Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 226 (2007) 534–570
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

An arbitrary Lagrangian–Eulerian discretization of MHD on 3D unstructured grids ☆

R.N. Rieben [a,*], D.A. White [b], B.K. Wallin [a], J.M. Solberg [c]

[a] *Scientific B-Division, Lawrence Livermore National Laboratory, United States*
[b] *Engineering Technologies Division, Lawrence Livermore National Laboratory, United States*
[c] *Defense Technologies Engineering Division, Lawrence Livermore National Laboratory, United States*

Received 16 June 2006; received in revised form 17 April 2007; accepted 24 April 2007
Available online 21 May 2007

Abstract

We present an arbitrary Lagrangian–Eulerian (ALE) discretization of the equations of resistive magnetohydrodynamics (MHD) on unstructured hexahedral grids. The method is formulated using an operator-split approach with three distinct phases: *electromagnetic diffusion, Lagrangian motion*, and *Eulerian advection*. The resistive magnetic induction equation is discretized using a compatible mixed finite element method with a second order accurate implicit time differencing scheme which preserves the divergence-free nature of the magnetic field. At each discrete time step, electromagnetic force and heat terms are calculated and coupled to the hydrodynamic equations to compute the Lagrangian motion of the conducting materials. By virtue of the compatible discretization method used, the invariants of Lagrangian MHD motion are preserved in a discrete sense. When the Lagrangian motion of the mesh causes significant distortion, that distortion is corrected with a relaxation of the mesh, followed by a second order monotonic remap of the electromagnetic state variables. The remap is equivalent to Eulerian advection of the magnetic flux density with a fictitious mesh relaxation velocity. The magnetic advection is performed using a novel variant of constrained transport (CT) that is valid for unstructured hexahedral grids with arbitrary mesh velocities. The advection method maintains the divergence-free nature of the magnetic field and is second order accurate in regions where the solution is sufficiently smooth. For regions in which the magnetic field is discontinuous (e.g. MHD shocks) the method is limited using a novel variant of algebraic flux correction (AFC) which is local extremum diminishing (LED) and divergence preserving. Finally, we verify each stage of the discretization via a set of numerical experiments.
Published by Elsevier Inc.

Keywords: Magnetohydrodynamics; Resistive MHD; Electromagnetic diffusion; Mixed finite element methods; $H(Curl)$ and $H(Div)$ – conforming methods; Discrete differential forms; Vector finite elements; Operator splitting; Electromagnetic advection; Constrained transport; TVD schemes; Algebraic flux correction; Unstructured grids

☆ This work was performed under the auspices of the US Department of Energy by the University of California, Lawrence Livermore National Laboratory under contract No. W-7405-Eng-48, UCRL-JRNL-222113.
* Corresponding author. Tel.: +1 925 422 3783.
E-mail address: rieben1@llnl.gov (R.N. Rieben).

0021-9991/$ - see front matter Published by Elsevier Inc.
doi:10.1016/j.jcp.2007.04.031

1. Introduction

We are interested in the simulation of magnetohydrodynamic events and electromechanical devices in three dimensions. Our primary goal is a numerical method that solves, in a self-consistent manner, the equations of electromagnetics (primarily statics and diffusion), heat transfer (primarily conduction), and nonlinear mechanics (motion, elastic–plastic deformation, and mechanical contact). Example applications for these simulations include magnetic flux compression generators, metal forming, and electromagnetic launchers. In this paper, we focus on the numerical discretization of the resistive electromagnetic induction equation in an arbitrary Lagrangian–Eulerian (ALE) fashion [1] for the purposes of computing $\vec{J} \times \vec{B}$ forces and $\vec{J} \cdot \vec{E}$ resistive energy losses for coupling to hydrodynamic and thermal calculations in an operator-split fashion. All computational results were obtained by incorporating ALE electromagnetics into a well-known ALE hydrodynamic code, ALE3D, which has been successfully used in a wide variety of computational physics applications including [2–8] In this paper, the discretization of the hydrodynamics is not discussed in detail.

The term *magnetohydrodynamics* (MHD) is used as a descriptor for a great variety of disparate phenomena. We are particularly interested in the simulation of pulse power devices that involve metal and air regions, and which are driven by known voltage or current sources. For example, explosively driven magnetic flux compression generators are devices that use explosives to amplify current, and can generate large current pulses with a much faster $\mathrm{d}I/\mathrm{d}t$ than a capacitor bank. Another application is industrial metal forming, where a current is injected into a metal sheet and the resulting $\vec{J} \times \vec{B}$ forces accelerate the metal sheet into a mold. A third application is rail-guns; these are essentially linear electric motors in which $\vec{J} \times \vec{B}$ forces accelerate the armature to very high velocities (several kilometers/second) useful for equation-of-state research. In these applications the metal regions undergo a significant plastic deformation, and in some situations the metal may transition to liquid. Our goal is accurate simulation of the evolving geometry of the metal, as well as accurate computation of electric currents, magnetic fields, temperatures, pressures, etc. Based on these applications of interest, we choose an existing arbitrary Lagrangian–Eulerian (ALE) hydrodynamics simulation code as our starting point. We acknowledge that ALE methods are not optimal for studying MHD turbulence such as occurs in magnetic fusion energy or astrophysical applications, but this is not our intended application.

In multi-physics ALE hydrodynamic codes, an operator-split method is typically employed where separate physics packages are run sequentially and update their variables in the Lagrangian frame. When the Lagrange motion of the mesh causes significant mesh distortion, that distortion is corrected with an equipotential relaxation of the mesh, followed by a second order monotonic remap of field quantities. This remap is equivalent to advection of field quantities through the mesh with a fictitious effective velocity determined by the amount of mesh relaxation. In our proposed ALE formulation of MHD, we will employ an operator-split method with three distinct steps:

- *Electromagnetic diffusion* – Solve the equations of electromagnetic diffusion in the Lagrangian frame at one discrete time step for fixed materials.
- *Lagrangian motion* – Move mesh nodes according to $\vec{J} \times \vec{B}$ forces assuming a $\frac{\mathrm{d}\vec{B}}{\mathrm{d}t} = 0$ "frozen-in-flux" condition.
- *Eulerian advection* – Only required if mesh is relaxed, advect (or transport) magnetic state variables to new mesh while preserving the divergence-free nature of the magnetic flux density.

While much progress has been made in obtaining numerical algorithms for coupled advection/diffusion of magnetic fields [9,10], there are several key obstacles to be overcome for a fully three-dimensional ALE finite element implementation on general unstructured hexahedral grids. One issue is the need to numerically preserve the divergence-free constraint of the magnetic flux density, $\vec{B}$ [11]. Failure to reproduce this fundamental physical property in any numerical discretization can lead to the non-physical acceleration of conducting materials due to the presence of fictitious magnetic charge. Methods for maintaining a divergence-free velocity field for incompressible flow, such as Lagrange multiplier constraints, penalty methods, elliptic projection (divergence cleaning), and relaxation-based elliptic projection (divergence damping), can in principle be applied to magnetic fields. However, more efficient and elegant approaches are based on the fact that $\vec{\nabla} \cdot \vec{B} = 0$ is not arbitrary, but is in fact a consequence of Ampere's law. If Ampere's law is discretized in a

particular manner, then $\vec{\nabla} \cdot \vec{B} = 0$ is satisfied exactly without any additional effort, and this is the basis of constrained transport methods [12–14]. Our proposed algebraic constrained transport discussed in Section 5.1 is a generalization of constrained transport for unstructured ALE simulations.

For our applications, an additional issue is the need to reproduce MHD shock fronts without introducing spurious oscillations in the magnetic field. Various flux limiters have been advocated for fluid dynamics, with the goal of limiting non-physical oscillations without introducing excessive artificial diffusion. These flux limiters can in principle be applied to magnetic fields, but care must be taken to not destroy the above mentioned divergence constraints. Some proposed methods and comparisons of various approaches can be found in [15–18,13,19–21]. Our proposed algebraic flux correction discussed in Section 5.4 limits an intermediate edge-based voltage in a manner that enforces a local extremum diminishing property on the magnetic flux. This algebraic flux correction is an intermediate step in the algebraic constrained transport algorithm and hence $\vec{\nabla} \cdot \vec{B} = 0$ is still satisfied exactly.

In this paper we review magnetic diffusion in the Lagrangian frame of a deforming region, and we argue that a mixed finite element method employing $H(Curl)$ and $H(Div)$ basis functions [22] is ideally suited for discretization of this partial differential equation. The algebraic constrained transport method and algebraic flux limiter are built upon the same topological curl operator that is used in the discrete diffusion equation. Computational experiments are performed to confirm the second order convergence of the method for smooth fields, and to quantify the ability to conserve energy and preserve discontinuities for strongly shocked problems.

2. Electromagnetic diffusion

As is typically the case for MHD, we solve Maxwell's equations under the so-called "good conductor" approximation and under this condition it is reasonable to neglect the displacement current, resulting in electromagnetic diffusion instead of wave propagation. The first step in our three part ALE formulation is to solve the equations of electromagnetic diffusion. We begin with a discussion of the relevant equations without material motion, then we discuss electromagnetic diffusion in moving materials.

2.1. Conductors at rest

We assume a charge-free three-dimensional domain Ω with a surface boundary Γ and an outwardly directed surface normal direction $\hat{n}$. The domain Ω consists of a set of materials, each specified by the values of the electromagnetic material properties σ and μ, the electrical conductivity and the magnetic permeability respectively. For our applications, we wish to drive an MHD simulation by a combination of known voltage and/or current sources. As such, we introduce a special source term to the Maxwell equations under the good conductor approximation. We therefore consider the following equations for a set of materials at rest:

$$\vec{\nabla} \cdot \sigma \vec{\nabla} \phi = 0 \tag{1}$$

$$\sigma \vec{E}^{\text{ind}} = \vec{\nabla} \times \frac{1}{\mu} \vec{B} + \sigma \vec{\nabla} \phi \tag{2}$$

$$\frac{\partial \vec{B}}{\partial t} = -\vec{\nabla} \times \vec{E}^{\text{ind}} \tag{3}$$

The divergence-free conduction current due to a scalar potential (or voltage) ϕ is first computed via (1), then added as a source term to Ampere's law (2), where we have introduced the term $\vec{E}^{\text{ind}}$ to denote the induced electric field. Finally, Faraday's law (3) is solved to obtain the time varying magnetic flux density $\vec{B}$ given the induced electric field $\vec{E}^{\text{ind}}$. It is common to eliminate either E or B from the above equations to yield a single equations involving the *Curl–Curl* operator, but we will not take that approach here. In instead, as in [23], we maintain two coupled first order equations, as this is beneficial for the advection step as discussed in Section 5 below.

2.2. Conductors in motion

We now consider electromagnetic diffusion in moving materials. We first consider the Eulerian case in which the material positions and field components are defined with respect to the fixed laboratory coordinate system. Let $\vec{x} = \{x_1, x_2, x_3\} \in \mathcal{E}$ denote the label of a point in Euclidean space $\mathcal{E}$ where the motion takes place. The coordinate system of $\mathcal{E}$ is called the *spatial* or *laboratory* system. Let each point in the material be labeled with $\vec{X} = \{X_1, X_2, X_3\} \in \mathcal{M}$. The coordinate system of $\mathcal{M}$ is the *material* system. We assume there exists a time dependent, bijective mapping which relates these two different labels of the same point,

$$\vec{x} = \vec{x}(\vec{X}, t), \vec{X} = \vec{X}(\vec{x}, t) \tag{4}$$

While not necessary, it is common to have these two coordinate systems be equal at time $t = 0$, the undeformed state. Let a vector field defined with respect to the laboratory frame be denoted with a prime, e.g. $F'(x,t)$, and the same vector field defined with respect to the material frame be unprimed, e.g. $F(X,t)$. In an Eulerian representation, the fields and operators of a partial differential equation (PDE) are functions of the fixed laboratory frame; in a Lagrangian representation the fields and operators of a PDE are functions of the moving material frame.

Faraday's law can be written in integral form as

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Gamma(t)} \vec{B}' \cdot \mathrm{d}a' = -\oint_{\partial\Gamma(t)} \vec{E}' \cdot \mathrm{d}l'$$

where the surface $\Gamma(t)$ is moving with the material. The material derivative $\frac{\mathrm{d}\vec{F}'}{\mathrm{d}t}$ of any flux-type quantity $\vec{F}$ is defined as

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Gamma(t)} \vec{F}' \cdot \mathrm{d}a' = \int_{\Gamma(t)} \frac{\mathrm{d}\vec{F}'}{\mathrm{d}t} \cdot \mathrm{d}a'$$

and, as shown in [24,25], a careful derivation gives

$$\frac{\mathrm{d}\vec{F}'}{\mathrm{d}t} = \frac{\partial \vec{F}'}{\partial t} - \vec{\nabla}' \times \vec{v}' \times \vec{F}' + \vec{v}'(\nabla' \cdot \vec{F}') \tag{5}$$

where the velocity of a material point $\vec{X}$ on the surface of integration is $\vec{v}' = \mathrm{d}\vec{x}/dt$. Combining (5) with (2), (3) yields the induction equation

$$\frac{\partial \vec{B}'}{\partial t} = -\vec{\nabla}' \times \frac{1}{\sigma}\vec{\nabla}' \times \frac{1}{\mu}\vec{B}' + \vec{\nabla}' \times \vec{v} \times \vec{B}' \tag{6}$$

2.3. Operator splitting

The first term on the right of (6) is magnetic diffusion while the second term is magnetic advection; the ratio of these is the magnetic Reynolds number $MRe = \frac{vL}{\lambda}$ where L is the characteristic size and $\lambda \equiv \frac{1}{\sigma\mu}$ is the magnetic diffusivity. For problems in which $MRe \approx 1$ and the velocity is such that advection and diffusion have opposite signs, we have near equilibrium $\frac{\partial \vec{B}'}{\partial t} \approx 0$ and time integration of the induction equation requires special care. However, for advection or diffusion dominated problems it is acceptable to employ an operator splitting of the equation. Let the induction equation be represented as

$$\frac{\partial \vec{B}}{\partial t} = L_\sigma(\vec{B}) + L_v(\vec{B})$$

where the operator L_σ denotes electromagnetic diffusion and the operator L_v denotes magnetic advection and consider the two separate equations

$$\frac{\partial \vec{B}_\sigma}{\partial t} = L_\sigma(\vec{B}_\sigma)$$
$$\frac{\partial \vec{B}_v}{\partial t} = L_v(\vec{B}_v)$$

Let $\mathcal{S}_\sigma$ denote the electromagnetic diffusion time integration operator that takes the field $\vec{B}_\sigma$ from a discrete time step n to time step $n+1$, and let $\mathcal{S}_v$ denote the magnetic advection time integration operator that takes $\vec{B}_v$ from a discrete time step n to time $n+1$. The generic operator splitting of the induction equation is then given by the composition

$$\vec{B}_{n+1} = [\mathcal{S}_\sigma \circ \mathcal{S}_v]\vec{B}^n$$

For this simple operator splitting the time accuracy is $O(\Delta t)$, but numerous alternatives exist that are $O(\Delta t^2)$ or better. The advantage of operator splitting is that different time integration can be used for $\mathcal{S}_\sigma$ and $\mathcal{S}_v$. Our approach for the induction equation is to perform diffusion in the Lagrangian system using implicit time integration, followed by updating the momentum equation in the Lagrangian system explicitly, followed by an optional mesh relaxation and advection step if the mesh becomes too distorted.

2.4. Material frame

If $\vec{A}$ is a vector in the material coordinate system $\mathcal{M}$ and $\vec{a}$ is the same vector in the laboratory system $\mathcal{E}$, then the components of these vectors are related by

$$a^i = \frac{\partial x^i}{\partial X^j} A^j$$

However, electromagnetic fields and fluxes do not simply transform as vectors. As shown in [24,25], voltage and flux are invariants with respect to the deformation transformation

$$\begin{array}{lcl} \textit{Material}(\textit{Lagrangian}) & & \textit{Laboratory}(\textit{Eulerian}) \\ \vec{E}\cdot \mathrm{d}\vec{x} & = & (\vec{E}' + \vec{v}' \times \vec{B}')\cdot \mathrm{d}\vec{x}' \\ \vec{B}\cdot \mathrm{d}\vec{a} & = & \vec{B}'\cdot \mathrm{d}\vec{a}' \end{array} \tag{7}$$

Differential arc length and surface area elements transform according to

$$\mathrm{d}\vec{x} = J^{\mathrm{T}}\mathrm{d}\vec{x}' \tag{8}$$
$$\mathrm{d}\vec{a} = |J|J^{-1}\mathrm{d}\vec{a}' \tag{9}$$

where our definition of the Jacobian matrix is $J_{ij} = \partial X_j/\partial x_i$. As a consequence, the electric field intensities and magnetic flux densities transform in a dual manner in order to maintain the invariance property of (7)

$$\vec{E} = J^{-1}(\vec{E}' + \vec{v}' \times \vec{B}') \tag{10}$$
$$\vec{B} = \frac{1}{|J|} J^{\mathrm{T}}\vec{B}' \tag{11}$$

In the material system the induction equation becomes

$$\frac{\partial \vec{B}}{\partial t} = -\nabla \times \frac{1}{\sigma}\nabla \times \frac{1}{\mu}\vec{B} \tag{12}$$

where it is understood that the *Curl* operator is with respect to the material coordinate system. Thus the form of the diffusion equation is invariant to material motion when the fields and operators are defined in the material frame. For the special case of a perfectly conducting material this equation gives $\frac{\partial \vec{B}}{\partial t} = 0$, the frozen-in-flux theorem. The operator splitting of the induction equation is particularly simple in the material frame: the first step is diffusion of the fields, the second step is to move the mesh nodes according to the resulting $\vec{J} \times \vec{B}$ force while maintaining $\frac{\mathrm{d}\vec{B}}{\mathrm{d}t} = 0$ during the mesh motion. When a mixed $H(\textit{Curl})$–$H(\textit{Div})$ discretization is used for the

equations, this latter step means that the magnetic degrees-of-freedom, which represent the net fluxes through each face of the mesh, are constant.

