Journal of Computational Physics **152,** 608–641 (1999)
Article ID jcph.1999.6248, available online at http://www.idealibrary.com on IDEAL®

A Discontinuous Galerkin Method for the Viscous MHD Equations

T. C. Warburton[1] and G. E. Karniadakis[2]

Center for Fluid Mechanics, Division of Applied Mathematics, Brown University,
Providence, Rhode Island 02912

Received September 21, 1998; revised February 22, 1999

We present a new high-order method for the unsteady viscous MHD equations in two and three dimensions. The two main features of this method are: (1) the discontinuous Galerkin projections for both the advection and diffusion components, and (2) the polymorphic spectral/*hp* elements for unstructured and hybrid discretizations. An orthogonal spectral basis written in terms of Jacobi polynomials is employed, which results in a matrix-free algorithm and thus high computational efficiency. We present several results that document the high-order accuracy of the method and perform a systematic *p*-refinement study of the compressible Orszag–Tang vortex as well as simulations of plasma flow past a circular cylinder. The proposed method, which can be thought of as a high-order extension of the finite volume technique, is suitable for direct numerical simulations of MHD turbulence as well as for other traditional MHD applications. © 1999 Academic Press

1. INTRODUCTION

There has been recently a renewed interest in developing numerical algorithms for the solution of compressible magnetohydrodynamics (MHD) equations [1–6]. This interest stems from a wide range of new applications and emerging technologies such as advanced plasma thrusters for space propulsion, high-power microwave and electro-magnetic pulse devices, processing of semiconductors, plasma-assisted drag reduction techniques in hypersonic flight, electro-magnetic turbulence control for conducting fluids, and models of the solar wind.

The governing equations describing both dense and sparse plasmas are time-dependent, and a realistic description of geometries involved in the aforementioned applications requires complex three-dimensional computational domains. In addition, these equations are

[1] Current address: Oxford University Computing Laboratory, Oxford, UK.

[2] To whom correspondence should be addressed.

strongly coupled and exhibit mixed hyperbolic/parabolic character depending on the parameter range, with a large range of temporal and spatial scales involved, and with complicated boundary conditions. Most of these issues have been adequately addressed in the published works, which are primarily concerned with the case of ideal MHD, thus neglecting the effect of viscosity and resistivity. However, one of the limitations of the current numerical methods is that they are of low-order (at most second-order) accurate. High-order accuracy is very important for accurate *long-time* integration of wave propagated phenomena, as it dictates the efficiency and thus the feasibility of predicting such long-term behavior [7, 8]. In addition, certain flows such as compressible MHD turbulence exhibit a very rich structure that includes features not observed in incompressible flows, e.g., small-scale structure massive jets [9]; high-order methods are more suitable for resolving such structures effectively.

Two of the main difficulties in employing high-order discretization for the solution of hyperbolic conservation laws are: (1) maintaining monotonicity for non-smooth solutions, and (2) preserving conservativity. Progress has been made in both collocation and Galerkin discretizations [10–12] but at the expense of extra computational complexity making these methods inefficient compared with their low-order counterparts. In the MHD framework, such difficulties are compounded by the imposition of the *divergence-free* condition for the magnetic field, which results in a loss of the hyperbolicity of the ideal MHD equations. This condition has been dealt with by employing staggered grids in the work of Evans and Hawley [13], which was extended more recently by Peterkin *et al*. [3]. However, such an approach cannot be easily incorporated in high-order discretizations. Alternative approaches include the operator-splitting algorithm proposed by Zachary *et al*. [2] and the development of extended Riemann solvers by Powell [4]; the latter is easily extended to multi-dimensions and also to high-order discretization. More recently, a different formulation for the MHD system has been proposed by Meir and Schmidt [14] who employed the *current flux* instead of the magnetic flux in their formulation.

The aforementioned difficulties of high-order discretization have been more recently addressed in the context of a discontinuous Galerkin formulation for the compressible Navier–Stokes equations in [15, 16]; a somewhat different version has also been proposed in [17, 18], and more for the Euler equations in [19]. The new formulation is *flux-based* and allows the incorporation of approximate Riemann solvers in the variational statement, providing in essence a variational (Galerkin) framework for what can be described as a finite volume formulation. In this context, it is then straightforward to incorporate a high-order basis in the discretization. The formulation also allows for finite jumps (L^2 continuity) across interfaces even for second-order operators, and thus a computationally efficient trial basis can be selected. To this end, we will employ spectral bases written as tensor-products in terms of one-dimensional Jacobi polynomials. We have obtained such bases for polymorphic domains including triangles and quadrilaterals in two dimensions, and tetrahedra, hexahedra, prisms, and pyramids in three dimensions. Conservativity is honored in the element-wise sense automatically, while monotonicity is satisfied by lowering the order of the expansion around discontinuities following a standard h-type refinement procedure [15].

In the following, we first review in Section 2 the spatial discretization we employ and we present more details in Appendixes I and II. We then formulate our version of the discontinuous Galerkin method for the advection and diffusion equations in Section 3. Subsequently, we present the specific algorithms for the MHD equations in Section 4. In Section 5 we first present two numerical examples that demonstrate exponential (spectral-like) convergence for smooth analytical solutions. We then perform a systematic refinement study of the Orszag–Tang vortex problem [20], as well as comparisons of incompressible and compressible MHD flows past a circular cylinder. We conclude in Section 6 with a brief summary.

2. SPATIAL HIGH-ORDER DISCRETIZATION: SPECTRAL/*hp* ELEMENT METHOD

We first review the spectral/*hp* element method that we employ in the proposed algorithm for spatial discretization. We break up the computational domain into subdomains or elements, which in two dimensions may be triangles or quadrilaterals or a combination of both. This is shown in Fig. 1 (left) where the domain is broken up into 38 quadrilaterals and 22 triangles. In three dimensions, we use tetrahedra, hexahedra, prisms, and pyramids or a combinations of these in order to accommodate the geometric complexity of the problem. We then expand unknowns and data within each one of these elements in terms of a suitable polynomial basis $\phi(\mathbf{x})$ using a local coordinate system (ξ_1, ξ_2) associated with that element. For example, the function $f(x, y)$ is approximated as

$$f(x, y) \approx \sum_p \sum_q a_{pq}\phi_{pq}(\xi_1, \xi_2),$$

where the a_{pq} are the unknown coefficients. The maximum value of the indices p, q is the order of the polynomial expansion which we will refer to in this paper as "p-order" and we will denote it by N. In order to achieve convergence we have two options: Either to keep the p-order fixed and increase the number of elements (the so-called "h-refinement") or to fix the number of elements and increase the p-order ("p-refinement"). The corresponding convergence rate is algebraic in the former case but exponential in the latter case assuming that smooth solutions are sought. Exponential convergence implies that by doubling the number of degrees of freedom the error will decay by at least two orders of magnitude whereas algebraic convergence implies that the error will decay by an algebraic factor, e.g., a factor of 4 for a second-order scheme. We will demonstrate this dual path to convergence in the results section (see Section 5). We also prove theoretically in Appendix I

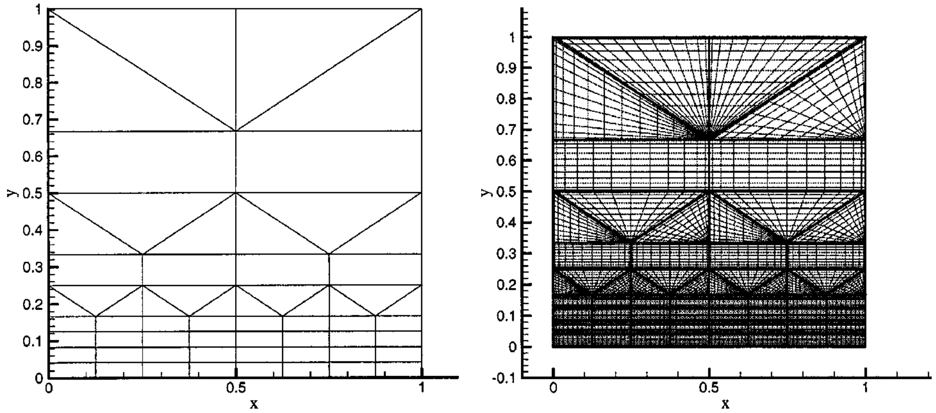

FIG. 1. Left, domain decomposition into triangular and quadrilateral elements. Right, quadrature grid indicating the location of quadrature points at the intersection of grid lines.

why the decay of coefficients is exponential which justifies the corresponding exponential convergence.

In addition to the element decomposition we also need a set of points where we evaluate all the integrals, i.e., inner products and boundary contributions in the discontinuous Galerkin formulation (see below). This quadrature grid is shown in Fig. 1 (right); the number of quadrature points (i.e., the "quadrature order") is related to the spectral order and is chosen so that the quadrature is exact (see Appendix II).

Returning now to the construction of a suitable basis $\phi_{pq}(\xi_1, \xi_2)$, we are interested in developing a computationally efficient *high-order* expansion which demonstrates attractive numerical properties such as matrix conditioning and, in the case of convection problems, appropriate explicit time step restrictions. Therefore, an appropriate starting point in developing the multi-dimensional expansions is to construct a set of polynomial expansions which are orthogonal in the Legendre inner product over each unstructured region. The following expansion was proposed in two dimensions by Dubiner [21] and extended to three dimensions in [22]. In Appendix I we present a theoretical derivation of this basis for polymorphic elements in two and three dimensions; see also [23]. Here we present the orthogonal basis for triangular elements,

$$\phi_{pq}(\xi_1, \xi_2) = \tilde{\psi}_p^a(\eta_1)\tilde{\psi}_{pq}^b(\eta_2),$$

where the coordinates ξ_1, ξ_2 are Cartesian whereas the η_1, η_2 are non-Cartesian and will be defined below. Figure 2 illustrates the construction of the two-dimensional expansion modes using this general form. To generate each mode the function $\tilde{\psi}_p^a(\eta_1)$ is combined with $\tilde{\psi}_{pq}^b(\eta_2)$. However, unlike a quadrilateral expansion, $\tilde{\psi}_{pq}^b(\eta_2)$ has a different form for every value of p of the principal function $\tilde{\psi}_p^a(\eta_1)$.

Let us now denote by $P_N^{\alpha,\beta}(z)$ the Nth-order Jacobi polynomial of weights α and β (see also Appendix I). Then, the principal functions, $\tilde{\psi}_i^a(z)$ and $\tilde{\psi}_{ij}^b(z)$, for orthogonal expansions are

$$\tilde{\psi}_i^a(z) = P_i^{0,0}(z), \qquad \tilde{\psi}_{ij}^b(z) = \left(\frac{1-z}{2}\right)^i P_j^{2i+1,0}(z).$$

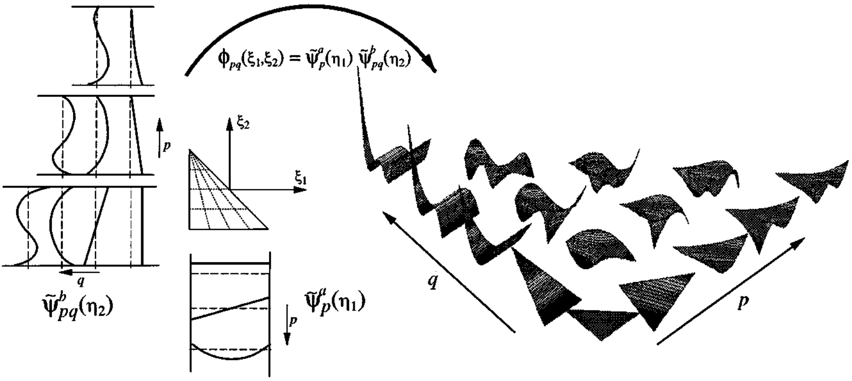
$\Phi_{pq}(\xi_1,\xi_2) = \tilde{\psi}_p^a(\eta_1)\,\tilde{\psi}_{pq}^b(\eta_2)$
ξ_2
ξ_1
p
q
$\tilde{\psi}_{pq}^b(\eta_2)$
p
$\tilde{\psi}_p^a(\eta_1)$
q
p

FIG. 2. Construction of two-dimensional expansion modes $\phi_{pq}(\xi_1, \xi_2)$ within a triangular region using the product of a one-dimensional tensor $\tilde{\psi}_p^a(\eta_1(\xi_1, \xi_2))$ and a two-dimensional tensor $\tilde{\psi}_{pq}^b(\eta_2(\xi_2))$.

The two-dimensional expansions in terms of the principal functions are defined as

$$\begin{aligned} \text{quadrilateral expansion,} \qquad & \phi_{pq}(\xi_1, \xi_2) = \tilde{\psi}^a_p(\xi_1)\tilde{\psi}^a_q(\xi_2) \\ \text{triangular expansion,} \qquad & \phi_{pq}(\xi_1, \xi_2) = \tilde{\psi}^a_p(\eta_1)\tilde{\psi}^b_{pq}(\eta_2), \end{aligned}$$

where

$$\eta_1 = \frac{2(1+\xi_1)}{(1-\xi_2)} - 1, \qquad \eta_2 = \xi_2.$$

The shape of all the triangular modes for a fourth-order polynomial expansion are shown in Fig. 2. Similarly, we can obtain the three-dimensional expansions (see Appendix I).

These expansions are all polynomials in terms of both system of coordinates, i.e., (ξ_1, ξ_2) *and* (η_1, η_2). The expansions in the quadrilateral and hexahedral domains are simply standard tensor products of Legendre polynomials in terms of Cartesian coordinates since $P^{0,0}_p(z) = L_p(z)$. The development of unstructured expansions using the local "collapsed" coordinates (η_1, η_2) is linked to the use of the more unusual function $\tilde{\psi}^b_{ij}(z)$. This function contains factors of the form $(\frac{1-z}{2})^n$ which are necessary to keep the expansions as polynomials in terms of the Cartesian coordinates (ξ_1, ξ_2).

