A Constrained-Gradient Method to Control Divergence Errors in Numerical MHD

Philip F. Hopkins¹*

¹TAPIR & The Walter Burke Institute for Theoretical Physics, Mailcode 350-17, California Institute of Technology, Pasadena, CA 91125, USA
E-mail: phopkins@caltech.edu


ABSTRACT

In numerical magnetohydrodynamics (MHD), a major challenge is maintaining \( \nabla \cdot \mathbf{B} = 0 \). Constrained transport (CT) schemes can achieve this at high accuracy, but have generally been restricted to very specific methods. For more general (meshless, moving-mesh, or ALE) methods, “divergence-cleaning” schemes reduce the \( \nabla \cdot \mathbf{B} \) errors; however they can still be significant, especially at discontinuities, and can lead to systematic deviations from correct solutions which converge way away slowly. Here we propose a new constrained gradient (CG) scheme which augments these with a hybrid projection step, and can be applied to any numerical scheme with a reconstruction. This iteratively approximates the least-squares minimizing, globally divergence-free reconstruction of the fluid. We emphasize that, unlike “locally divergence free” methods, this actually minimizes the numerically unstable \( \nabla \cdot \mathbf{B} \) terms, without affecting the convergence order of the method. We implement this in the mesh-free code GIZMO and compare a wide range of test problems. Compared to state-of-the-art cleaning schemes, our CG method reduces the maximum \( \nabla \cdot \mathbf{B} \) errors in each problem by \( \sim 1 - 3 \) orders of magnitude (\( \sim 2 - 5 \) dex below the typical errors if no \( \nabla \cdot \mathbf{B} \) cleaning is used). By preventing large \( \nabla \cdot \mathbf{B} \) even at unresolved discontinuities, the method eliminates systematic errors at jumps. In every problem, the accuracy of our CG results is comparable to CT methods; for practical purposes, the \( \nabla \cdot \mathbf{B} \) errors are eliminated. The cost is modest, \( \sim 30\% \) of the hydro algorithm, and the CG correction can be easily implemented in a wide range of different numerical MHD methods. While for many problems, we find Dedner-type cleaning schemes are sufficient for good results, we identify a wide range of problems where using only the simplest Powell or “8-wave” cleaning can produce systematic, order-of-magnitude errors.

Key words: methods: numerical — hydrodynamics — instabilities — turbulence — cosmology: theory

1 INTRODUCTION

Magnetohydrodynamics (MHD) is essential to many physical problems, and (because the equations are non-linear) often requires numerical simulations. But this poses unique challenges. Naive discretizations of the MHD equations lead to violations of the “divergence constraint” \( (\nabla \cdot \mathbf{B} = 0) \); unfortunately, certain errors related to non-zero \( \nabla \cdot \mathbf{B} \) are numerically unstable (they corrupt the solution even at infinite resolution). As such, many methods have been developed to control them.

The CT method of Evans & Hawley (1988), and related vector-potential/flux-central difference methods can conserve an initial \( \nabla \cdot \mathbf{B} = 0 \) to machine precision each timestep; however, while it may be possible to extend these methods in principle (see Mocz et al. 2014), it has thus far only been practical to implement for real problems in regular, fixed-grid (Eulerian), Godunov-type schemes. It is also (often) computationally expensive, and only conserves \( \nabla \cdot \mathbf{B} \), so has difficulty dealing with the injection/removal of magnetic energy, volume elements, or mass.

For other (e.g. mesh-free, moving-mesh, arbitrary Lagrangian-Eulerian, smoothed-particle) methods, “divergence-cleaning” schemes are popular. The Powell et al. (1999) or “8-wave” cleaning simply subtracts the unstable \( \nabla \cdot \mathbf{B} \) terms from the equation of motion; this cures the instability and restores Galilean invariance, but does not actually reduce \( \nabla \cdot \mathbf{B} \) (it only “transports” the errors). Many studies have shown that certain types of problems, treated only with this method, will converge to the wrong solution (Toth 2000; Mignone & Tzeferacos 2010; Mocz et al. 2014; Hopkins & Raives 2015); moreover the subtraction necessarily violates momentum conservation, so one would like to minimize the subtracted terms. More sophisticated cleaning schemes have been proposed; many of which follow Dedner et al. (2002) and add a scalar-field and set of source terms which transport the divergence in waves and damp it (and correct behavior in shock jumps). However, this still requires a finite “response time” to damp \( \nabla \cdot \mathbf{B} \) (so may act “too slowly” in discontinuities), and is reactive (dissipating, rather than preventing errors); as such it is less than ideal. Nevertheless, these schemes have been applied across a wide range of methods.

Alternatively, projection schemes following Brackbill & Barnes (1980) take the solution at each timestep, and project it onto a globally divergence-free basis (or equivalently, solve for the divergence-free component of the fluxes, and subtract off the other components). This can reduce \( \nabla \cdot \mathbf{B} \) below a desired tolerance at each timestep, acts “instantly,” and as shown in Toth (2000), preserves the convergence order of the method. However, it is expensive (requires a global sparse matrix inversion every timestep), can become unacceptably inefficient when adaptive/hierarchical timesteps and/or non-regular mesh geometries are used, is not compatible with arbitrary slope/flux limiters, and the inversion itself can become unstable under certain circumstances. Hence the application of these methods has been limited.