3. Discretization of electromagnetic diffusion equations in the Lagrangian frame

3.1. Mixed variational formulation and time discretization

The first step in applying a mixed finite element method is to cast the relevant PDEs into variational form. Our particular mixed variational formulation is derived from a combination of methods originally presented in [23,26]. Our goal is to obtain a numerical formulation where the primary discrete field is the magnetic flux density as found in the $\vec{B}$ field formulation of [23] and where the discrete time integration is performed using a generalized Crank–Nicholson method as found in the $\vec{A}$–ϕ potential formulation of [26]. The advantage of the $\vec{B}$ formulation of [23] becomes apparent during the advection phase of the ALE formulation, as magnetic flux is the best electromagnetic quantity to use for magnetic transport [27]. The advantages of the $\vec{A}$–ϕ potential formulation of [26] are that it can be up to second order accurate in time, and voltage sources can be explicitly added to a problem by specifying them as essential boundary conditions on the additional elliptic PDE, making this method well suited for coupling to an external RLC circuit model.

We begin by multiplying Ampere's law of (2) by a test function $\vec{W}^1 \in H(Curl)$ and integrate over the three-dimensional problem domain Ω to obtain the variational form

$$\int_\Omega \sigma \vec{E}^{\text{ind}} \cdot \vec{W}^1 = \int_\Omega \vec{\nabla} \times \frac{1}{\mu}\vec{B} \cdot \vec{W}^1 + \int_\Omega \sigma \vec{\nabla}\phi \cdot \vec{W}^1 \tag{13}$$

Now we perform integration by parts on (13) and apply the Gauss divergence theorem to obtain

$$\int_\Omega \sigma \vec{E}^{\text{ind}} \cdot \vec{W}^1 = -\int_\Omega \frac{1}{\mu}\vec{B} \cdot \vec{\nabla} \times \vec{W}^1 + \int_\Omega \sigma \vec{\nabla}\phi \cdot \vec{W}^1 + \oint_\Gamma \hat{n} \times \frac{1}{\mu}\vec{B} \cdot \vec{W}^1 \tag{14}$$

The resulting surface integral term has units of electrical current and will become important when we discuss boundary conditions; but for the sake of clarity, we will omit this term for the remaining derivation. Now we assume that the fields $\vec{E}$, $\vec{B}$ and ϕ are known at discrete time intervals denoted by the subscript integer n. At time $n+1$ we now have the following:

$$\int_\Omega \sigma \vec{E}^{\text{ind}}_{n+1} \cdot \vec{W}^1 = -\int_\Omega \frac{1}{\mu}\vec{B}_{n+1} \cdot \vec{\nabla} \times \vec{W}^1 + \int_\Omega \sigma \vec{\nabla}\phi_{n+1} \cdot \vec{W}^1 \tag{15}$$

We apply a generalized trapezoidal approximation for the time derivative of the magnetic field such that

$$\vec{B}_{n+1} = \vec{B}_n + (1-\alpha)\Delta t \left.\frac{\partial \vec{B}}{\partial t}\right|_n + \alpha \Delta t \left.\frac{\partial \vec{B}}{\partial t}\right|_{n+1} \tag{16}$$

The averaging parameter α determines the nature of the numerical time integration such that

$$\alpha = \begin{cases} 0 & \text{Explicit, First Order Accurate Forward Euler} \\ 1/2 & \text{Implicit, Second Order Accurate Crank Nicolson} \\ 1 & \text{Implicit, First Order Accurate Backward Euler} \end{cases}$$

Applying this discretization to Faraday's law (3), we obtain

$$\vec{B}_{n+1} = \vec{B}_n - \Delta t \vec{\nabla} \times ((1-\alpha)\vec{E}^{\text{ind}}_n + \alpha \vec{E}^{\text{ind}}_{n+1}) \tag{17}$$

Now we substitute $\vec{B}_{n+1}$ on the right hand side of (15) with the expression from (17) and move all quantities at time step n to the right hand side to obtain

$$\begin{aligned} &\int_\Omega \left(\sigma \vec{E}^{\text{ind}}_{n+1} \cdot \vec{W}^1 + \alpha \Delta t \frac{1}{\mu}\vec{\nabla} \times \vec{E}^{\text{ind}}_{n+1} \cdot \vec{\nabla} \times \vec{W}^1 \right) \\ &\quad = \int_\Omega \left(\frac{1}{\mu}\vec{B}_n \cdot \vec{\nabla} \times \vec{W}^1 - (1-\alpha)\Delta t \vec{\nabla} \times \vec{E}^{\text{ind}}_n \cdot \vec{\nabla} \times \vec{W}^1 \right) + \int_\Omega \sigma \vec{\nabla}\phi_{n+1} \cdot \vec{W}^1 \end{aligned} \tag{18}$$

We now introduce the time averaged variable, $\widetilde{\vec{B}}$, defined as

$$\widetilde{\vec{B}}_n \equiv \vec{B}_n - (1-\alpha)\Delta t \vec{\nabla} \times \vec{E}_n^{\text{ind}} \tag{19}$$

Next, we rewrite (18) in terms of our new time averaged variable $\widetilde{\vec{B}}$ to obtain

$$\int_\Omega \left(\sigma \vec{E}_{n+1}^{\text{ind}} \cdot \vec{W}^1 + \alpha \Delta t \frac{1}{\mu} \vec{\nabla} \times \vec{E}_{n+1}^{\text{ind}} \cdot \vec{\nabla} \times \vec{W}^1 \right) = \int_\Omega \frac{1}{\mu} \widetilde{\vec{B}}_n \cdot \vec{\nabla} \times \vec{W}^1 + \int_\Omega \sigma \vec{\nabla} \phi_{n+1} \cdot \vec{W}^1 \tag{20}$$

Note that the variational formulation of (20) is incomplete due to the presence of the ϕ_{n+1} term on the right hand side. In order to fully define the problem, we must add an additional variational equation to define the scalar potential. To do this, we multiply (1) by a scalar test function $W^0 \in H(Grad)$ and integrate over the domain Ω

$$\int_\Omega (\vec{\nabla} \cdot \sigma \vec{\nabla} \phi_{n+1}) W^0 \, \mathrm{d}\Omega = 0$$

and employ Green's first scalar identity to obtain

$$\int_\Omega \sigma \vec{\nabla} \phi_{n+1} \cdot \vec{\nabla} W^0 \, \mathrm{d}\Omega = \oint_\Gamma \hat{n} \cdot \sigma \vec{\nabla} \phi_{n+1} W^0 \, \mathrm{d}\Gamma \tag{21}$$

for all test functions W^0.

3.2. Boundary conditions

The natural and essential boundary conditions for (20) are

$$\begin{aligned} &\text{Natural} \quad \hat{n} \times \frac{1}{\mu} \vec{B} = 0 \\ &\text{Essential} \quad \hat{n} \times \vec{E}^{\text{ind}} = 0 \end{aligned} \tag{22}$$

These are sometimes referred to as the Neumann and Dirichlet boundary conditions, respectively. Recall that the *essential* boundary condition is a constraint that is enforced manually, whereas the *natural* boundary condition is automatically satisfied in the variational (weak) sense. In general, the inhomogeneous versions of these two boundary conditions require vector valued functions $\vec{g}_{\text{N}}(\Gamma)$ and $\vec{g}_{\text{D}}(\Gamma)$ such that

$$\begin{aligned} &\hat{n} \times \frac{1}{\mu} \vec{B} = \vec{g}_{\text{N}} \quad \text{on } \Gamma_{\text{N}} \\ &\hat{n} \times \vec{E} = \vec{g}_{\text{D}} \quad \text{on } \Gamma_{\text{D}} \end{aligned}$$

The natural and essential boundary conditions for (21) are

$$\begin{aligned} &\text{Natural} \quad \hat{n} \cdot \sigma \vec{\nabla} \phi = 0 \\ &\text{Essential} \quad \phi = 0 \end{aligned} \tag{23}$$

In other words, the normal component of the conduction current density $\sigma \vec{\nabla} \phi$ is the natural boundary condition while the surface scalar potential (or voltage) ϕ is the essential boundary condition. In general, the inhomogeneous versions of these two boundary conditions require scalar valued functions $g_{\text{N}}(\Gamma)$ and $g_{\text{D}}(\Gamma)$ such that

$$\begin{aligned} &\hat{n} \cdot \sigma \vec{\nabla} \phi = g_{\text{N}} \quad \text{on } \Gamma_{\text{N}} \\ &\phi = g_{\text{D}} \quad \text{on } \Gamma_{\text{D}} \end{aligned}$$

3.3. Curl and divergence conforming basis functions

We assume the three-dimensional domain Ω has been partitioned into a set of discrete hexahedral elements Σ_i, the union of which forms the finite element mesh Ω_h. Furthermore, we assume the surface boundary Γ has

been partitioned into the sets Γ_N and Γ_D, denoting surfaces to which either a Natural or Essential boundary condition is applied.

We will discretize the variational form of the electromagnetic diffusion equations of (20) and (21) using a mixed finite element method. In the context of Galerkin approximations of saddle-point variational problems, the choice of the finite element space plays a crucial role in the stability and convergence of the discretization. For the case of Maxwell's equations, mixed finite element methods which use $H(Curl)$ and $H(Div)$ conforming spaces to model the electric field intensities and magnetic flux densities respectively are preferred over traditional nodal vector spaces $H^0(\Omega)^3$. In simple terms, nodal basis functions enforce too much continuity, leading to spurious, non-physical solutions. The $H(Curl)$ and $H(Div)$ basis functions can be considered the finite element equivalent of so-called *compatible* finite difference and finite volume discretization methods that also utilize edge and face based degrees-of-freedom [28].

The salient features of the $H(Curl)$ and $H(Div)$ basis functions on a hexahedral element are as follows. We will denote the space of $H(Curl)$ basis functions as W^1 and $H(Div)$ basis functions as W^2. The functional form of the $H(Curl)$ basis functions on the reference hexahedral element $0 < (x,y,z) < 1$ are

$$\begin{array}{lll} x\text{-directed} & y\text{-directed} & z\text{-directed} \\ \vec{W}_1^1 = (1-y)(1-z) & \vec{W}_5^1 = (1-x)(1-z) & \vec{W}_9^1 = (1-x)(1-y) \\ \vec{W}_2^1 = y(1-z) & \vec{W}_6^1 = x(1-z) & \vec{W}_{10}^1 = x(1-y) \\ \vec{W}_3^1 = (1-y)z & \vec{W}_7^1 = (1-x)z & \vec{W}_{11}^1 = (1-x)y \\ \vec{W}_4^1 = yz & \vec{W}_8^1 = xz & \vec{W}_{12}^1 = xy \end{array} \tag{24}$$

Within an element, an $H(Curl)$-conforming field $\vec{F}^1(x,y,z)$ will be approximated by the basis function expansion

$$\vec{F}_h^1(x,y,z) = \sum_{i=1}^{12} f_i \vec{W}_i^1(x,y,z) \tag{25}$$

where the degrees-of-freedom f_i are given by

$$f_i = \int \vec{F}^1 \cdot \hat{t}_i \, \mathrm{d}l \tag{26}$$

where $\hat{t}_i$ is the unit tangent to edge i, i.e. the degrees-of-freedom are associated with the edges of the element. This is due to the fact that $\int \vec{W}_j^1 \cdot \hat{t}_i \, \mathrm{d}l = \delta_{ij}$, the basis functions "interpolate on edges". Clearly, across adjacent mesh elements the tangential component of $\vec{F}_h^1$ is continuous across the shared face (a requirement for $\vec{F}_h^1 \in H(Curl)$) whereas the normal component is discontinuous. These basis functions are well suited for electric fields, as electric fields have a jump discontinuity in normal component across material interfaces, and as shown in Section 3.2 electromagnetic boundary conditions involve only the tangential component of the electric field on the bounding surface.

The functional form of the $H(Div)$ basis functions on the reference hexahedral element are

$$\begin{array}{lll} x\text{-directed} & y\text{-directed} & z\text{-directed} \\ \vec{W}_1^2 = (1-x) & \vec{W}_3^2 = (1-y) & \vec{W}_5^2 = (1-z) \\ \vec{W}_2^2 = x & \vec{W}_4^2 = y & \vec{W}_6^2 = z \end{array} \tag{27}$$

Within an element, an $H(Div)$-conforming field $\vec{F}^2(x,y,z)$ will be approximated by the basis function expansion

$$\vec{F}_h^2(x,y,z) = \sum_{i=1}^{6} f_i \vec{W}_i^2(x,y,z) \tag{28}$$

where the degrees-of-freedom f_i are given by

$$\vec{f}_i = \int F \cdot \hat{n}_i \, \mathrm{d}a \tag{29}$$

where $\hat{n}_i$ is the unit normal to face i, i.e. the degrees-of-freedom are associated with the faces of the element. This is due to the fact that $\int \vec{W}_j^2 \cdot \hat{n}_i \, da = \delta_{ij}$, the basis functions "interpolate on faces." Clearly, across adjacent mesh elements the normal component of $\vec{F}_h^2$ is continuous across the shared face (a requirement for $\vec{F}_h^2 \in H(Div)$) whereas the tangential component is discontinuous. These basis functions are well suited for flux density or current density , as flux densities have a jump discontinuity in tangential component across material interfaces.

Consider the basis function $\vec{W}_1^1$ in (24). The curl of this function is $\vec{\nabla} \times \vec{W}_1^1 = \hat{y}(1-y) - \hat{z}(1-z)$. Notice that this can be written as the combination of $\vec{W}^2$ basis functions $\vec{W}_3^2 - \vec{W}_5^2$, and that this combination is divergence free. Now consider the basis function expansions for $\vec{F}_h^1(x,y,z)$ and $\vec{F}_h^2(x,y,z)$ on a single reference element, with the respective degrees-of-freedom being given by vectors **e** and **b** of length 12 and 6, respectively. Given ***e***, we can compute **b** such that $\vec{F}_h^2(x,y,z) = \nabla \times \vec{F}_h^1(x,y,z)$ using the matrix operation

$$\mathbf{b} = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & -1 & 0 & 1 \\ -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 1 & -1 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \mathbf{e} \tag{30}$$

The divergence of the resulting field $\vec{F}_h^2(x,y,z)$ is given by

$$\nabla \cdot \vec{F}_h^2(x,y,z) = -b_1 + b_2 - b_3 + b_4 - b_5 + b_6 \tag{31}$$

and this is exactly zero, for any given vector **e**. The matrix in (30) above is an edge-face incidence matrix, with the signs of the entries determined by the (arbitrary) definition of the directions of the edges and faces of the element.

