

# A constrained-transport magnetohydrodynamics algorithm with near-spectral resolution.

Jason L. Maron<sup>1</sup>

Mordecai-Mark Mac Low<sup>2</sup>

Jeffrey S. Oishi<sup>3</sup>

*Department of Astrophysics, American Museum of Natural History, 79th Street at Central Park West, New York, NY 10024-5192*

## ABSTRACT

Numerical simulations including magnetic fields have become important in many fields of astrophysics. Evolution of magnetic fields by the constrained transport algorithm preserves magnetic divergence to machine precision, and thus represents one preferred method for the inclusion of magnetic fields in simulations. We show that constrained transport can be implemented with volume-centered fields and hyperresistivity on a high-order finite difference stencil. Additionally, the finite-difference coefficients can be tuned to enhance high-wavenumber resolution. Similar techniques can be used for the interpolations required for dealiasing corrections at high wavenumber. Together, these measures yield an algorithm with a wavenumber resolution that approaches the theoretical maximum achieved by spectral algorithms. Because this algorithm uses finite differences instead of fast Fourier transforms, it runs faster and isn't restricted to periodic boundary conditions. Also, since the finite differences are spatially local, this algorithm is easily scalable to thousands of processors. We demonstrate that, for low-Mach-number turbulence, the results agree well with a high-order, non-constrained-transport scheme with Poisson divergence cleaning.

---

<sup>1</sup>badger@amnh.org

<sup>2</sup>mordecai@amnh.org

<sup>3</sup>Also Department of Astronomy, University of Virginia, Charlottesville, VA; joishi@amnh.org

## 1. Introduction

Many astrophysical flows involve dynamically significant magnetic fields, such as molecular clouds, accretion disks, the galactic dynamo, jets, galaxy clusters, stellar dynamos and coronae, the solar wind and the interstellar medium. These problems tend to be three-dimensional, multiscale and turbulent, so there is an ongoing interest in developing high-resolution and efficient magnetohydrodynamics (MHD) algorithms for them. In this paper, we outline an extension of the constrained transport algorithm (Evans & Hawley 1988) to the combination of higher spatial order and zone-centered grids, and with resolution-enhanced tuned derivatives. We then describe how these measures fit together to yield an algorithm that closely approaches the theoretical maximum wavenumber resolution of spectral algorithms.

### 1.1. Constrained transport

The induction equation for a magnetic field  $\mathbf{B}$  and a velocity field  $\mathbf{V}$  in ideal MHD is

$$\partial_t \mathbf{B} = \nabla \times (\mathbf{V} \times \mathbf{B}). \quad (1)$$

Analytically, this equation conserves magnetic divergence:  $\partial_t(\nabla \cdot \mathbf{B}) = 0$ . However, this may or may not be the case for a finite-difference treatment of this equation. Tóth (2000) reviews the methods taken by various algorithms to treat the divergence in MHD simulations. A spectral code explicitly projects the Fourier components so that  $\hat{\mathbf{B}}(\mathbf{k}) \cdot \mathbf{k} = 0$ . For a finite difference code, the magnetic field can be evolved by a constrained transport scheme that preserves the magnetic divergence to machine precision (Evans & Hawley 1988). Alternatively, if the discretization doesn't conserve magnetic divergence, the divergence can be removed with measures such as periodic use of a Poisson solver (Brackbill & Barnes 1980), adding a divergence diffusion term  $\partial_t \mathbf{B} = \nabla \nabla \cdot \mathbf{B}$  to the magnetic evolution, or following an artificial and independently evolving divergence field (Dedner et al. 2002) to propagate divergence away from where it is produced and then dissipate it. The Powell scheme (Powell et al. 1999) adds a source term to advect divergence rather than let it grow in place. A finite difference code can also employ a vector potential  $\mathbf{A}$  such that  $\mathbf{B} = \nabla \times \mathbf{A}$ , in which case the magnetic divergence is automatically zero. This requires the use of a higher-order advection algorithm to ensure accurate second-derivatives, as is done in the Pencil code (Brandenburg & Dobler 2002).

We denote any finite difference scheme for MHD that explicitly conserves the magnetic divergence to machine precision as constrained transport (CT), and any scheme that does not as unconstrained transport (UT). Several variations of CT are possible. If the electric

field is differenced as a curl:  $\partial_t \mathbf{B} = -\nabla \times \mathbf{E}$ , then the magnetic divergence is preserved to machine precision for most grid types (see Appendix). Evans & Hawley (1988) introduced CT for staggered grids and Tóth (2000) showed that it works for centered grids as well (see the Appendix for explanation of centered and staggered grids). Londrillo & Del Zanna (2000) further showed that high-order CT is possible on staggered grids with a radius-two stencil. In this paper, we show that volume-centered CT is possible on arbitrarily large stencil sizes, with hyperresistivity, and that the resolution of this algorithm at moderately high order approaches the theoretical maximum exhibited by a spectral code.

In § 2, we discuss the specifics of the algorithm, and in § 3, we describe test simulations that demonstrate the capabilities of CT with high-order spatial derivatives.

## 2. Algorithms

Our algorithm is based on a constrained transport scheme, plus measures to enhance the resolution and maintain stability. High wavenumber resolution is achieved by a combination of high-order and tuned finite differences plus hyperdiffusivity, and stability is achieved by Runge-Kutta timestepping and hyperdiffusivity.

### 2.1. Timestepping

A high-order timestepping scheme for the evolution equations is essential for the stability of most algorithms. The time update for a variable  $Q(t)$  is  $Q(\Delta t) = Q(0) + \Delta t Q^*$ , where  $Q^*$  represents some estimate of  $\int_0^{\Delta t} \partial_t Q(t) dt$ . One example is the second-order Runge-Kutta scheme, which estimates  $Q(\Delta t/2) \approx Q(0) + (\Delta t/2)Q'(0)$ , and then identifies  $Q^* = Q'(\Delta t/2)$ . Another class of algorithms maintains the conservation of mass and momentum by computing fluxes through zone boundaries. A variety of techniques exist for time-extrapolating the fluxes at  $t = \Delta t/2$ , such as piecewise parabolic advection (Colella and Woodward 1984), Total Variation Diminishing (Harten 1983), Riemann solvers (Toro 1999), the method of characteristics (Stone & Norman 1992a, Hawley & Stone 1995), and many more. MHD poses a challenge to time extrapolation because there are seven or eight wavemode characteristics, depending on the technique used for treating magnetic divergence. In particular, the well-known method-of-characteristics algorithm interpolates along the Alfvén characteristic while neglecting the fast and slow mode characteristics. For our simulations, we use Runge-Kutta for time-extrapolation because it doesn't invoke any diffusive spatial interpolations (§ 2.2), and because it automatically captures all three MHD wavemode types. In our demonstration

implementation, we use second-order Runge-Kutta while the Pencil code (Brandenburg & Dobler 2002) uses third-order, although either order has proven successful.

## 2.2. Diffusivity

A common class of algorithms is based on momentum fluxes that are time-extrapolated with upwind spatial interpolations. The errors from the interpolations required for these flux transport algorithms produce an intrinsic diffusivity that can stabilize the evolution, even in the absence of any explicit diffusive terms. The nature and magnitude of the diffusivity has been characterized in Zhong (1998) and Dobler et al (2006).