3. DISCONTINUOUS GALERKIN FORMULATION

Discontinuous Galerkin projections provide great flexibility in domain-decomposition methods as they allow L^2 jumps across subdomains even for operators of order higher than first. This, in turn, implies that any convenient complete set of trial functions can be employed. In our method, we have chosen to employ the aforementioned high-order representations using hierarchical and orthogonal bases that lead to high computational efficiency. In the Galerkin framework this is not possible as, for example, for second-order operators a C^0 continuity requirement is imposed across the subdomains. This constraint results in at least partial loss of orthogonality if hierarchical spectral bases are employed for spatial discretization [24]. However, if the spectral bases are not constrained as in the current formulation full orthogonality is restored.

We present first the discontinuous Galerkin (DG) formulation for a generic system of advection-diffusion equations of the form

$$\vec{U}_t + \nabla \cdot \mathbf{F}^{\text{Ideal}} = \upsilon \nabla \cdot \mathbf{F}^{\text{Visc}}, \tag{1}$$

where $\mathbf{F}^{\text{Ideal}}$ and $\mathbf{F}^{\text{Visc}}$ correspond to inviscid and viscous flux contributions, respectively. Specific implementation issues for the MHD system will be discussed separately in Section 4. Splitting the advection-diffusion operator in this form allows for a separate treatment of the inviscid and viscous contributions, which in general exhibit different mathematical properties. In the following, we review briefly the discontinuous Galerkin formulations employed in the proposed method. A rigorous analysis of the advection operator was presented in [15], where a mixed formulation was used to treat the diffusion terms. No flux limiters are necessary as has been found before in [25] and has been justified theoretically in [26]. We present a formulation similar to the one developed by Cockburn and Shu [27] for finite elements but with important modifications both in the variational form as well as in the time-stepping algorithm.

3.1. *Discontinuous Galerkin for Advection*

To explain the formulation we consider the linear two-dimensional equation for advection of a conserved quantity u in a region Ω

$$\frac{\partial u}{\partial t} + \nabla \cdot \mathbf{F}(u) = 0, \tag{2}$$

where $\mathbf{F}(u) = \mathbf{F}(u)^{\text{Ideal}} = (f(u), g(u))$ is the *flux* vector which defines the transport of $u(\mathbf{x}, t)$. We start with the variational statement of the standard Galerkin formulation of (2) by multiplying by a test function v and integrating by parts

$$\int_\Omega \frac{\partial u}{\partial t} v\, dx + \int_{\partial\Omega} v\hat{n} \cdot \mathbf{F}(u)\, ds - \int_\Omega \nabla v \cdot \mathbf{F}(u)\, dx = 0. \tag{3}$$

The solution $u \in \mathcal{S}$ (approximation space) satisfies this equation for all $v \in \mathcal{V}$ (test space). The requirement that $\mathcal{S}$ consist of continuous functions naturally leads to a basis consisting of functions with overlapping support, which implies that Eq. (3) becomes a banded matrix equation. Solving the corresponding large system is not a trivial task for parallel implementations, and therefore a different type of formulation is desirable. Another consideration from the point of view of advection is that continuous function spaces are not the natural place to pose the problem. Mathematically, hyperbolic problems of this type tend to have solutions in spaces of bounded variation. In physical problems, the solutions are piecewise continuous, that is, are smooth in regions separated by discontinuities (shocks). An additional consideration is that the formulation presented next preserves automatically conservativity in the element-wise sense.

These considerations suggest immediately a formulation where $\mathcal{S}$ may contain discontinuous functions. The discrete space $\mathcal{S}^\delta$ contains polynomials within each "element," but zero outside the element. Here the "element" is, for example, an individual triangular region T_i in the computational mesh applied to the problem. Thus, the computational domain $\Omega = \cup_i T_i$, and T_i, T_j overlap only on edges. In the applications in Section 5 we will use hybrid discretizations, i.e., combinations of triangles and quadrilaterals, as well.

In the discontinuous Galerkin formulation, each element (E) is treated separately corresponding to a variational statement (after integrating by parts once more),

$$\frac{\partial}{\partial t}(u, v)_E + \int_{\partial T_E} v(\tilde{f}(u_i, u_e) - \boldsymbol{F}(u_i)) \cdot \mathbf{n}\, ds + (\nabla \cdot \boldsymbol{F}(u), v)_E = 0, \tag{4}$$

where $\boldsymbol{F}(u_i)$ is the flux of the interior values. Computations on each element are performed separately, and the connection between elements is a result of the way boundary conditions are applied. Boundary conditions are enforced via the numerical surface flux $\tilde{f}(u_i, u_e)$ that appears in Eq. (4). Because this value is computed at the boundary between adjacent elements, it may be computed from the value of u given at either element. These two possible values are denoted here as u_i in the interior of the element under consideration and u_e in the exterior (see Fig. 3). Upwinding considerations dictate how this flux is computed as we explain in the one-dimensional example discussed below. In the more complicated case of a hyperbolic system of equations, an approximate Riemann solver should be used to compute a value of f, g based on u_i and u_e.

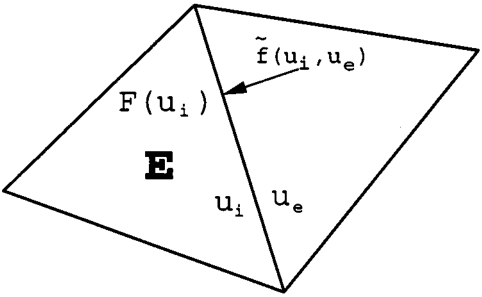

FIG. 3. Interface conditions between two adjacent triangles.

Specifically, we compute the flux $\tilde{f}(u_i, u_e)$ using upwinding, i.e.,

$$\tilde{f}(u) = R\Lambda^+ L u_i + R\Lambda^- L u_e,$$

where A (the Jacobian matrix of **F**) is written in terms of the left and right eigenvectors, i.e., $A = R\Lambda L$ with Λ containing the corresponding eigenvalues in the diagnoal; also, $\Lambda^\pm = (\Lambda \pm |\Lambda|)/2$. Alternatively, we can use a standard Roe-splitting for the flux of the form

$$\tilde{f}(u) = \frac{1}{2}(f(u_e) + f(u_i)) - \frac{1}{2} R|\Lambda| L(u_e - u_i)$$

or its appropriate extension for the MHD equations [4]. Other simplified Riemann solvers such as the one derived by Linde [28] or of kinetic-type [29] can also be employed, similar to the implementation in standard finite volume methods (B. van Leer, private communications).

3.1.1. One-dimensional example. To illustrate how the discontinuous Galerkin formulation works, we consider the one-dimensional equation for a scalar $u(x, t)$

$$\partial_t u + [f(u)]_x = 0, \qquad \text{where } x \in [x_L, x_R],$$

which we put in weak form and integrate by parts

$$(v, \partial_t u) - (v_x, f(u)) + v f(u)|_{x_L}^{x_R} = 0, \tag{5}$$

where $x \in [x_L, x_R]$ with x_L, x_R the left and right boundaries of a single element.

The treatment of the boundary terms is important as it justifies the *conservativity property*. To wit, the last term in (5) expands to

$$v_R^- f_R^- - v_L^+ f_L^-,$$

where f_R^- denotes the flux evaluated at the right boundary on the interior side, and similarly for the other terms. The above expression implies an *upwind* treatment (see the flux of the second term with f_L^- replacing f_L^+), and the test function v is evaluated inside the interval $[x_L, x_R]$. Note that f_L^- is a function of (u_L^-, u_L^+) and similarly for f_R^-. Integrating Eq. (5) by parts again we obtain

$$(\partial_t u, v) + (f_x(u), v) + v_R^- f_R^- - v_L^+ f_L^- - v_R^- f_R^- + v_L^+ f_L^+, \tag{6}$$

which reduces to the form

$$(\partial_t u, v) + (f_x(u), v) + v_L^+(f_L^+ - f_L^-). \tag{7}$$

The last term in this equation represents the so-called weak imposition of boundary conditions (through the jump term). If we use test functions which are constants along each element (in an equidistant mesh with spacing Δx), we recover the upwind (Euler backwards) finite difference formulation for the familiar *linear advection equation*, that is,

$$(\partial_t u)_j + V\frac{u_j - u_{j-1}}{\Delta x} = 0,$$

where V is the constant advection velocity.

3.2. *Discontinuous Galerkin for Diffusion*

The main idea in our version of discontinuous Galerkin formulation is similar to the one in mixed methods [30], i.e., the use of an auxiliary variable. A special version of this method was first proposed in [31] for the compressible Navier–Stokes equations. A new variant of the method without the introduction of auxiliary variables has been presented and analyzed in [32, 18].

Here, we consider as a model problem the parabolic equation with variable coefficient $v(\mathbf{x})$ to demonstrate the treatment of the viscous contributions:

$$\begin{aligned} u_t &= \nabla \cdot (\nu \nabla u) + f, && \text{in } \Omega,\ u \in L^2(\Omega)\\ u &= g(\mathbf{x}, t), && \text{on } \partial\Omega. \end{aligned}$$

We then introduce the flux variable

$$\mathbf{q} = -\nu \nabla u$$

with $\mathbf{q}(\mathbf{x}, t) \in \mathbf{L}^2(\Omega)$, and re-write the parabolic equation

$$\begin{aligned} u_t &= -\nabla \cdot \mathbf{q} + f, && \text{in } \Omega\\ 1/\nu \mathbf{q} &= -\nabla u, && \text{in } \Omega\\ u &= g(\mathbf{x}, t), && \text{on } \partial\Omega. \end{aligned}$$

The weak formulation of the problem is then as follows. Find $(\mathbf{q}, u) \in \mathbf{L}^2(\Omega) \times L^2(\Omega)$ such that

$$\begin{aligned} (u_t, w)_E &= (\mathbf{q}, \nabla w)_E - \langle w, \mathbf{q_b} \cdot \mathbf{n}\rangle_E + (f, w)_E, && \forall w \in L^2(\Omega)\\ 1/\nu(\mathbf{q}, \mathbf{v})_E &= (u, \nabla \cdot \mathbf{v})_E - \langle u_b, \mathbf{v} \cdot \mathbf{n}\rangle_E, && \forall \mathbf{v} \in \mathbf{L}^2(\Omega)\\ u &= g(\mathbf{x}, t), && \text{on } \partial\Omega, \end{aligned}$$

where the parentheses denote the standard inner product in an element (E) and the angle brackets denote boundary terms on each element, with $\mathbf{n}$ denoting the unit outwards normal. The surface terms contain weighted boundary values of v_b, q_b, which can be chosen as the arithmetic mean of values from the two sides of the boundary, i.e., $v_b = 0.5(v_i + v_e)$, and

$q_b = 0.5(q_i + q_e)$. The subscripts (i) and (e) denote contributions from the interior and exterior of element (E), respectively (see Fig. 3).

By integrating by parts once more, we obtain an equivalent formulation which is easier to implement and it is actually used in the computer code. The new variational problem is

$$\begin{aligned}
(u_t, w)_E &= (-\nabla\cdot\mathbf{q}, w)_E - \langle w, (\mathbf{q_b} - \mathbf{q_i})\cdot\mathbf{n}\rangle_E + (f, w)_E, \qquad \forall w \in L^2(\Omega)\\
1/\nu(\mathbf{q}, \mathbf{v})_E &= (-\nabla u, \mathbf{v})_E - \langle u_b - u_i, \mathbf{v}\cdot\mathbf{n}\rangle E, \qquad \forall \mathbf{v} \in \mathbf{L}^2(\Omega)\\
u &= g(\mathbf{x}, t), \qquad \text{in } \partial\Omega,
\end{aligned}$$

where the subscript (i) denotes contributions evaluated at the interior side of the boundary. The above system is currently solved explicitly but iterative solution schemes (implicit) are also under consideration.

4. THE VISCOUS MHD EQUATIONS

The equations for compressible magnetohydrodynamics (MHD) can be expressed in conservative form in compact notation as

$$\begin{aligned}
\frac{\partial\rho}{\partial t} &= -\nabla\cdot(\rho\mathbf{v})\\
\frac{\partial(\rho\mathbf{v})}{\partial t} &= -\nabla\cdot\left(\rho\mathbf{v}\mathbf{v}^t - \mathbf{B}\mathbf{B}^t + \left(p + \frac{1}{2}|\mathbf{B}|^2\right)I - \frac{1}{S_v}\tau\right)\\
\frac{\partial\mathbf{B}}{\partial t} &= -\nabla\times\left(\mathbf{B}\times\mathbf{v} + \frac{1}{S_r}\nabla\times\mathbf{B}\right)\\
\frac{\partial E}{\partial t} &= -\nabla\cdot\left((E+p)\mathbf{v} + \left(\frac{1}{2}|\mathbf{B}|^2\mathbf{I} - \mathbf{B}\mathbf{B}^t\right)\cdot\mathbf{v} - \frac{1}{S_v}\mathbf{v}\cdot\tau\right.\\
&\qquad \left. + \frac{1}{S_r}\left(\mathbf{B}\cdot\nabla\mathbf{B} - \nabla\left(\frac{1}{2}|\mathbf{B}|^2\right)\right) - \frac{1}{S_v Pr}\nabla T\right)\\
\nabla\cdot\mathbf{B} &= 0\\
\tau &= (\partial_j v_i + \partial_i v_j) - \frac{2}{3}\nabla\cdot\mathbf{v}\delta_{ij}.
\end{aligned}$$