This is a specific example of the inclusion condition $\vec{\nabla} \times \vec{W}^1 \subset \vec{W}^2$, and this inclusion condition is a component of the overarching deRham complex that is satisfied by the basis functions. The $H(Curl)$ and $H(Div)$ basis functions satisfy a discrete deRham diagram as shown in (32). In this diagram the top row indicates the continuous fields and the bottom row represents the discrete finite element approximation. The right arrows represent differentiation, from left to right: gradient, curl, and divergence, respectively. The vertical arrows represent projection from the continuous field to the discrete. The projection operator is the machinery that computes the degrees-of-freedom given the continuous field. The bottom row means that the vector identities $\vec{\nabla} \times \vec{\nabla}\phi_h = 0$ and $\vec{\nabla} \cdot \vec{\nabla} \times \vec{F}_h = 0$ are satisfied exactly, regardless of mesh distortion or mesh resolution h. Given an arbitrary discrete scalar field $\phi_h \in W^0$, we have $\vec{\nabla}\phi_h \in \{\vec{g} : \vec{g} \in W^1, \vec{\nabla} \times \vec{g} = 0\}$. Likewise, for an arbitrary discrete vector field $\vec{E}_h \in \vec{W}^1$, we have $\vec{\nabla} \times \vec{E}_h \in \{\vec{g} : \vec{g} \in \vec{W}^2, \vec{\nabla} \cdot \vec{g} = 0\}$. This allows the divergence conditions of the fields to be satisfied without any need for Lagrange or penalty constraints. One last point to be made is that the discrete deRham diagram commutes, and this is important in obtaining mathematical convergence proofs of stability and convergence.

$$\begin{array}{ccccccc} H(Grad) & \rightarrow^{\vec{\nabla}} & H(Curl) & \rightarrow^{\vec{\nabla}\times} & H(Div) & \rightarrow^{\vec{\nabla}\cdot} & L^2 \\ \downarrow \Pi_h^0 & & \downarrow \Pi_h^1 & & \downarrow \Pi_h^2 & & \downarrow \Pi_h^3 \\ W^0 & \rightarrow^{\vec{\nabla}_h} & \vec{W}^1 & \rightarrow^{\vec{\nabla}_h\times} & \vec{W}^2 & \rightarrow^{\vec{\nabla}_h\cdot} & W^3 \end{array} \tag{32}$$

The above formulas (24) and (27) for $\vec{W}^1$ and $\vec{W}^2$, respectively, are for the reference hexahedral element only. For a distorted element the basis functions transform in such a manner to maintain the definition of degrees-of-freedom, and hence maintain the proper continuity (tangential or normal) across elements. The $\vec{W}^1$ and $\vec{W}^2$ basis functions transform dual to the edge tangent and face normal vectors, respectively. Note that the matrix operation (30) representing the discrete curl operator is still valid for a distorted element when the basis functions are defined to transform in this manner. Hence this matrix operation is referred to as a *topological derivative*, meaning that the matrix entries depend upon the connectivity of the mesh, but not upon the actual value of the mesh coordinates. The transformations are shown in Table 1, visual examples of the basis functions and their transformation properties are shown in Figs. 1 and 2. In this table we also include the basis functions W^0

Table 1
Basis function transformation rules

Object	AKA	Transformation rule	Units
W^0	Node basis	$W^0(\vec{X}) = W^0(\vec{x})$	h^0
$\vec{W}^1$	Edge basis	$\vec{W}^1(\vec{X}) = J^{-1}\vec{W}^1(\vec{x})$	h^{-1}
$\vec{W}^2$	Face basis	$\vec{W}^2(\vec{X}) = \frac{1}{\lvert J\rvert}J^{\mathrm{T}}\vec{W}^2(\vec{x})$	h^{-2}
W^3	Cell basis	$W^3(\vec{X}) = \frac{1}{\lvert J\rvert}\vec{W}^3(\vec{x})$	h^{-3}

and W^3, which correspond to scalar nodal and cell-center basis functions, respectively. The transformations in Table 1 preserve the deRham diagram, with the discrete differential operators on the bottom row of the deRham diagram corresponding topological derivative matrices. Interestingly, and critically, the transformations for $\vec{W}^1$ and $\vec{W}^2$ are the same as that of electric fields and magnetic flux densities in a laboratory frame (vs material frame) in (10) and (11). As discussed in Section 2.4, a perfectly conducting material has $\frac{\partial B}{\partial t} = 0$ in the material frame, and this condition is trivial to satisfy when using an $H(Div)$ representation for $\vec{B}$, as the condition is simply to keep the degrees-of-freedom constant during mesh motion.

In our proposed ALE formulation the scalar potential ϕ will be discretized using W^0 basis functions, the electric field will be discretized using W^1 basis functions, and the magnetic flux density $\vec{B}$ will be discretized using W^2 basis functions. For a hexahedral element in the Lagrangian frame we have

$$\phi(\vec{X},t) \approx \sum_{i=1}^{8} v_i(t) W_i^0(\vec{X}), W^0 \in H(Grad) \tag{33}$$

$$\vec{E}(\vec{X},t) \approx \sum_{i=1}^{12} e_i(t) \vec{W}_i^1(\vec{X}), \vec{W}^1 \in H(Curl) \tag{34}$$

$$\vec{B}(\vec{X},t) \approx \sum_{i=1}^{6} b_i(t) \vec{W}_i^2(\vec{X}), \vec{W}^2 \in H(Div) \tag{35}$$

For the case of (33), the degrees-of-freedom $v_i(t)$ are time dependent voltages at element nodes and the basis functions are unit-less. For the case of (34), the degrees-of-freedom $e_i(t)$ are time dependent induced voltages along element edges and the basis functions have units of inverse distance. Finally, for the case of (35), the degrees-of-freedom $b_i(t)$ are time dependent magnetic fluxes through element faces and the basis functions have units of inverse area.

3.4. Mixed finite element discretization

We employ the finite element library FEMSTER [29,30,26] for computation of the local "mass", "stiffness", and "derivative" matrices, where γ denotes an arbitrary symmetric tensor function of time and space (for material constitutive relations) and the superscript $l = 0, 1, 2, 3$ denotes the degree of the form, where *form* refers to the type of basis function, i.e. $H(Grad)$ $(l = 0)$, $H(Curl)$ $(l = 1)$, $H(Div)$ $(l = 2)$, or L_2 $(l = 3)$. The matrices can be written generically as

$$\mathbf{M}^l(\gamma)_{ij} = \int_\Omega \gamma W_i^l W_j^l \,\mathrm{d}\Omega \tag{36}$$

$$\mathbf{S}^l(\gamma)_{ij} = \int_\Omega \gamma \,\mathrm{d}W_i^l \cdot \mathrm{d}W_j^l \,\mathrm{d}\Omega \tag{37}$$

$$\mathbf{D}^{l(l+1)}(\gamma)_{ij} = \int_\Omega \gamma \,\mathrm{d}W_i^l \cdot W_j^{l+1} \,\mathrm{d}\Omega \tag{38}$$

Note that the d operator denotes *Gradient*, *Curl*, or *Divergence*, for $l = 0, 1, 2$ respectively. The "mass" matrices $\mathbf{M}$ and the "stiffness" matrices $\mathbf{S}$ are square and map l-forms to l-forms, the "derivative" matrices $\mathbf{D}$ are rectangular and map l-forms to $(l+1)$-forms. It can be shown that

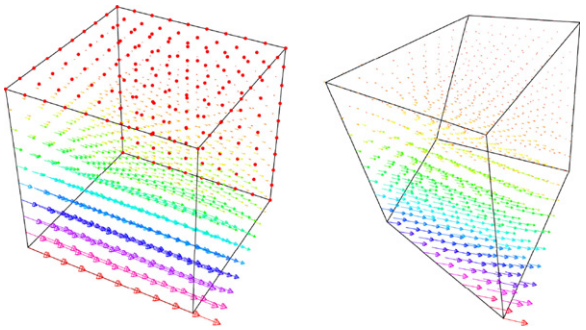

Fig. 1. Vector plot of $H(Curl)$ basis function on reference element (*left*) and on a distorted element (*right*). Note how function has a tangential component along one and only one edge in the element, regardless of element shape.

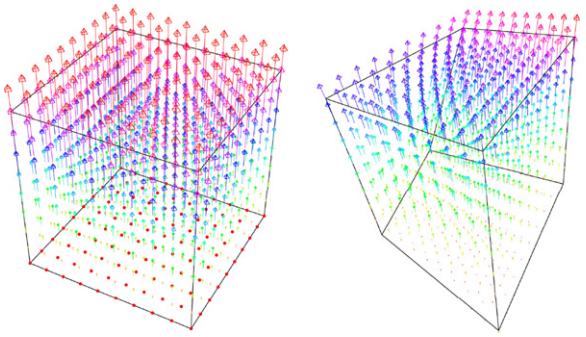

Fig. 2. Vector plot of $H(Div)$ basis function on reference element (*left*) and on a distorted element (*right*). Note how function has a normal component along one and only one face in the element, regardless of element shape.

$$\mathbf{D}^{l(l+1)} = \mathbf{M}^{l+1}\mathbf{K}^{l(l+1)} \tag{39}$$

$$\mathbf{S}^{l} = (\mathbf{K}^{l(l+1)})^{\mathrm{T}}\mathbf{M}^{l+1}\mathbf{K}^{l(l+1)} \tag{40}$$

where $\mathbf{K}^{l(l+1)}$ is a "topological derivative" matrix. This matrix is the discretization of the exterior derivative operator d from differential geometry, $\mathrm{d}W^{l} = W^{(l+1)}$. This matrix depends upon the mesh connectivity, but is independent of the nodal coordinates. It does not involve an integral over the element, and it does not involve any material properties. While seemingly abstract, it is enormously valuable in practice. Given an l-form quantity X with basis function expansion

$$X = \sum_{i=1}^{n} x_i W_i^{l}, \tag{41}$$

and an $(l+1)$-form quantity Y with basis function expansion

$$Y = \sum_{i=1}^{n} y_i W_i^{(l+1)}, \tag{42}$$

the exterior derivative (*Gradient*, *Curl*, *Divergence* for $l = 0, 1, 2$ respectively) is given by

$$\mathbf{y} = \mathbf{K}^{l(l+1)}\mathbf{x}. \tag{43}$$

It can be shown that

$$\mathbf{K}^{12}\mathbf{K}^{01} = 0 \tag{44}$$

$$\mathbf{K}^{23}\mathbf{K}^{12} = 0 \tag{45}$$

which are the discrete versions of $\mathrm{d}(\mathrm{d}W^l) = 0$. In terms of standard vector calculus, these matrix relations correspond to the identities $\vec{\nabla} \times \vec{\nabla} f = 0$ and $\vec{\nabla} \cdot \vec{\nabla} \times \vec{F} = 0$, respectively. These identities are satisfied in the discrete sense, exactly (to machine precision), for any mesh and any order basis function. While FEMSTER supports arbitrary order elements, basis functions, and quadratures, only linear basis functions will be employed here.

Using the matrix notation previously defined, we can now write the fully discrete formulation of the magnetic induction equation in the Lagrangian frame as

$$\mathbf{S}^0 \mathbf{v}_{n+1} = \mathbf{g}^0 \tag{46}$$

$$(\mathbf{M}^1(\sigma) + \alpha \Delta t \mathbf{S}^1(\mu^{-1})) \mathbf{e}^{\mathrm{ind}}_{n+1} = (\mathbf{D}^{12}(\mu^{-1}))^{\mathrm{T}} \tilde{\mathbf{b}}_n + \mathbf{D}^{01}(\sigma) \mathbf{v}_{n+1} - \mathbf{g}^1 \tag{47}$$

$$\mathbf{b}_{n+1} = \tilde{\mathbf{b}}_n - \alpha \Delta t \mathbf{K}^{12} \mathbf{e}_{n+1} \tag{48}$$

$$\tilde{\mathbf{b}}_{n+1} = \mathbf{b}_{n+1} - (1 - \alpha) \Delta t \mathbf{K}^{12} \mathbf{e}_{n+1} \tag{49}$$

where $\mathbf{v}$ is an array consisting of the time dependent voltage degrees-of-freedom from the scalar potential solve of (33) for every node in the mesh, $\mathbf{e}^{\mathrm{ind}}$ is an array consisting of the time dependent induced voltage degrees-of-freedom of (34) for every edge in the mesh and $\mathbf{b}$ is an array consisting of the time dependent magnetic flux degrees-of-freedom of (35) for every face in the mesh. The terms $\mathbf{g}^0$ and $\mathbf{g}^1$ are voltage and current density source terms respectively, which can be added to drive a problem. Note that the face based array $\tilde{\mathbf{b}}$, the discrete analog of the secondary variable we introduced in (19), is the only state variable required to be known at time n. This is a critical feature of our discretization; it means the only state variable that needs to be remapped during our Eulerian advection phase in Section 5 is this face based time averaged flux.

Note that in (47) the rectangular derivative matrices $\mathbf{D}^{12}(\mu^{-1})$ and $\mathbf{D}^{01}(\sigma)$ are discrete versions of the *Curl* and *Gradient* defined with respect to the Lagrangian frame (i.e. they have metric information encoded in them by virtue of the mass matrix). As such, they will change as the mesh is moved by $\vec{J} \times \vec{B}$ forces. Furthermore, the discrete divergence constraints on the fields are given by

$$(\mathbf{D}^{01}(\sigma))^{\mathrm{T}} \mathbf{e} = 0 \tag{50}$$

$$\mathbf{K}^{23} \mathbf{b} = 0 \tag{51}$$

and from the identities (45) and (44) these constraints are implicitly satisfied for all time, assuming the initial conditions and the source terms are divergence free.

4. Lagrangian motion

In this section we review methods for coupling the electromagnetic force and heat terms to the equations of Lagrangian motion. This phase of the calculation can be viewed as the Lagrangian treatment of the advection operator L_v of the magnetic induction equation. Given an electromagnetic force, we move the mesh nodes according to this force, keeping the magnetic degrees-of-freedom (face fluxes) constant. The new node locations affect the basis functions, so while the magnetic degrees-of-freedom are constant the magnetic field is in fact properly advected. We begin with the general continuum equation of motion derived from Newton's second law. In a Lagrangian reference frame, this is given by

$$\rho \frac{\partial^2 \vec{u}}{\partial t^2} = \vec{\nabla} \cdot \overleftrightarrow{S} + \vec{F} \tag{52}$$

where ρ is the material mass density, $\vec{u}$ is the displacement vector, $\overleftrightarrow{S}$ is the Cauchy stress tensor, and $\vec{F}$ is an independent volumetric body force density. The variational form of (52) is constructed by multiplying by a test vector $\vec{w}$ and integrating over the entire domain Ω

$$\frac{\partial^2}{\partial t^2} \int_\Omega \rho \vec{u} \cdot \vec{w} \, \mathrm{d}\Omega = \int_\Omega (\vec{\nabla} \cdot \overleftrightarrow{S}) \cdot \vec{w} \, \mathrm{d}\Omega + \int_\Omega \vec{F} \cdot \vec{w} \, \mathrm{d}\Omega \tag{53}$$

If the test vector $\vec{w}$ is considered to have units of distance, then each term in the above equation has units of work; hence this variational method is often referred to as the method of virtual work. For a valid variational

method, each component of the test vector w_i must be a fully continuous function, i.e. $w_i \in H(Grad)$, and the stress tensor must satisfy certain symmetry conditions. Integration by parts is employed to yield

$$\frac{\partial^2}{\partial t^2}\int_\Omega \rho\vec{u}\cdot\vec{w}\,\mathrm{d}\Omega = \int_\Gamma (\overleftrightarrow{S}\cdot\hat{n})\cdot\vec{w}\,\mathrm{d}\Gamma - \int_\Omega \overleftrightarrow{S} : (\vec{\nabla}\otimes\vec{w})\,\mathrm{d}\Omega + \int_\Omega \vec{F}\cdot\vec{w}\,\mathrm{d}\Omega \tag{54}$$

where $\hat{n}$ is the outward normal of the surface Γ. The common boundary conditions are the *displacement* condition $\vec{u} = \vec{d}$ on Γ and the *traction* condition $\overleftrightarrow{S}\cdot\hat{n} = \vec{t}$ on Γ. It is common practice in ALE hydrocodes to decompose the stress tensor into a sum of deviatoric and hydrostatic components such that

$$S_{ij} = \tau_{ij} - P\delta_{ij} \tag{55}$$

where P is the hydrostatic pressure defined to be the mean of the principle stresses $P = \frac{1}{3}S_{ii}$, and δ_{ij} is the Kronecker delta. The deviatoric stress components τ_{ij} are determined by a material's strength model (if present) while the hydrostatic components are determined by the material's equation-of-state (EOS).

4.1. Computation of electromagnetic force

There are multiple options for coupling the electromagnetic force to the elastic equation of motion (52). The conceptually simplest approach is to compute $\vec{F} = \vec{J}\times\vec{B}$ and use this as the body force in (53). As shown in [24], the $\vec{J}\times\vec{B}$ body force density is equivalent to the divergence of a Maxwell stress tensor plus a term involving the divergence of $\vec{B}$ such that

$$\vec{J}\times\vec{B} = \frac{1}{\mu}\vec{B}\cdot(\vec{\nabla}\otimes\vec{B}) - \vec{\nabla}\left(\frac{|B|}{2\mu}\right) = \vec{\nabla}\cdot\overleftrightarrow{T} - \vec{B}(\vec{\nabla}\cdot\vec{B})$$

Under the good conductor approximation (i.e. ignoring energy stored in displacement current), the Maxwell stress tensor (MST) is given by

$$T_{ij} = \frac{1}{\mu}\left(B_iB_j - \frac{1}{2}\delta_{ij}B_kB_k\right) \tag{56}$$

Provided that $\vec{\nabla}\cdot\vec{B} = 0$ (implying no forces due to the presence of magnetic charge), then the $\vec{J}\times\vec{B}$ body force and the MST approach will yield identical accelerations of a conducting body. It is interesting to point out the similarities between the Cauchy stress tensor decomposition of (55) and the Maxwell stress tensor of (56). The MST consists of a deviatoric component $\frac{1}{\mu}B_iB_j$ and a pressure component consisting of the principle Maxwell stresses $\frac{1}{2}\delta_{ij}B_kB_k$. For MHD problems, the mean of the principle Maxwell stress terms is equivalent to the magnetic pressure $\frac{1}{2\mu}|\vec{B}|^2$. The deviatoric components of the MST can add an effective "magnetic strength" to materials that might otherwise have no strength. This gives rise to the physical phenomena of shear Alfven waves, an example of which is given in Section 6. From a discretization standpoint it is very straightforward to implement the MST approach. We simply evaluate the components of (56) at element quadrature points in the Lagrangian frame at the discrete time level $n+1$ via the face based representation of (35) and add these values to the corresponding component of the Cauchy stress tensor in the discretization of (54). This is consistent with the time centering of the hydrodynamic variables in ALE3D, where the Cauchy stress divergence terms (which are used to compute accelerations) are known at the discrete time step $n+1$, since stress rates are integrated at $n+\frac{1}{2}$.

