589, 458
Fragile, P. C., Gillespie, A., Monahan, T., Rodriguez, M., & Anninos, P. 2012, ApJS, 201, 9
Fragile, P. C., Olejar, A., & Anninos, P. 2014, ApJ, 796, 22
Gottlieb, S., & Shu, C. W. 1998, MaCom, 67, 73
Guermond, J.-L., Pasquetti, R., & Popov, B. 2011, JCoPh, 230, 4248
Harten, A., Lax, P. D., & van Leer, B. 1983, SIAMR, 25, 35
Hartmann, R., & Houston, P. 2002, JCoPh, 183, 508
Johnson, C., Naerti, U., & Pitkaranta, I. 1984, CMAME, 45, 285