Alternatively, in flux form with the explicitly stated fluxes they are expressed as

$$\frac{\partial\mathbf{U}}{\partial t} = -\frac{\partial\mathbf{F}_x^{\text{Ideal}}}{\partial x} - \frac{\partial\mathbf{F}_y^{\text{Ideal}}}{\partial y} - \frac{\partial\mathbf{F}_z^{\text{Ideal}}}{\partial z} + \frac{\partial\mathbf{F}_x^{\text{Visc}}}{\partial x} + \frac{\partial\mathbf{F}_y^{\text{Visc}}}{\partial y} + \frac{\partial\mathbf{F}_y^{\text{Visc}}}{\partial z}$$

$$\nabla\cdot\mathbf{B} = 0$$

$$\begin{aligned}
\mathbf{U} &= (\rho, \rho u, \rho v, \rho w, B_x, B_y, B_z, E)\\
\mathbf{F}_x^{\text{Ideal}} &= \big(\rho u, \rho u^2 - B_x^2 + \bar{p}, \rho uv - B_x B_y, \rho uw - B_x B_z, 0, uB_y - vB_x, uB_z - wB_x,\\
&\qquad (E + \bar{p})u - (\mathbf{v}\cdot\mathbf{B})B_x\big)^T\\
\mathbf{F}_y^{\text{Ideal}} &= \big(\rho v, \rho vu - B_y B_x, \rho v^2 - B_y^2 + \bar{p}, \rho vw - B_y B_z, vB_x - uB_y, 0, vB_z - wB_y,\\
&\qquad (E + \bar{p})v - (\mathbf{v}\cdot\mathbf{B})B_y\big)^T\\
\mathbf{F}_z^{\text{Ideal}} &= \big(\rho w, \rho wu - B_z B_x, \rho wu - B_z B_y, \rho w^2 - B_x^2 + \bar{p}, wB_x - uB_z, wB_y - vB_z, 0,\\
&\qquad (E + \bar{p})w - (\mathbf{v}\cdot\mathbf{B})B_z\big)^T
\end{aligned}$$

$$
\begin{aligned}
\mathbf{F}_x^{\text{Visc}} = \Bigg(&0,\ \frac{2}{S_v}\left(\frac{\partial u}{\partial x} - \frac{1}{3}\nabla\cdot\mathbf{v}\right),\ \frac{1}{S_v}\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right),\ \frac{1}{S_v}\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right),\\
&0,\ \frac{1}{S_r}\left(\frac{\partial B_y}{\partial x} - \frac{\partial B_x}{\partial y}\right),\ \frac{1}{S_r}\left(\frac{\partial B_z}{\partial x} - \frac{\partial B_x}{\partial z}\right),\\
&\frac{1}{S_v}\left(-\frac{2}{3}(\nabla\cdot\mathbf{v})u + \mathbf{v}\cdot\nabla u + \frac{1}{2}\frac{\partial\mathbf{v}^2}{\partial x} + \frac{1}{Pr}\frac{\partial T}{\partial x}\right) + \frac{1}{S_r}\left(\frac{1}{2}\frac{\partial(|\mathbf{B}|^2)}{\partial x} - \mathbf{B}\cdot\nabla B_x\right)\Bigg)^T\\
\mathbf{F}_y^{\text{Visc}} = \Bigg(&0,\ \frac{1}{S_v}\left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\right),\ \frac{2}{S_v}\left(\frac{\partial v}{\partial y} - \frac{1}{3}\nabla\cdot\mathbf{v}\right),\ \frac{1}{S_v}\left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right),\\
&\frac{1}{S_r}\left(\frac{\partial B_x}{\partial y} - \frac{\partial B_y}{\partial x}\right),\ 0,\ \frac{1}{S_r}\left(\frac{\partial B_z}{\partial y} - \frac{\partial B_y}{\partial z}\right),\\
&\frac{1}{S_v}\left(-\frac{2}{3}(\nabla\cdot\mathbf{v})v + \mathbf{v}\cdot\nabla v + \frac{1}{2}\frac{\partial\mathbf{v}^2}{\partial y} + \frac{1}{Pr}\frac{\partial T}{\partial y}\right) + \frac{1}{S_r}\left(\frac{1}{2}\frac{\partial(|\mathbf{B}|^2)}{\partial y} - \mathbf{B}\cdot\nabla B_y\right)\Bigg)^T\\
\mathbf{F}_z^{\text{Visc}} = \Bigg(&0,\ \frac{1}{S_v}\left(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial z}\right),\ \frac{1}{S_v}\left(\frac{\partial w}{\partial y} + \frac{\partial v}{\partial z}\right),\ \frac{2}{S_v}\left(\frac{\partial w}{\partial z} - \frac{1}{3}\nabla\cdot\mathbf{v}\right),\\
&\frac{1}{S_r}\left(\frac{\partial B_x}{\partial z} - \frac{\partial B_z}{\partial x}\right),\ \frac{1}{S_r}\left(\frac{\partial B_y}{\partial z} - \frac{\partial B_z}{\partial y}\right),\ 0,\\
&\frac{1}{S_v}\left(-\frac{2}{3}(\nabla\cdot\mathbf{v})w + \mathbf{v}\cdot\nabla w + \frac{1}{2}\frac{\partial\mathbf{v}^2}{\partial z} + \frac{1}{Pr}\frac{\partial T}{\partial z}\right) + \frac{1}{S_r}\left(\frac{1}{2}\frac{\partial(|\mathbf{B}|^2)}{\partial z} - \mathbf{B}\cdot\nabla B_z\right)\Bigg)^T
\end{aligned}
$$

with the variables and parameters defined in Table I. Note that here we use a different non-dimensionalization than the one used in the ideal MHD equations; the important non-dimensional parameters are the viscous and resistive Lundquist numbers.

TABLE I
Variables and Parameters Used in the Equations of Compressible MHD

Variable	Description
$\rho(\mathbf{x}, t)$	Density
$\mathbf{v}(\mathbf{x}, t) = (u, v, w)(\mathbf{x}, t)$	Velocity
$\mathbf{B}(\mathbf{x}, t) = (B_x, B_y, B_z)(\mathbf{x}, t)$	Magnetic fields
$E = \frac{p}{(\gamma-1)} + \frac{1}{2}(\rho\mathbf{v}\cdot\mathbf{v} + \mathbf{B}\cdot\mathbf{B})$	Total energy
p	Pressure
$\bar{p} = p + \frac{1}{2}\mathbf{B}\cdot\mathbf{B}$	Pressure plus magnetic pressure
$T = \frac{p}{R\rho}$	Temperature
$Pr = \frac{c_p\mu}{\kappa}$	Prandtl number
R	Ideal gas constant
η	Magnetic resistivity
μ	Viscosity
$S_v = \frac{\rho_0 V_A L_0}{\mu}$	Viscous Lundquist number
$S_r = \frac{V_A L_0}{\eta}$	Resistive Lundquist number
c_p	Specific heat at constant pressure
$V_A^2 = \frac{\mathbf{B}\cdot\mathbf{B}}{\rho}$	Alfven wave speed
$A = \sqrt{V_A^2/V_0^2}$	Alfven number

4.1. *The* $\nabla \cdot \mathbf{B} = 0$ *Constraint*

The presence of the $\nabla \cdot \mathbf{B} = 0$ constraint implies that the equations do not have a strictly hyperbolic character. It has been shown in [33] that even a small divergence in the magnetic fields can dramatically change the character of results from numerical simulations. In our formulation, we will demonstrate two approaches in dealing with this constraint. One formulation uses a streamfunction $\phi(x, y)$ for the magnetic fields (in two dimensions),

$$\mathbf{B} = \nabla \times (\phi \mathbf{k}),$$

where $\mathbf{k}$ is the unit vector perpendicular to the plane (x, y). We have used this approach in velocity-pressure-magnetic streamfunction formulation (v, p, ϕ) for incompressible MHD in [34]. There we showed that it is a good approach in two dimensions for dealing with the zero divergence constraint for the magnetic fields. We used the same basic approach here, as was used in [9]. However, we coupled the magnetic fields to the rest of the state vector in the following way: First, we updated the magnetic potential using a semi-implicit integrator, from which we calculated the magnetic field components. Then, we used these values to linearize around in the usual Riemann solver for the other components. This means we used modified left and right eigenvectors for the Riemann solver for the density, momentum, and energy flux functions in the ideal (inviscid) part, and similarily for the viscous fluxes. We will outline the full Riemann solver in the next section.

An alternative approach to the magnetic stream function was developed by Powell in [4]. The idea is to re-formulate the Jacobian matrix to include an "eighth-wave," the divergent mode that corresponds to velocity u. This way the degeneracy associated with the divergence-free condition is avoided while the rest of the eigenvalues of the Jacobian remain the same. This modification effectively corresponds to adding to the MHD equations a source term proportional to $\nabla \cdot \mathbf{B}$,

$$\mathrm{S}_P = -(\nabla \cdot \mathbf{B})(0, B_x, B_y, B_z, u, v, w, \mathbf{v} \cdot \mathbf{B})^T$$

to the right-hand-side of the evolution equation. This source term will compromise the conservation formulation of the scheme. In practice, however, this is a very small contribution, especially in the high-order discretization, and thus the conservativity condition is essentially satisfied. From another point of view, this source term effectively changes the evolution equation of the $\nabla \cdot \mathbf{B}$ from $\partial_t \nabla \cdot \mathbf{B} = 0$ to an advection equation. This implies that in certain situations, such as in stagnation-type flows, there may be some small accumulation of divergence of magnetic flux. However, in such cases a projection to a divergence-free field (using, for example, Hodge decomposition) can be employed as suggested in [33].

In [4] this extra term was incorporated into the Riemann solver for the inviscid flux terms; a similar implementation was followed in [5]. In the present work, we incorporate this term as a true *source term* without modifying the Riemann solver. The divergence of the magnetic field is calculated consistently using the discontinuous Galerkin formalism to compute the derivatives of the magnetic fields.

4.2. *Implementation of the Inviscid Flux Terms*

We evaluate the inviscid fluxes and their derivatives in the interior of the elements and add correction terms (jumps) for the discontinuities in the flux between any two adjacent elements as shown in Subsection 3.1. In order to evaluate the inviscid flux at an element interface we use a one-dimensional Riemann solver to supply a numerical flux there. At a domain boundary we use the specified conditions and treat the exterior boundary as the boundary of a "ghost" element. This way we can use the same Riemann solver at all element boundaries.

We linearize the one-dimensional flux $\mathbf{F}_n^{\text{Ideal}}$ in the normal direction to a shared element boundary using the average of the state vector at either side of the element boundary. That is, since $\mathbf{F}_n^{\text{Ideal}}$ is a nonlinear function of the state vector we use the average state to form an approximation to the Jacobian of the flux vector A_c. The Jacobian matrix for the flux vector for the evolution equations expressed in primitive variables is simpler than the conserved form. Thus, we perform the linearization for the primitive form and transform to the conserved form. The primitive Jacobian matrix A_p has the form

$$\begin{bmatrix} u & \rho & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & u & 0 & 0 & -\frac{B_x}{\rho} & \frac{B_y}{\rho} & \frac{B_z}{\rho} & \frac{1}{\rho} \\ 0 & 0 & u & 0 & -\frac{B_y}{\rho} & \frac{B_x}{\rho} & 0 & \frac{1}{\rho} \\ 0 & 0 & 0 & u & -\frac{B_z}{\rho} & 0 & \frac{B_x}{\rho} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & B_y & -B_x & 0 & -v & u & 0 & 0 \\ 0 & B_z & 0 & -B_x & -w & 0 & u & 0 \\ 0 & \gamma p & 0 & 0 & -(\gamma-1)\mathbf{u}\cdot\mathbf{B} & 0 & 0 & u \end{bmatrix}.$$

The scaled left and right eigenvectors of the primitive Jacobian matrix A_p, due to Powell [4], are

entropy wave,

$$\begin{aligned} \lambda_e &= u \\ l_e &= \left(1, 0, 0, 0, 0, 0, 0, -\frac{1}{a^2}\right) \\ r_e &= (1, 0, 0, 0, 0, 0, 0, 0)^t; \end{aligned}$$

Alfven waves,

$$\begin{aligned} \lambda_a &= u \pm \frac{B_x}{\sqrt{\rho}} \\ l_a &= \frac{1}{\sqrt{2}}\left(0, 0, -\beta_z, \beta_y, 0, \pm\frac{\beta_z}{\sqrt{\rho}}, \mp\frac{\beta_y}{\sqrt{\rho}}, 0\right) \\ r_a &= \frac{1}{\sqrt{2}}\left(0, 0, -\beta_z, \beta_y, 0, \pm\beta_z\sqrt{\rho}, \mp\beta_y\sqrt{\rho}, 0\right)^T; \end{aligned}$$

fast waves,

$$\begin{aligned} \lambda_f &= u \pm c_f \\ l_f &= \frac{1}{2a^2}\left(0, \pm\alpha_f c_f, \mp\alpha_s c_s \beta_x \beta_y, \mp\alpha_s c_z \beta_x \beta_z, 0, \frac{\alpha_s \beta_y a}{\sqrt{\rho}}, \frac{\alpha_s \beta_z a}{\sqrt{\rho}}, \frac{\alpha_f}{\rho}\right) \\ r_f &= (\rho\alpha_f, \pm\alpha_f c_f, \mp\alpha_s c_s \beta_x \beta_y, \mp\alpha_s c_s \beta_x \beta_z, 0, \alpha_s \beta_y a\sqrt{\rho}, \alpha_s \beta_z a\sqrt{\rho}, \alpha_f \gamma p)^t; \end{aligned}$$

slow waves,

$$\lambda_s = u \pm c_s$$
$$l_s = \frac{1}{2a^2}\left(0, \pm\alpha_s c_s, \pm\alpha_f c_f \beta_x \beta_y, \pm\alpha_f c_f \beta_x \beta_z, 0, -\frac{\alpha_f \beta_y a}{\sqrt{\rho}}, -\frac{\alpha_f \beta_z a}{\sqrt{\rho}}, \frac{\alpha_s}{\rho}\right)$$
$$r_s = (\rho\alpha_s, \pm\alpha_s c_s, \pm\alpha_f c_f \beta_x \beta_y, \pm\alpha_f c_f \beta_x \beta_z, 0, -\alpha_f \beta_y a\sqrt{\rho}, -\alpha_f \beta_z a\sqrt{\rho}, \alpha_s \gamma p)^t,$$

where

$$(a^*)^2 = \frac{\gamma p + \mathbf{B}\cdot\mathbf{B}}{\rho}$$
$$c_f^2 = \frac{1}{2}\left((a^*)^2 + \sqrt{(a^*)^4 - 4\frac{\gamma p B_x^2}{\rho^2}}\right)$$
$$c_s^2 = \frac{1}{2}\left((a^*)^2 - \sqrt{(a^*)^4 - 4\frac{\gamma p B_x^2}{\rho^2}}\right)$$
$$\alpha_f^2 = \frac{a^2 - c_s^2}{c_f^2 - c_s^2}$$
$$\alpha_s^2 = \frac{c_f^2 - a^2}{c_f^2 - c_s^2}$$
$$\beta_x = \operatorname{sgn}(B_x)$$
$$\beta_y = \frac{B_y}{\sqrt{B_y^2 + B_z^2}}$$
$$\beta_z = \frac{B_z}{\sqrt{B_y^2 + B_z^2}}.$$