1 This is generally done by representing \( \mathbf{B} \) as \( \mathbf{B} = \nabla \times \mathbf{A} + \nabla \phi \), solving for the vector/scalar fields \( \mathbf{A} \) and \( \phi \), subject to some constraints (e.g. \( \phi \) is constrained to minimize its least-squares value/integral over volume), then taking \( \mathbf{B} = \mathbf{B} - \nabla \phi \).

2 Of course, many further divergence-control schemes have been proposed (see e.g. Sweeny et al. 1995; Monaghan 2000; Barve et al. 2001; Marion & Howes 2003; Price & Rosswog 2006; Rosswog & Price 2007; Price & Bate 2008; Dolag & Stasyszyn 2009). However, many of these examples either fail to cure the numerical tensile instability, are zeroth-order inconsistent...
In this paper, therefore, we propose a hybrid constrained-gradient (CG) scheme which combines some advantages of the schemes above. We implement this in the multi-method Lagrangian MHD code GIZMO\(^4\) and test the scheme in a wide range of problems.

2 NUMERICAL METHODOLOGY

2.1 The Problem

In most finite-volume Godunov-type methods, the MHD equations are a set of hyperbolic partial differential conservation equations for an element (particle or cell) \(i\), surrounded by elements \(j\), which take the discrete form

\[
\frac{d}{dt}(V U_i) + \sum_j F_{ij} \cdot A_{ij} = (V S)_i
\]

(1)

where \(V\) is the element volume, \(U_i = (\rho, \rho v, \rho e, B, \rho \psi)\) is a vector of primitive variables (mass/momentum/energy/density, magnetic field, scalar fields), \(S\) is a vector of source terms, \(A_{ij}\) is an oriented "effective face area" defining the interaction surface between the elements, and \(F_{ij}\) is the relevant flux, computed by solving the appropriate Riemann problem at the face.\(^5\)

The inputs to the Riemann problem at \(A_{ij}\) are the reconstructed quantities \(U_{R,i}\) and \(U_{L,j}\), which are the extrapolated values of \(U\) on the "i-side" and "j-side" of the face, respectively. In second-order methods, if the point at which the Riemann problem is solved (usually the midpoint of the face) lies at coordinates \(x_{ij}\), then \(U_{R,i} = U_i + \phi (\nabla \otimes U_i) \cdot (x_{ij} - x_i)\), where \(\nabla \otimes U_i\) is the gradient tensor calculated at element \(i\), and \(\phi\) is an appropriate slope-limiter which restricts the gradient values to prevent the creation of new maxima/minima, and \(U_{L,j}\) denotes the value of \(U\) which is appropriately time-centered for the numerical time integration scheme.

In MHD, we wish to preserve \(\nabla \cdot B = 0\). Since there are an infinite number of valid definitions of the discrete gradient operator, it is impossible for any non-trivial field configuration to satisfy this under all operators. The relevant definition of \(\nabla \cdot B\), in most second-order finite-volume schemes, is something like

\[
(V \nabla \cdot B)_i \equiv -\frac{1}{2} \sum_j \left( B_{R,i} + B_{L,j} + \frac{\psi_L - \psi_R}{\delta x_{ij}} A_{ij} \right) \cdot A_{ij} \]

(2)

where the (optional) \(\psi\) terms arise from divergence-cleaning schemes such as that in\(^6\) Dedner et al. (2002), following a source function proportional to \(V \nabla \cdot B\). The averaging of \(B_R\) and \(B_L\) appears because the one-dimensional Riemann problem requires that the \(B\) component normal to the face be constant. This definition represents the numerically unstable terms, which are subtracted in cleaning schemes\(^7\) (Powell et al. 1999; Dedner et al. 2002); also, since this represents a surface integral, maintenance of \(V \nabla \cdot B = 0\) according to this definition is physically equivalent to magnetic flux conservation. Ideally, we would always have

\begin{align*}
B_R &= B_L, \quad (V \nabla \cdot B)_i = 0 \quad \text{(hence also} \psi = 0\text{). This is exactly what CT schemes try to ensure.}
\end{align*}

Note that variations of Eq.\(^2\) are possible, and can (if carefully defined) represent valid gradient definitions. We will use Eq.\(^2\) as the basis of our subsequent derivations because it is common to many codes, and specifically it is the definition of relevance for the GIZMO code which we will use for our tests. However it is straightforward to modify our subsequent derivations for a modified form of Eq.\(^2\).