Runge-Kutta timestepping, on the other hand, has no spatial interpolations, and thus no intrinsic diffusivity. One then generally needs an explicit stabilizing diffusivity. One has various options for the form of this diffusivity, with Laplacian or hyper-Laplacian typically chosen. These diffusivities have the benefit that their magnitude is easily characterized, and the diffusive coefficient can be tuned to have the minimum value necessary to preserve stability. Consider:

$$\partial_t \mathbf{V} = \nu_2 \nabla^2 \mathbf{V} - \nu_4 \nabla^4 \mathbf{V} + \nu_6 \nabla^6 \mathbf{V} - \nu_{[4]} \nabla^{[4]} \mathbf{V}. \quad (2)$$

Let the Fourier components be  $\hat{\mathbf{V}}(\mathbf{k})$ , where  $\mathbf{k} = \{k_x, k_y, k_z\}$  is the wavenumber. They evolve according to

$$\partial_t \hat{\mathbf{V}}(\mathbf{k}) = -\nu_2 \mathbf{k}^2 \hat{\mathbf{V}}(\mathbf{k}) - \nu_4 \mathbf{k}^4 \hat{\mathbf{V}}(\mathbf{k}) - \nu_6 \mathbf{k}^6 \hat{\mathbf{V}}(\mathbf{k}) - \nu_{[4]} (k_x^4 + k_y^4 + k_z^4) \hat{\mathbf{V}}(\mathbf{k}). \quad (3)$$

The  $\nu_2$  term is the Laplacian viscosity and the others are higher-order hyperdiffusivities. Specifically,  $\nabla^4 = \nabla^2 \nabla^2$ ,  $\nabla^6 = \nabla^2 \nabla^2 \nabla^2$ , and  $\nabla^{[4]} = \partial_x^4 + \partial_y^4 + \partial_z^4$ . They differ in that the higher the order, the more selective they are in diffusing high-k structure while preserving low-k structure. Equivalently stated, high-order hyperdiffusivities erase small-scale structure without affecting the larger scales. For many physical applications, such as the turbulent magnetic dynamo, the large scale structure is unaffected by the nature of the small-scale diffusivity (Haugen & Brandenburg 2004), and so in these instances, the use of hyperdiffusivity instead of Laplacian diffusivity enhances the wavenumber resolution.

The  $\nabla^4$  operator is spherically symmetric in Fourier space while the  $\nabla^{[4]}$  operator is not. This affects the maximum-possible timestep because in order to be advectively stable, the Courant condition implies that the product  $|k| \Delta t$  must be less than a given value, and so the high-k corners of the 3D Fourier cube are the most vulnerable to advective instability. In these corners, the  $\nabla^4$  term delivers more diffusion than the  $\nabla^{[4]}$  term, but with a cost of twice as many finite difference operations.

Hyperdiffusivity acts together with high-order finite differences to enhance the resolution of a simulation. High-order finite differences allows structure to be finite differenced with less error, and hyperdiffusivity allows this structure to evolve with less dissipation than it would with Laplacian diffusivity.

Hyperdiffusivity can benefit the timestep as well as the resolution. Suppose one chooses the Laplacian term to provide the dominant fraction of the diffusivity. One typically then evaluates by trial and error the minimum value of  $\nu_2$  and the maximum timestep  $\Delta t$  that one can get away with to maintain stability. Once these values are chosen, the addition of a small measure of hyperdiffusivity, small enough so as not to contribute significantly to the total diffusivity, but large enough to affect the highest-k structure, increases the maximum stable timestep by about 50%. This technique works because the highest-wavenumber structure is the most vulnerable to instability, and so the timestep depends most critically on the value of the diffusivity for these wavenumbers.

One could additionally note that “maximum stable timestep” is not sharply defined. For instance, a timestep at the cusp of stability might be unstable, but only after perhaps more than 10 timesteps. Also, the maximum stable timestep depends on the maximum value of the velocity in the simulation, which is changing in time.

### 2.3. Magnetohydrodynamic equations

The equations of incompressible MHD with diffusivity and hyperdiffusivity are

$$\partial_t \mathbf{V} = -\mathbf{V} \cdot \nabla \mathbf{V} + \mathbf{B} \cdot \nabla \mathbf{B} + \nu_2 \nabla^2 \mathbf{V} - \nu_4 \nabla^4 \mathbf{V} - \nu_{[4]} \nabla^{[4]} \mathbf{V} + \nu_6 \nabla^6 \mathbf{V} + \nu_{[6]} \nabla^{[6]} \mathbf{V} + \nu_D \nabla \nabla \cdot \mathbf{V} \quad (4)$$

$$\partial_t \mathbf{B} = \nabla \times \left[ \mathbf{V} \times \mathbf{B} - \eta_2 \mathbf{J} + \eta_4 \nabla^2 \mathbf{J} - \eta_6 \nabla^4 \mathbf{J} - \eta_{2,[4]} \nabla^{[4]} \mathbf{J} \right] + \eta_D \nabla \nabla \cdot \mathbf{B} \quad (5)$$

$$\mathbf{J} = \nabla \times \mathbf{B} \quad (6)$$

The variables are defined in Table (1). The magnetic equation (5) is written as a curl so that the constrained transport scheme can preserve the magnetic divergence (Appendix). The remaining term proportional to  $\eta_D$  outside of the curl has the effect of diffusing away any magnetic divergence present, and does nothing otherwise (§ 2.4). Similarly, the term proportional to  $\nu_D$  term can be used to help quell kinetic divergence in incompressible simulations, although for the simulations in this paper we use a spectral projection to accomplish this.

We clarify the meaning of equation (5), by assuming that  $\nabla \cdot \mathbf{B} = 0$  :

$$\partial_t \mathbf{B} = \mathbf{B} \cdot \nabla \mathbf{V} - \mathbf{V} \cdot \nabla \mathbf{B} - \mathbf{B} \nabla \cdot \mathbf{V}$$

$$+ \eta_2 \nabla^2 \mathbf{B} - \eta_4 \nabla^4 \mathbf{B} + \eta_6 \nabla^6 \mathbf{B} + \eta_{2,[4]} \nabla^2 \nabla^{[4]} \mathbf{B} + \eta_D \nabla \nabla \cdot \mathbf{B} \quad (7)$$

The  $\mathbf{J}$  terms in equation (5) are seen to have the role of diffusivities in equation (7). In the simulations in § 3, we use equation (5) for constrained transport and equation (7) for unconstrained transport.

Time centering of the staggered-grid and centered-grid constrained transport equations encounter a similar circumstance as with the kinetic equation. In the staggered-grid constrained transport configuration employed by Hawley and Stone (1995), the electric fields are spatially interpolated from the nearest 8 velocity and magnetic field vectors, and they are also time-interpolated to the next half step with an Alfvén wave Method of Characteristics scheme (Hawley and Stone 1995). The implicit diffusivity inherent in these interpolations stabilizes the magnetic field evolution, even in the absence of explicit diffusivity. For Runge-Kutta timestepping on a centered grid, there are no spatial interpolations in constructing the electric field, and no accompanying intrinsic diffusivity. Some form of diffusivity is then generally required to maintain stability.

## 2.4. Divergence

In a constrained transport simulation, the magnetic divergence remains zero to machine precision. For unconstrained transport, we set the magnetic divergence to zero with a Poisson projection in Fourier space once every four timesteps. In tests, we found that the evolution is virtually identical whether the divergence is removed once every timestep or once every four timesteps (Maron 2004), and in both cases the fractional magnetic divergence remains below one percent. For both constrained and unconstrained transport simulations of incompressible flow that we describe below, the kinetic divergence is removed once every timestep in Fourier space, although for quasi-incompressible flow, it suffices to do this only once every few timesteps (Maron 2004).