4.2. Computation of resistive energy loss

Due to the resistive nature of the coupled magnetic induction equations of (2) and (3), the energy stored in the magnetic fields is subject to dissipation due to Joule heating. To account for this energy loss, we need to compute a resistive energy loss term and couple this to the internal energy update equation. This can be accomplished by computing the resistive energy loss density

$$e_\sigma = \vec{J}\cdot\vec{E} = \sigma\vec{E}\cdot\vec{E} = \sigma(\vec{E}^{\mathrm{ind}} - \vec{\nabla}\phi)\cdot(\vec{E}^{\mathrm{ind}} - \vec{\nabla}\phi) \tag{57}$$

Again, from a discretization standpoint it is very straightforward to implement the resistive energy loss term. We simply evaluate the term of (57) at element centroids at the discrete time level $n+1$ via the edge-based

representation of (34) and the node based representation of (33) and add these values to existing zonal energies at time $n+1$, which is consistent with the time centering of energy density variables in ALE3D.

5. Eulerian advection

The ALE3D code performs an optional equipotential relaxation of the mesh. This is important for problems with gross deformation of the mesh, it prevents the mesh from becoming tangled. If relaxation is performed, fields defined on the "old" mesh must be remapped to the "new" mesh. This remapping is equivalent to Eulerian advection, but with a fictitious mesh velocity $\vec{v}_m$. It is interesting to note that it is possible to implement a pure Eulerian formulation as a Lagrange step followed by a complete remap step in which the mesh snaps back to its original configuration at every time step. We consider only new grids which are "nearby" in the sense that only small perturbations of the grid are allowed (i.e. the mesh nodes should not travel farther than one mesh element in any one relaxation step). This is known as the continuous remap approximation (CRA). This is in contrast to general remapping methods (a.k.a. interpolation methods, see [31]) whose goal is to remap quantities between two arbitrary grids. Under the CRA, the nodes of the old mesh are displaced to new locations; the topology (or connectivity) of the mesh does not change. Furthermore we restrict the relaxation process to interior mesh nodes, keeping all boundary nodes fixed. A key point is that the divergence of the magnetic flux density should be preserved during the advection process, this is referred to as constrained transport or constrained interpolation. The definition of the Maxwell stress tensor assumes a zero divergence field, so if the advection step does not preserve divergence then some additional post processing (projection, filtering) would be required to prohibit unlimited growth of magnetic monopoles and the resulting non-physical forces.

5.1. Constrained transport of magnetic flux

We assume the frozen flux condition (the diffusion of the fields has already been computed) and now our goal is to compute the rate of change of the magnetic flux density due to advective "transport" caused by the mesh motion. In essence, we are holding the magnetic field $\vec{B}$ fixed in space and letting the mesh relax around it; this is opposite in sense to advecting a magnetic field across a fixed Eulerian mesh (as is the case with the original CT method of [12]). The change in magnetic flux density due to mesh relaxation is therefore

$$\frac{\partial}{\partial t}\vec{B} = -\vec{\nabla} \times \vec{v}_m \times \vec{B} \tag{58}$$

where v_m denotes the mesh velocity. It is imperative that this process maintain the solenoidal nature of the $\vec{B}$ field by satisfying the constraint

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

Now consider an arbitrarily oriented surface S with differential surface area $\mathrm{d}\vec{a}$. We integrate (58) over the surface S

$$\int_S \frac{\partial}{\partial t}\vec{B} \cdot \mathrm{d}\vec{a} = -\int_S \vec{\nabla} \times (\vec{v}_m \times \vec{B}) \cdot \mathrm{d}\vec{a}$$

Now we apply Stokes' theorem to obtain

$$\frac{\partial \Phi}{\partial t} = -\oint_C (\vec{v}_m \times \vec{B}) \cdot \mathrm{d}\vec{x} \tag{59}$$

where Φ denotes total magnetic flux through the surface S, and C represents the boundary of the surface S with differential arc length $\mathrm{d}\vec{x}$. Eq. (59) states that a voltage in a circuit loop C is induced by a time rate of change of flux through this loop due to the motion of the mesh across the "frozen-in" $\vec{B}$ field.

Now suppose the surface S, in the presence of a fixed background $\vec{B}$ field, moves arbitrarily (including distortion, stretching, re-orientation, etc.) in a time Δt_m. We can approximate the time derivative for the magnetic flux using a simple finite difference

$$\frac{\partial \Phi}{\partial t} \approx \frac{\Phi^{\text{new}} - \Phi^{\text{old}}}{\Delta t_m}$$

where Φ^{old} is the flux through the original surface S at time t and Φ^{new} is the flux through the surface S at time $t + \Delta t_m$. We know from a Taylor series analysis that this simple finite difference will be second order accurate if $\frac{\partial \Phi}{\partial t}$ is known at time $t + \frac{1}{2}\Delta t_m$. This provides us with a numerical method for computing the new flux

$$\Phi^{\text{new}} \approx \Phi^{\text{old}} - \Delta t_m \oint_C \left(\frac{\vec{u}}{\Delta t_m} \times \vec{B} \right) \cdot \mathrm{d}\vec{x} = \Phi^{\text{old}} - \oint_C (\vec{u} \times \vec{B}) \cdot \mathrm{d}\vec{x} \tag{60}$$

where $\vec{u} = \Delta t_m \vec{v}_m$ is the displacement of the surface S. Stated another way, we can approximate the flux through the new surface by "measuring" the voltage in the closed circuit loop C. This approximation is most accurate if the location of the loop C is halfway between the old face and the new face. See Fig. 3 for a depiction of this.

5.2. Algebraic constrained transport on 3D unstructured grids

Let $\vec{x}^{\text{old}}$ denote the positions of the mesh nodes after a Lagrangian time step and let $\vec{x}^{\text{new}}$ denote the mesh nodes after one mesh relaxation step. We define the nodal displacement as

$$\vec{u} \equiv \vec{x}^{\text{new}} - \vec{x}^{\text{old}} \tag{61}$$

Furthermore, we define an intermediate nodal position $\vec{x}^{\text{mid}}$ as

$$\vec{x}^{\text{mid}} = \vec{x}^{\text{old}} + \frac{1}{2}\vec{u} \tag{62}$$

Since the topology (or connectivity) of a mesh is constant for all time, there is a one to one correspondence between mesh entities such as edges and faces at the old, intermediate and new locations. This allows us to define an *intermediate mesh* with unique edges and faces, topologically identical to the old and new faces. These intermediate quantities differ geometrically from their old and new counterparts by virtue of the nodal positions $\vec{x}^{\text{mid}}$. A schematic representation of this is shown in Fig. 4.

Now suppose we have calculated the magnetic flux density $\vec{B}$ in a Lagrangian time step via the proposed method of (48). Recall that $\vec{B}$ is a 2-form and is approximated by 2-form basis functions according to the expansion

$$\vec{B}^{\text{old}} \approx \sum_{i=1}^{6} b_i^{\text{old}} \vec{W}_i^{2,\text{old}} \tag{63}$$

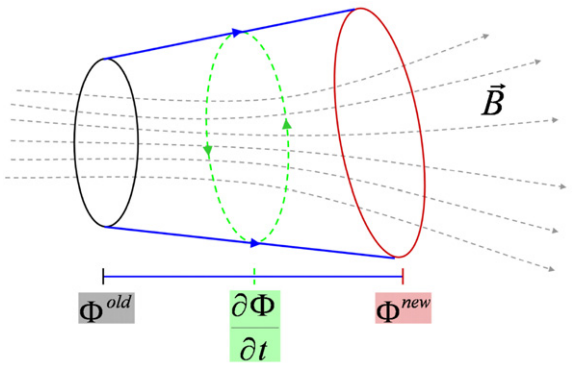

Fig. 3. Schematic diagram depicting the relationship between magnetic flux through two arbitrary faces and the corresponding time rate of change of magnetic flux.

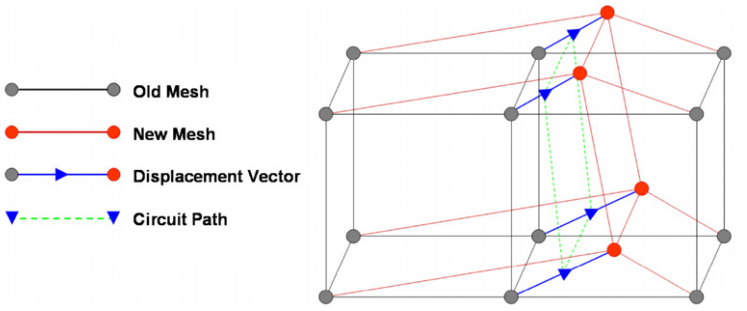

Fig. 4. Schematic diagram depicting a simple two element mesh with one face displaced. Topologically speaking, the old and new meshes are identical. They differ geometrically by the location of the nodes in space. The voltage update circuit is depicted in dashed-green and corresponds to a set of four edges, determined by the four intermediate nodes, which forms the boundary of an intermediate face.

The degrees-of-freedom b_i^{old} in this expansion carry the units of magnetic flux; this implies that we know the magnetic flux through every face in the Lagrangian (or old) mesh. Our goal is to compute new values of the magnetic flux b_i^{new} which will allow us to represent the magnetic flux density on the new mesh as

$$\vec{B}^{\text{new}} \approx \sum_{i=1}^{6} b_i^{\text{new}} \vec{W}_i^{2,\text{new}} \tag{64}$$

where $\vec{W}_i^{2,\text{new}}$ denote the basis functions for the new mesh (which are known once the locations of the new mesh nodes are computed).

Using (60) as a starting point, we can compute the flux through a given face in the new mesh by

$$b_i^{\text{new}} = b_i^{\text{old}} + \Delta b_i \tag{65}$$

The flux change Δb_i we are adding to each face in the old mesh is computed by numerically integrating the voltage along the closed circuit path C, defined by four intermediate edges, which in turn are defined by the four intermediate nodes $\vec{x}^{\text{mid}}$ associated with each face. Specifically, we can compute the flux change as

$$\Delta b_i = -\sum_{j=1}^{4} \left(\vec{x}_{j+1}^{\text{mid}} - \vec{x}_j^{\text{mid}}\right) \cdot \left(\frac{(\vec{u}_j \times \vec{B}|_{\vec{x}=\vec{x}_j^{\text{mid}}}) + (\vec{u}_{j+i} \times \vec{B}|_{\vec{x}=\vec{x}_{j+1}^{\text{mid}}})}{2} \right) \tag{66}$$

where the index j is cyclic (modulo 4). A detailed schematic representation of this is shown in Fig. 5. Each term in the sum of (66) is a line integral of the voltage along one of the intermediate edges. This integral is computed with the trapezoid rule. A critical point is that this computation requires evaluation of the magnetic flux density $\vec{B}$ at the intermediate nodes; since the $H(Div)$ representation of $\vec{B}$ is discontinuous at nodes, some interpolation may be necessary if $\vec{x}^{\text{mid}}$ does not lie entirely within an old element, this is discussed in the next section.

The flux update of (66) relies on defining a circulation around the four intermediate edges. The direction of the circulation will determine the sign of Δb. Either of both directions can be used; however, it is imperative that the choice is made consistently in order to compute b_i^{new} for each face. On a general unstructured grid, it can become difficult to enforce such a rule, especially if one has no control over the source of the mesh topology. As such, a more robust (and ultimately more revealing) method for updating the fluxes can be obtained by considering the rectangular topological derivative matrix $\mathbf{K}^{12}$ of (43) which is a sparse rectangular matrix representing an incidence map between edges and faces of a mesh. The first step is to introduce an edge-based array $\mathbf{e}'$ representing the edge-based flux contributions defined by a line integral along that edge. For every edge in the mesh, we have

$$\mathbf{e}'_j = \left(\vec{x}_b^{\text{mid}} - \vec{x}_a^{\text{mid}}\right) \cdot \left(\frac{(\vec{u}_a \times \vec{B}|_{\vec{x}=\vec{x}_a^{\text{mid}}}) + (\vec{u}_b \times \vec{B}|_{\vec{x}=\vec{x}_b^{\text{mid}}})}{2} \right) \tag{67}$$

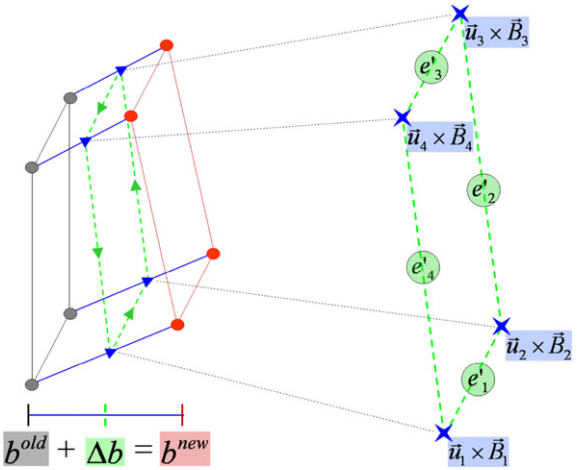

Fig. 5. Schematic diagram detailing the process of computing the update voltage along the closed circuit C defined by four intermediate edges. Given the values of $\vec{u} \times \vec{B}$ at each intermediate node, a voltage contribution from each intermediate edge can be computed. The appropriately signed sum of each edge contribution is the flux change for the face.

where the generic integers a and b denote the unique integer IDs of the intermediate mesh nodes associated with edge j such that $a < b$. Therefore, the direction of the line integral is uniquely defined according to a global standard like that originally proposed in [32] (i.e. the line integral path is always from the node with low integer ID to the node with high integer ID). We can now write the flux update in terms of global mesh arrays as

$$\mathbf{b}^{\text{new}} = \mathbf{b}^{\text{old}} - \mathbf{K}^{12}\mathbf{e}' \tag{68}$$

Written in this form, it is clear that the flux update method will preserve the solenoidal nature of the magnetic field. Taking the discrete divergence of (68) yields

$$\mathbf{K}^{23}\mathbf{b}^{\text{new}} = \mathbf{K}^{23}\mathbf{b}^{\text{old}} + \mathbf{K}^{23}(\mathbf{K}^{12}\mathbf{e}') = 0 \tag{69}$$

Therefore, the divergence-free constraint is satisfied to machine precision for every mesh relaxation step. Compare the flux update of (68) to the discrete Faraday's law of (48).

5.3. Patch recovery process for nodal magnetic field representation

In order to compute the edge-based voltage contributions of (67), we need to evaluate the $\vec{B}$ field at the points $\vec{x}^{\text{mid}}$. By virtue of the continuous remap approximation, the points $\vec{x}^{\text{mid}}$ are guaranteed to lie inside (or possibly on the side of) an upwind element of the old (or Lagrangian) mesh. An example of this is depicted in Fig. 6.

Once the upwind elements are known for each intermediate node, we can use a finite element representation to evaluate $\vec{B}$ inside of the upwind element at the location of the intermediate node $\vec{x}^{\text{mid}}$. However, we cannot use the face representation of (35) since, by construction, this representation is tangentially discontinuous across element boundaries. Instead, we perform a type of patch recovery to obtain a fully continuous (or smooth) representation of $\vec{B}$ which we will denote as $\vec{B}^{\text{avg}}$. We define the smooth representation as

$$\vec{B}^{\text{avg}} = \sum_{i=1}^{8} \sum_{j=1}^{3} b_{i,j}^{\text{avg}} \vec{V}_{i,j} \tag{70}$$

This representation has 24 degrees-of-freedom, corresponding to three vector components located at each of the 8 element nodes, and is fully continuous at element boundaries. This nodal vector field representation is equivalent to the tri-linear interpolation commonly used for FEM discretization of fluid velocities. The vector

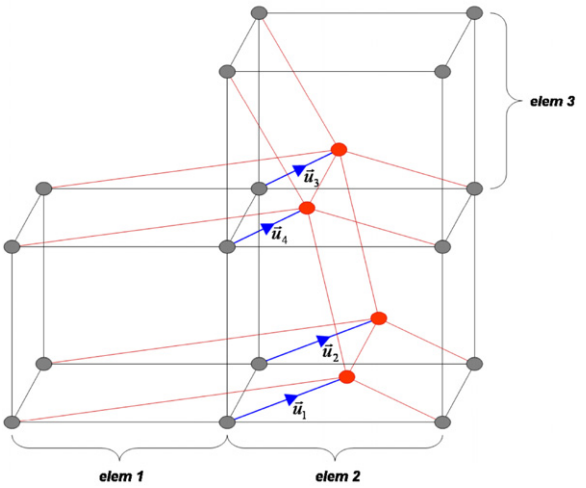

Fig. 6. Schematic diagram depicting the upwind locations of the intermediate mesh nodes $\vec{x}_i^{\text{mid}} = \vec{x}_i^{\text{old}} + \frac{1}{2}\vec{u}_i$. In this example, $\vec{x}_1^{\text{mid}}$ and $\vec{x}_2^{\text{mid}}$ lie in element 2 of the old mesh while $\vec{x}_3^{\text{mid}}$ and $\vec{x}_4^{\text{mid}}$ lie in element 3 of the old mesh

valued basis functions $\vec{V}_{i,j}$ can be viewed as three sets of 0-form (or scalar nodal) basis functions, one for each component of the vector field.

There are several options for computing the degrees-of-freedom $b_{i,j}^{\text{avg}}$ for this representation. The simplest and most efficient method is to first compute a cell centered value of the magnetic field for every element in the mesh using the face based representation of (35), then to apportion a volume weighted average of this value to each node. This is a cumulative process, any given node will receive a contribution from every element it is connected to. This cumulative nodal value is then divided by a "nodal volume." Alternatively, it is possible to employ a local polynomial patch recovery process, or a global least squares process. But these alternative approaches are expensive and are not evaluated here.