2.2 Locally Divergence-Free Methods

We have some freedom in the choice of discrete approximation for the \(B\)-field gradients, \(\nabla \otimes B\). For second-order methods, we must choose a definition such that the errors in the reconstruction scale \(\propto h^2\) (where \(h\) is the linear element size) in smooth flows, but this still allows considerable flexibility. To make progress, we will adopt the gradient estimator in GIZMO, a moving least-squares estimator. For a scalar \(f\), this is

\[
(V f)_i = \sum_j (f_j - f_i) \left( \frac{W_j^{-1}}{|h|} \right)^{\alpha \beta} (x_i - x_j)^{\alpha \omega_{i}(x)}
\]

(3)

\[
W_i^{\alpha \beta} \equiv \sum_j (x_i - x_j)^{\alpha} (x_j - x_i)^{\beta} \omega_{i}(x)
\]

(4)

here we assume an Einstein summation convention over the indices \(\beta\) corresponding to the spatial dimensions, and \(\omega_{i}(x)\) is an (arbitrary) weight function defined in Paper I. This estimator is second-order accurate for an arbitrary mesh configuration, minimizes the (weighted) least-squares deviation \(\sum_{i,j} \omega_{i}(x_j - x_i)^{2}\), and has been applied in a wide range of different numerical methods (see e.g. Oñate et al. [1996]; Kuhnert 2003; Maron & Howes 2003; Luo et al. 2008; Larson & Vlajić 2008).

It is straightforward to constrain this to obtain the locally divergence-free solution: i.e. \((\nabla \otimes B)_i\) such that \((\nabla \cdot B)_i = \sum_{j} (\nabla \otimes B)_{ij} = 0\). In the least-squares formulation, this is just a constrained least-squares problem – we seek the matrix \((\nabla \otimes B)_{ij}\) constrained to have \((\nabla \cdot B)_i = 0\) which minimizes the (weighted) square deviation of \((\nabla \cdot B)_i + (x_i - x_j) \cdot (\nabla \otimes B)_{ij} - B_{i,j}\).

A similar approach is to calculate the gradient projected onto a set of divergence-free basis functions. The matrix formulation above implicitly adopts the Cartesian polynomial basis functions \((1, x, y, z, x^{2}, xy, xz, ...\), but a divergence-free basis can be chosen instead and used for the reconstruction.

The problem with both of these is as follows. If we temporarily ignore the slope-limiter and \(\psi\) terms in Eq.\(^2\) then \((V \nabla \cdot B)_i\) is just

\[
(V \nabla \cdot B)_i = -\frac{1}{2} \sum_j \left[ B_{i,j} + (\nabla \otimes B) \cdot (x_j - x_i) \right. \\

+ B_{ij} + (\nabla \otimes B) \cdot (x_i - x_j) \left. \right] \cdot A_{ij} + O(\phi, \psi)
\]

(5)

It is obvious from this expression that ensuring \((\nabla \cdot B)_i = 0\) does not ensure \((V \nabla \cdot B)_i = 0\;\text{in};\;\text{it}\;\text{does}\;\text{not}\;\text{even}\;\text{necessarily}\;\text{decrease}\;\text{(V \nabla \cdot B)}_i\).

\(^{5}\) It is easy to verify this. Consider a trivial case: a 2D, perfectly regular lattice of equally-spaced elements so \(A_{ij} = A\) and \(|x_i - x_j| = \delta x\), and take \(x_0 = 0\). Assuming \((\nabla \cdot B)_i = 0\;\text{in}\;\text{Eq.\(^5\)}\) simplifies to \((V \nabla \cdot B)_i = -A[(\nabla \cdot B)_{ij} - B_{ij}(x_i - x_j) + (B_{ij}(0, x_j) - B_{ij}(0, -x_j))]/2.\) Assume \(B_{i,j} = f(y)\), so \(B_{ij} = B_{0,j}(y) - (df/dy)x_j\), then \((V \nabla \cdot B)_i = A[(df/dy)x + (f(x_j) - f(-x_j))/2] \neq 0\) for any non-linear \(f\). Or simply assume a noisy, but constant-mean field, such that \((\nabla \cdot B)_i = 0\;\text{in};\;\text{then}\;\text{(V \nabla \cdot B)}_i = 0\), but \((V \nabla \cdot B)_i \propto \sum_j (B_{ij} + B_{j,i}) \cdot A_{ij},\;\text{which} \;\text{can} \;\text{only} \;\text{vanish} \;\text{for} \;\text{special} \;\text{configurations} \;\text{of} \;B_{ij} \;\text{and} \;A_{ij}.

\(^{6}\) Users are encouraged to modify and extend the capabilities of this code; the development version of the code is available upon request from the author.

\(^{7}\) Throughout, "\(\otimes\)" denotes the outer product, "\(\cdot\)" the inner (dot) product, and "\(\sum\)" the double-dot product \(A \cdot B = \sum_{ij} A_{ij} B_{ij}\). We use \sum_{ij} to denote double-summation \(\sum_{ij} \equiv \sum_{i} \sum_{j} \).
2.3 An Approximate, Globally Divergence-Free Method

Still, these locally divergence-free methods suggest a solution. Assume, for now, that we are estimating the gradient $(\nabla \cdot B)_i$ of element $i$, and all other gradients in the system have been determined; also for now neglect the slope-limiter and $\psi$ terms so $(V \nabla \cdot B)_i$ follows Eq. 5. Then we see that we can, in fact, ensure $(V \nabla \cdot B)_i = 0$, provided that $(\nabla \cdot B)_i$ satisfies:

$$(\nabla \cdot B)_i : \sum (x_j - x_i) \otimes A_{ij} = 0$$

and more computationally efficient solution is to iteratively calculate $(\nabla \cdot B)_i$:

$$(\nabla \cdot B)_i^{(n)} = (\nabla \cdot B)_i^{(n-1)} - Q \left( \frac{S_0^{(n)} + (\nabla \cdot B)_i^{(n-1)} \cdot Q}{Q \cdot Q} \right)$$

where in the first-pass ($n = 1$), we take $(\nabla \cdot B)_i^{(0)}$ to be the value of $(\nabla \cdot B)_i$ from the most recent time/drift step, calculate the new $(\nabla \cdot B)_i^{(1)}$ for all elements, then use these values to calculate the updated $(\nabla \cdot B)_i^{(2)}$, etc.