We can transform between the primitive and conserved variables with the transform

$$A_c = \frac{\partial \mathbf{U}}{\partial \mathbf{W}} A_p \frac{\partial \mathbf{W}}{\partial \mathbf{U}},$$

where

$$\mathbf{U} = (\rho, \rho u, \rho v, \rho w, B_x, B_y, B_z, E)$$

are the conserved variables, and

$$\mathbf{W} = (\rho, u, v, w, B_x, B_y, B_z, p)$$

are the primitive variables. This gives

$$\frac{\partial \mathbf{U}}{\partial \mathbf{W}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ u & \rho & 0 & 0 & 0 & 0 & 0 & 0 \\ v & 0 & \rho & 0 & 0 & 0 & 0 & 0 \\ w & 0 & 0 & \rho & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \frac{\mathbf{v}\cdot\mathbf{v}}{2} & \rho u & \rho v & \rho w & B_x & B_y & B_z & \frac{1}{\gamma-1} \end{bmatrix}$$

and

$$\frac{\partial \mathbf{W}}{\partial \mathbf{U}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{u}{\rho} & \frac{1}{\rho} & 0 & 0 & 0 & 0 & 0 & 0 \\ -\frac{v}{\rho} & 0 & \frac{1}{\rho} & 0 & 0 & 0 & 0 & 0 \\ -\frac{w}{\rho} & 0 & 0 & \frac{1}{\rho} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ \frac{\bar{\gamma}}{2}\mathbf{v}\cdot\mathbf{v} & -\bar{\gamma}u & -\bar{\gamma}v & -\bar{\gamma}w & -\bar{\gamma}B_x & -\bar{\gamma}B_y & -\bar{\gamma}B_z & \bar{\gamma} \end{bmatrix},$$

where $\bar{\gamma} = \gamma - 1$.

We are now in a position to evaluate the numerical flux at the element boundaries. We use the following formulation for the upwinded flux:

$$\hat{\mathbf{F}}(\mathbf{U}_I, \mathbf{U}_E) = \frac{1}{2}\left(\mathbf{F}(\mathbf{U}_I) + \mathbf{F}(\mathbf{U}_E) - \frac{\partial \mathbf{U}}{\partial \mathbf{W}} \sum_{k=1}^{k=7} \alpha_k |\lambda_k| \mathbf{r_k}\right) \tag{8}$$

$$\alpha_k = \mathbf{l_k} \cdot \frac{\partial \mathbf{W}}{\partial \mathbf{U}} (\mathbf{U}_E - \mathbf{U}_I). \tag{9}$$

Here the $\mathbf{l_k}$ and $\mathbf{r_k}$ are the ordered left and right eigenvectors of the primitive Jacobian matrix. We have to apply the $\frac{\partial \mathbf{U}}{\partial \mathbf{W}}$ operator to the right eigenvectors to calculate the conserved flux. The λ_k are the wave speeds associated with the eigenvectors.

4.3. *Implementation of the Viscous Terms*

The viscous terms are evaluated in two steps. First, we obtain the spatial derivatives of the primitive variables using the discontinuous Galerkin approach. Then, we repeat the process for each of the viscous fluxes using these derivatives. Dirichlet boundary conditions for the momentum and energy characteristic variables can be imposed weakly as discussed in Subsection 3.2 or explicitly after the fluxes have been evaluated and then project the result using the orthogonal basis.

4.4. *Summary of the Algorithm*

The main idea in the implementation of this algorithm is to consider a set of quadrature points at the interface Q^I and a set of quadrature points at the edges Q^E (see right plot of Fig. 1). The set Q^I, on which the approximate Riemann solver is applied, corresponds to Gauss quadrature, i.e., it does not include the end points, thus avoiding complications associated with multiplicities of vertices. On the other hand, the set Q^E depends on the type of element that is used. Details on this as well as on computing the flux integrals involved in the discontinuous Galerkin formulation are presented in Appendix II. In the following we present the main steps of the proposed algorithm:

- Step 1. Read in initial conditions $\mathbf{U}(\mathbf{x}, 0)$ and evaluate all fields at all element quadrature points. Set $n = 0$.
- Step 2. Calculate the fluxes $\hat{\mathbf{F}}_\mathbf{n}$ from Eqs. (8) and (9) at the Gauss quadrature points Q^I on the element interfaces. At domain boundaries use the prescribed boundary conditions

for the exterior values of the fields. Interpolate the fluxes $\hat{\mathbf{F}}_{\mathbf{n}}$ to the quadrature points Q_E. Scale the fluxes with the edge Jacobian divided by the volume Jacobians.

- Step 3. Calculate the inviscid flux terms $\mathbf{F}_x^{\text{Ideal}}$, $\mathbf{F}_y^{\text{Ideal}}$, and $\mathbf{F}_z^{\text{Ideal}}$ at the element quadrature points.
- Step 4. For each component of the state vector $U_k = (\mathbf{U}(\mathbf{x}, t^n))_k$ calculate $(\partial \mathbf{F}_x^{\text{Ideal}}/\partial x + \partial \mathbf{F}_y^{\text{Ideal}}/\partial y + \partial \mathbf{F}_z^{\text{Ideal}}/\partial z)_k$.
- Step 5. Form the difference $(\hat{\mathbf{F}}_{\mathbf{n}} - \mathbf{F}_I)$ (where $\mathbf{F}_I$ is the flux at interior edge side) and add this to the divergence of the inviscid fluxes calcualted in Step 4.
- Step 6. Calculate the spatial derivatives of the primitive fields. For example, we will need $\frac{\partial u}{\partial x}$. We evaluate this as described in Subsection 3.2 using the discontinuous Galerkin formalism to compensate for the jumps across element interfaces.
- Step 7. Use the derivatives of the primitive fields to construct the viscous flux terms $\mathbf{F}_x^{\text{Visc}}$, $\mathbf{F}_y^{\text{Visc}}$, and $\mathbf{F}_z^{\text{Visc}}$.
- Step 8. Take the divergence of the viscous flux terms and subtract the results of Step 5.
- Step 9. Take the inner-product of the result from Step 8 with the orthogonal basis. Evaluate the resulting polynomials at the quadrature points and place it in $\mathbf{Uf}(\mathbf{x}, t^{n-q})$.
- Step 10. Update the state vector $\mathbf{U}(\mathbf{x}, t^{n+1}) = \mathbf{U}(\mathbf{x}, t^n) + \Delta t \sum_q \beta_q \mathbf{Uf}(\mathbf{x}, t^{n-q})$ using an Adams–Bashforth integration scheme.
- Step 11. Increase n by one. If t^n is less than the termination time return to Step 2.
- Step 12. Output final values of the state vector $\mathbf{U}(\mathbf{x}, t^{\text{end}})$.

5. CONVERGENCE AND SIMULATION EXAMPLES

In the following we first test the accuracy of the proposed method using analytical solutions and verify its exponential convergence. We then present simulations of more complex flows that demonstrate how accurate solutions can be obtained and verified without the need for re-meshing but by simply increasing the p-order N, i.e., the number of modes per element.

5.1. *Two-Dimensional Magnetohydrostatic Test Case*

A simple test for the two-dimensional MHD code is to consider a steady irrotational magnetic field and zero velocity. The test was performed as an *initial value problem* and the following exact solution was used:

$$\begin{aligned} \rho &= 1 \\ u &= 0 \\ v &= 0 \\ E &= 19.84 + \frac{e^{(-2\pi y)}}{2} \\ B_x &= -\cos(\pi x) e^{(-\pi y)} \\ B_y &= \sin(\pi x) e^{(-\pi y)}. \end{aligned}$$

The above solution was first derived by Priest [35]. We used Dirichlet boundary conditions by specifying the exact solution on all boundaries. We also used the exact solution as initial conditions. Therefore, after convergence to steady state, the discrete solution will

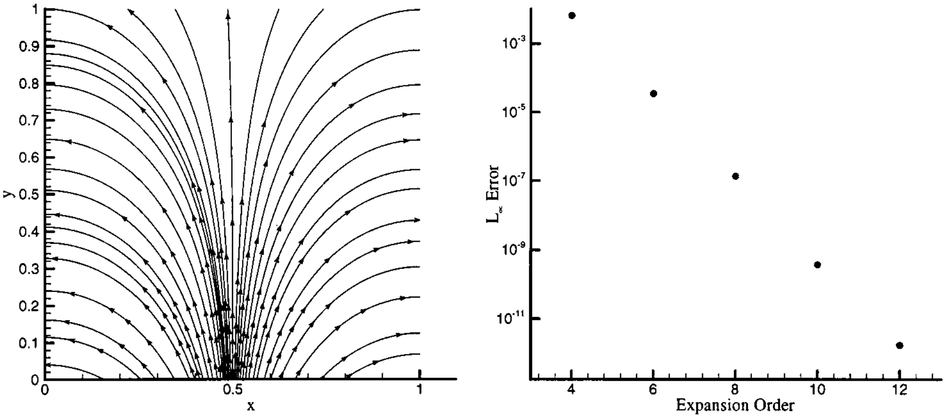

FIG. 4. Magnetohydrostatic test case for the two-dimensional code. Left, magnetic streamlines of steady solution at p-order $N = 12$; right, maximum pointwise error versus p-order for a fixed number of elements.

deviate from the initial condition and that deviation will give us a measure of the spatial discretization error. In steady state there are no temporal errors unless there is operator splitting involved in the formulation, which is not the case here. The irrotational magnetic field implies that the Lorentz force is zero so the momentum equations are trivially satisfied. The magnetoviscous term is zero and the $\mathbf{v} \times \mathbf{B}$ term is also zero. Thus, the compressible MHD equations are satisfied. The domain and hybrid discretization we used consisted of triangular and quadrilateral elements and is depicted in Fig. 1. In Fig. 4 we also show that the numerical error decreases exponentially fast to zero with increasing p-order while keeping the number of elements fixed. This is indicated by the straight line in the convergence plot (linear-logarithmic) axes of Fig. 4.

In summary, in this example we have demonstrated the exponential ("spectral") convergence of the method in agreement with the analysis in Appendix I that presents the decay rate of the expansion coefficients.

5.2. *Three-Dimensional Magnetohydrostatic Test Case*

We modified the two-dimensional test case, used in the previous section, to be a three-dimensional test case for the three-dimensional MHD code. Again we used a steady irrotational magnetic field and zero velocity solution to the MHD equations. The test was performed as an initial value problem and the following exact solution

$$\begin{aligned}
\rho &= 1\\
u &= 0\\
v &= 0\\
w &= 0\\
B_x &= (\cos(\pi(y+1)) - \cos(\pi z))e^{-\pi(x+1)}\\
B_y &= \cos(\pi z)e^{-\pi(y+1)} + \sin(\pi(y+1))e^{-\pi(x+1)}\\
B_z &= \sin(\pi z)\left(e^{-\pi(y+1)} - e^{-\pi(x+1)}\right)\\
E &= 5 + 0.5\left(B_x^2 + B_y^2 + B_z^2\right)
\end{aligned}$$

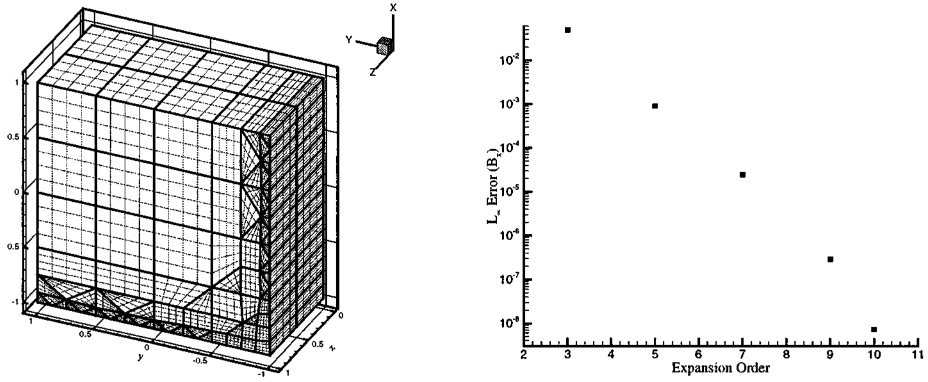

FIG. 5. Three-dimensional magnetohydrostatic test case. Left, mesh of prisms and hexahedra used. Right, convergence plot showing exponential decrease in maximum pointwise error with increasing *p*-order.

was used as the boundary conditions and as the initial condition. By construction, the magnetic field is irrotational and the Lorentz force is zero so the momentum equations are trivially satisfied. The magnetoviscous term is zero and the $\mathbf{v} \times \mathbf{B}$ term is also zero. The domain and discretization, which consisted of a mix of prisms and hexahedra, are depicted in Fig. 5 (left plot). Although such a hybrid discretization consisted of heterogeneous elements is not needed here for this simple computational domain, this example demonstrates the flexibility of the method in discretizing complex geometry domains using different types of elements. We also plot on the right the numerical (maximum pointwise) error showing that it decreases exponentially fast to zero with increasing expansion order (*p*-order) while keeping the number of elements fixed. This numerical result is again in agreement with the theoretical result of the exponential decay of the expansion coefficients as discussed in Appendix I.