5.4. Algebraic flux correction for magnetic shocks

The transport method of Section 5.2 is second order accurate and will therefore exhibit non-monotonic solution behavior (aka spurious oscillations, "ringing", or overshoots and undershoots) for solutions with discontinuities or shock fronts. We must impose a form of limiting that will suppress the non-monotonic solution behavior. Limiting schemes for the scalar advection equation are prevalent and well understood as a result of many years of research in the computational fluid dynamics (CFD) community [33]. Nevertheless, the design of genuinely multidimensional schemes for finite element discretizations on unstructured meshes has proved to be a particularly challenging task [21]. Here, we derive a limiting procedure that is based upon our topological curl operator $\mathbf{K}^{12}$, and hence is inherently three-dimensional and unstructured. Furthermore, to our knowledge there is no published method for limiting the vector valued magnetic advection equation on a general unstructured grid.

The limiting procedure can be interpreted as reducing a high order numerical method to first order accuracy in the vicinity of a sharp discontinuity while maintaining the high order accuracy in the remaining regions where the solution is smooth. This requires a numerical procedure for detecting a shock (i.e. a "smoothness sensor") and a procedure for limiting (or reducing the order of) the advection update method by the proper amount to prevent spurious overshoots and undershoots. The theoretical foundations of this process were originally developed for 1D finite difference solutions to scalar conservation laws [34,35], where the notion of a Total Variation Diminishing (TVD) method was introduced in order to guarantee a monotonic solution. As pointed out in [21], the generalization of the TVD criterion to finite element discretizations on 3D

unstructured grids is the so-called local extremum diminishing (LED) criterion [36]. The LED criterion is an algebraic statement which enforces the rule that local solution maxima cannot increase (thereby preventing spurious *overshoots*) and local minima cannot decrease (thereby preventing spurious *undershoots*). In this section, we follow the algebraic approach of [21] of modifying the discrete magnetic transport method so as to render the discretization local extremum diminishing.

Suppose we apply a limiting (or correction) term to the local face flux update equation (65) of the form

$$b_i^{\lim} = b_i^{\text{old}} + \theta_i \Delta b_i$$

where the correction term θ_i has been introduced for each face flux. When $\theta_i = 1$, no limiting is performed and the flux update method is second order accurate. Now suppose we had a shock detector or smoothness sensor which could tell us whether or not the resulting face flux would result in a spurious overshoot or undershoot in the computation of $\vec{B}$. Our goal then is to compute the value of θ_i which would correct the flux update and prevent this from happening. However, we are immediately confronted with a problem. We cannot simply change the value of each face based flux independently, as this will clearly destroy the discrete divergence-free property which we have worked so hard to obtain.

The key to overcoming this obstacle is to limit the edge-based voltages rather than the face based fluxes. We therefore propose a divergence preserving limited update method of the form

$$\mathbf{b}^{\lim} = \mathbf{b}^{\text{old}} - \mathbf{K}^{12}(\boldsymbol{\theta}\mathbf{e}') \tag{71}$$

Since we are limiting (or correcting) independent edge-based voltages, the update of (71) is guaranteed to be divergence preserving to machine precision. The general rule of thumb is that the edge-based voltages which border the shock front are the ones responsible for the spurious overshoots/undershoots in the magnetic flux and must therefore be limited. However, in order to determine which edges are on the shock front and how much they need to be limited by, we need information from the face based fluxes, since we are ultimately concerned with obtaining a limited value of the discrete magnetic flux density $\vec{B}$ which is a face based quantity. An overview of this process is presented in Fig. 7.

The details of the process can be broken down into four steps:

- *Step 1* – Compute the unlimited flux change via (68)
- *Step 2* – Compute the face limited flux change via Algorithm 1

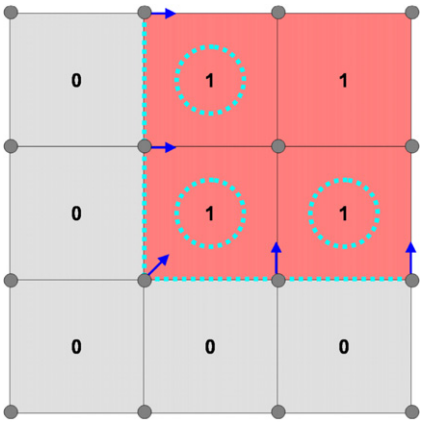

Fig. 7. Consider a patch of element faces in the presence of a discontinuous magnetic field oriented out of the page. In this example, the top four corner faces have unit magnetic flux (1's) while the remaining faces have no flux (0's). If the nodes bordering the discontinuity were to move in the direction indicated, the unlimited flux update method would generate spurious overshoots in the three faces bordering the shock front (indicated with circles). These are the faces that require flux correction. To compute the divergence preserving flux correction, we limit the voltages on the edges which border the shock front.

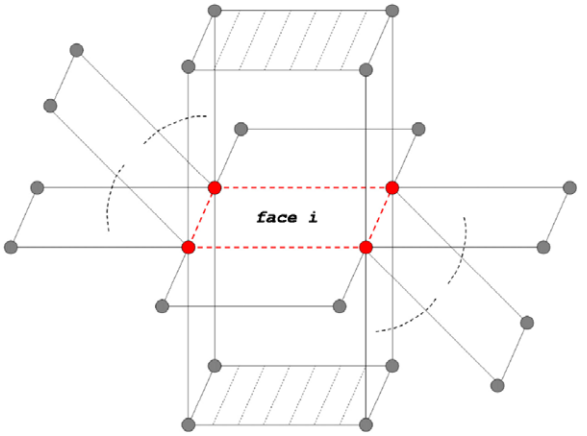

Fig. 8. Topological data structure used to detect discontinuities in face based fluxes. Note that the top and bottom faces are not used.

- *Step 3* – Loop over limited faces to determine the edges which lie along the shock front and compute the edge limiting factor via Algorithm 2
- *Step 4* – Compute the edge limited flux change by taking the limited curl via (71)

We begin by computing the unlimited flux change via the algebraic constrained transport update of (68). Next, we compute a face limited flux change $\overline{\Delta b}$ by searching the topological data structure of Fig. 8 using the method outlined in Algorithm 1. In other words, we check to see if the unlimited flux change will result in an overshoot or undershoot by searching all of the connected faces; if so, then we simply compute the limited value to be the maximum/minimum connected value of the data structure. The resulting limited flux change $\overline{\Delta b}$ could be used in the algebraic constrained transport update of (68) and it would result in a properly limited $\vec{B}$ field; however the resulting discrete $\vec{B}$ field would no longer be divergence free. Therefore, the next step is to determine which edges in the limited faces are responsible for the over/undershoots. In logical u–v (or reference) space, we can decompose a face into two sets of edges: the two edges parallel to the local u-direction and the two edges parallel to the local v-direction. Given a face that lies on the border of the shock front, our goal is to compute which edge in each of these two sets requires limiting. This is accomplished by computing the "edge curl" using the topological data structure of Fig. 9 according to the method outlined in Algorithm 2. Edges which border the shock front will have a large "edge curl" relative to the opposite edge in logical space.

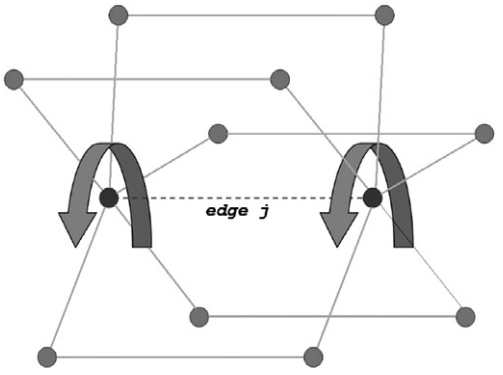

Fig. 9. Topological data structure used to detect discontinuities in edge-based voltages.

Algorithm 1: Face Based Shock Detection and Flux Correction

for $i = 1$ ***to*** *NumFaces* **do**
  //Compute the area of the old and new faces given their nodal coordinates
  A_i^{old} = `ComputeFaceArea(`*OldNodes*`);`
  A_i^{new} = `ComputeFaceArea(`*NewNodes*`);`
  $|B|^{old} = \frac{b_i}{A_i^{old}}$
  $|B|^{new} = \frac{b_i + \Delta b_i}{A_i^{new}}$
  //Initialize the max and min values
  $Max = Min = |B|^{old}$
  for $j = 1$ ***to*** 4 **do**
    for $k = 1$ ***to*** *NumConnectedFaces* **do**
      //Calculate local face extrema connected to edge j
      $A_{j,k}^{con}$ = `ComputeFaceArea(`*OldNodes*`);`
      $|B|^{con} = \frac{b_{j,k}}{A_{j,k}^{con}}$
      if $|B|^{con} > Max$ **then**
        $Max = |B|^{con}$
      else if $|B|^{con} < Min$ **then**
        $Min = |B|^{con}$
    end
  end
  //If face i is a local min or max, compute the limited value
  if $|B|^{new} > Max$ **then**
    $b_i^{lim} = Max * A_i^{new}$;
  else if $|B|^{new} < Min$ **then**
    $b_i^{lim} = Min * A_i^{new}$;
  $\Delta \bar{b}_i = b_i^{lim} - \Delta b_i$
end

For example, suppose we have identified a face that requires limiting via the method outlined in Algorithm 1. Furthermore, suppose that the 4 edges in this face are such that edges 1 and 3 are parallel to the local u-direction and edges 2 and 4 are parallel to the local v-direction. We now compute the edge curl for each of the 4 local edges and compare relative magnitudes between pairs $1,3$ and $2,4$. By doing so, we determine which, if any, of the 4 edges lies on the shock front. Suppose we have determined that local edge 3 (from the u set) and local edge 4 (from the v set) both have a large edge curl with respect to their counterparts. The unlimited flux change for this face is given by

$$\Delta b = -(K_1 e_1 + K_2 e_2 + K_3 e_3 + K_4 e_4)$$

We now introduce the edge limiting factor θ into the equation for the face limited flux change $\overline{\Delta b}$ as computed by the method outlined in Algorithm 1

$$\overline{\Delta b} = -(K_1 e_1 + K_2 e_2 + \theta(K_3 e_3 + K_4 e_4))$$

Solving for θ, we get

$$\theta = \frac{\Delta b - \overline{\Delta b}}{K_3 e_3 + K_4 e_4} + 1$$

We apply the same limiting factor θ to each of the two global edges corresponding to the local edges 3 and 4. This process is performed for every limited face until the global edge-based array $\boldsymbol{\theta}$ is formed. Once this value has been computed for each limited edge, we can then compute the edge limited flux change via (71). The details for identifying the edges in a face which require limiting are outlined in Algorithm 2. Note that all of the information required for the data structures depicted in Figs. 8 and 9 is encoded in the topological derivative

matrix $\mathbf{K}^{12}$ of (43), since this purely topological quantity is simply an incidence map which designates the connectivity between edges and faces.

Algorithm 2: Edge Based Shock Detection and Voltage Correction

Input: $reltol$, Δb, $\bar{\Delta b}$
for $i = 1$ ***to*** *NumLimitedFaces* **do**
  //Compute "edge-curl" for each edge
  for $j = 1$ ***to*** *4* **do**
    $\Delta e_j = 0$
    for $k = 1$ ***to*** *NumConnectedFaces* **do**
      //Compute the curl of $\vec{B}$ around edge j
      $A_{j,k}^{con}$ = `ComputeFaceArea(OldNodes);`
      $|B|^{con} = \frac{b_{j,k}}{A_{j,k}^{con}}$
      $\Delta e_j = \Delta e_j + K_{j,k} * |B|^{con}$
    end
  end
  //Check for shock in local u-direction
  $r1 = \frac{|\Delta e_1|}{|\Delta e_1|+|\Delta e3|}$;
  $r3 = \frac{|\Delta e_3|}{|\Delta e_1|+|\Delta e3|}$;
  Δe_u = `Max(r1,r3);`
  if $\Delta e_u > 0.5 + reltol$ **then**
    if $r1 > r3$ **then**
      Limit edge 1
    else if $r3 > r1$ **then**
      Limit edge 3
  else
    No limiting for u edges
  //Check for shock in local v-direction
  $r2 = \frac{|\Delta e_2|}{|\Delta e_2|+|\Delta e4|}$;
  $r4 = \frac{|\Delta e_4|}{|\Delta e_2|+|\Delta e4|}$;
  Δe_v = `Max(r2,r4);`
  if $\Delta e_v > 0.5 + reltol$ **then**
    if $r2 > r4$ **then**
      Limit edge 2
    else if $r4 > r2$ **then**
      Limit edge 4
  else
    No limiting for v edges
  //Given IDs of limited edges, compute edge correction factor
  `ComputeTheta(`$\Delta b, \bar{\Delta b}$`);`
end

6. Numerical verification experiments

In this section we present a series of numerical experiments which are designed to verify the individual components of our operator-split discretization of MHD. For the first two examples of Section 6.1 and 6.2, we need to solve the linear system of (47) where the right hand side consists of an edge-based finite element mass and stiffness matrix. For these examples we use a simple diagonally scaled pre-conditioned conjugate gradient

(PCG) method which is sufficient for most applications. However we should point out that more advanced and efficient methods for solving linear systems arising from mixed finite element discretizations using $H(Curl)$ and $H(Div)$ basis functions exist, such as those described in [23,37].

6.1. Electromagnetic diffusion in a coaxial cylinder at rest

The purpose of this computational experiment is to verify the discrete electromagnetic diffusion operator L_σ of our operator splitting of the induction equation. Since we are ignoring the advection operator, we choose a simple test problem in which the conducting materials are at rest (i.e. $\vec{v}' = 0$). Furthermore, this test is designed to validate our approach for treating electromagnetic diffusion in highly heterogeneous conducting regions (i.e. regions consisting of conductors immersed in insulating vacuum like regions) using only a voltage source boundary condition. This test problem was developed in the spirit of the first test problem from [23]; however in this case we drive the problem with a voltage source boundary condition and we have an analytic solution to compare with.

In this computational experiment we apply a 1 V potential difference across the ends of a conducting coaxial cylinder and compute the steady state conduction current and magnetic field via the mixed FEM formulation of Section 3.3. The electrical resistance of the coaxial cylinder is given by

$$R = \frac{l}{\sigma A} \tag{72}$$

where l is the length of the coaxial cylinder and A is the cross sectional surface area of the coaxial cylinder determined by its inner radius R_i and outer radius R_o. The potential difference across the coaxial cylinder will result in a steady state conduction current density $\vec{J} = \sigma\vec{\nabla}\phi$ where ϕ is the scalar potential inside the conductor. We fix the geometry and conductivity σ of the problem such that the total resistance is 1 Ω and the total conduction current $I = 1$ A. To facilitate the magnetic fields in the vacuum around the cylinder, the computational domain is a cylinder of radius $R_b = 2R_\mathrm{o}$ and length l oriented along the $\hat{z}$ direction, divided into two material regions as shown in Fig. 10. The cylinder is assigned a conductivity value $\sigma_\mathrm{c} = 2$ S/m while the vacuum region is assigned a very small conductivity value $\sigma_\mathrm{vac} = 10^{-7}\sigma_\mathrm{c}$. The computational mesh consists of 5760 hexahedral elements.

The steady state magnetic field will have azimuthal symmetry which can be determined analytically from Ampere's law

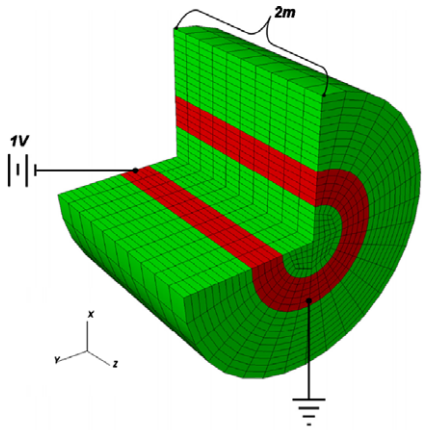

Fig. 10. Computational domain for conducting coaxial cylinder immersed in a vacuum like material.

$$B_\theta(r) = \frac{\mu I_{\text{enc}}(r)}{2\pi r} \tag{73}$$

$$I_{\text{enc}}(r) = \begin{cases} 0 & r \leqslant R_{\text{i}} \\ I\frac{r^2 - R_{\text{i}}^2}{R_{\text{o}}^2 - R_{\text{i}}^2} & R_{\text{i}} \leqslant r < R_{\text{o}} \\ I & R_{\text{o}} < r \end{cases} \tag{74}$$

For the scalar potential solve of (46), we apply the inhomogeneous Dirichlet boundary condition $\phi = +1$ at the surface $z = 0$ and $\phi = 0$ at the surface $z = l$. For the discrete Ampere solve of (47), we apply the homogeneous Dirichlet boundary condition $\hat{n} \times \vec{E}^{\text{ind}} = 0$ over the entire surface of the problem domain. For both solves, a simple diagonally scaled PCG method with a residual error tolerance of 10^{-10} is used. We run the problem for a total time $t_{\text{fin}} = 3\tau$ where τ is one diffusion time constant such that $\tau = \sigma\mu(R_{\text{o}} - R_{\text{i}})^2$, this will ensure that the fields reach steady state. We use a fixed time step $\Delta t = \frac{t_{\text{fin}}}{100}$. In Fig. 11 we plot the scalar potential as well as the steady state conduction current density $\vec{J} = \sigma\vec{E}$ and magnetic field $\vec{B}$. In Fig. 12 we plot the analytic solution for the azimuthal magnetic field as a function of radius and compare it with our mixed FEM solution. The total current can be computed by numerically integrating the z-component of the current density at any plane normal to the z-axis. In Fig. 13 we plot the total computed current as a function of time and verify that it reaches its correct steady state value of $I = 1$ A.