The divergence diffusivity term  $\partial_t \mathbf{B} = \eta_D \nabla \nabla \cdot \mathbf{B}$  is helpful for reducing the effect of magnetic divergence in the timesteps between Poisson projections. It diffuses any magnetic divergence present, but otherwise does nothing. If each Fourier component has the form  $\hat{\mathbf{B}}(\mathbf{k}) = \hat{\mathbf{B}}_{\parallel} \mathbf{k}/|\mathbf{k}| + \hat{\mathbf{B}}_{\perp}$ , then the  $\nabla \nabla \cdot \mathbf{B}$  term evolves  $\hat{\mathbf{B}}$  as  $\partial_t \hat{\mathbf{B}}_{\parallel} = -k^2 \hat{\mathbf{B}}_{\parallel}$  and  $\partial_t \hat{\mathbf{B}}_{\perp} = 0$ . The divergence diffusivity is also helpful for a constrained transport simulation. Without it, CT preserves the magnetic divergence but does nothing to remove it. With the divergence diffusivity, any divergence present is diffused away.

## 2.5. Initialization

The constrained transport algorithm evolves the magnetic field in such a way as to conserve magnetic divergence. If the initial conditions have zero divergence, the divergence remains zero indefinitely. Even if any monopoles do grow slowly, they can be removed at negligible computational expense with a Fourier projection, say, once every thousand timesteps. Care must be taken with the initial conditions, however, because the divergence depends on the method for approximately evaluating derivatives. One may use, for example, a spectral or a finite difference divergence operator. For constrained transport to work, the derivatives used for making the initial conditions divergenceless must be the same as those used in the simulation. Even an analytic function with vanishing divergence may not have vanishing numerical divergence.

To initialize the magnetic field, we apply the following procedure. In three dimensions, let the wavenumbers for the magnetic field Fourier components  $\hat{\mathbf{B}}$  be  $\mathbf{k}$ , the length of each side of the simulation volume be  $\mathbf{L}$ , and the number of grid zones on each side be  $\mathbf{N}$ . If the magnetic divergence is defined spectrally, then the constraint on the magnetic field is  $\hat{\mathbf{B}}(\mathbf{k}) \cdot \mathbf{k} = 0$ . For a finite difference derivative the constraint is slightly different. If we use the high-order finite difference from § 2.6 (eq. 11), we can define an adjusted dimensionless wavenumber by

$$k_i^* = \sum_{j=-S}^S m_j \sin(k_i L_i j / N_i), \quad (8)$$

and then the finite difference condition for zero divergence is

$$\hat{\mathbf{B}}(\mathbf{k}) \cdot \mathbf{k}^* = 0. \quad (9)$$

The initial conditions for a constrained transport simulation should satisfy this condition, which is easily implemented in Fourier space.

## 2.6. High-wavenumber finite differences

High-order spatial derivatives can enhance the wavenumber resolution of a simulation. To quantify this, define a function  $f_j(x_j)$  on a periodic grid  $x_j = j$ , with  $j$  an integer. Then construct a finite difference derivative  $f'(0)$  at  $x = 0$  using a radius- $S$  stencil. The familiar result for the gradient on a radius-one stencil is  $f'(0) \sim (f_1 - f_{-1})/2$ , which is obtained from fitting a polynomial of order 2 to  $f_j$  at  $j = 0$ . For an order 4 polynomial on a radius-two stencil,

$$f'(0) \sim \frac{1}{12}f_{-2} - \frac{2}{3}f_{-1} + \frac{2}{3}f_1 - \frac{1}{12}f_2. \quad (10)$$

For a stencil of order  $S$ ,

$$f'(0) \sim \sum_{j=-S}^S m_j f_j \quad (11)$$

where  $m_{-j} = -m_j$ . The coefficients for a radius-three stencil are  $\{m_1, m_2, m_3\} = \{3/4, 3/20, 1/60\}$ .

Consider the finite-difference error at  $x = 0$  for a Fourier mode  $\sin(\pi kx)$ . (Cosine modes can be ignored because they don't contribute to the derivative at  $x = 0$ .) We scale the wavenumber  $k$  to grid units so that  $k = 1$  corresponds to the maximum (Nyquist) wavenumber expressible on the grid. The finite difference formula (eq. 11) gives

$$f'_k(0) \sim 2 \sum_{j=1}^S m_j \sin(\pi jk) \quad (12)$$

whereas the correct value is  $k\pi$ . The spectral procedure of taking the derivative by transforming to Fourier space and back gives the correct value up to  $k = 1$ , but at a cost of  $5 \log_2 N$  floating point operations per grid point per transform, where  $N$  is the number of grid points, whereas the finite difference derivative (equation 11) on a radius  $S$  stencil costs  $3S - 1$  floating point operations. Figure (1) exhibits the accuracy of finite difference derivatives of different orders as a function of wavenumber. The wavenumber resolution increases with polynomial order.

The polynomial fit can be tuned by adjusting the coefficients to enhance high-wavenumber accuracy substantially at the expense of a negligible loss of accuracy at low wavenumbers (Maron 2004). Table 2 gives the coefficients for finite difference operators at various orders, along with the maximum wavenumber for which they are 0.5% accurate. In Figure 1 we also show the wavenumber accuracy for a radius-three stencil using tuned coefficients, designed to have a relative precision of  $< 0.5\%$  for  $k = 0$  to 0.50. We will see in § 3.1.2 that a simulation based on this tuned radius-three scheme performs better than a radius-four simulation with conventional coefficients. We note that power spectra do not reveal a difference between turbulence simulations with tuned and untuned coefficients, because errors in the derivative manifest as an advective dispersion rather than as a diffusivity. One has instead to examine the fields in real space rather than in Fourier space.

## 2.7. Dealiasing

Nonlinear terms in the MHD equations such as  $\mathbf{V} \cdot \nabla \mathbf{V}$  suffer an aliasing error for high-wavenumber structure. We first illustrate this with a 1D example before treating the 3D case (Canuto et al. 1987). Let  $A_j$  and  $B_j$  be discrete functions on a one-dimensional grid

<b>V</b>	Velocity	<b>B</b>	Magnetic field
		<b>J</b>	Current ( $\nabla \times \mathbf{B}$ )
$\nu_2$	Laplacian Viscosity	$\eta_2$	Laplacian resistivity
$\nu_4$	Hyperviscosity for $\nabla^4$	$\eta_4$	Hyperresistivity for $\nabla^4$
$\nu_{[4]}$	Hyperviscosity for $\nabla^{[4]}$	$\eta_{[4]}$	Hyperresistivity for $\nabla^{[4]}$
$\nu_6$	Hyperviscosity for $\nabla^6$	$\eta_6$	Hyperresistivity for $\nabla^6$
$\nu_{2,[4]}$	Hyperviscosity for $\nabla^2 \nabla^{[4]}$	$\eta_{2,[4]}$	Hyperresistivity for $\nabla^2 \nabla^{[4]}$
$\nu_{[6]}$	Hyperviscosity for $\nabla^{[6]}$	$\eta_{[6]}$	Hyperresistivity for $\nabla^{[6]}$
$\nu_D$	Divergence viscosity	$\eta_D$	Divergence resistivity