In practice, we find that given the errors sourced by the slope-limiters, we converge to nearly best-case accuracy in just two iterations $(n = 2)$. And since all the quantities here can be calculated in the same path that is used to calculate $(\nabla \cdot B)_i^{(0)}$, the entire iteration series only requires one additional element sweep, compared to the “standard” method.

2.4 Complications: Dealing with Slope Limiters and Cleaning Terms

Issue (2) is more challenging. The $\psi$ terms in Eq. 2 should be minimized along with $\nabla \cdot B$, so they are not particularly problematic.

We can include them explicitly in our constraint solution for $G$, using the same iterative approach to account for the fact that $\psi_{t_i}$, $\psi_r$, and $c_{b,i}$ themselves depend on the gradients in the problem. This “mixing” of $\psi$ and $B$ essentially defeats the purpose of the damping $\psi$ terms, and can introduce more serious numerical instabilities in the (rare) cases where $\psi/\psi_r \gg |B|$. We therefore leave them, since their purpose is to damp $\nabla \cdot B$ where present. But we do reduce the contribution of the $\psi_{t} - \psi_r$ term in Eq. 2 by minimizing the least-squares deviation between $\psi$ extrapolated from the $i$ and $j$ “sides” at the face locations, rather than at the particle $j$ locations (i.e. our preferred gradient is minimizing the squared deviation of $[\psi_i + (\nabla \psi)_i \cdot (x_j - x_i)] - [\psi_j + (\nabla \psi)_j \cdot (x_i - x_j)]$, rather than $\psi_i + (\nabla \psi)_i \cdot (x_j - x_i) - \psi_j$). This is numerically consistent at the same order and trivial to implement using the same matrix based least-squares formulation, and we find it slightly reduces the divergences and improves the cleaning accuracy, so we use it throughout.

To deal with the slope-limiters $\tilde{\phi}$, we take advantage of our iterative approach. Generally speaking, there are two types of slope-limiters in most of the methods of interest.

First, slope-limiters that are applied to the gradient after its calculation loop (and apply to all subsequent operations): $\tilde{\phi}$ is $\text{MIN}(1, \tilde{\phi'})$ where $\tilde{\phi'}$ is chosen such that the reconstruction value of a field does not exceed the maximum/minimum neighbor values by more than some tolerance [Balsara 2001]. These are straightforward: we calculate our “preferred” gradient and then apply this

In GIZMO, this takes the form:

$$\tilde{\phi'} \equiv \text{MIN} \left[ 1, \beta \text{MIN} \left( U_{i,j,\text{nb}}^{\text{mid}} - U_j, \frac{U_i - U_{i,j,\text{nb}}^{\text{mid}}}{U_i - U_{i,j,\text{nb}}^{\text{mid}}} \right) \right]$$

where $U_{i,j,\text{nb}}^{\text{mid}}$ and $U_{i,j,\text{nb}}^{\text{max}}$ are the maximum and minimum values of $U_j$ among all neighbors $j$ of the particle $i$, and $U_{i,j,\text{nb}}^{\text{mid}}$, $U_{i,j,\text{nb}}^{\text{min}}$ are the maximum and minimum values (over all pairs $ij$ of the $j$ neighbors of $i$) of $U$ re-constructed on the “$i$ side” of the interface between particles $i$ and $j$. The constant $\beta = 1 - 2$ depending on local particle order. In our iterative CG implementation, we first limit with the “normal” $\beta = 2/\beta_0$, then correct the gradient, then re-limit only if it exceeds the slightly weaker limiter with $\beta = 2/\beta_0$.\footnote{In GIZMO, this takes the form:}
limiter, and treat this as the new “preferred” gradient. After correction, the new gradient may violate this condition, so we can optionally re-limit it, and treat this as the new “preferred” gradient, and iterate until convergence (this iteration is outside the neighbor loop so has negligible cost). This converges to the gradient satisfying the desired slope limiter which comes as close as possible to the desired CG-corrected gradient.

Second, another class of slope-limiters can (optionally) be additionally applied in pair-wise fashion between every interacting element pair in the flux computations; this ensures no local maxima/minima are created. Here the limiter \( \delta_i \) is unique to the element pair. We account for these limiters explicitly in our calculation of \( S_i \); using both the current values of \( (\nabla \otimes \mathbf{B})_i \) and \( (\nabla \otimes \mathbf{B})_{\hat{i}} \), we apply the limiters between each pair, and thus obtain a more accurate guess for the correction to \( (\nabla \otimes \mathbf{B})_i \).