5.3. *Simulation of the Orszag–Tang Vortex*

We have performed a series of detailed simulations in order to investigate the small-scale structure exhibited in MHD turbulence. In particular, we consider a problem first studied by Orszag and Tang (1979) [20] in the compressible case and later extended by Dahlburg and Picone (1989) [9] to the compressible case. The initial conditions are non-random, periodic fields with the velocity field being solenoidal. The total initial pressure consists of the superposition of appropriate incompressible pressure distribution upon a flat pressure field corresponding to an initial average. Mach number below unity. It was found in [20, 9] that the coupling of the two-dimensional flow with the magnetic field causes the formation of singularities, i.e., excited small-scale structure, which although not as strong as the singularities in three-dimensional turbulence, they are certainly much stronger than two-dimensional hydrodynamic turbulence. Moreover, it was found in [9] that compressibility causes formation of additional small-scale structures such as massive jets and bifurcation of eddies. Our interest here is to investigate if we can capture these fine features both on structured and unstructured meshes, as shown in Fig. 6.

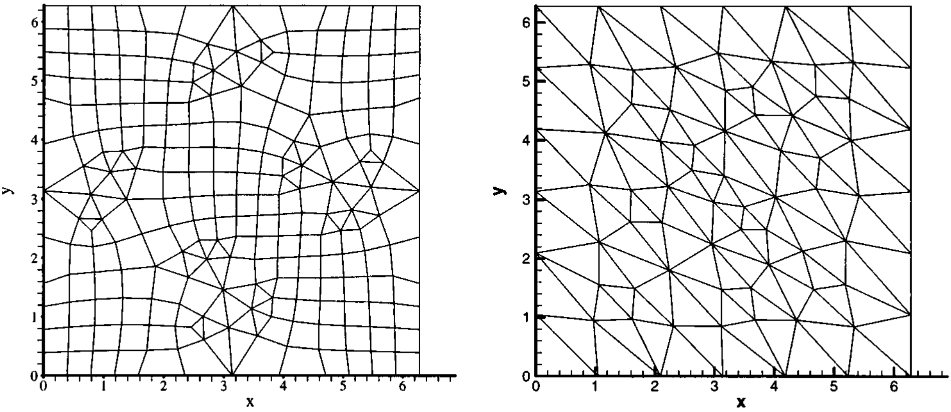

FIG. 6. Hybrid mesh on the left and unstructured mesh on the right used for the Orszag–Tang vortex simulations.

The initial conditions we used were

$$\begin{aligned}
\rho &= 1\\
u &= -\sin\left(\frac{2\pi y}{L}\right)\\
v &= \sin\left(\frac{2\pi x}{L}\right)\\
B_x &= -\sin\left(\frac{2\pi y}{L}\right)\\
B_y &= \sin\left(\frac{4\pi x}{L}\right)\\
p &= C + \frac{1}{4}\cos\left(\frac{8\pi x}{L}\right) + \frac{4}{5}\cos\left(\frac{4\pi x}{L}\right)\cos\left(\frac{2\pi y}{L}\right)\\
&\quad - \cos\left(\frac{2\pi x}{L}\right)\cos\left(\frac{2\pi y}{L}\right) + \frac{1}{4}\cos\left(\frac{4\pi y}{L}\right),
\end{aligned}$$

where C fixes the initial average Mach number and p is the instantaneous pressure for the equivalent incompressible flow.

We first simulate this MHD flow on a hybrid grid consisting of quadrilaterals and triangles as shown in Fig. 6. We perform the simulations using the formulation of Powell [4] for the magnetic field as well as the streamfunction formulation with the objective of investigating divergence errors in the magnetic field. The rest of the parameters of this simulation are listed in Table II. In Fig. 7 we plot streamlines of the incompressible flow as well as the compressible flow at Mach number 0.4 and non-dimensional time $t = 2.0$. These results agree very well with the simulations of [9] at the same set of parameters. We note here that the compressible flow exhibits structures of finer features compared to the incompressible flow but the differences in the magnetic field are less obvious. Next, we examine errors in $\nabla \cdot \mathbf{B}$ by comparing the two implementations corresponding to the eighth-wave formulation and the streamfunction formulation for compressible flow and identical conditions as above.

TABLE II
Simulation Parameters for the Compressible Orszag–Tang Vortex Problem (Hybrid Mesh)

Parameter	Value
Dimension	2D
S_v	100
S_r	100
A (Alfven number)	1.0
Mach	0.4
N	12
K_{Quad}	176
K_{Tri}	64
Method	Discontinuous Galerkin

In Fig. 8 we plot contours of the divergence of the magnetic field. We see that it is mostly zero except for certain regions associated with very large gradients in the magnetic field, especially for the eighth-wave formulation. To examine these regions we also plot in Fig. 8 the $\nabla \times \mathbf{B}$ (which is proportional to current density) and we see that very steep **B** gradients are indeed induced, especially in the middle of the domain (where divergence errors of

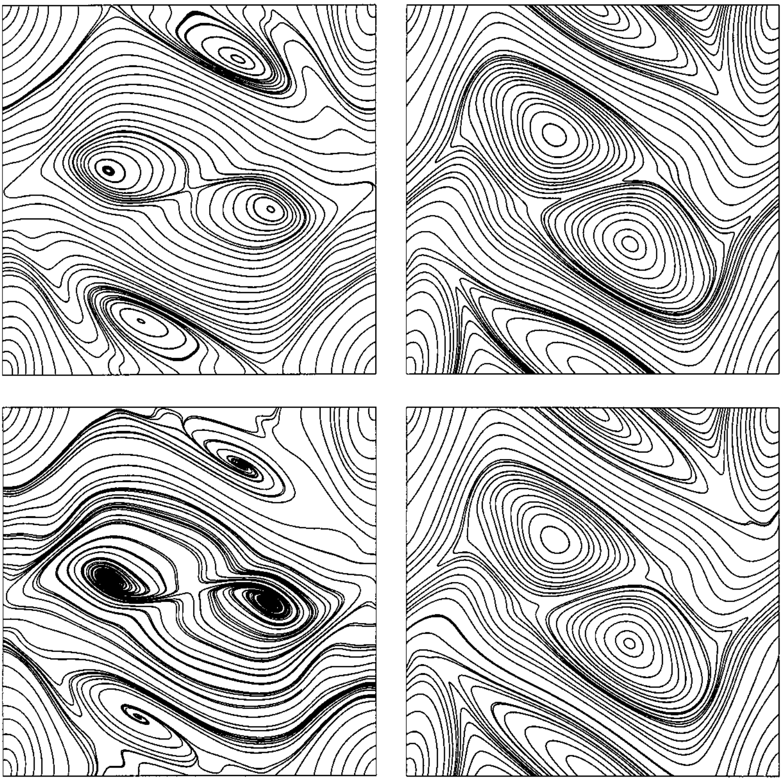

FIG. 7. Compressible Orszag–Tang vortex ($t = 2$, instantaneous fields, Mach $= 0.4$). Top, incompressible flow; left, flow streamlines; right, magnetic streamlines. Bottom, compressible flow; left, flow streamlines; right, magnetic streamlines.

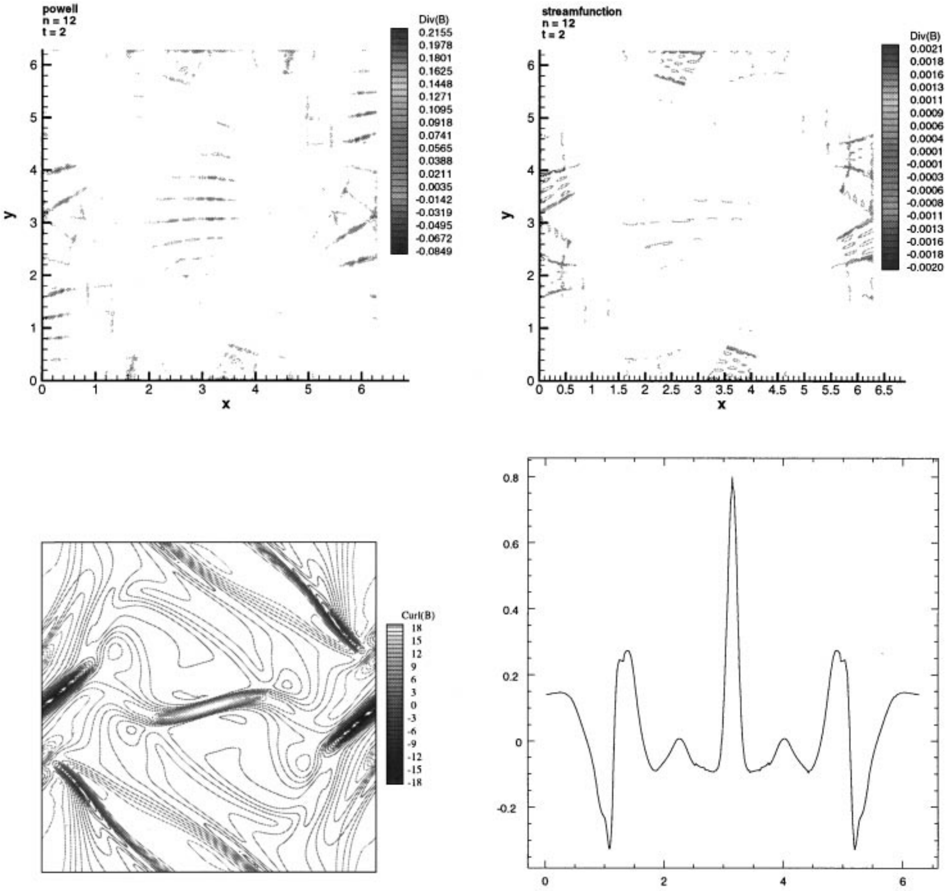

FIG. 8. Compressible Orszag–Tang vortex ($t = 2$, instantaneous fields, Mach $= 0.4$). Top, isocontours of $\nabla \cdot \mathbf{B}$ corresponding to the eighth-wave formulation (left) and streamfunction formulation (right). Bottom, isocontours of $\nabla \times \mathbf{B}$ corresponding to the eighth-wave formulation (left) and horizontal profile across the middle of the domain (right).

order $\mathcal{O}(1)$ are present), as shown in the profile taken across the domain in the same figure (bottom right). The errors in divergence seen in the streamfunction formulation are obviously spatial discretization errors and are approximately two orders of magnitude lower than the divergence errors of the eighth-wave formulation. Despite these differences in divergence errors between the two formulations, the resulting velocity and magnetic fields are identical within discretization error.

We now consider the effect of p-refinement on accuracy using the unstructured mesh of Fig. 6 (right). Vorticity is a good indicator of "noise" in the solution of unsteady simulations as it reflects errors in derivatives. More specifically, we examine the *curl of momentum* as low resolution simulations result in non-smoothness in this vorticity-like quantity. In Fig. 9 we compare the vorticity at time $t = 1$ for the unstructured mesh shown in Fig. 6 (right) run with $N = 4$ and $N = 16$. The top figures show how the vorticity profile varies across the lower-left to top-right diagonal. The vorticity in this direction should be symmetric about the mid-point. We see that at $N = 4$ the profile is noisy, the peeks are not well resolved, and the symmetry is not very well represented. The results are improved very quickly as we increase the polynomial order N. Here we present the final results at $N = 16$ and we

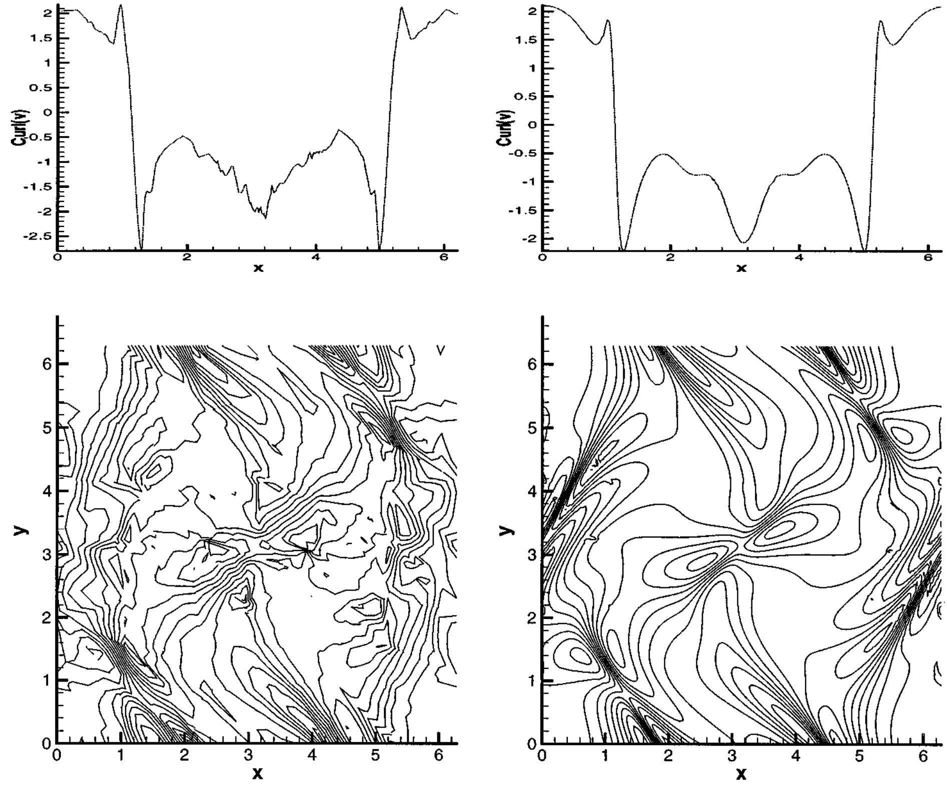

FIG. 9. Simulation of the compressible Orszag–Tang vortex ($t = 1$, instantaneous fields, Mach $= 0.2$, $K = 132$) on the unstructured mesh. Top left, curl of momentum along the diagonal ($N = 4$). Bottom left, iso-contours of curl of momentum ($N = 4$). Top right, curl of momentum along the diagonal ($N = 16$). Bottom right, iso-contours of curl of momentum ($N = 16$).

see that symmetry is restored and the profile is very smooth. Results for the intermediate values of the p-order N can be found in [34].