Table 1: Variables in the equations of MHD

Operation	$m_0$	$m_1$	$m_2$	$m_3$	$m_4$	$m_5$	$m_6$	$m_7$	$m_8$	$k_{max}$
$\partial/\partial x$ (P1)	0	0.50000								0.12
$\partial/\partial x$ (P2)	0	0.66667	-0.08333							0.24
$\partial/\partial x$ (P3)	0	0.75000	-0.15000	0.01667						0.36
$\partial/\partial x$ (P4)	0	0.80000	-0.20000	0.03810	-0.00357					0.48
$\partial/\partial x$ (P5)	0	0.83333	-0.23810	0.05952	-0.00992	0.00079				0.60
$\partial/\partial x$ (P8)	0	0.88889	-0.31111	0.11313	-0.03535	0.00870	-0.00155	0.00018	-0.00001	0.50
$\partial/\partial x$ (T3)	0	0.81796	-0.21324	0.03683						0.50
$\partial/\partial x$ (T8)	0	0.95951	-0.42312	0.22746	-0.12450	0.06461	-0.03004	0.01156	-0.00273	0.70
$\partial/\partial x$ (T8)	0	0.96685	-0.43635	0.24410	-0.14168	0.07976	-0.04148	0.01884	-0.00536	0.80
$\partial^2/\partial x^2$ (P3)	-2.72222	1.5	-0.15	0.01111						
$\partial^4/\partial x^4$ (P3)	9.33333	-6.5	2.	-0.16667						
$\partial^6/\partial x^6$ (P3)	-20.	15.	-6.	1.						
Interpolation (P3)	0	0.58594	-0.09766	0.01172						0.33
Interpolation (T3)	0	0.60103	-0.12312	0.02214						0.50
Interpolation (T8)	0	0.63099	-0.19580	0.10149	-0.05770	0.03248	-0.01709	0.00793	-0.00233	0.80

Table 2: Coefficients for finite difference operations. The column “ $k_{max}$ ” indicates the maximum wavenumber for which the operator is no worse than 0.5% accurate. For the odd-order derivatives,  $m_{-j} = -m_j$ , and for the even-order derivatives as well as the interpolations,  $m_{-j} = m_j$ . The interpolation coefficients are for interpolation to a point halfway between two grid zones using the nearest  $S$  grid points on each side (§ 2.8)

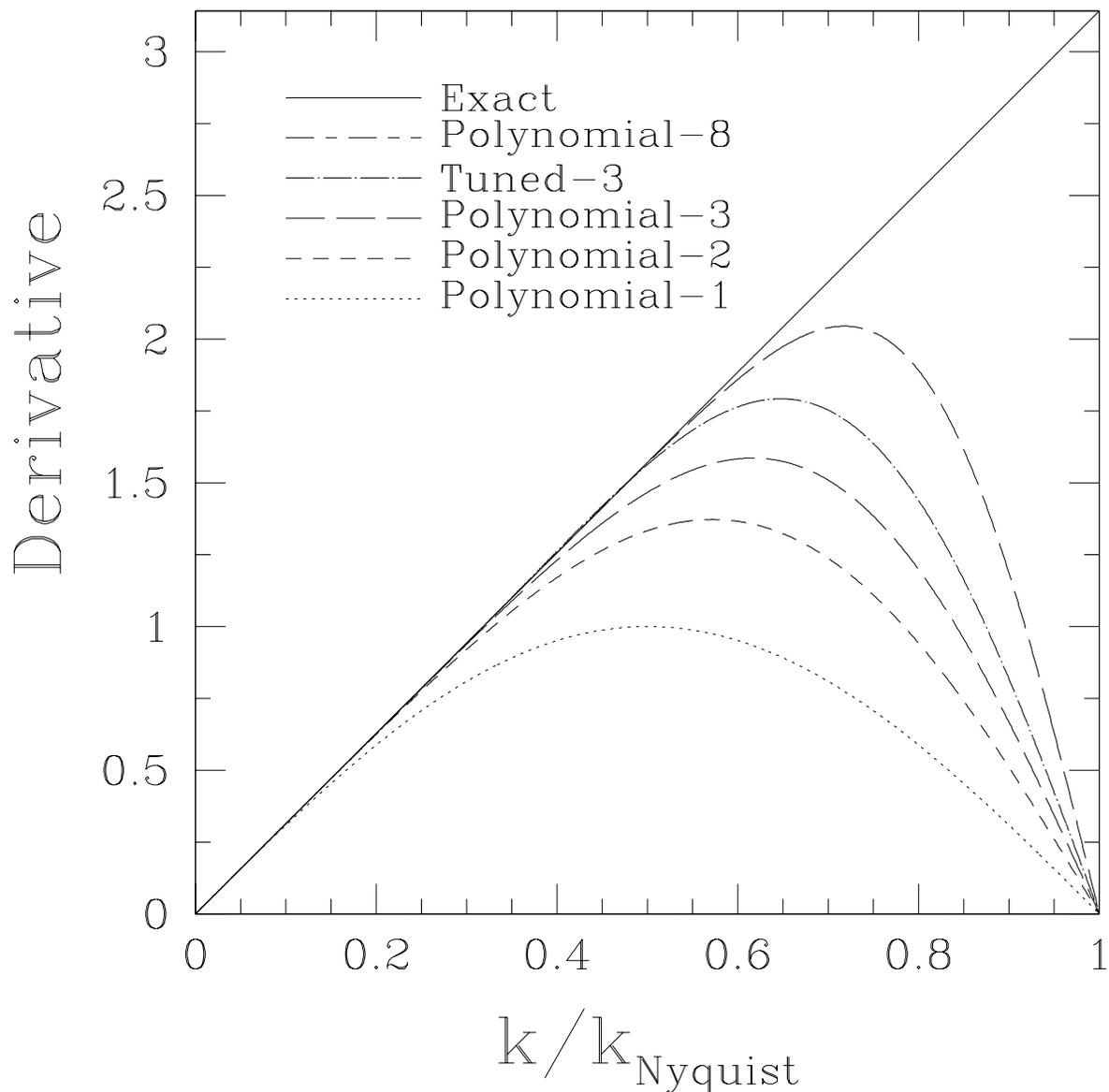


Fig. 1.— The accuracy of finite-difference derivatives as a function of the Nyquist-scaled wavenumber  $k$ . “Polynomial- $N$ ” denotes a radius- $N$  stencil with polynomial-based coefficients, and “Tuned- $N$ ” denotes a radius- $N$  stencil with tuned coefficients (§ 2.6 and Table 2). The “Exact” curve is the derivative as evaluated by a Fourier transform. A radius- $N$  stencil with polynomial-based coefficients is accurate to order  $2N$ .

with  $N = 16$  grid points:  $1 \leq j \leq N$ , and with accompanying Fourier modes ranging from  $-N/2 \leq s \leq (N/2) - 1$ . The Fourier expansion of  $A$  is

$$A_j = \sum_{s=-N/2}^{N/2-1} \hat{A}_s \exp(2\pi i j s / N) \quad (13)$$

and similarly for  $B_j$ . Let  $A$  and  $B$  each be composed of a single Fourier mode with  $s = 6$ . The product  $C = AB$  is then composed of a single Fourier mode with  $s = 12$ , but in this discrete representation, this is equivalent to  $s = -4$ . This remapping of high wavenumber modes to low wavenumber modes is known as aliasing error. Note that  $s = 6$  is within the Nyquist limit of  $|s| \leq N/2$  while the product mode  $s = 12$  is not. While  $A_j$  and  $B_j$  might be resolvable on the grid, their product need not be. This can be fixed by truncating before and after the product all modes outside  $|s| \leq N/3$  to ensure that no modes in the product will exceed the Nyquist limit. A spectral code can make this truncation because the fields are in Fourier space, while a finite difference code cannot. As a practical matter it suffices to set the diffusivities high enough so that negligible structure exists outside the  $N/3$  aliasing limit.