These approaches allow us to handle arbitrary slope limiters and still return some result. But it is easy to see that application of any slope limiter can, under some circumstances, disallow the corrected value of \( (\nabla \otimes \mathbf{B})_i \), needed to actually ensure \( (\nabla \cdot \mathbf{B})_i = 0 \). This is why our procedure ceases to significantly improve after a couple iterations. And clearly, a stricter slope-limiter prevents \( (\nabla \cdot \mathbf{B})_i \), correction under a wider range of circumstances. Therefore, when we implement our CG method, we “weaken” our normal pair-wise slope limiter for \( \mathbf{B} \), to allow more flexible CG correction. This is important because, as we will show, if we do this and do not include any divergence-damping terms, it (unsurprisingly) produces large oscillations. Note, though, that we still retain the standard slope-limiter applied after the gradient calculation \( (\hat{\phi}^i) \), so the second slope limiter is somewhat redundant anyways, and we only alter the slope limiters for \( \mathbf{B} \).

2.5 Implementation

We implement the CG method above in the code GIZMO (Hopkins 2015). GIZMO is a mesh-free, finite-volume Godunov code, built on the gravity solver and domain decomposition algorithms of GADGET-3 (Springel 2005). In Hopkins (2015), Hopkins & Raives (2015) we consider extensive surveys of test problems in both hydrodynamics and MHD (using the Dedner scheme) with this code, and demonstrate accuracy and convergence in good agreement with well-studied regular-mesh finite-volume Godunov methods. Because GIZMO is a multi-method code, it allows us to compare directly the effects of different divergence-control methods with an otherwise entirely identical code. Since we are not comparing hydro solvers here but the divergence control method, in what follows, we run GIZMO always in its MFM (Meshless Finite-Mass) mode, but we note that we have run several problems in the MFV (Meshless Finite-Volume) mode, and find nearly identical results (as expected from the comparisons in the methods paper).

We have implemented a limited, 2D version of CG in the public moving-mesh code FVMHD3D (Gaburov et al. 2012), as expected from our previous comparisons, this is very similar to the GIZMO MFV results. For reasons shown below, when we implement our CG method, we still retain the Powell and Dedner source terms in the MHD equations, to deal with imperfect minimization of \( (\nabla \cdot \mathbf{B})_i \).

We have made public a version of the GIZMO code with the CG implementation used in this paper. Users interested in the details of our implementation (for example, the exact numerical form of the slope-limiters used, kernel weights used for the least-squares calculation, etc.) are encouraged to examine the source code.

3 TEST PROBLEMS

We now consider a series of test problems. Each of these has been studied in detail in Hopkins & Raives (2015), where we undertook a systematic comparison of different algorithms (MFM, MFV, SPH-MHD, and moving meshes, using the Dedner et al. 2002 divergence-cleaning scheme, and grid/AMR schemes using constrained transport). Therefore we will not describe them in detail here.

3.1 Shocktubes

Fig. 1 shows two shocktube standard shocktube tests: the sub-sonic, magnetically dominated Brio & Wu (1988) and super-sonic Toth (2000) shocktubes. We compare different divergence-control...
schemes, with resolution $\approx 256 \times 56$ across the domain plotted (a 2D grid, with the shock propagating at an angle $\pi/6$ to the grid).

As discussed at length in Hopkins & Raives (2015), with no divergence-control at all, the schemes are unstable and crash (developing negative pressures). The minimal correction to restore stability is the Powell et al. (1999) or “8-wave” cleaning; however, this only subtracts the tensile terms from non-zero $\nabla \cdot B$, it does not actually control $\nabla \cdot B$; as a result we see therein that the terms driving corrections to the gradients are not being damped. If we apply our CG scheme but ignore the Powell et al. (1999) correction terms, the tensile instability appears and the oscillations grow to unacceptable levels.

Figure 2. Shocktubes from Fig. 1 with alternative divergence-control schemes. If, instead of approximating the global, divergence-free reconstruction according to Eq. 8, we simply constrain the system to be “locally divergence free” (i.e. $\nabla \cdot B = 0$ for the particle-centered gradient estimate; as § 2.2), we see there is essentially no reduction in the numerically problematic $h_i |\nabla \cdot B_i|/|B_i|$ term compared to the Dedner et al. (2002) without this constraint, and the systematic shock jump errors are actually increased. If we apply our CG scheme without the Dedner et al. (2002) cleaning, we recover the mean solutions but see large oscillations since the terms driving corrections to the gradients are not being damped. If we apply our CG scheme but ignore the Powell et al. (1999) correction terms, the tensile instability appears and the oscillations grow to unacceptable levels.