5.4. *MHD Flow Past a Cylinder*

In this test we consider uniform flow past a circular cylinder. We perform simulations using an unstructured mesh shown in Fig. 10 consisting of $K = 490$ triangular elements. A

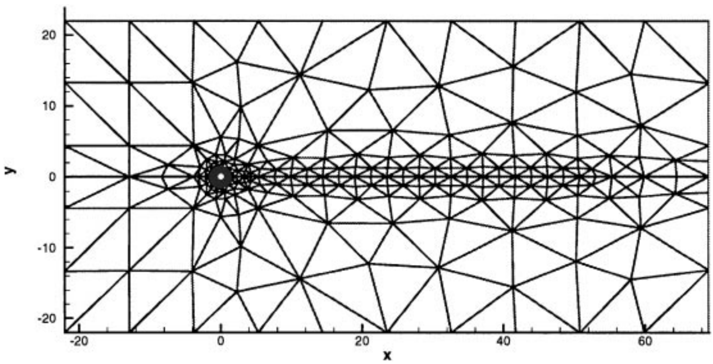

FIG. 10. Unstructured mesh used in simulations for flow past a cylinder.

TABLE III
Simulation Parameters for Compressible Flow Past a Cylinder with a Streamwise Magnetic

Parameter	Value
Dimension	2D
S_v	100
S_r	100
A	0.1
Mach	0.5
Δt	1e-4
N-range	1 to 8
K_{Tri}	490
Method	Discontinuous Galerkin

uniform magnetic field (streamwise direction) is imposed at the inflow and zero values are prescibed at the cylinder curface. We have simulated both compressible and incompressible flow with and without the presence of a magnetic field. The relevant list of parameters is shown in Table III. Note that external fields can either suppress or enhance the vortex street as has been found in experimental and numerical work for incompressible flows [36–38].

In Fig. 11 we compare the two components of the magnetic field for incompressible and compressible flow ($M = 0.5$). We see that the fields look similar although finer scale features are present in the compressible case. Next, we compare the velocity fields of a compressible flow with and without the presence of the uniform magnetic field at the inflow; all other parameters are similar. The results are plotted in Fig. 12. We see that up-down symmetry of the x-component of velocity has been broken in the wake and the regular pattern of the y-component of velocity also breaks down ten diameters from the cylinder. This is an indication that the specified magnetic fields is causing the von Karman street to become unstable.

6. SUMMARY

We have developed a new method for solving the compressible viscous MHD equations on standard unstructured and hybrid meshes in two and three dimensions and presented several convergence tests and flow simulations for validation. The new method is based on a discontinuous Galerkin treatment of the advective and diffusive components. This, in turn, allows the use of orthogonal tensor-product spectral bases in these non-orthogonal and polymorphic subdomains, which results in high computational efficiency. In particular, the computational cost is KN^{d+1} (where $d = 2$ or 3 in 2D and 3D, respectively) with K the number of elements and N the polynomial order in an element. This cost corresponds to differentiation and integration cost on the entire domain and is similar to the cost of such operations in standard global methods in simple separable domains [39]. The method is essentially *matrix-free* as the only matrix inversion required is that of a *local* mass matrix, which is diagonal, and thus trivial to invert.

Such a discretization based here on discontinuous Galerkin projections is similar to the finite volume formulation, and in particular the implementations reported in [4, 5]. In

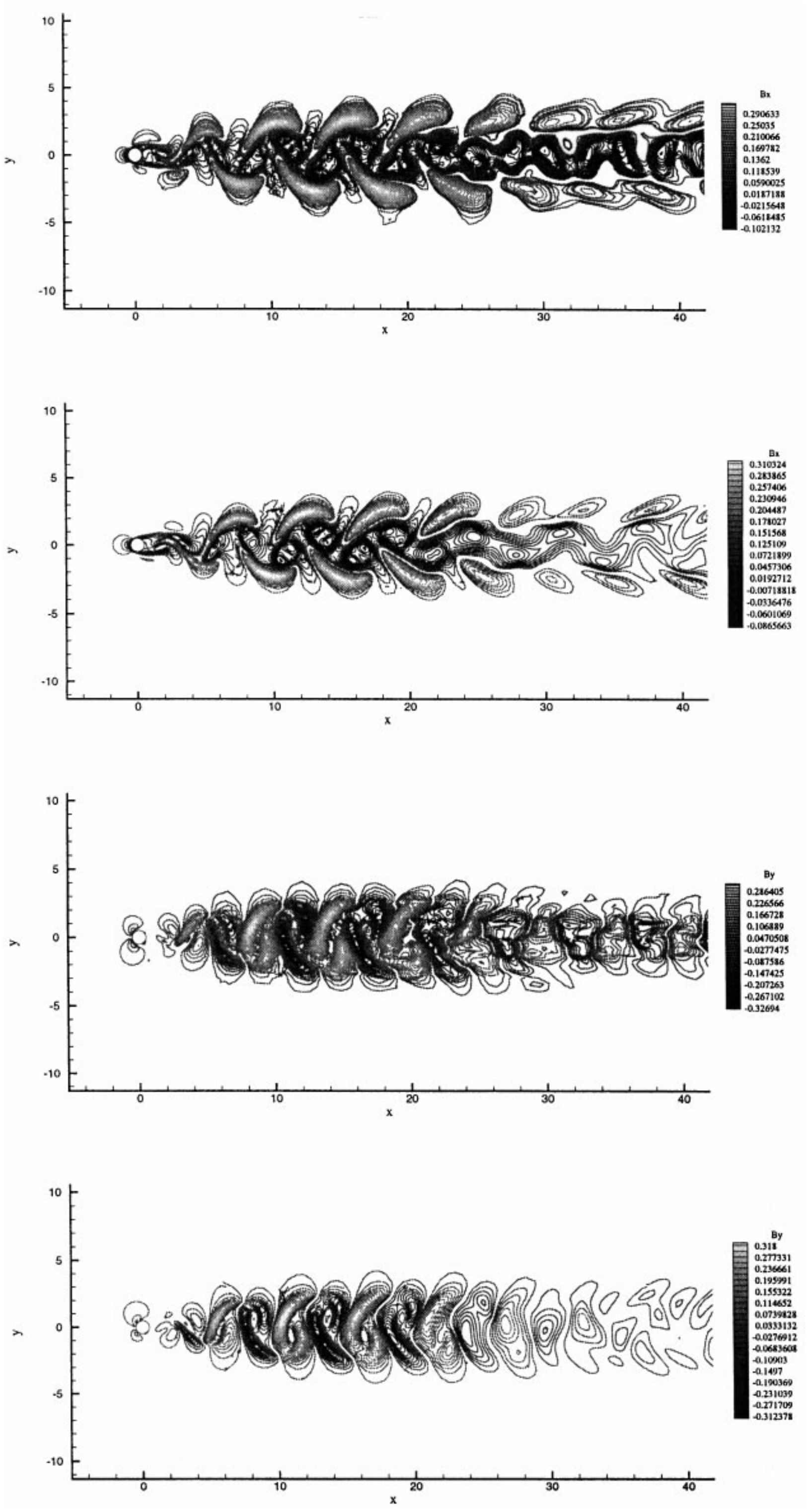

FIG. 11. Instantaneous iso-contours of the magnetic field for compressible (top and third from top) and incompressible (second and fourth from top) flow.

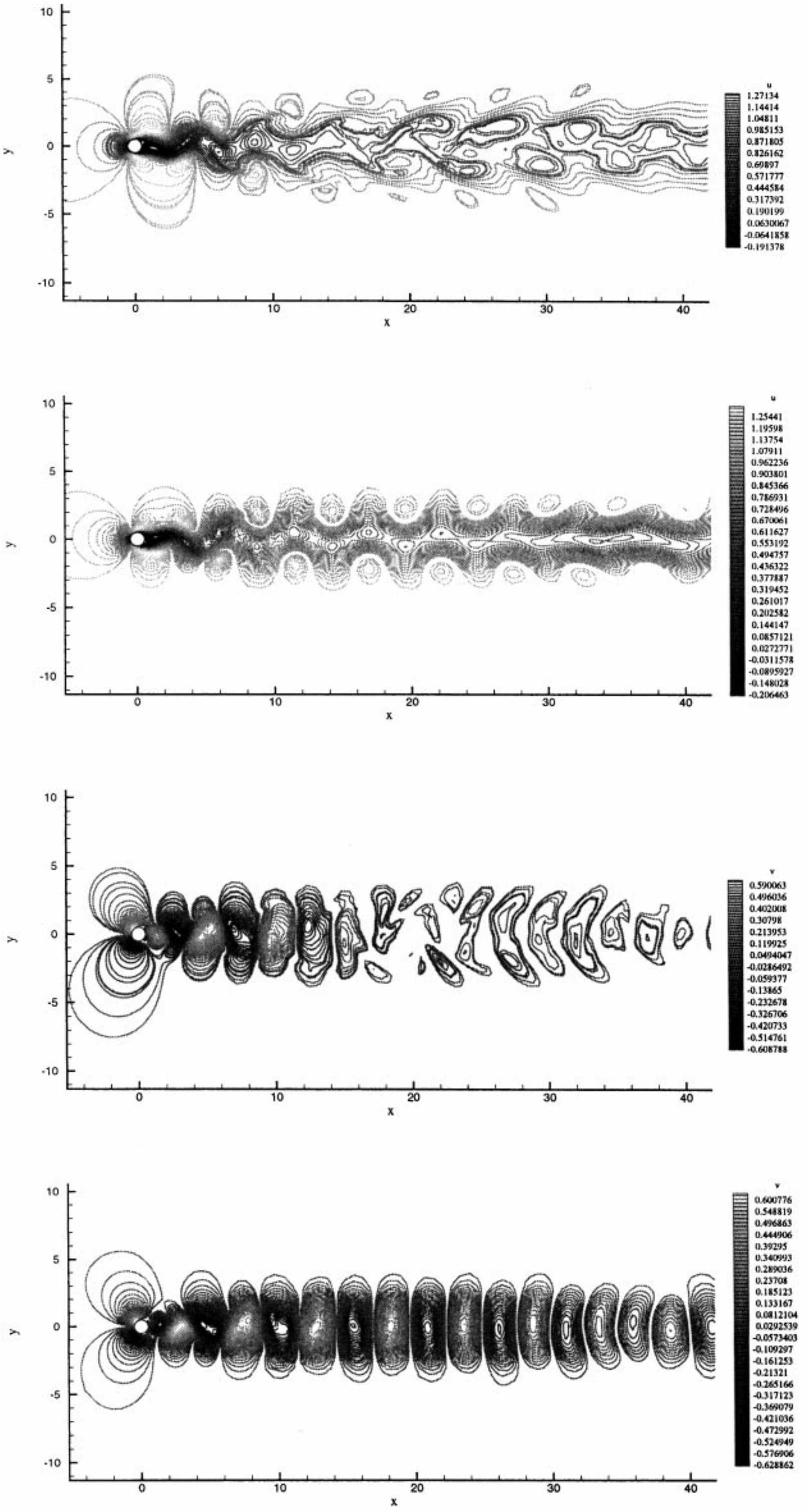

FIG. 12. Instantaneous iso-contours of the velocity field for compressible MHD flow (top and third from top) and compressible flow without magnetic fields (second and fourth from top).

particular, the degenerate form of the proposed *discontinuous* Galerkin spectral/*hp* element method for constant polynomial order ($N = 0$) reduces to the standard finite volume method. In contrast, degenerate forms of the *continuous* spectral/*hp* element Galerkin method obtained by setting $N = 1$ reduce to the standard finite element method [40].

The flow examples presented here were all subsonic cases. However, the method can also handle supersonic cases by handling strong discontinuities with adaptive h-refinement and by lowering the polynomial order locally. Although this adaptive procedure has not yet been implemented for the compressible MHD code, it has been shown to be very successful in simulation of supersonic flows without magnetic fields in [15, 16]; see also [17]. This approach avoids filtering, limiters, or non-oscillatory reconstruction algorithms, which add substantially to computational cost and are not generally robust for most aerodynamic applications (see, for example, [41, 12]).

The method presented here borrows from features of finite volumes (flux-based), finite-elements (variational statement), and spectral methods (high-order basis), and is both robust and flexible as it is conservative. It does not rely on flux-limiters, and it works on standard unstructured and hybrid meshes. Although the examples presented are for relatively low Reynolds number, numerical experience has shown that discontinuous Galerkin methods combined with high-order spatial discretization are suitable in simulating high Reynolds number turbulent flows without the need for *ad hoc* turbulence modeling. In particular, a very useful feature of the method is the capability of using *variable p*-order per element thus distributing resolution resources efficiently in resolving the multiple boundary layers encountered in wall-bounded MHD flows.

APPENDIX I

Spectral Bases in Polymorphic Domains

In this appendix we will show that it is possible to construct tensor-product spectral bases for polymorphic non-orthogonal domains. Specifically, we will show that it is possible to find a generalized singular Sturm–Liouville operator for a d-dimensional simplex and present its eigenfunction/eigenvalue pairs explicitly. The eigenfunctions for the cases $d = 1$ (a segment), $d = 2$ (a triangle), and $d = 3$ (a tetrahedron) will prove to be useful in the context of polynomial-based approximation on these simplices. Similar results have also been obtained for some of the domains we consider here in [42, 43] following different derivations.