Alternatively, the aliasing problem can be remedied with a staggered-grid correction (Canuto et al. 1987), and then the truncation isn't necessary. Additionally, this allows us to simulate structure beyond  $k = 2/3$ . We first exhibit this procedure in 1D and then extend it to 3D below. To implement the staggered-grid correction, first construct the usual product  $C_j = A_j B_j$ . Then interpolate the centered grids  $A_j$  and  $B_j$  to the staggered grids  $A_{j+1/2}$  and  $B_{j+1/2}$ , multiply to produce  $C_{j+1/2}$  and transform back to the centered grid  $C'_j$ . Finally, combine the centered and staggered results as  $C_j \leftarrow (C_j + C'_j)/2$ . This result is free from aliasing error for all Fourier modes, and so no Fourier-space truncation is necessary. In Fourier space, the fields on the staggered grid can be computed exactly by applying a phase shift to each Fourier mode. A finite difference scheme can accomplish this with the high-order interpolation discussed in § 2.8.

In three dimensions the aliasing error can be eliminated by truncating the Fourier modes outside  $|\vec{s}| < \vec{N}/3$ , where  $\vec{s} = \{s_x, s_y, s_z\}$  and  $\vec{N} = \{N_x, N_y, N_z\}$ . The grid-shift correction is more complicated. Seven grid shifts are needed to completely correct the error, but it is more efficient to settle for a more limited correction involving just one shift (Canuto et al. 1987). Interpolate  $A$  and  $B$  to the staggered grid  $\{j_x + 1/2, j_y + 1/2, j_z + 1/2\}$ , multiply, return to the centered grid  $\{j_x, j_y, j_z\}$ , and as in the one-dimensional case, average the result with the product carried out on the original centered grid. This yields an alias-free result if accompanied by a Fourier truncation of all modes outside  $|\vec{s}| < \sqrt{3}\vec{N}/2$ , which is 94 % of the Nyquist limit. For a finite difference code, the diffusivity can be set to minimize the energy outside the truncation zone. When the three-dimensional, staggered-grid, aliasing correction

is combined with high-order, volume-centered, constrained transport, magnetic divergence is still preserved to machine precision. This scheme uses the high-order finite differences and interpolations discussed in § 2.6 and § 2.8.

## 2.8. Interpolation

High-order interpolation allows implementation of the staggered aliasing correction discussed in § 2.7. It can also be used for doubling the grid size. In this case, the new points in the doubled three-dimensional grid can be generated with a set of interpolations along and diagonal to the three grid axes. Doubling the grid is also useful for interpolating to points that are not exactly halfway between grid points. For example, the grid can be doubled with the high-order interpolation discussed above, and then a simpler algorithm can be used to further interpolate to a point anywhere within the refined grid. We used this technique to trace the path of magnetic fieldlines in our studies of cosmic ray diffusion (Maron 2003), and also for doubling the resolution of turbulent dynamo simulations in Maron et al. (2004). Lastly, we have found that high-order tuned derivatives and interpolations are useful for post-simulation analysis of timeslice data without the need for computationally expensive fast Fourier transforms.

A high-order interpolation to a point halfway between two grid points can be accomplished with

$$f_{j+1/2} = \sum_{i=1}^S m_i (f_{j+i} + f_{j+1-i}). \quad (14)$$

The error in the interpolation as a function of wavenumber can be calculated by applying equation (14) to a cosine mode centered on  $f_{j+1/2}$ :  $f_i = \cos(\pi k(i - (j + 1/2)))$ . The interpolated value is

$$f_{j+1/2}^* = \sum_{i=1}^S m_j \cos(\pi i k) \quad (15)$$

whereas the correct value is  $f_{j+1/2} = 1$ . Table 2 gives coefficients for an interpolation based on a radius-three stencil from a polynomial fit to  $f_j$ , in the row labeled “Interpolation (P3).” Tuning the coefficients can improve the wavenumber resolution of the interpolation in a similar manner as was done for the derivative. Tuned coefficients for radius 3 and 8 stencils are given in Table (2) in the rows labeled T3 and T8, with the column labeled  $k_{max}$  giving the maximum wavenumber for which the interpolation is 0.5% accurate.

## 2.9. Operation count

A finite difference derivative on a radius  $S$  stencil costs  $S$  multiplies and  $2S - 1$  adds. Since add and multiply units tend to come in pairs on most modern machines, this is effectively  $4S - 2$  floating point operations per grid cell. A code with  $F$  finite difference convolutions per timestep involves  $F(4S - 2)$  floating point operations.

The computation cost per timestep scales as the number of derivatives computed per grid element per time update. For constrained transport, one needs to calculate three derivatives for  $\partial_i \rho$ , three for  $\partial_i e$ , nine for  $\partial_i \mathbf{V}_j$ , six for  $\nabla \times \mathbf{B}$  and six for  $\nabla \times \mathbf{E}$ . Diffusion terms such as  $\nabla^2 \mathbf{V}$  and  $\nabla^2 \mathbf{J}$  aren't included in the tally because they need only be applied every few timesteps (Maron 2004). The same is true for the Fourier-space Poisson projection to make the fields divergenceless. Maron (2004) found that the results were effectively identical whether these terms were applied once every timestep or once every four timesteps.

In Table 3 we list the number of derivatives computed per timestep for various classes of algorithms. Pencil is a vector potential code (Brandenburg & Dobler 2002) that calculates  $\mathbf{J}$  directly from  $\mathbf{A}$  to take advantage of the cache memory. This involves 15 distinct 2-level derivatives computed from a 2D stencil. For a radius- $S$  finite difference, a 2D derivative involves  $4S^2 - 1$  adds, compared to  $2S - 1$  adds for a conventional 1D derivative. We don't consider multiples because they are always less numerous than adds, and because CPUs tend to feature add and multiply units in pairs. As a result, a 2D derivative corresponds to  $2S + 1$  times the computational effort of a conventional 1D derivative. Thus, the calculation of  $\mathbf{J}$  with  $S = 3$  corresponds effectively to about 105 conventional derivatives. Alternatively, a vector potential algorithm can instead calculate  $\mathbf{B}$  from  $\mathbf{A}$  and then  $\mathbf{J}$  from  $\mathbf{B}$  in separate stages, at a possible cost of extra memory latency. In table 3, we denote a vector potential algorithm that calculates  $\mathbf{J}$  directly from  $\mathbf{A}$  as “1-stage,” and a vector potential algorithm that calculates  $\mathbf{B}$  from  $\mathbf{A}$  and then  $\mathbf{J}$  from  $\mathbf{B}$  as “2-stage.” The constrained transport algorithm evolves  $\mathbf{V}$  and  $\mathbf{B}$  according to equations (4) and (5), and the unconstrained transport algorithm evolves  $\mathbf{V}$  and  $\mathbf{B}$  according to equations (4) and (7). The spectral algorithm computes 15 Fourier transforms instead of derivatives and thus doesn't directly compare with the finite difference codes.

A vector potential code such as Pencil can be adapted to the algorithm described here by substituting equation (5) for the vector potential equation, and by changing the finite difference coefficients from the polynomial-based values to the tuned values. This allows one to compute boundary conditions using the magnetic field instead of the vector potential. However, there is an extra round of inter-processor communication associated with calculating  $\mathbf{J}$  from  $\mathbf{B}$  and then applying the curl.