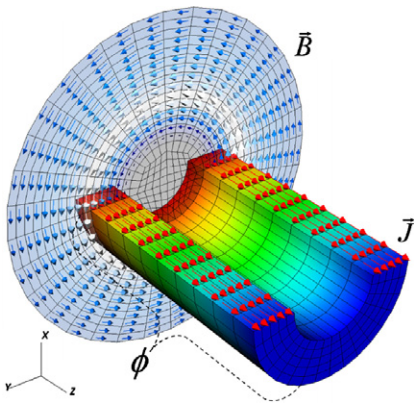

Fig. 11. Computed steady state electromagnetic fields for the conducting coaxial cylinder problem.

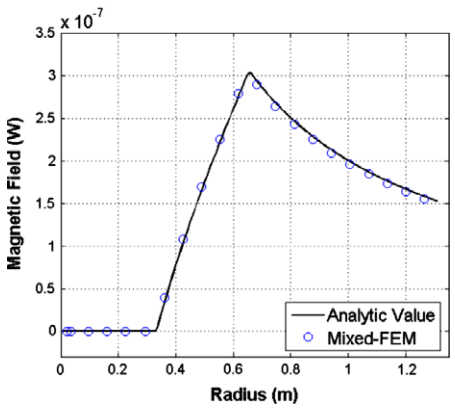

Fig. 12. Azimuthal magnetic field as a function of radius at the final time step.

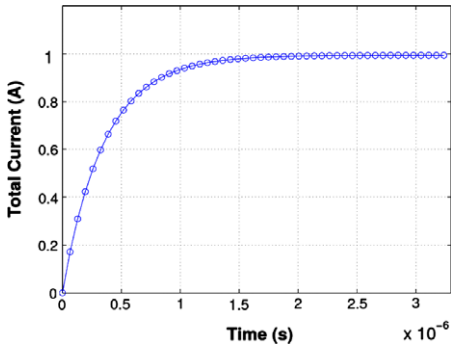

Fig. 13. Total computed current as a function of time. Note that the total current reached the correct steady state value.

6.2. MHD wave propagation in an ideal gas

In this computational experiment we verify the coupling of the electromagnetic force to the equations of Lagrangian motion, which in turn will verify our Lagrangian treatment of the advection operator L_v. Our goal for this experiment is to launch waves and verify their computed velocity. A simple way to do this is to fix the velocity of the wave to some desired value, then to scale the domain size and total time for the problem such that the wave front just reaches the end of the computational domain at time $t = t_{\text{fin}}$. We consider the case of a rectangular "slab" mesh (i.e. one element thick in the z-direction) of dimension $2L_x$ by $2L_y$ centered at the origin $x = y = 0$ representing an ideal gas. We use a simple gamma-law model for the EOS of the gas given by

$$P = (\gamma - 1)\frac{\rho}{\rho_0}E \tag{75}$$

For an ideal monatomic gas, $\gamma = \frac{5}{3}$.

For reference, we first consider the case of a pure sound (or acoustic) wave. This is a purely hydrodynamic calculation and does not involve any electromagnetic properties. The sound speed is determined by the relation

$$v_s = \sqrt{\frac{\gamma P}{\rho}} \tag{76}$$

We choose a sound speed $v_s = 0.5$ m/s and an initial density $\rho = 1.0$. We excite the wave by applying a time dependent velocity perturbation to a face in the mesh that is normal the x-axis and located at the center of the mesh. Specifically, we have

$$\vec{v}_{\text{per}} = A\cos(\omega t)\hat{x} \tag{77}$$

which gives us a displacement perturbation equal to

$$\vec{x}_{\text{per}} = \frac{A}{\omega}\sin(\omega t)\hat{x} \tag{78}$$

We set the velocity perturbation amplitude to be very small, specifically $A = 10^{-2}$. This keeps the relative change in pressure $\vec{\nabla}P/P$ small enough to eliminate the need for an artificial viscosity (i.e. there are no shocks in the problem). For this experiment, the computational domain has dimensions $L_x = L_y = v_s t_{\text{fin}}$. We set the total time to be $t_{\text{fin}} = 1$ s. This implies that the perturbation velocity (and displacement) will oscillate for two full periods during the simulation. In this example, the velocity perturbation should propagate outward from the center of the mesh via compression and rarefaction waves traveling in the x-direction at the sound speed v_s as shown in Fig. 14.

Now we consider the case of an MHD wave. We begin by applying an initial magnetic field to the problem domain oriented in the y-direction such that $\vec{B} = B_y\hat{y}$. For this case, we add an electrical conductivity to the ideal gas. We set the conductivity very high ($\sigma = 10^8$ S/m) so that the conducting gas effectively has no

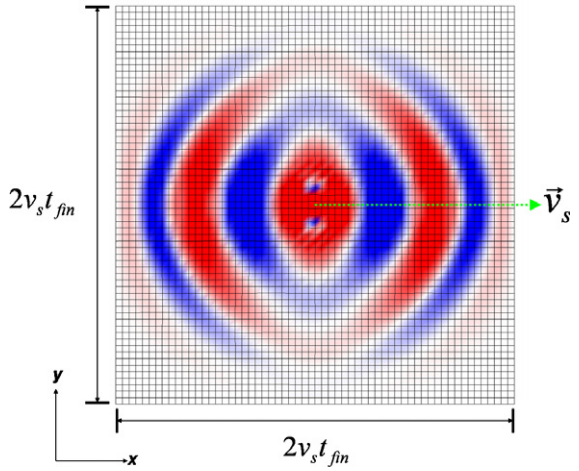

Fig. 14. Pseudocolor plot of velocity wave at time $t = t_{\text{fin}}$ for the case of a pure sound (or acoustic) wave. Since the initial velocity perturbation is oriented in the x-direction and the ideal gas has no strength, the velocity perturbation travels in the x-direction via compression waves at the sound speed v_{s}. The computational domain is a single element thick "slab" of dimension $2v_{\text{s}}t_{\text{fin}}$ by $2v_{\text{s}}t_{\text{fin}}$.

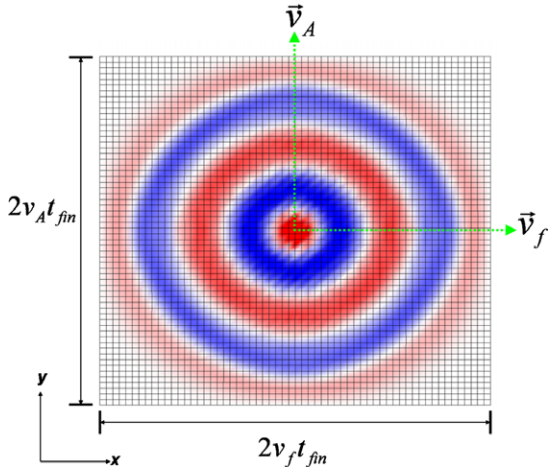

Fig. 15. Pseudocolor plot of velocity wave at time $t = t_{\text{fin}}$ for the case of an MHD wave, consisting of a fast (or magnetosonic) wave and a shear (or Alfven) wave. This is accomplished by adding a $\vec{B}$ field oriented in the y-direction to the problem. The velocity perturbation now travels in the x-direction via compression waves at the fast (or magnetosonic) speed v_{f} and in the y-direction via shear waves at the Alfven speed v_{A}. The computational domain is a single element thick rectangular "slab" of dimension $2v_{\text{f}}t_{\text{fin}}$ by $2v_{\text{A}}t_{\text{fin}}$.

electrical resistance. This implies that our MHD diffusion equations are effectively loss-less for the time scales we are considering, meaning we are in the ideal MHD limit with the frozen-in-flux condition. We apply the same time dependent velocity perturbation in the x-direction to a face in the middle of the mesh. Because the magnetic field is "frozen in" to the material, it will be dragged along with the material as it moves. However, the $\vec{J} \times \vec{B}$ restoring force will work to resist this motion and effectively add strength to the gas in the from of the Maxwell stress tensor. We now expect to see two types of waves, a fast (or magnetosonic) compression wave traveling in the x-direction at the speed v_{f} and a shear Alfven wave traveling in the y-direction at the speed v_{A} as shown in Fig. 15. The shear Alfven wave velocity is given by

$$v_{\text{A}} = \frac{|\vec{B}|}{\sqrt{\rho}} \tag{79}$$

while the fast (or magnetosonic) wave velocity is given by

$$v_f = \sqrt{v_s^2 + v_A^2} \tag{80}$$

We fix the Alfven speed to be $v_A = 1.0$ m/s (which subsequently determines B_y) and keep the sound speed fixed at $v_s = 0.5$ m/s as before; this implies the magnetosonic speed will be $v_f = \frac{\sqrt{5}}{2}$ m/s. For the linear solve of (47) we apply the homogeneous boundary condition $\hat{n} \times \frac{1}{\mu}\vec{B} = 0$ to the x and y boundary planes (i.e. we are enforcing the constraint $\vec{v} \times \vec{B} = 0$ on the perimeter of the mesh). Since this is a three-dimensional problem (i.e. it has finite depth in the z-direction), we apply the homogeneous boundary condition $\hat{n} \times \vec{E} = 0$ on the top and bottom of the mesh defined by the planes $z = z_{\min}$ and $z = z_{\max}$. The linear solve is performed using a diagonally scaled PCG method with a residual error tolerance of 10^{-8}.

In Fig. 16 we plot the magnetic field vectors and magnitude along with the computational mesh using an exaggerated displacement (scale factor of 300) to emphasize the characteristics of the Lagrangian calculation. Note how the mesh lines move with the magnetic field. In addition, note how the magnetic field lines compress and expand in the y-direction and undulate due to shearing motion in the x-direction. In Fig. 17 we track the velocity wave amplitude to its first peak value for six different spatial locations along the Alfven wave axis (the y-axis). Note how the velocity wave amplitude decays at a rate proportional to $\frac{1}{\sqrt{r}}$, in direct agreement with the expected results for wave propagation in two dimensions (recall the Green's function for 2D wave propagation is proportional to $\frac{1}{\sqrt{r}}$). The peak to peak separation of the velocity wave amplitude at different points in time can be used to measure the instantaneous numerical velocity of the wave as shown in Fig. 18. Note how the numerical Alfven wave travels at a non-constant rate which is slower than the expected constant rate, indicating the effects of numerical dispersion.

Finally, we perform the Lagrangian MHD wave calculation on a very unstructured mesh to test the robustness of the numerical method. In Fig. 19 we compare the final results at time $t = t_{\text{fin}}$ for both mesh types, indicating that the proposed method can support MHD waves on highly unstructured grids with arbitrary connectivity.

6.3. Eulerian advection of smooth fields

The purpose of this computational experiment is to demonstrate the second order accuracy of the algebraic constrained transport method of (68), and therefore verify our Eulerian treatment of the advection operator L_v. We consider the case of a "smooth" (i.e. infinitely differentiable) magnetic field initially projected onto a

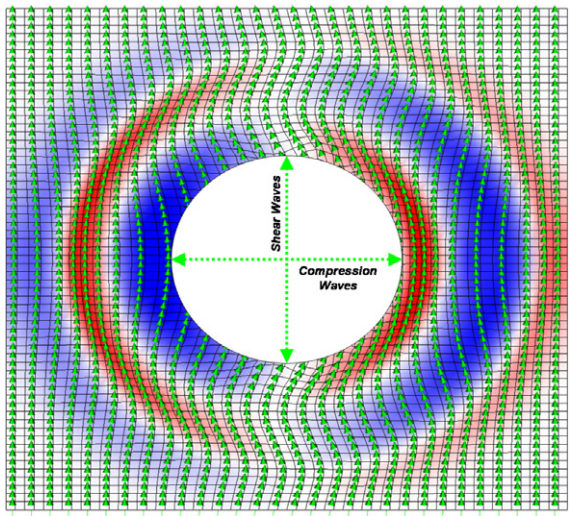

Fig. 16. MHD wave problem with exaggerated displacement (scale factor of 300) to emphasize the features of the Lagrangian calculation. Note how the mesh lines are parallel to the magnetic fields lines. The pseudocolor plot represents the magnitude of the magnetic field, $|\vec{B}|$. Here we can more clearly see the components of the Maxwell stress tensor in action. The pressure terms contribute to the peaks and nulls in the magnetic field magnitude along the y-direction due to compression waves while the deviatoric components give rise to the shearing motion of the vector field along the x-direction.

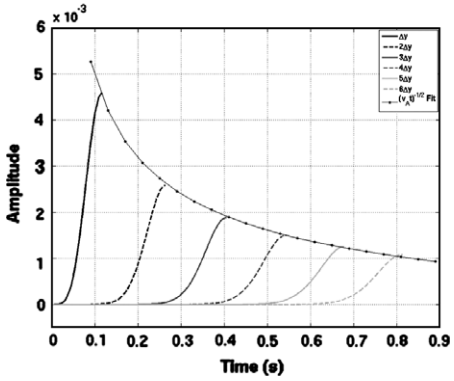

Fig. 17. Velocity wave amplitude as a function of time (up to first peak value) for six different spatial locations along the Alfven wave axis (y-direction) and a $\frac{1}{\sqrt{r}}$ fit to the amplitude.

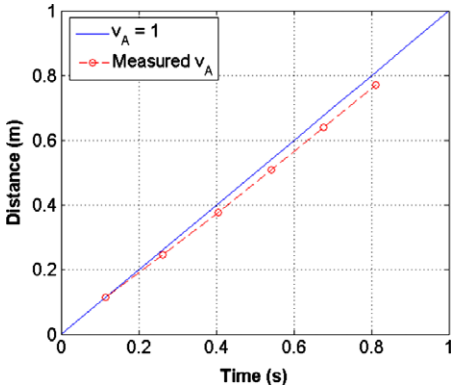

Fig. 18. Location of first peak value in velocity space for six different spatial locations, providing a measure of the instantaneous Alfven velocity as well as numerical dispersion.

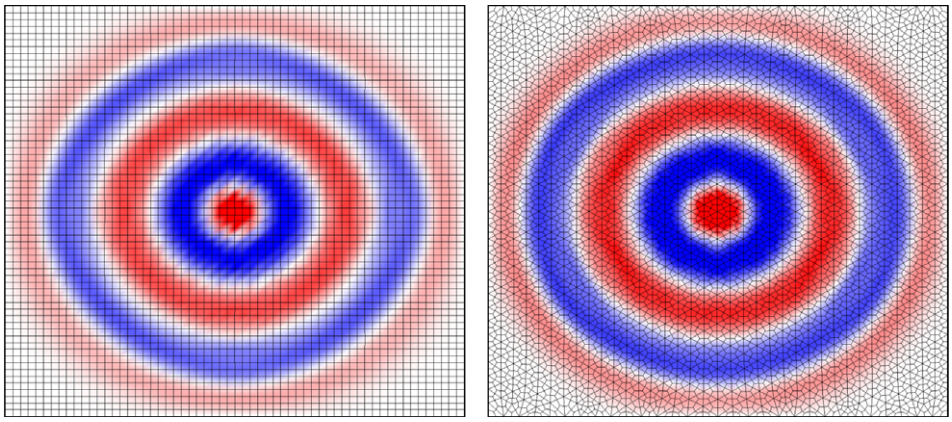

Fig. 19. Comparison of Lagrangian MHD wave results at time $t = t_{\text{fin}}$ for the case of a structured mesh (*left*) and a very unstructured mesh (*right*).

significantly distorted mesh. We then let the mesh relax to equilibrium while applying the constrained transport update of (68) at each mesh relaxation step to update the magnetic flux values. A sequence of images depicting this process is shown in Fig. 20. We consider a solenoidal $\vec{B}$ field that can be expressed as the curl

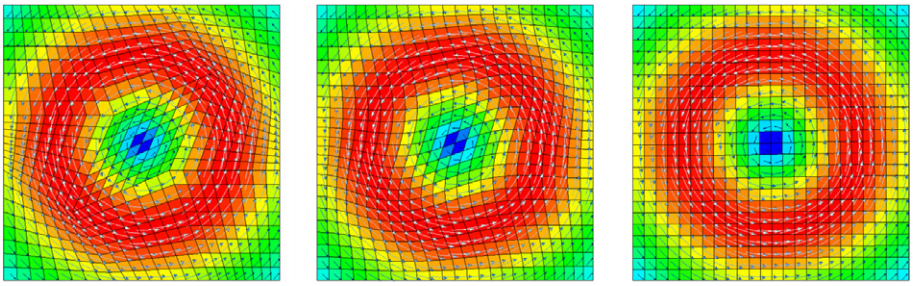

Fig. 20. A solenoidal magnetic field $\vec{B}$ is projected onto an initially distorted mesh (*left*). The mesh is then relaxed to equilibrium (*right*) via a sequence of steps (*middle*) while the magnetic flux is updated using the algebraic constrained transport method.

of a vector potential which represents a vector valued "Gaussian Hill" oriented in the $\hat{z}$ direction (i.e. out of the page)

$$\vec{A} = \exp(-\alpha(x^2 + y^2))\hat{z}$$
$$\vec{B} = \vec{\nabla} \times \vec{A}$$

The corresponding $\vec{B}$ field will have non-zero $\hat{x}$ and $\hat{y}$ components. We perform the calculation on a sequence of refined meshes and at each mesh relaxation step, the $L2$ finite element error is computed for each element, as shown in Fig. 21. Note that the convergence of the method indicates second order accuracy.