A d-dimensional simplex S^d can be defined as a set of constraints on the entries of a d-dimensional vector,

$$S^d = \left\{ \mathbf{r} \in \mathbb{R}^d \mid 0 \leq \sum_{i=1}^{i=j} r_i \leq 1;\ \ j = 1, 2, \ldots, d \right\}.$$

We define an operator on a d-dimensional space of at least twice differentiable functions of d variables,

$$L^d_{\mathbf{r}} = \sum_{i=1}^{i=d} \partial_{r_i} \left(r_i \partial_{r_i} - r_i \sum_{j=1}^{j=d} r_j \partial_{r_j} \right).$$

We also define a $(d+1)$ vector, $\mathrm{s} = [(1-\omega)r_1, (1-\omega)r_2, \ldots, (1-\omega)r_d, \omega]$, where $\omega \in [-1, 1]$. Using the identities

$$\begin{aligned} \partial s_i &= \frac{1}{1-\omega}\partial_{r_i}, \qquad i = 1, \ldots, d \\ \partial s_{d+1} &= \frac{1}{1-\omega}\sum_{i=1}^{i=d} r_i \partial_{r_i} + \partial_\omega \\ s_i \partial s_i &= r_i \partial_{r_i}, \qquad i = 1, \ldots, d \\ \sum_{i=1}^{i=d+1} s_i \partial s_i &= \frac{1}{1-\omega}\sum_{i=1}^{i=d} r_i \partial_{r_i} + \omega\partial_\omega \end{aligned}$$

it is straightforward to show the relationship

$$L_s^{d+1} = \frac{1}{1-\omega}L_{\mathbf{r}}^d + \frac{1}{(1-\omega)^d}\partial_\omega(\omega(1-\omega)^{d+1}).$$

From this we can repeat the recurrence relation ending up with the operator in terms of the **a** coordinates,

$$L_{\mathbf{r}}^d = \sum_{i=1}^{i=d} \frac{1}{\prod_{j=i+1}^{j=d}(1-a_j)}[\partial_{a_i}(a_i(1-a_i)\partial_{a_i}) - (i-1)a_i\partial_{a_i}],$$

where **a** is defined by the canonical transform

$$\begin{aligned} r_1 &= a_1(1-a_2)(1-a_3)\cdots(1-a_d) \\ r_2 &= a_2(1-a_3)\cdots(1-a_d) \\ \vdots \;&\vdots\; \vdots \\ r_d &= a_d. \end{aligned}$$

This form of the operator shows that $L_{\mathbf{r}}^d$ is self-adjoint in the inner product taken over the simplex S^d. This is because S^d maps to a d-dimensional unit box $U^d = \{\mathbf{r} \in \mathbb{R}^d \mid 0 \le r_i \le 1 \; i = 1, 2, \ldots, d\}$ and using integration by parts we notice that all the surface integral terms are zero.

For example, we consider the ith operator in the sum for $L_{\mathbf{r}}^d$,

$$\begin{aligned} \left(u, \left(L_{\mathbf{r}}^d\right)^i v\right)_{S^d} &= \left(u, \frac{\prod_{i=2}^{k=d}(1-a_k)^{k-1}}{\prod_{j=i+1}^{j=d}(1-a_j)}[\partial_{a_i}(a_i(1-a_i)\partial_{a_i}) - (i-1)a_i\partial_{a_i}]v\right)_{U^d} \\ &= \left(u, \frac{\prod_{k=2}^{k=d}(1-a_k)^{k-1}}{\prod_{j=i+1}^{j=d}(1-a_j)} \frac{1}{(1-a_i)^{i-1}}\left[\partial_{a_i}\left(a_i(1-a_i)^i\partial_{a_i}\right)\right]v\right)_{U^d} \\ &= \left(u, \left(\frac{\prod_{k=2,k\neq i}^{k=d}(1-a_k)^{k-1}}{\prod_{j=i+1}^{j=d}(1-a_j)}\right)\left[\partial_{a_i}\left(a_i(1-a_i)^i\partial_{a_i}\right)\right]v\right)_{U^d} \\ &= \left(\left[\partial_{a_i}(a_i(1-a_i)^i\partial_{a_i})\right]u, \left(\frac{\prod_{k=2,k\neq i}^{k=d}(1-a_k)^{k-1}}{\prod_{j=i+1}^{j=d}(1-a_j)}\right)v\right)_{U^d} \\ &= \left(\left(L_{\mathbf{r}}^d\right)^i u, v\right)_{S^d}. \end{aligned}$$

We can also use the definition of the Jacobi polynomials to show that $L^d_{\mathbf{r}}$ has eigenfunctions,

$$\phi^d_{\mathbf{d}} = \prod_{i=1}^{i=d} (1-a_i)^{c_i} P^{2c_i+i-1,0}_{d_i}(a_i),$$

where $\mathbf{d} \in \mathbb{N}^d,\ c_i = \sum_{j=0}^{i-1} d_j$. The eigenvalues for these eigenfunctions are

$$\lambda^d_{\mathbf{d}} = c_{d+1}(c_{d+1}+d).$$

In summary, the operators we have defined in the d-dimensional simplex have simple tensor product eigenfunctions when mapped to a d-dimensional unit box. Also, their eigenvalues are $n(n+d)$ where n is the total degree of the eigenfunction in both the mapped and original coordinates $\mathbf{a}$ and $\mathbf{r}$.

Orthogonal (Modal) Bases

The results we now present for the triangle and the tetrahedron are special cases of the previous section, but for clarity we will outline some of the steps used to find these results.

We first consider the tetrahedron as this has the most complex mapping between the local cartesian coordinates and the tensor coordinates. Using the scaled coordinates $(a, b, c) \in [0, 1] \times [0, 1] \times [0, 1]$ for the tetrahedron (for simplicity) we can express the local orthogonal coordinates $(r, s, t) \in \{0 \le r, s, t, r+s+t \le 1\}$ as

$$\begin{aligned} r &= a(1-b)(1-c) \\ s &= b(1-c) \\ t &= c. \end{aligned}$$

We consider the operator

$$\begin{aligned} L_{\text{Tet}} :=\ & \partial_r(r(1-r)\partial_r - rs\partial_s - rt\partial_t) + \partial_s(s(1-s)\partial_s - sr\partial_r - st\partial_t) \\ & + \partial_t(t(1-t)\partial_t - tr\partial_r - ts\partial_s). \end{aligned}$$

Using the following identities it is straightforward to express the operator with respect to the (a, b, c) coordinates:

$$\begin{aligned} r\partial_r &= a\partial_a \\ s\partial_s &= \frac{ab}{(1-b)}\partial_a + b\partial_b \\ t\partial_t &= \frac{ac}{(1-b)(1-c)}\partial_a + \frac{bc}{(1-c)}\partial_b + c\partial_c \\ r\partial_r + s\partial_s + t\partial_t &= \frac{a}{(1-b)(1-c)}\partial_a + \frac{bc}{(1-c)}\partial_b + c\partial_c. \end{aligned}$$

After some manipulation the operator can be expressed in terms of the (a, b, c) coordinates and is

$$\begin{aligned} L_{\text{Tet}} =\ & \frac{1}{(1-b)}\frac{1}{(1-c)}[\partial_a(a(1-a)\partial_a)] + \frac{1}{(1-b)}\frac{1}{(1-c)}\left[\partial_b\left(b(1-b)^2\partial_b\right)\right] \\ & + \frac{1}{(1-c)^2}\left[\partial_c\left(c(1-c)^3\partial_c\right)\right]. \end{aligned}$$

This demonstrates that the operator maintains tensor form in the (a, b, c) coordinate system. It is now trivial to show that this is a self-adjoint operator by applying one-dimensional integration by parts to each of the three tensor parts. Also, by using the definition of the Jacobi polynomials we can show that the orthogonal basis is a set of eigenfunctions of L_{Tet} and find their eigenvalues.

We will now show that the polynomial functions ϕ_{ijk} defined by

$$\phi_{ijk} = P_i^{0,0}(a)\left(\frac{1-b}{2}\right)^i P_j^{2i+1,0}(b)\left(\frac{1-c}{2}\right)^{(i+j)} P_k^{2(i+j)+2,0}(c)$$

are eigenfunctions of the L_{Tet} operator. We consider the first part of the operator. The definition of the Jacobi polynomial directly implies the relationship

$$\partial_a(a(1-a)\partial_a\phi_{ijk}) = -i(i+1)\phi_{ijk}.$$

We now consider the first two terms of the operator. Using the previous result we can remove the dependency on a and then use the definition of the Jacobi polynomials with non-zero (α, β) to show that the polynomials are indeed eigenfunctions of the first two terms of the operator:

$$\begin{aligned} L_b\phi_{ijk} &= \partial_b(b(1-b)\partial_b\phi_{ijk}) - b\partial_b\phi_{ijk} - \frac{i(i+1)}{1-b}\phi_{ijk} \\ &= (1-b)^i P_i^{0,0}(a) P_k^{2(i+j)+2,0}(c) \\ &\quad \times \left[b(1-b)\frac{\partial^2}{\partial b^2} + (1-(3+2i)b)\frac{\partial}{\partial b} + (-i(2+i))\right] P_j^{2i+1,0}(b) \\ &= \phi_{ijk}[-j(j+(2i+1)+1) - i(2+i)] \\ &= -(i+j)(i+j+2)\phi_{ijk}. \end{aligned}$$

Applying the same technique again we come to the relationship

$$\begin{aligned} L_{\mathrm{Tet}}\phi_{ijk} &= \partial_c(c(1-c)\partial_c\phi_{ijk}) - 2c\partial_c\phi_{ijk} - \frac{(i+j)(i+j+2)}{1-c}\phi_{ijk} \\ &= \lambda_{ijk}\phi_{ijk}. \end{aligned}$$

Thus, the tensor product of Jacobi polynomials ϕ_{ijk} are eigenfunctions of the total operator L_{Tet} with eigenvalues

$$\lambda_{ijk} = -(i+j+k)(i+j+k+3).$$

This approach clearly applies to the triangular, quadrilateral, prismatic, and hexahedron elements as well.

The Orthogonal Triangle Basis

The triangle is a special case of the tetrahedron with $c = -1$. As before, we can specify a self-adjoint operator that has the orthogonal basis proposed by Dubiner [21] as its eigenfunctions. We set out the scaled coordinate system, operator and eigenfunction/eigenvalue pairs:

$$\begin{aligned}
r &= a(1-b)\\
s &= b\\
L_{\text{Tri}} &= \partial_r(r(1-r)\partial_r - rs\partial_s) + \partial_s(s(1-s)\partial_s) - rs\partial_r)\\
&= \frac{1}{1-b}[\partial_a(a(1-a)\partial_a)] + \frac{1}{(1-b)}\partial_b\big(b(1-b)^2\partial_b\big)\\
\phi_{ij} &= P_i^{0,0}(a)(1-b)^i P_j^{2i+1,0}(b)\\
\lambda_{ij} &= -(i+j)(i+j+2).
\end{aligned}$$

The Orthogonal Quadrilateral Basis

The straight tensor product property of the quadrilateral gives a very simple form for the operator; it is the sum of two one-dimensional operators. Similarly, each eigenfunction is a straight tensor product of one-dimensional eigenfunctions and their eigenvalue is a sum of the one-dimensional eigenvalues.

$$\begin{aligned}
r &= a\\
s &= b\\
L_{\text{Quad}} &= \partial_r(r(1-r)\partial_r) + \partial_s(s(1-s)\partial_s))\\
&= \partial_a(a(1-a)\partial_a) + \partial_b(b(1-b)\partial_b)\\
\phi_{ij} &= P_i^{0,0}(a)P_j^{0,0}(b)\\
\lambda_{ij} &= -i(i+1) - j(j+1).
\end{aligned}$$

The Orthogonal Prism Basis

The prism is simply a tensor product of a triangle and a uniform third direction. Hence, we sum the operator for the triangle in the (r, t) directions and the one-dimensional operator in the s direction.

Likewise, we can obtain the eigenfunction/eigenvalue pairs from the above analysis for the triangles,

$$\begin{aligned}
r &= a(1-c)\\
s &= b\\
t &= c\\
L_{\text{Prism}} &= \partial_r(r(1-r)\partial_r - rt\partial_t) + \partial_s(s(1-s)\partial_s) + \partial t(t(1-t)\partial_t - rt\partial_r)\\
&= \frac{1}{1-c}[\partial_a(a(1-a)\partial_a)] + [\partial_b(b(1-b)\partial_b)] + \frac{1}{1-c}\big[\partial_c\big(c(1-c)^2\partial_c\big)\big]\\
\phi_{ijk} &= P_i^{0,0}(a)P_j^{0,0}(b)(1-c)^i P_k^{2i+1,0}(c)\\
\lambda_{ijk} &= -(i+k)(i+k+2) - j(j+1).
\end{aligned}$$

The Orthogonal Hexahedral Basis

The hexahedral analysis is trivial since, like the quadrilateral, the hexahedron is a straight tensor product of three one-dimensional directions. The operators and eigenfunction and eigenvalue pairs are

$$\begin{aligned} r &= a \\ s &= b \\ t &= c \\ L_{\text{Hex}} &= \partial_r(r(1-r)\partial_r) + \partial_s(s(1-s)\partial_s) + \partial t(t(1-t)\partial_t) \\ &= \partial_a(a(1-a)\partial_a) + \partial_b(b(1-b)\partial_b) + \partial_c(c(1-c)\partial_c) \\ \phi_{ijk} &= P_i^{0,0}(a)P_j^{0,0}(b)P_k^{0,0}(c) \\ \lambda_{ijk} &= -i(i+1) - j(j+1) - k(k+1). \end{aligned}$$

The Decay of Basis Coefficients

Thus, each basis is a set of eigenfunctions of a singular Sturm–Liouville operator which leads us to the observations

$$\begin{aligned} \hat{u}_{ijk} &= (u, \phi_{ijk}) \\ &= \left(u, \frac{1}{\lambda_{ijk}} L(\phi_{ijk})\right) \\ &= \frac{1}{\lambda_{ijk}}(L(u), \phi_{ijk}) \\ &= \frac{1}{\lambda_{ijk}^m}\left(L^m(u), \phi_{ijk}\right). \end{aligned}$$

Hence, if the function is infinitely smooth we see that the coefficients $\hat{u}_{ijk}$ must decrease faster than any polynomial power of i, j, k. Thus, the sum

$$\tilde{u}^N = \sum_i^N \sum_j^N \sum_k^N \hat{u}_{ijk}\phi_{ijk}$$

must converge exponentially fast to u as N increases for all infinitely smooth u.