9 Here and in all subsequent plots, the CT results are calculated with the grid code ATHENA (Stone et al. 2008), run in its most accurate mode (PPM, CT, CTU). The accuracy and convergence properties of this code are well-studied. All other results are from GIZMO. Further comparison of the codes is found in Hopkins (2015).
Figure 3. Two-dimensional MHD tests. For each test (column), we compare four methods: CT, CG, Dedner, & Powell (see Fig. 1), as labeled (top-to-bottom). Each pair of columns shows a map of a fluid quantity (left), and the corresponding map of $\log_{10}(h_i |\nabla \cdot B_i|/|B_i|)$ (right), with values following the colorbar. From left to right we show: Left: Field loop advection. We show magnetic pressure at time $t = 20$. Methods should preserve a perfect circle at the maximum amplitude; numerical diffusion is visible at the center and edges. Middle Left: Orszag-Tang vortex, showing density at $t = 0.5$. Middle Right: MHD rotor, showing gas pressure at $t = 0.15$. Right: MHD blastwave, showing density at $t = 0.2$. In each case Dedner, CG, and CT solutions are nearly-identical (the extra diffusion in CT in e.g. the field loop, is only because it uses an Eulerian, not Lagrangian code). The Powell scheme produces visibly incorrect features in the blastwave shock jump; and in the field loop test the $h_i |\nabla \cdot B_i|/|B_i|$ errors self-interfere and grow unstably. In all cases $h_i |\nabla \cdot B_i|/|B_i|$ decreases dramatically from Powell to Dedner and Dedner to CG, where it remains at values $\ll 0.01$.

Figure 4. Additional 2D tests, as Fig. 3: MHD Rayleigh-Taylor instability at times $t = 6$ (Left) & $t = 16$ (Middle Left), and growth of the magnetorotational instability (MRI) in a shearing sheet at $t = 10$ (Middle Right) and $t = 19$ (Right). All methods capture the linear growth of the RT and MRI, and breakup of the non-linear MRI into turbulence at late times. In each, Dedner, CG, and CT agree well (even into non-linear stages); CG maintains $h_i |\nabla \cdot B_i|/|B_i| \ll 0.01$ even well into non-linear and turbulent evolution. Powell schemes show small deviations in the MRI and linear RT growth, but the errors build up in the non-linear RT and destroy the solution.
Figure 5. Quantitative comparison of the 2D tests in Figs.3-4. Left: Values of \( B_x \) in horizontal slices, for the rotor (top), Orszag-Tang vortex (middle), and blastwave (bottom) tests (at the same time as Fig.3). We compare methods at 256\(^2\) resolution to an exact solution. All other fluid quantities show comparable or smaller deviations from the exact solution at this resolution. Right: Values versus time of the box-averaged \( |B|^2 \) in the field loop test (top), low-density plume height in the RT test (middle), and magnetic energy density in the MRI test (bottom). In most tests Powell cleaning produces small deviations (offset shock positions, slower MRI growth); but in the blastwave and field loop tests the failure is dramatic. All other methods agree well and exhibit similar convergence rates. CG shows slightly smaller errors at fixed resolution compared to Dedner. Difference between CG & CT (more diffusion for CT in the field loop & RT tests, slightly sharper shock-capturing in CT in the blastwave, & different late-time decay of the MRI) owe to the difference between grid methods (the CT results here) and Lagrangian methods (all others), not to divergence errors.

well with the analytic predictions). Note that in the field loop problem, the CG method is slightly more dissipative compared to Dedner (CT is more dissipative still, but this is primarily due to the CT code being an Eulerian, not Lagrangian, code). This is because there is a real non-zero \( \nabla \cdot B \) set up in the ICs at the “edge” of the circle, as we numerically implement them. This forces the gradients to correct for this, and dissipate away the divergence.

Variations of the above tests with different seed \( B \)-fields produce qualitatively identical conclusions. And because the GIZMO methods are Lagrangian, “boosted” or rotated versions of the tests are trivially identical to those shown. We have compared the MHD Kelvin-Helmholtz instability and “blob” test, but the qualitative differences between methods are identical to the RT test shown. The “current sheet” test in Hawley & Stone (1995) is a test of numerical stability which all methods here pass similarly (see Paper I); however as with the field loop we see greater dissipation in CG because the ICs contain a real non-zero \( \nabla \cdot B \). We have also simulated low-resolution 3D versions of the RT, KH, field loop, and MRI problems with qualitatively similar results in all cases to the 2D tests here.

3.3 Non-Linear Dynamics with External Forces

Next we consider a set of non-linear dynamics tests where the dominant forces are not MHD, but gravity. These are especially challenging for divergence-control methods because elements are being constantly re-arranged by non-MHD forces in a manner that is often faster than the local fast magnetosonic crossing (hence fluid response) time.