In addition, we compute the numerical energy stored in the magnetic fields at each mesh relaxation step as

$$e_{mag} = \mathbf{b}^T \mathbf{M}^2(\mu^{-1})\mathbf{b} \tag{81}$$

In Fig. 22 we plot the measured magnetic energy at each mesh relaxation cycle using three different limiting methods: no flux limiting (i.e. pure algebraic CT), face based flux limiting (i.e. non-divergence preserving) and edge-based flux limiting (i.e. divergence preserving). Strict conservation of energy is not explicitly built into the method, and Fig. 22 shows that some energy is lost during advection, even when no limiting is applied. The amount of lost energy is a function of the mesh, and as the mesh is refined the amount of lost energy decreases with second order convergence. The fact that our advection method conserves magnetic flux exactly and conserves magnetic energy approximately is consistent with the hydrodynamics advection in ALE3D which conserves momentum exactly and conserves energy approximately. It is also consistent with the interpretation of flux limiting as an artificial local diffusion. In Fig. 23 we plot the total magnetic charge at each mesh relaxation cycle using the same three limiting methods. As expected, the unlimited and the edge limited algebraic CT method preserves the divergence of the $\vec{B}$-field to machine precision, independent of the mesh distortion. Note also how the face limited approach destroys the $\vec{\nabla} \cdot \vec{B} = 0$ property as expected.

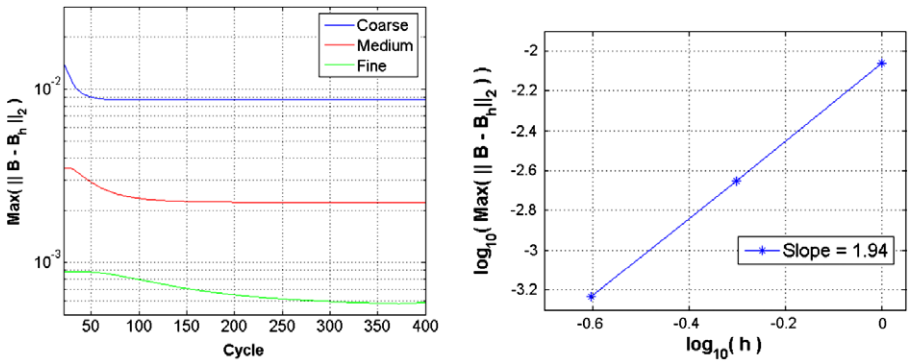

Fig. 21. Convergence analysis of finite element error for the magnetic advection equation using a coarse (144 element), medium (576 element) and fine (2304 element) hexahedral mesh. The maximum $L2$ error vs. mesh relaxation cycle is plotted *left* while the maximum $L2$ error vs. element size at the final cycle is plotted *right* on a log scale, indicating the convergence rate is second order.

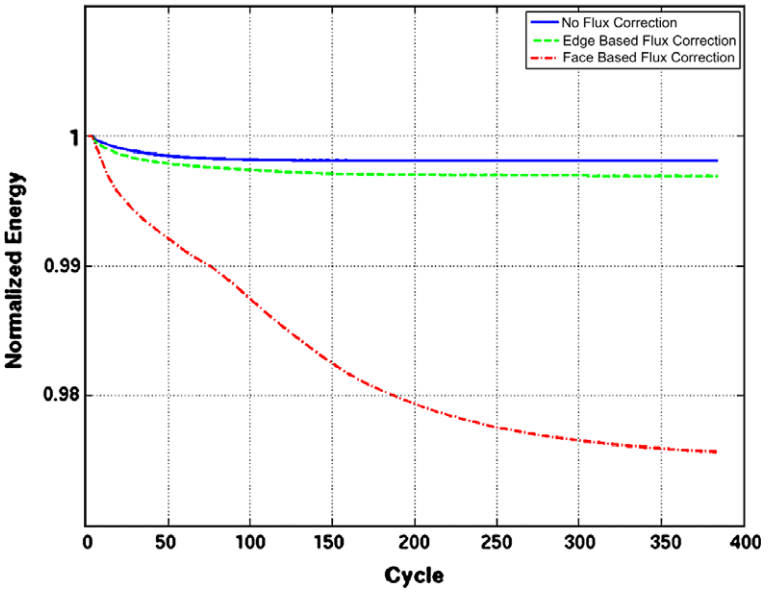

Fig. 22. Normalized magnetic energy at each mesh relaxation step using three different limiting methods.

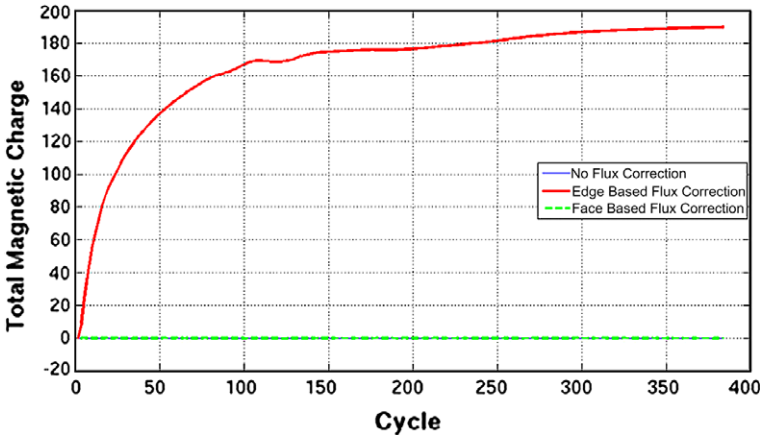

Fig. 23. Total magnetic charge at each mesh relaxation step using three different limiting methods.

6.4. Eulerian advection of discontinuous fields

In these computational experiments we verify the algebraic flux correction method by performing mesh relaxation over discontinuous fields and verifying that the LED criterion is satisfied. We begin with a classic one-dimensional advection problem adapted for our ALE treatment of magnetic flux. The computational domain and initial fields are depicted in Fig. 24. We project onto the region two different initial vector fields. The first is a smooth Gaussian hill for reference purposes while the second is a unit step function. We apply an initial mesh density gradient to the computational domain such that one end has a high concentration mesh elements while the other end has a low concentration of mesh elements. When mesh relaxation is applied, the mesh will flow from the high density region to the low density region with the non-uniform mesh velocity $\vec{v}_m$ until equilibrium is reached. This is equivalent to advecting the initial data in the opposite direction $-\vec{v}_m$. The results of this calculation with and without algebraic flux correction are shown in Fig. 25. Note how spurious oscillations occurs at the leading and trailing edge of the shock front when no flux correction is applied. These

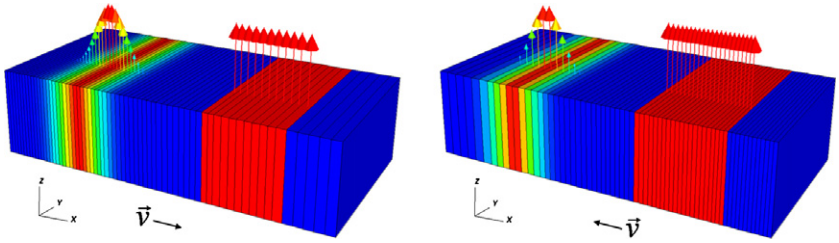

Fig. 24. Computational domain and initial field data for mesh relaxation over a smooth "Gaussian-hill" vector field and a discontinuous vector field. In both cases, the mesh is given an initial density gradient. When the mesh relaxation process is applied, the elements will flow from the region of high density to low density. In the first case (*left*) the mesh velocity will be from left to right while in the second case (*right*) the mesh velocity will be from right to left.

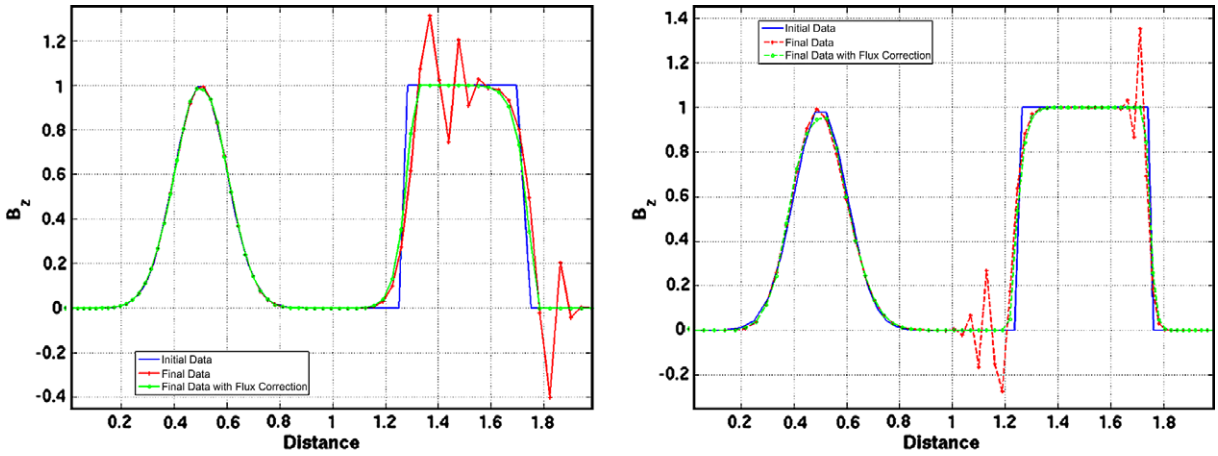

Fig. 25. Computational results for algebraic constrained transport of magnetic flux with and without algebraic flux correction for two different mesh velocity directions.

results are in direct agreement with numerical advection of a square pulse using the textbook second order accurate Lax–Wendroff method. Note how the flux corrected result has succeeded in suppressing the oscillations by directly enforcing the LED criterion. Furthermore, note that with and without flux correction, the results for the smooth function are in good agreement with the initial data. This indicates that the proposed algebraic flux correction method does not introduce excess diffusion for smooth fields.

In Fig. 26 we perform a similar experiment, except this time the computational domain is an unstructured cylinder mesh with a radial step function. Again, we apply an initial mesh density gradient to the computational domain. When mesh relaxation is applied, the mesh will flow radially inward/outward from high density region to the low density region with the non-uniform mesh velocity $\vec{v}_m$ until equilibrium is reached. The results of this calculation with and without algebraic flux correction are shown in Fig. 27. For the case of radial expansion, note how spurious undershoots in the magnetic flux density $\vec{B}$ occur at the shock boundary when no flux correction is applied while for the case of radial compression spurious overshoots occur. Note that in both cases, these spurious oscillations are suppressed when algebraic flux correction is applied.

6.5. Eulerian treatment of MHD wave propagation

As mentioned previously, it is possible to implement a pure Eulerian formulation of MHD by computing a Lagrange step followed by a complete remap step in which the mesh snaps back to its original configuration at every time step. This is how ALE3D performs pure Eulerian calculations. In this section we revisit the MHD wave problem from Section 6.2 which was solved in a pure Lagrangian fashion (i.e. the computational mesh flowed with the conducting material). Here we perform the same calculation using a pure Eulerian process,

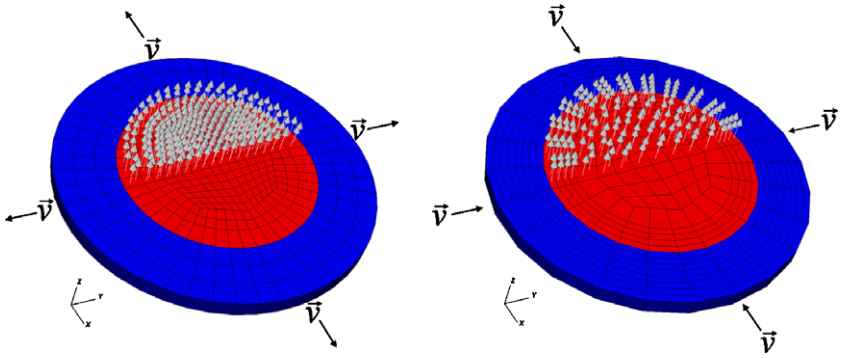

Fig. 26. Computational domain and initial field data for mesh relaxation over discontinuous vector field on an unstructured cylindrical mesh. In both cases, the mesh is given an initial density gradient. In the first case (*left*) the mesh will flow radially outward while in the second case (*right*) the mesh will flow radially inward.

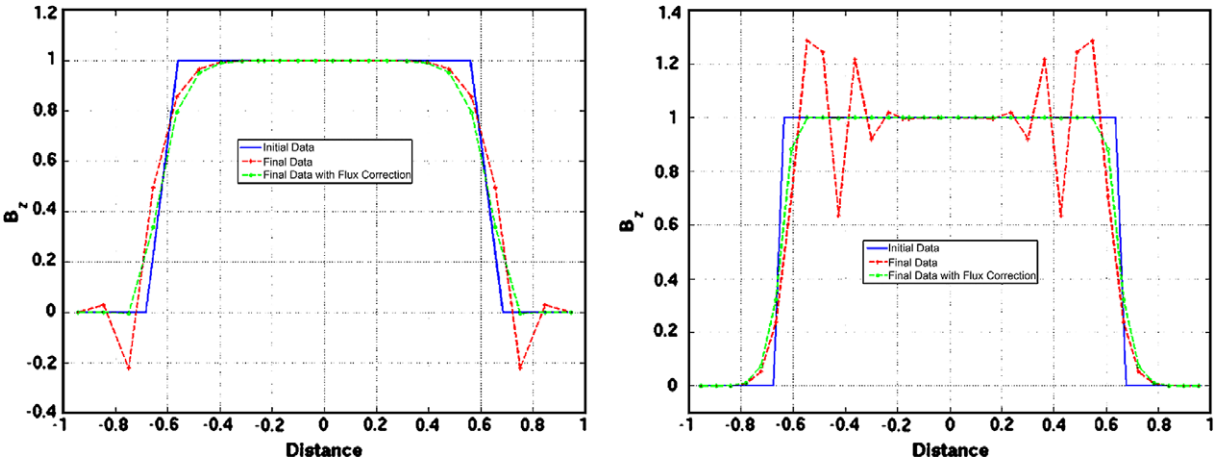

Fig. 27. Computational results for algebraic constrained transport of magnetic flux on an unstructured mesh with and without algebraic flux correction for two different mesh velocity directions.

meaning that at every time step, a Lagrange step is performed followed by an advective remap to the original unperturbed mesh. This is challenging test of the advection algorithm, as it is applied over and over again. In Fig. 28 we compare values for the amplitude of the magnetosonic wave at time $t = t_{\text{fin}}$ obtained in a pure

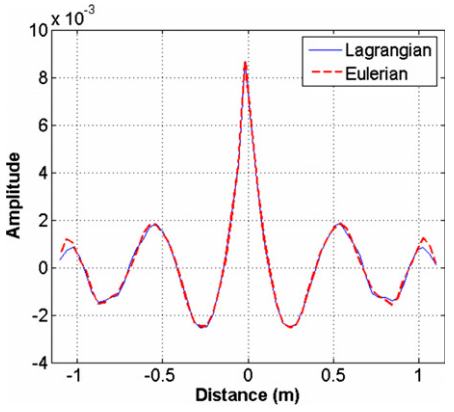

Fig. 28. Comparison of magnetosonic wave amplitude along the x-axis at time $t = t_{\text{fin}}$ for both Lagrangian and pure Eulerian calculations.

Eulerian fashion against the original Lagrangian results of Section 6.2. Note how the two are in excellent agreement.

7. Magnetic flux compression generation

As mentioned in the introduction, the primary motivation for this simulation capability is understanding the behavior of pulse power devices that involve gross deformation of solid metal. An Explosive Magnetic Flux Compression Generator (EMFCG) is an example of a device for which the electromagnetics, solid mechanics, and heat transfer are tightly coupled. The metal starts out solid, but may transition to liquid or even a gas/plasma state. In this section we present simulation results for the EMFCG proposed and tested in [38], the geometry is shown in Fig. 29. The generator is connected to a capacitor bank that provides the seed current, and to a detonation system for the high explosive. The generator is essentially a current amplifier, the input current is amplified in magnitude and compressed in time. The device operates as follows: at $t = 0$ the capacitor bank is switched, current flows into the coaxial device, and a large B-field is established between the conductors. A few microseconds later the high explosive is detonated and the thin copper outer conductor (the armature) is compressed. At some point the armature contacts the inner conductor (the stator) and at this point the magnetic flux is trapped between the conductors. As the compression continues, due to Lenz's law the current in the armature and stator and the load is amplified (in this simulation the load, connecting armature to stator, is a simple copper plate). The geometry is designed such that the device "zippers shut" generating a current pulse with a fast dI/dt. Snapshots of the geometry, current density, magnetic field, and electric field are shown in Figs. 30 and 31. Note that Fig. 30 corresponds to the "crowbar time", the time at which the armature contacts the stator and the magnetic flux is trapped.