It is important to notice that since straight-sided tetrahedra and triangles have constant geometric mapping Jacobians these results hold for arbitrarily stretched tetrahedra and triangles. This does not follow for the other elements since their geometric Jacobians are quadratic for non-perpendicular elements. This backs up the findings that the simplicial elements handle deformation better than the other types.

Unfortunately, it does not appear that this method generalizes to the pyramid in a straightforward way. However, a suitable orthogonal basis is known for a pyramid:

$$\begin{aligned} r &= a(1-c) \\ s &= b(1-c) \\ t &= c \\ \phi_{ijk} &= P_i^{0,0}(a)P_j^{0,0}(b)(1-c)^{i+j}P_k^{2(i+j)+2,0}(c). \end{aligned}$$

This basis is only appropriate for supporting P^n since the c component is of the order $i+j+k$. Thus, if each $i, j, k \leq N$ but $i+j+k > N$ then it is necessary to use high-order quadrature to integrate these modes exactly. Also, if $i + j + k < N$ the function is a polynomial in r, s, and t.

APPENDIX II

Numerical Quadrature and Flux Computation

Here we provide some details on how the numerical quadrature is performed on polymorphicelements and specifically how the flux terms involved in the discontinuous Galerkin formulation are computed. To this end, we take advantage of the tensor product in the transformed space (a, b, c) to perform integration. Note that in the following we have changed the range of the variables to $(-1, 1)$ from the range $(1, 0)$ used in Appendix I. The integrations over each element can be performed as a set of one-dimensional integrals using Gauss quadrature. If we used the reference coordinate systems this would be very difficult since the limits of the "collapsed" elements are not constant.

We first describe the choice of quadrature type for integrating each direction. We will then motivate the inclusion of quadrature with non-constant weights in order to reduce the number of points we use.

In two dimensions we consider integrals of the form

$$\begin{aligned}\int_{\text{Physical}} f(\mathbf{x})\,dx\,dy &= \int_{\text{Reference}} f(\mathbf{x}(\mathbf{r}))\frac{\partial(\mathbf{x})}{\partial(\mathbf{r})}\,dr\,ds\\ &= \int_{\text{Tensor}} f(\mathbf{x}(\mathbf{r}(\mathbf{a})))\frac{\partial(\mathbf{x})}{\partial(\mathbf{r})}\frac{\partial(\mathbf{r})}{\partial(\mathbf{a})}\,da\,db,\end{aligned}$$

and in three dimensions,

$$\begin{aligned}\int_{\text{Physical}} f(\mathbf{x})\,dx\,dy\,dz &= \int_{\text{Reference}} f(\mathbf{x}(\mathbf{r}))\frac{\partial(\mathbf{x})}{\partial(\mathbf{r})}\,dr\,ds\,dt\\ &= \int_{\text{Tensor}} f(\mathbf{x}(\mathbf{r}(\mathbf{a})))\frac{\partial(\mathbf{x})}{\partial(\mathbf{r})}\frac{\partial(\mathbf{r})}{\partial(\mathbf{a})}\,da\,db\,dc.\end{aligned}$$

We use the Gauss weights that will perform the discrete integral of a function as a sum:

$$\int_{-1}^{1}(1-z)^{\alpha}(1+z)^{\beta} f(z)\,dz = \sum_{i=0}^{N-1} f\left(z_i^{\alpha,\beta}\right)w_i^{\alpha,\beta}.$$

This will be used in each of the d directions in the d-dimensional elements. In Table IV we show the type of Gaussian quadrature we use in each of the a, b, and c directions.

For the discontinuous Galerkin formulation it is necessary to evaluate terms of the form

$$\int_{\partial\Omega} f\phi_n + \int_{\Omega} F\phi_n,$$

where $\partial\Omega$ is the boundary of an element Ω, for all the ϕ_n test functions in the elemental basis. There are $N\frac{(N+1)}{2}$ test functions for a triangle so the boundary integral is an $O(N^3)$ operation. This means that the flux integration is as expensive as the volume integral. We

TABLE IV

Element	a	b	c
Triangle	GLL	$\text{GRJ}_{0,0}$	—
Quadrilateral	GLL	GLL	—
Tetrahedron	GLL	$\text{GRJ}_{0,0}$	$\text{GRJ}_{0,0}$
Pyramid	GLL	GLL	$\text{GRJ}_{0,0}$
Prism	GLL	GLL	$\text{GRJ}_{0,0}$
Hexahedron	GLL	GLL	GLL

Note. GLL implies Gauss–Lobatto–Legendre which is the Gauss quadrature for a constant weight function with both $x = \pm 1$ endpoints included. $\text{GRJ}_{\alpha,\beta}$ implies Gauss–Radau–Jacobi quadrature with (α, β) weights and one of the endpoints included.

can reduce the cost of this integral by examining the discrete sum form

$$\int_{\partial\Omega} f\phi_n = \sum_{i=0}^{N} \phi_n(a_i, 0)\, f^1(a_i) w_i^a J^1(a_i) + \sum_{i=0}^{N} \phi_n(1, b_i)\, f^2(b_i) w_i^b J^2(b_i) + \sum_{i=0}^{N} \phi_n(-1, b_i)\, f^3(b_i) w_i^b J^3(b_3),$$

where J^n and f^n are the Jacobian and flux function for the nth edge.

We can rewrite the edge_1 flux as

$$\int_{\text{edge}_1} f\phi_n = \sum_{j=0}^{N}\sum_{i=0}^{N} \left(\frac{J^1(a_i)}{w_0^b}\right) f^e(a_i)\delta_{j0}\phi_n(a_i, b_j) w_i^a w_j^b,$$

where

$$\delta_{ij} = \begin{matrix} 0 & \text{if } i \neq j \\ 1 & \text{if } i = j. \end{matrix}$$

The fluxes for the other edges can be constructed in similar ways. Using this summation representation we can now evaluate the surface flux integral by adding the edge fluxes scaled by weight and Jacobians to the F field and then evaluating one volume integral.

ACKNOWLEDGMENTS

This work was supported by AFOSR Grant F49620-97-1-0185 and computations were performed at NCSA of University of Illinois at Urbana-Champaign and on the IBM SP2 at Maui High Performance Computing Center and at Brown University.

REFERENCES

1. W. Dai and P. R. Woodward, A simple Riemann solver and high-order Godunov schemes for hyperbolic systems of conservation laws, *J. Comput. Phys.* **121**, 51 (1995).

2. A. L. Zachary, A. Malagoli, and P. Colella, A higher-order Godunov method for multidimensional ideal magnetohydrodynamics, *SIAM J. Sci. Stat. Comp.* **15** (1994).
3. R. E. Peterkin, M. H. Frese, and C. R. Sovinec, Transport magnetic flux in an arbitrary coordinate ALE code, *J. Comput. Phys.* **140**, 148 (1998).
4. K. G. Powell, *An Approximate Riemann Solver for Magnetohydrodynamics* (*That Works in More Than One Dimension*), Technical Report ICASE Report 94-24, ICASE, NASA Langley, 1994.
5. O. S. Jones, U. Shumlak, and D. S. Eberhardt, An implicit scheme for nonideal magnetohydrodynamics, *J. Comput. Phys.* **130**, 231 (1997).
6. P. Colella, M. Dorr, and D. D. Wake, *A Conservative Finite Difference Method for the Numerical Solution of Plasma Fluid Equations*, Technical Report UCRL-JC-129912, Lawrence Livermore National Laboratory, 1998.
7. H. O. Kreiss and J. Olinger, Methods for the approximate solution of time-dependent problems, in *GARP Publ. Ser. (GARP, Geneva, 1973)*, Vol. 10.
8. G. E. Karniadakis, Towards a numerical error bar in CFD, *J. Fuids Eng.* **117**, 7 (1995).
9. R. B. Dahlburg and J. M. Picone, Evolution of the Orszag–Tang vortex system in a compressible medium. I. Initial average subsonic flow, *Phys. Fluids B* **1**(11), 2153 (1989).
10. J. S. Hesthaven and D. Gottlieb, A stable penalty method for the compressible Navier–Stokes equations. I. Open boundary conditions, *SIAM J. Sci. Comp.* **17**(3), 579 (1996).
11. D. A. Kopriva, Multidomain spectral solutions of compressible viscous flows, *J. Comput. Phys.* **115**, 184 (1994).
12. J. Giannakouros and G. E. Karniadakis, A spectral element-FCT method for the compressible Euler equations, *J. Comput. Phys.* **115**, 65 (1994).
13. C. R. Evans and J. F. Hawley, Simulation of magnetohydrodynamic flows: A constrained transport method, Astro. *Phys. J.* **332**, 659 (1988).
14. A. J. Meir and P. G. Schmidt, Analysis and numerical approximation of a stationary MHD flow problem with nonideal boundary, *SIAM J. Numer. Anal.*, in press.
15. I. Lomtev, C. B. Quillen, and G. E. Karniadakis, Spectral/hp methods for viscous compressible flows on unstructured 2d meshes, *J. Comput. Phys.* **144**(2), 325 (1998).
16. I. Lomtev and G. E. Karniadakis, A discontinuous Galerkin method for the Navier–Stokes equations, *Int. J. Numer. Methods Fluids* **29**, 587 (1999).
17. C. E. Baumann and T. J. Oden, A discontinuous hp finite element method for the Euler and the Navier–Stokes equations, *Int. J. Numer. Method Fluids*, in press.
18. J. T. Oden, I. Babuska, and C. E. Baumann, A discontinuous hp finite element method for diffusion problems, *J. Comput. Phys.* **146**, 491 (1998).
19. P. Houston, C. Schwab, and E. Suli, Stabilized hp-finite element approximations of hyperbolic problems, *SIAM J. Numer. Anal.*, in press.
20. S. A. Orszag and C. Tang, Small-scale structure of two-dimensional magnetohydrodynamic turbulence, *J. Fluid Mech.* **90**(1), 129 (1979).
21. M. Dubiner, Spectral methods on triangles and other domains, *J. Sci. Comput.* **6**, 345 (1991).
22. S. J. Sherwin and G. E. Karniadakis, A new triangular and tetrahedral basis for high-order finite element methods, *Int. J. Numer. Methods Eng.* **123**, 3775 (1995).
23. S. J. Sherwin, Hierarchical hp finite elements in hybrid domains, *Finite Elem. Anal. Des.* **27**, 109 (1997).
24. S. J. Sherwin and G. E. Karniadakis, Tetrahedral hp finite elements: Algorithms and flow simulations, *J. Comput. Phys.* **124**, 14 (1996).
25. C. Johnson, *Numerical Solution of Partial Differential Equations by the Finite Element Method* (Cambridge Univ. Press, Cambridge, UK, 1994).
26. G. Jiang and C. W. Shu, On a cell entropy inequality for discontinuous Galerkin methods, *Math. Comp.* **62**, 531 (1994).
27. B. Cockburn and C. W. Shu, The local discontinuous Galerkin for time dependent convection-diffusion systems, *SIAM J. Numer. Anal.*, in press.

28. T. Linde, *A Three-Dimensional Adaptive Multifluid MHD Model of the Heliosphere*, Ph.D. thesis, University of Michgan, 1998.

29. J. Croisille, R. Khanfir, and G. Chanteur, Numerical simulation of the MHD equations by a kinetic-type method, *J. Sci. Comput.* **10**(1), 81 (1995).

30. V. Girault and P. A. Raviart, *Finite Element Methods for Navier–Stokes Equations* (Springer-Verlag, New York/Berlin, 1986).

31. I. G. Giannakouros, *Spectral Element/Flux-Corrected Methods for Unsteady Compressible Viscous Flows*, Ph.D. thesis, Department of Mechanical and Aerospace Engineering, Princeton University, 1994.

32. I. Babuska, C. E. Baumann, and T. J. Oden, A discontinuous hp finite element method for diffusion problems: 1D analysis, *Comput. Math. Appl.*, in press.

33. J. U. Brackbill and D. C. Barnes, The effect of nonzero $\nabla \cdot \mathbf{B}$ on the numerical solution of the magnetohydrodynamic equations, *J. Comput. Phys.* **35**, 426 (1980).

34. T. C. Warburton, *Spectral/hp Methods on Polymorphic Multi-Domains: Algorithms and Applications*, Ph.D. thesis, Division of Applied Mathematics, Brown University, 1998.

35. E. Priest, *Solar Magnetohydrodynamics* (Reidel, Dordrecht, 1982).

36. C. H. Crawford and G. E. Karniadakis, Control of unsteady flows via electro-magnetic fields, in *26th AIAA Fluid Dynamics Conference, San Diego, June 19–22, 1995*, AIAA-95-2185.

37. V. Shatrov, G. Mutschke, and G. Gerbeth, Numerical simulation of the two-dimensional MHD flow around a circular cylinder, *Magnetohydrodynamics* **33**, 3 (1997).

38. T. Weier, G. Gerbeth, G. Mutschke, E. Platacis, and O. Lielausis, Experiments on cylinder wake stabilization in an electrolyte solution by means of electromagnetic forces localized on the cylinder surface, *Exp. Therm. Fluid Sci.* **16**, 84 (1998).

39. D. Gottlieb and S. A. Orszag, *Numerical Analysis of Spectral Methods* (Soc. for Industr. & Appl. Math., Philadelphia, 1977).

40. S. J. Sherwin and G. E. Karniadakis, Tetrahedral hp finite elements: Algorithms and flow simulations, *J. Comput. Phys.* **122**, 191 (1995).

41. W. Cai, D. Gottlieb, and C. W. Shu, Non-oscillatory spectral Fourier methods for shock wave calculations, *Math. Comp.* **52**, 389 (1989).

42. R. G. Owens, Spectral approximations on the triangle, *Proc. R. Soc. London A* **454**, 857 (1998).

43. B. A. Wingate and M. A. Taylor, The natural function space for triangular and tetrahedral spectral elements, *SIAM J. Numer. Anal.*, in press.