We consider (1) collapse of a rotating, self-gravitating protostellar core, to form a protostar and accretion disk which winds up a seed \( B \) field and launch an MHD jet, following Hennebelle & Fromang (2008). (2) the MHD version of the “Santa Barbara cluster,” in which a cosmological simulation of gas+dark matter is followed using adiabatic (non-radiative) gas physics, from high redshift until the Lagrangian region being simulated forms an object with the mass of a galaxy cluster at \( z = 0 \) (Frenk et al. 1999), (3) an isolated (non-cosmological) galaxy disk, with gas, stars, and dark matter, radiative cooling, star formation, and stellar feedback, following the simple sub-grid Springel & Hernquist (2003) “effective equation of state” model (in which the phase structure of the ISM is not resolved but replaced with a simple barytropic equation of state, as used in large-volume cosmological simulations; Vogelsberger et al. 2013), and (4) the same disk, treating feedback explicitly according to the FIRE (Feedback in Realistic Environments) project physics (Hopkins et al. 2011, 2012, 2013, 2014), which explicitly follow the multi-phase ISM, turbulence, and feedback from stellar winds, radiation, and supernovae.
These tests do not have known exact solutions, however there are specific qualitative behaviors that should be observed in each case. And they are valuable “stress tests” for divergence control (indeed, many algorithms cannot run these problems without crashing). Fig. 6 summarizes the results. In all cases the behavior is very similar between our Dedner and CG results. In the jet test, we see the jet launched efficiently and the system evolves stably to late times; the mass-loading of these jets is in good agreement with much higher-resolution CT-based AMR results (see Hennebelle & Fromang 2008), as discussed in detail in Hopkins & Raives (2015). Likewise, in the Santa Barbara test, we see the cluster form, and non-linear field amplification in the cluster center; the radiative simulation of dark matter and gas which forms a massive galaxy cluster-hosting halo), at redshift $z = 0$ (slice through the cluster center shown).}

\footnote{Unfortunately, we cannot rigorously compare MT methods in Fig. 6, since the relevant physics for these problems is not implemented in the same manner in any CT code. Moreover the gravity solvers are different.}
profiles of density, temperature, magnetic field strength, and velocity also agree well with higher-resolution CT-based AMR runs (compare Mink & Martin 2011), up to subtle differences at small radii that depend on whether the methods used are Lagrangian or Eulerian (see Hopkins 2015).

In the disk problem, the disk remains smooth (as it should) with the “effective EOS” model, while with the FIRE model it rapidly develops super-sonic turbulence, multi-phase structure, and a strong galactic wind. In both cases the field is amplified; amplification is slower in the “effective EOS” case because (by construction) there is no sub-structure, turbulence, or galactic wind (so only global disk winding amplifies B).

In the FIRE case, the molecular gas tracing spiral structure and GMCs is clearly evident in the $\beta$ map as regions where the thermal pressure is sub-dominant. Although the qualitative behavior is similar, the CG method reduces $h_i |\nabla \cdot \mathbf{B}|/|\mathbf{B}|_i$ by $\sim 1 - 2$ dex, relative to the Dedner case (which, as expected, can reach large $h_i |\nabla \cdot \mathbf{B}|/|\mathbf{B}|_i \geq 0.1$ in sharp discontinuities).

In all tests here, the Powell-only case is problematic: divergence errors reach $h_i |\nabla \cdot \mathbf{B}|/|\mathbf{B}|_i \gg 1$, and even $h_i |\nabla \cdot \mathbf{B}|/(|\mathbf{B}|_i^2 + 2P_{\text{thermal}})^{1/2} \sim 1$. This produces some qualitatively erroneous features: the jet disconnects from its launching zone, and the disk on small scales exhibits a “bending” because the protostar is actually moving with a net $z$-velocity that grows in time, until it “self-ejects” (owing to the momentum conservation error that comes with subtracting large $h_i |\nabla \cdot \mathbf{B}|/|\mathbf{B}|_i$ terms); this is seen in SPH and grid-based codes using Powell-only control as well (see Price et al. 2012). In the cluster and disk problems, the same errors seen in the field loop problem lead to artificial, unstable growth of magnetic energy: the B field is higher by about $\sim 1$ dex at the times shown compared to the Dedner, CG, and CT solutions, and it grows at an unphysical rate (faster than the disk dynamical time, even in the smooth-disk case).

We have also compared the 3D MHD Zeldovich pancake test from Paper I with various initial B (see Zel’dovich 1970; Li et al. 2008). This test, unlike those above, actually has an exact solution. Unfortunately, that is possible because of the highly simplified problem geometry which makes $\nabla \cdot \mathbf{B}$ control unnecessary (even Powell schemes maintain $h_i |\nabla \cdot \mathbf{B}|/|\mathbf{B}|_i \ll 10^{-4}$), so (not surprisingly), all the methods here produce indistinguishable results from CT.

4 DISCUSSION

We have introduced a new method, Constrained-Gradient MHD, to control the $\nabla \cdot \mathbf{B}$ errors associated with numerical MHD. This involves an iterative approximation to the least-squares minimizing, globally divergence-free reconstruction of the fluid. We implement this in the code GIZMO and show that, compared to state-of-the-art MHD implementations using divergence-cleaning schemes, this is able to further reduce the $\nabla \cdot \mathbf{B}$ errors by orders of magnitude, and improves numerical convergence and accuracy at fixed resolution.

The performance cost of this method is small, compared to constrained transport on irregular grids. It requires one additional neighbor element sweep after the main gradient sweep (the iteration step), in which the quantity $S_0$ is re-calculated based on the updated gradients. In our implementation in GIZMO, this increases the CPU cost of the hydro operations by $\sim 20 - 30\%$, with essentially no memory cost.