A convenient unit for execution speed is Kilo-grid Elements per CPU GigaFLOPS per Second (KEGS). A constrained transport code with a radius-three stencil and two Runge-Kutta steps per timestep would run ideally at a speed of 1850 KEGS. The actual speed is slower of course because of overhead and because the finite difference convolutions don't run at the peak floating point speed. In benchmarks, we have observed that the Pencil code, which has 3 Runge-Kutta stages, runs at a speed of around 100 KEGS for serial operation and

50 KEGS for massively parallel operation. For comparison, the highly-optimized spectral-MHD code Tullu (Maron & Goldreich 2001), has a speed of around 80 KEGS in serial and 40 KEGS in parallel (Maron 2004).

### 3. Simulations

#### 3.1. The turbulent nonhelical dynamo

Our first test model is the turbulent, nonhelical, MHD dynamo (Batchelor 1950), which is the magnetic analogue of the Kolmogorov cascade for hydrodynamic turbulence. The Kolmogorov cascade is the long-term, steady state of hydrodynamic turbulence that is forced at the large scale and dissipated by viscosity at the small scale, and it has an energy spectrum of  $E(k) \sim k^{-5/3}$ . The nonhelical MHD dynamo has the same setup but includes a spatially homogeneous magnetic field with unit magnitude and zero mean. Maron et al. (2004) found that the steady state of this system has a kinetic energy spectrum of  $E(k) \propto k^{-2}$ , where the energy is dominantly in large-scale eddys, and a magnetic spectrum with the energy predominantly in the smallest-scale (resistive-scale) magnetic structures. The kinetic and magnetic spectra are shown in figure 2.

Algorithm	Terms	Derivatives
Constrained transport	$\partial_i \rho, \partial_i e, \partial_i \mathbf{V}_j, \nabla \times \mathbf{B}, \nabla \times \mathbf{E}$	27
Unconstrained transport	$\partial_i \rho, \partial_i e, \partial_i \mathbf{V}_j, \partial_i \mathbf{B}_j, \partial_i (\nabla \cdot \mathbf{B})$	30
2-stage vector potential	$\partial_i \rho, \partial_i e, \partial_i \mathbf{V}_j, \nabla \times \mathbf{A}, \nabla \times \mathbf{B}$	27
1-stage vector potential (Pencil)	$\partial_i \rho, \partial_i e, \partial_i \mathbf{V}_j, \nabla \times \mathbf{A}, \nabla \times \nabla \times \mathbf{A}$	120
Spectral	15 Fourier transforms	N/A

Table 3: The number of derivatives computed per time update (see § 2.9).

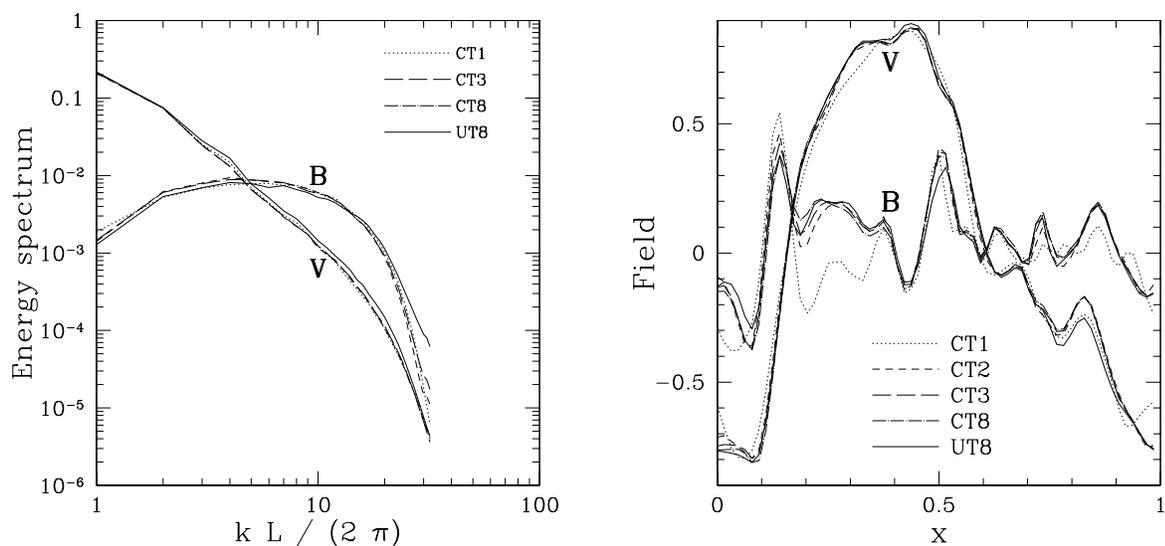


Fig. 2.— (a) The kinetic (**V**) and magnetic (**B**) spectra for a set of simulations with identical initial conditions, after two crossing times. (b) The value of  $B_y$  along a grid line parallel to the  $x$ -axis after 0.4 crossing times, for the same set of simulations. The high-order constrained and unconstrained transport simulations (CT3, CT8 & UT8) more closely resemble each other than they do the low-order constrained transport simulations (CT1 & CT2).

### 3.1.1. Models

We ran a set of simulations of forced homogeneous MHD turbulence (Table 4) with and without constrained transport and at various spatial orders to test the effectiveness of high-order constrained transport. Each simulation has the same timestep, viscosity and resistivity. The grid used in all cases has  $64^3$  zones covering a periodic unit cube. The forcing power, density, initial RMS velocity and magnetic field are all unity, and they remain so in the long-term steady state.

The initial state is the same for all simulations. It was taken from a simulation of forced magnetized turbulence (§ 3.1) in the long-term steady state. The magnetic divergence is zero in Fourier space:  $\hat{\mathbf{B}}(\mathbf{k}) \cdot \mathbf{k} = 0$ . The initial state was modified slightly for the constrained transport simulations to make it magnetically divergenceless according to the finite-difference derivative (§ 2.5). This amounts to a very small adjustment in the fields, much less than the difference between the fields after a crossing time of evolution.

Constrained transport needs some form of diffusivity, either Laplacian or hyperdiffusivity, to maintain stability. With diffusivity, it is stable indefinitely, whereas it is unstable without it. To establish this, we evolved the MHD equations with constrained transport for ten crossing times stably. The diffusivities used in these models are Laplacian viscosity  $\nu_2 = 10^{-3}$ , hyperviscosity  $\nu_{[4]} = 2.5 \times 10^{-8}$  and hyperresistivity  $\eta_{2,[4]} = 7 \times 10^{-12}$ , which we empirically find are the minimum values that maintain stability for  $V_{RMS} = B_{RMS} = 1$  on a  $64^3$  grid. Expressed in dimensionless form:  $\bar{\nu}_n = \nu_n / (V_{RMS} dx^{n-1})$  and  $\bar{\eta}_n = \eta_n / (V_{RMS} dx^{n-1})$ , these are  $\bar{\nu}_2 = .064$ ,  $\bar{\eta}_{[4]} = 6.6 \cdot 10^{-3}$  and  $\bar{\eta}_{2,[4]} = 7.5 \cdot 10^{-3}$ .

The forcing is the same as used by Maron et al. (2004). A random forcing field is added to the velocity every timestep. The spectrum of the forcing field is  $k^{-5/3}$ , truncated 2.5 lattice units from the origin in Fourier space, and the Fourier components have random phases. The forcing power, simulation volume and density are unity, which yields RMS velocity and magnetic fields of order unity (Maron 2004).

### 3.1.2. Results

The diffusivities are given in § 3. We plot the kinetic and magnetic spectra in figure 2(a). The spectra are very similar for constrained and unconstrained transport simulations, and also for different orders. However the spectra alone do not distinguish between simulations of different orders because an error in the derivative manifests itself as an advective dispersion rather than as a diffusivity (§ 2.6). One instead has to examine the fields in real space.