One issue with EMFCG's is that the inductive electric field can become large, and if the field exceeds some critical value breakdown will occur and the device will be short circuited. This device developed a peak electric field strength of 10^5 V/cm, hence it is possible that an arc forms between the armature and stator. This peak field occurred late in time (after 80 ms) and hence will not significantly impact the current gain. Interestingly, the copper armature exceeds its melting point at approximately the same location and time.

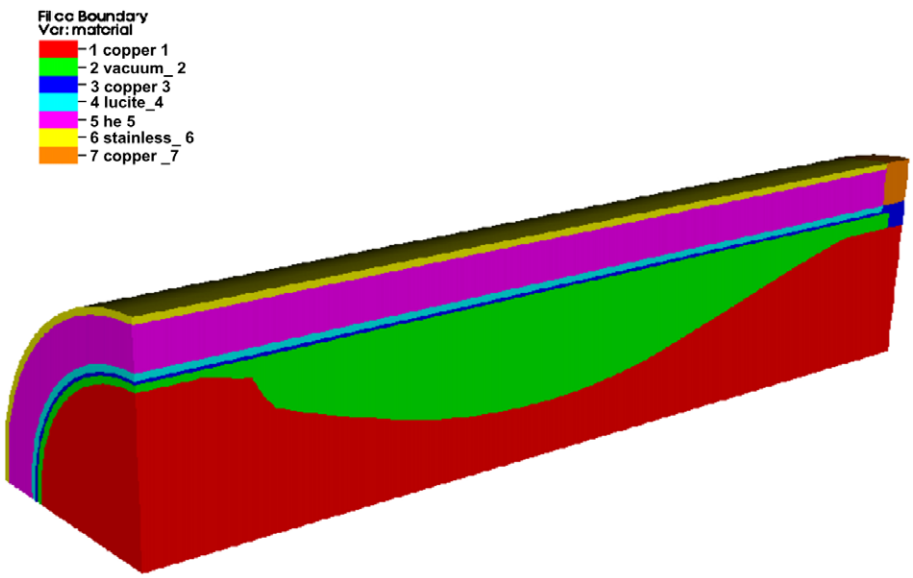

Fig. 29. Geometry of the Shearer coaxial explosive magnetic flux compression generator. This is 1/4 of the geometry. From inside to outside, the materials consist of copper, air, copper, Lucite, high explosive, and stainless steel.

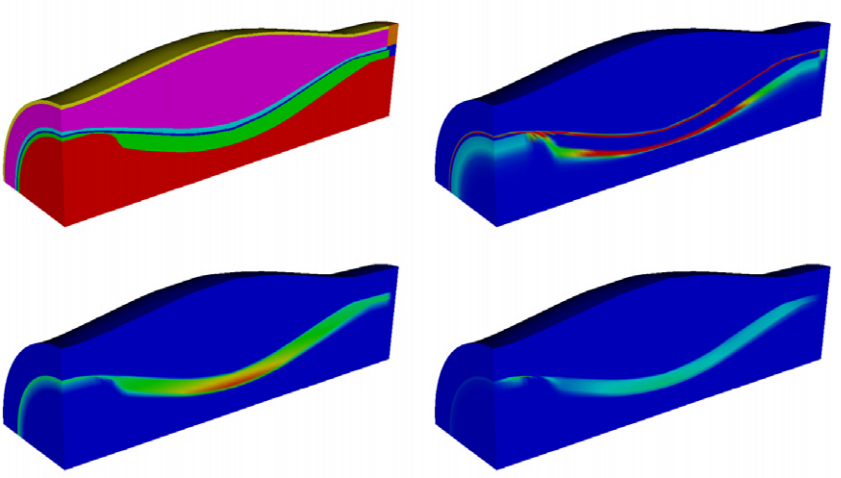

Fig. 30. Shearer generator results at $t = 61$ μs. Snapshots of the materials (*top-left*), current density (*top-right*), magnetic flux density (*bottom-left*) and electric field (*bottom-right*).

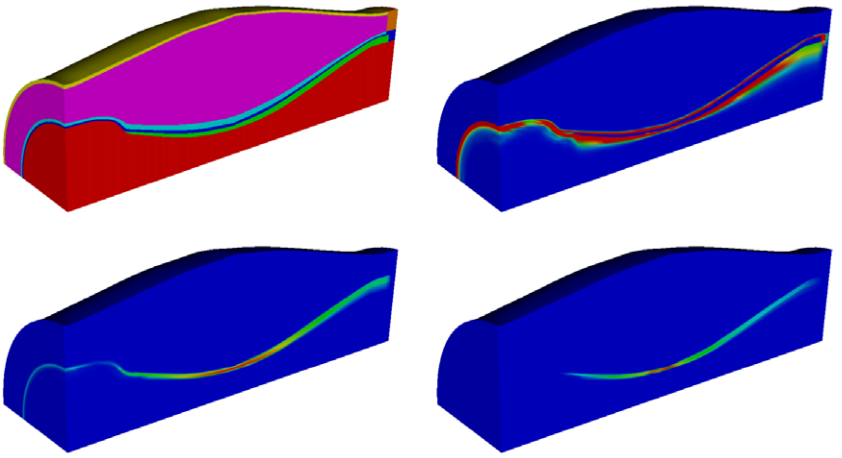

Fig. 31. Shearer generator results at $t = 75$ μs. Snapshots of the materials (*top-left*), current density (*top-right*), magnetic flux density (*bottom-left*) and electric field (*bottom-right*).

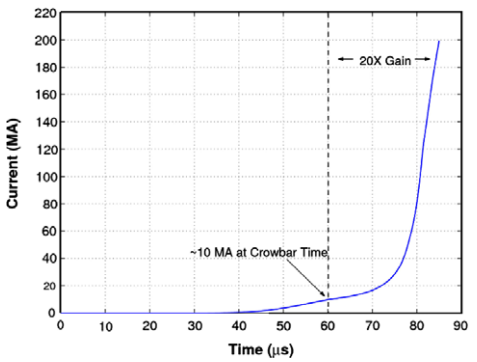

Fig. 32. Computed total current vs. time, these results agree with the 20× gain reported in [38].

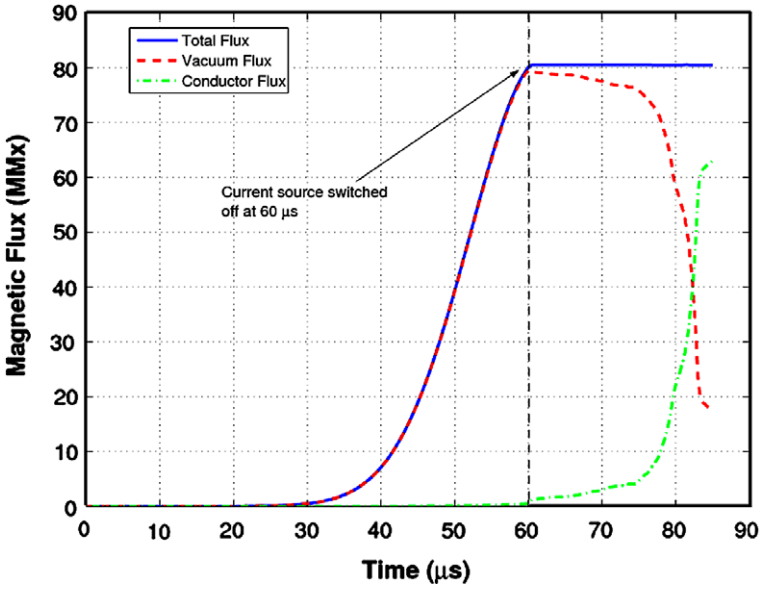

Fig. 33. Computed magnetic flux vs. time along with flux in the vacuum region and conducting region.

The computed output current is shown in Fig. 32, this agrees with the approximately 20× gain described in the original experiments [38]. This is less than predicted by simple circuit models because of lost flux in the conductors due to electromagnetic diffusion. In Fig. 33 we plot the magnetic flux vs. time for the entire problem. Note that at exactly $60\mu s$ (roughly equal to the crowbar time), we switch of the external voltage source implying that we have now deposited a finite amount of electromagnetic energy into the generator. From this point on, total magnetic flux should be conserved and as evidenced by Fig. 33, our numerical method is doing a good job of conserving the total magnetic flux in the problem. In Fig. 33, we also plot the components of the flux in the conducting region and in the vacuum region enclosed by the conductors. If the generator were lossless, then all of the flux would remain trapped in the vacuum region and the total gain would be significantly greater. However, due to finite electrical resistivity, the flux in the vacuum region quickly diffuses into the conducting region, therefore degrading the overall performance of the generator.

Clearly, this simulation could have been performed using a 2D body-of-revolution code, and in fact we have compared results to the established 2D CALE code [9,10]. The results were virtually identical. The advantage of the full 3D simulation is that we can investigate 3D perturbations to the coaxial device, and we can also investigate EMFCG geometries such as flat plate and helical devices.

8. Conclusions

We have developed an arbitrary Lagrangian–Eulerian (ALE) discretization of resistive MHD on 3D unstructured grids. The method was formulated in an operator-split manner with three distinct phases. We have argued that the operator splitting of the magnetic induction equation is particularly simple in the material (or Lagrangian) frame, and therefore a mixed finite element discretization using $H(Curl)$ and $H(Div)$ basis functions is naturally suited for such a problem. This is accomplished by constructing rectangular derivative matrices which represent the curl operator defined with respect to the material frame. We have employed the methodology of the finite element library FEMSTER to perform this discretization as it was designed specifically for these types of representations. Furthermore, we have developed a mixed finite element formulation of the induction equation that has a 2-form magnetic flux as its only state variable (making it amenable to advection), is second order accurate in time and supports explicit voltage source boundary conditions via an additional elliptic solve. For problems that require mesh relaxation and the subsequent remapping of state variables to the new mesh, via Eulerian advection, we have developed an algebraic constrained transport

method which makes use of the rectangular topological derivative matrix. As such, the method is valid for 3D unstructured grids with arbitrary mesh velocity and is second order accurate for smooth magnetic fields. For discontinuous magnetic fields (e.g. MHD shocks), we have developed an algebraic flux correction method which limits an intermediate edge-based voltage in a manner that enforces a local extremum diminishing property on the magnetic flux. This algebraic flux correction is an intermediate step in the algebraic constrained transport algorithm and hence $\vec{\nabla} \cdot \vec{B} = 0$ is still satisfied exactly. We have presented a series of numerical verification experiments which demonstrate the properties and accuracy of the proposed method.

Acknowledgements

We wish to thank colleagues P. Bochev and A. Robinson at Sandia National Laboratory, and M. Shashkov at Los Alamos National Laboratory, for numerous suggestions and review of this work.

References

[1] D.J. Benson, Computational methods in Lagrangian and Eulerian hydrocodes, Comput. Methods Appl. Mech. Eng. 99 (1992) 235–394.
[2] T. Pierce, G. Rodrigue, A parallel two-sided contact algorithm in ALE3D, Comput. Methods Appl. Mech. Eng. 194 (27) (2005) 3127–3146.
[3] A. Nichols, R. Couch, J. Matlby, R. McCallen, I. Otero, R. Sharp, Coupled thermal/chemcial/mechanical modeling of energetic materials in ALE3D, Technical Report UCRL-JC-124706, Lawrence Livermore National Laboratory, 1996.
[4] R. Couch, R. Sharp, I. Otero, R. Tipton, R. McCallen, Application of ALE techniques to metal forming simulations, Technical Report UCRL-JC-114851, Lawrence Livermore National Laboratory, 1993.
[5] M. McClelland, J. Maienschein, A. Nichols, J. Wardell, A. Atwood, P. Curran, ALE3D model predictions and materials charcterization for the cookoff response of pbxn-109, Technical Report UCRL-JC-145756, Lawrence Livermore National Laboratory, 2002.
[6] R. Couch, D. Faux. , Simulation of underwater benchmark experiments with ALE3D. Technical Report UCRL-JC-123819, Lawrence Livermore National Laboratory, 1997.
[7] D. Stevens, Compressible multiphase flows in an ALE framework, Technical Report UCRL-JC-201373, Lawrence Livermore National Laboratory, 2003.
[8] C. Aro, E. Dube, W. Futral, Coupled mechanical/heat transfer simulation on MPP platforms using a finite element linear solver interface, Technical Report UCRL-JC-131746, Lawrence Livermore National Laboratory, 1999.
[9] R. Tipton, A 2D lagrange MHD code, Technical Report UCRL-JC-94277, Lawrence Livermore National Laboratory.
[10] R. Tipton, Modeling flux compression generators with a 2D ALE code, Technical Report UCRL-JC-99900, Lawrence Livermore National Laboratory.
[11] G. Toth, The $\nabla \cdot B = 0$ constraint in shock-capturing magnetohydrodynamics codes, J. Comput. Phys. 161 (2) (2000) 605–652.
[12] C.R. Evans, J.F. Hawley, Simulation of magnetohydrodynamic flows: a constrained transport method, Astrophys. J. 332 (1988) 659–677.
[13] J.M. Stone, M.L. Norman, ZEUS 2D – a radiation magnetohydrodynamics code for astrophysical flows in 2 space dimensions: II. The magnetohydrodynamic algorithms and tests, Astrophys. J. Suppl. Ser. 80 (2) (1992) 791–818.
[14] D.S. Balsara, Divergence-free adaptive mesh refinement for magnetohydrodynamics, J. Comput. Phys. 174 (2) (2001) 614–648.
[15] D.S. Balsara, D.S. Spicer, A staggered mesh algorithm using high order Godunov fluxes to ensure solenoidal magnetic fields in magnetohydrodynamic simulations, J. Comput. Phys. 149 (1999) 270–292.
[16] D.S. Balsara, Second-order accurate schemes for magnetohydrodynamics with divergence-free reconstruction, Astrophys. J. Suppl. Ser. 151 (2004) 149–184.
[17] D.S. Balsara, J. Kim, A comparison between divergence-cleaning and staggered mesh formulations for numerical magnetohydrodynamics, Astrophys. J. 602 (2004) 1079–1090.
[18] P. Londrillo, L. Del Zanna, High-order upwind schemes for multidimensional magnetohydrodynamics, Astrophys. J. 530 (2000) 508–524.
[19] J.M. Stone, J.F. Hawley, C.R. Evans, M.L. Norman, A test suite for magnetohydrodynamic simulations, Astrophys. J. 388 (2) (1992) 415–437.
[20] R.E. Peterkin, M.H. Frese, C.R. Sovinec, Transport of magnetic flux in an arbitrary coordinate ALE code, J. Comput. Phys. 140 (1) (1998) 148–171.
[21] D. Kuzmin, S. Turek, High-resolution FEM-TVD schemes based on a fully multidimensional flux limiter, J. Comput. Phys. 198 (1) (2004) 131–158.
[22] J.C. Nédélec, Mixed finite elements in R3, Numer. Math. 35 (1980) 315–341.
[23] P. Bochev, J. Hu, A. Robinson, R. Tuminaro, Towards robust Z-pinch simulations: discretization and fast solvers for magnetic diffusion in heterogenous conductors, Electron. Trans. Numer. Anal. 15 (2003) 186–210.

[24] J.D. Jackson, Classical Electrodynamics, 1962.
[25] M. Lax, D.F. Nelson, Maxwell equations in material form, Phys. Rev. B 13 (4) (1975) 1777–1784.
[26] R. Rieben, D. White, Verification of high-order mixed FEM solution of transient magnetic diffusion problems, IEEE Trans. Mag. 42 (1) (2006) 25–39.
[27] A. Robinson, Personal communication, 2005.
[28] D. Arnold et al., Compatible Spatial Discreizations, IMA Volumes in Mathematics and its Applications, Springer-Verlag, 2006.
[29] P. Castillo, J. Koning, R. Rieben, D. White, A discrete differential forms framework for computational electromagnetics, Comput. Model. Eng. Sci. 5 (4) (2004) 331–346.
[30] P. Castillo, R. Rieben, D. White, FEMSTER: An object oriented class library of higher-order discrete differential forms, ACM Trans. Math. Soft. 31 (4) (2005) 425–457.
[31] P. Bochev, M. Shaskov, Constrained interpolation (remap) of divergence-free fields, Comput. Methods Appl. Mech. Eng. 194 (2005) 511–530.
[32] R. Rieben, D. White, G. Rodrigue, A high order mixed vector finite element method for solving the time dependent Maxwell equations on unstructured grids, J. Comput. Phys. 204 (2005) 490–519.
[33] M. Berger, M. Aftosmis, Analysis of slope limiters on irregular grids, Technical Report 2005-0490, American Institute of Aeronautics and Astronautics.
[34] A. Harten, High resolution schemes for hyperbolic conservation laws, J. Comput. Phys. 49 (1983) 357–393.
[35] R.J. LeVeque, Numerical Methods for Conservation Laws, Birkhauser-Verlag, 1990.
[36] A. Jameson, Computational algorithms for aerodynamic analysis and design, Appl. Numer. Math. 13 (1993) 383–422.
[37] T.V. Kolev, P.S. Vassilevski, Parallel H1-based auxiliary space AMG solver for $H(curl)$ problems, Technical Report UCRL-TR-222763, Lawrence Livermore National Laboratory.
[38] J.W. Sehearer et al., Explosive driven magnetic field compression generators, J. Appl. Phys. 39 (4) (1968) 2102–2116.