This method is motivated in spirit by projection and locally divergence-free reconstruction schemes. However, it avoids the large overhead expense of projection schemes and does not modify the fluxes, and unlike traditional projection schemes it can operate with complicated slope limiters, non-linear gradient estimators, irregular mesh geometries, and adaptive timestepping, and preserves the convergence order of the code. Unlike locally divergence-free schemes, the method here actually minimizes the numerically unstable $\nabla \cdot \mathbf{B}$ terms, as opposed to those from a different estimator.

We consider a large suite of test problems, and compare a variety of divergence-control schemes. With only Powell or “8-wave” cleaning, typical $h_i |\nabla \cdot \mathbf{B}|/|\mathbf{B}|_i \sim 1$ in non-linear problems/discontinuities, and this sources zeroth-order systematic errors at discontinuities which do not decrease with increasing resolution. This causes serious problems on a wide range of problems: large systematic errors in shock jumps, catastrophic noise, exponential growth of magnetic energy, and order-unity violations of energy/momentum conservation and symmetry all appear even at high resolution.

Adding a more sophisticated Dedner hyperbolic-parabolic cleaning-damping term allows the method to converge properly, and reduces $h_i |\nabla \cdot \mathbf{B}|/|\mathbf{B}|_i \sim 1 - 2$ orders of magnitude. In most tests we find that this is sufficient for ideal behavior and high accuracy; however, non-negligible $\nabla \cdot \mathbf{B}$ errors can still appear at large discontinuities, which lead to small systematic offsets in jump conditions that converge away relatively slowly.

In every case, the CG method further reduces the problematic $\nabla \cdot \mathbf{B}$ terms (by another $\sim 1 - 3$ orders of magnitude). Critically, while the Dedner scheme requires some finite time and resolution to dissipate non-zero $\nabla \cdot \mathbf{B}$, the CG scheme instantaneously controls $\nabla \cdot \mathbf{B}$ across single resolution elements. As a result, in almost all cases, this means that the maximum $h_i |\nabla \cdot \mathbf{B}|/|\mathbf{B}|_i < 0.01$, even across arbitrarily sharp discontinuities. This eliminates the systematic offsets present in the Dedner scheme, and gives results which are, for all practical purposes in the tests here, indistinguishable from those obtained using CT methods which maintain $h_i |\nabla \cdot \mathbf{B}|/|\mathbf{B}|_i = 0$ to machine precision.

The CG method is easy to implement and quite general: it is applicable to any finite-volume or finite-element method with a well-defined reconstruction procedure (this includes our arbitrary Lagrangian-Eulerian FMF/MFV methods, Godunov and Galerkin-type finite-volume grid/AMR codes, moving-mesh methods, and finite-pointset methods). And it is independent of the “default” procedure used for gradient estimation: all that is needed is a sweep to calculate the terms needed for the correction tensor $\mathbf{G}$. Although the formal maintenance of $\nabla \cdot \mathbf{B} = 0$ is not as accurate as CT, it is compatible with arbitrary gradient definitions and slope limiters (these are usually implicitly limited to highly prescribed forms for the magnetic field in CT), is relatively inexpensive, and can trivially handle arbitrary mesh/point geometries, different numbers of spatial dimensions, and adaptive (non-uniform) timesteps. And because CT-type and flux-projection methods attempt to conserve (not minimize) $|\nabla \cdot \mathbf{B}|$, the injection/removal of magnetic energy, volume, or mass, and/or existence of complicated boundary conditions must be handled extremely carefully to avoid introducing divergences; our method actively corrects for divergences so can trivially handle these situations.

Finally, we have considered one simple implementation of the method, but it allows considerable freedom which merits further exploration. Different “preferred” gradients (we used a kernel-weighted least-squares estimate), penalty functions, iteration/convergence schemes (e.g. semi-implicit matrix methods), slope-limiters, and cleaning-field ($\psi$) terms in $S_0$ could be easily
considered and may well provide superior performance. Higher-order generalizations should also be straightforward, although the equations in this paper would need to be modified for a higher-order reconstruction used in determining the $\nabla \cdot \mathbf{B}$ error.

ACKNOWLEDGMENTS

Support for PFH was provided by the Gordon and Betty Moore Foundation through Grant #776 to the Caltech Moore Center for Theoretical Cosmology and Physics, an Alfred P. Sloan Research Fellowship, NASA ATP Grant NNX14AH35G, and NSF Collaborative Research Grant #1411920. Numerical calculations were run on the Caltech compute cluster “Zwicky” (NSF MRI award #PHY-0960291) and allocation TG-AST130039 granted by the Extreme Science and Engineering Discovery Environment (XSEDE) supported by the NSF.

REFERENCES

Brio, M., & Wu, C. C. 1988, Journal of Computational Physics, 75, 400
Dedner, A., Kemm, F., Kröner, D., Munz, C.-D., Schnitzer, T., & Wesenberg, M. 2002, Journal of Computational Physics, 175, 645
Mignone, A., & Tzeferacos, P. 2010, Journal of Computational Physics, 229, 2117
Monaghan, J. J. 2000, Journal of Computational Physics, 159, 290
Price, D. J., & Rosswog, S. 2006, Science, 312, 719
Tóth, G. 2000, Journal of Computational Physics, 161, 605
Stuff