Tag	Method	Stencil radius	Finite difference technique	$k_{max}$	$L_2$ norm
CT1	Constrained transport	1	Polynomial	0.12	0.561
CT2	Constrained transport	2	Polynomial	0.24	0.193
CT3	Constrained transport	3	Polynomial	0.34	0.083
CT4	Constrained transport	4	Polynomial	0.40	0.042
CT5	Constrained transport	5	Polynomial	0.44	0.023
CT8	Constrained transport	8	Polynomial	0.56	0
CT3t1	Constrained transport	3	Tuned	0.50	0.034
CT3t2	Constrained transport	3	Tuned	0.52	0.037
UT8	Unconstrained transport	8	Polynomial	0.56	0.267

Table 4: Index of simulations. CT and UT denote constrained and unconstrained transport. The column labeled “finite difference technique” specifies whether the finite difference coefficients are from a polynomial fit or if they have been tuned to enhance high-wavenumber accuracy (§ 2.6).  $k_{max}$  denotes the largest wavenumber for which derivatives are 0.5% accurate. The  $L_2$  norms are taken with respect to UT8, as discussed in § 3.1.2 The simulations with tuned coefficients are CT3t1, which has  $k_{max}$  set to 0.50, and CT3t2, which has  $k_{max}$  set to 0.52.

In Figure 2(b), we compare the magnetic fields at  $t=0.4$  crossing times. For the comparison, we examine the difference between the fields integrated over space by computing the  $L_2$  norm between simulations  $i$  and  $j$  :

$$A_{ij}^2 = \frac{\int [B_y(i) - B_y(j)]^2 d(\text{Vol})}{\int B_y(i)^2 d(\text{Vol}) + \int B_y(j)^2 d(\text{Vol})}. \quad (16)$$

Stone et. al. (1992b) argue that this kind of comparison is more meaningful than merely plotting the overlay of both fields. The constrained transport simulation with polynomial-based finite differences on a radius-eight stencil (CT8 in Table 4) serves as the basis of comparison. We compare the constrained transport simulations to an unconstrained transport simulation on a radius-eight tuned finite-difference stencil (UT8). We use UT8 as a stand-in for the spectral algorithm because of its high wavenumber resolution.

The spectral algorithm delivers the highest-attainable resolution because spectral derivatives are exact for all wavenumbers. With this, a 3D spectral simulation without an aliasing grid-shift correction can resolve structure up to  $k=2/3$ , and with a grid-shift correction it can resolve up to  $k=.94$  (Canuto 1987). The spectral algorithm can also set the magnetic divergence to zero in Fourier space at negligible cost. Unconstrained transport does not explicitly conserve magnetic divergence and so in model UT8 the divergence is cleaned with a Fourier projection every timestep. We also tried applying the correction every fourth timestep and with virtually identical results. The radius-eight stencil of UT8 yields derivatives that are accurate up to  $k=.56$ .

The  $L_2$  norms given in Table (2) show how the simulations progressively approach the CT8 result as the stencil size increases. The match is poor for CT1 and better for CT3. We also note that the radius-three simulation with tuned derivatives (CT3t) performs better than the radius-four simulation with polynomial-based derivatives, establishing the effectiveness of tuned derivatives. This can also be qualitatively seen in Figure (2), where we see that the fields for CT8 and UT8 are closely aligned (Fig. 2), and that they also closely resemble those for CT3.

We attribute the remaining differences between CT8 and UT8 to the fact that the magnetic divergence is removed spectrally in UT8, while it is handled by constrained transport in CT8. Collectively, the high-order constrained and unconstrained transport simulations (CT3, CT8 & UT8) more closely resemble each other than they do the low-order constrained transport simulations (CT1 & CT2). We conclude that CT3 is already a good approximation to the spectral algorithm.

### 3.2. Comparison of the constrained transport and vector potential techniques

We adapted the vector potential code Pencil to run in CT mode and used it to compare the vector potential and CT techniques. We ran an Alfvén wave on a  $32^3$  grid with zero viscosity and resistivity. (Figure 3). After ten crossing times, both the vector potential and CT techniques yield wave profiles that agree with each other to within 1 percent. The shape of the profiles are also well-matched with the initial conditions, with a phase error of 10 percent.

We also used both techniques to run a turbulent dynamo simulation. We started with an initially weak magnetic field in the form of a Beltrami wave and applied helical forcing until it grew to a steady state. The box size is  $(2\pi)^3$ , the density is unity, the forcing power is equal to 0.07, the viscosity is equal to  $5 \cdot 10^{-3}$ , and the resistivity is equal to  $5 \cdot 10^{-3}$ . The RMS magnetic field strength is plotted in Figure (4). After 30 crossing times, the values for  $B_{RMS}$  for the CT and vector potential techniques agree to 1 percent (Figure 4).

### 3.3. Oblique Alfvén wave test

We ran an Alfvén wave test where the propagation axis is oblique to the grid axes, with the initial conditions in Gardiner & Stone (2005):

$$\mathbf{V} = \{0, 0.1 \sin(2\pi x + 4\pi y), 0.1 \cos(2\pi x + 4\pi y)\} \quad (17)$$

$$\mathbf{B} = \{1, 0.1 \sin(2\pi x + 4\pi y), 0.1 \cos(2\pi x + 4\pi y)\} \quad (18)$$

The simulation volume is a unit cube, modeled on a grid of size  $16^3$ . The velocity field is quasi-incompressible, with the divergence removed spectrally every 4 timesteps. The kinetic and magnetic diffusivities are all set to zero for this linear problem. We ran two simulations: one with third-order polynomial finite differences and another with third-order tuned finite differences from Table (2). After the wave has traveled 16 times around the periodic box, the waveform remains almost indistinguishable from the initial conditions, with the tuned finite differences yielding a more precise result than the polynomial finite differences (figure 5).

## 4. Summary

We have developed a new version of the constrained transport algorithm that uses volume-centered fields and hyperresistivity on a high-order finite difference stencil, with

tuned finite difference coefficients to enhance high-wavenumber resolution. High-order interpolation allows implementation of staggered dealiasing. Together, these measures yield a wavenumber resolution that approaches the ideal value achieved by the spectral algorithm.

Volume centered fields are desirable because then  $\mathbf{V}$ ,  $\mathbf{B}$  and  $\mathbf{E}$  all reside at the same grid location, allowing  $\mathbf{E}$  to be constructed directly from the cross product of  $\mathbf{V}$  and  $\mathbf{B}$  without interpolation. For staggered fields,  $\mathbf{V}$  and  $\mathbf{B}$  reside at the zone faces and  $\mathbf{E}$  on the edges, and so constructing  $\mathbf{E}$  involves spatial interpolation, which reduces wavenumber resolution.

High-order stencils and tuned finite difference coefficients both enhance the wavenumber resolution of finite differences. For a radius-three stencil with tuned coefficients, derivatives can be computed to a relative precision of 0.5% up to a Nyquist-scaled wavenumber of  $k = 0.5$ . Without tuning, this would be  $k = 0.34$  for a radius-three stencil. A radius-one stencil derivative such as is used in Zeus (Stone & Norman 1992a) is only accurate up to  $k = 0.12$ . The spectral derivative is precise up to  $k = 1.00$ , although in practice it is limited to  $k = 0.94$  because of aliasing. Aliasing limits a finite-difference code to  $k = 0.66$  unless the finite-difference grid shift aliasing correction is used (§ 2.7).

Hyperresistivity is desirable because it is more effective than Laplacian resistivity in diffusing high-wavenumber modes while at the same time preserving low-wavenumber modes. The fact that hyperresistivity can be written as a curl allows its inclusion into CT. If Laplacian diffusivity were used instead, too much high-wavenumber structure would be diffused for the high-order or tuned derivatives to matter.

The resolution of the algorithm described here approaches that of a spectral code, but because it uses finite differences, it runs faster than a spectral code and isn't restricted to periodic boundary conditions. Also, since the finite differences are local, it is easily scalable to thousands of processors. The spectral algorithm is more difficult to scale to large numbers of processors because it involves all-to-all communications between processors. A finite difference code only passes information between processors whose subgrids are adjacent in physical space. Lastly, because the code works with the magnetic field rather than the vector potential, boundary conditions are often easier to implement.

We received support for this work from NSF Career grant AST99-85392, NSF grants AST03-07854 and AST06-12724, and NASA grant NAG5-10103. We acknowledge stimulating discussions with E. Blackman, A. Brandenburg, B. Chandran, and J. Stone, and we also acknowledge the referee, Wolfgang Dobler, for thorough comments that improved the paper.”

## A. Appendix

Constrained transport expresses the magnetic induction equation as a pure curl plus a divergence diffusivity:  $\partial_t \mathbf{B} = \nabla \times \mathbf{F} + \eta_b \nabla \nabla \cdot \mathbf{B}$ , where  $\mathbf{F}$  is defined in equation (5). The  $\eta_b$  term serves to diffuse away magnetic divergence, and the finite differences are arranged so that  $\nabla \cdot \nabla \times \mathbf{F} = 0$ . Thus, the curl term does not contribute to the evolution of the magnetic divergence, and if the initial conditions are divergence-free, the magnetic divergence remains zero throughout the evolution.

To see how constrained transport works, denote the vector field by  $\mathbf{F}_{i,j,k} = \{F_{i,j,k}^x, F_{i,j,k}^y, F_{i,j,k}^z\}$ , where  $\{i, j, k\}$  are integers specifying the locations of grid cell centers. There are two basic grid types: “centered” and “staggered” (figure 6). For a centered grid, scalar and vector quantities are located at cell centers. For a staggered grid, scalar quantities are located at cell centers and vector quantities at cell faces. For instance, we would index the components of  $\mathbf{F}$  as  $\{F_{i+1/2,j,k}^x, F_{i,j+1/2,k}^y, F_{i,j,k+1/2}^z\}$ .

The finite divergence divergence of the curl of  $\mathbf{F}$  is  $\nabla \cdot \nabla \times \mathbf{F} = \epsilon_{ijk} \partial_i \partial_j \mathbf{F}_k$ , which consists of terms such as  $(\partial_1 \partial_2 - \partial_2 \partial_1) \mathbf{F}_k$ . One can straightforwardly see that this is zero for finite differences of the form eq. 11 for both centered and staggered grids. Thus, constrained transport can be coordinated with high-order and tuned finite differences, as well as with hyperresistivity.

For a staggered grid, the “ $\mathbf{F}$ ” vectors are located at cell edges, whereas the  $\mathbf{V}$  and  $\mathbf{B}$  vectors from which they are constructed are found at cell faces. A staggered grid CT scheme therefore involves spatial interpolation, one example being the method of characteristics scheme for time-interpolating Alfvén waves. We use volume-centered fields and Runge-Kutta timestepping because, among other reasons, no interpolation is required.

## REFERENCES

- Batchelor, G. K. 1950, Proc. Roy. Soc. Lon., A201, 405
- Brackbill, J. & Barnes, D. 1980, J. Comput. Phys., 35, 426
- Brandenburg, A., & Dobler, W. 2002, Comput. Phys. Comm., 147, 471
- Canuto, C., Hussaini, M. Y., Quarteroni, A., Zang, T. A., Jr. 1987, Spectral Methods in Fluid Dynamics (Heidelberg: Springer-Verlag)
- Colella, P., & Woodward, P. R. 1984, J. Comput. Phys., 54, 174

- Dedner, A., Kemm, F., Kroner, D., Munz, C.-D., Schnitzer, T. & Wesenberg, M. 2002, *J. Comput. Phys.*, 175, 645
- Dobler, W., Stix, M. & Brandenburg, A. 2006, *ApJ* 638, 336
- Evans, C.R. & Hawley, J.F. 1988, *ApJ*, 332, 659
- Gardiner, T. & Stone, J., *J. Comput. Phys.*, 205, 509
- Harten, A. 1983, *J. Comput. Phys.*, 49, 357
- Haugen N. E. L., & Brandenburg, A. 2004, *PRE* 70, 026405
- Hawley, J. F., & Stone, J. M. 1995, *Comp. Phys. Comm.*, 89, 127
- Londrillo, P. & Del Zanna, L. 2000, *ApJ*, 530, 508
- Maron, J., 2004, *J. Comput. Phys.*, submitted (astro-ph/0402606)
- Maron, J., Chandran, B. & Blackman, E. 2004, *PRL*, 92, 045001
- Maron, J. & Goldreich, P. 2001, *ApJ*, 554, 1175
- Powell, K.G., Roe, P.L., Linde, T.J., Gombosi, T.J. & De Zeeuw, D.L. 1999, *J. Comput. Phys.*, 154, 284
- Stone, J. M. & Norman, M. L. 1992a, *ApJS*, 80, 791
- Stone, J., Hawley, J.F., Evans, C.R. & Norman, M. 1992b, *ApJ*, 388, 415
- Tóth, G. 2000, *J. Comput. Phys.*, 161, 605
- Toro, E. F. 1999, *Riemann Solvers and Numerical Methods for Fluid Dynamics* (Heidelberg: Springer-Verlag)
- Zhong, X. 1998, *J. Comp. Phys.* 144, 662

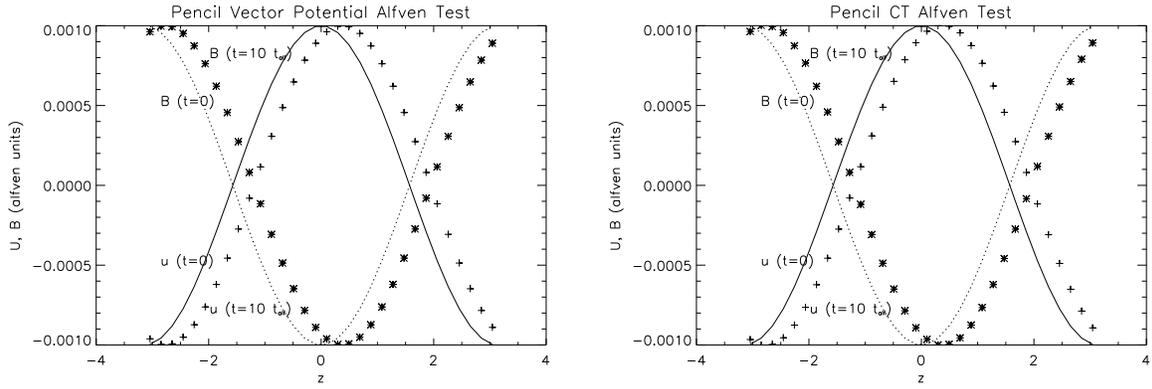


Fig. 3.— The Alfvén wave profile after ten crossing times, with (a) the Pencil code using the vector potential form of the induction equation and (b) using CT (§ 3.2). Alfvén units denote the fluctuating field strengths compared to the uniform component of the magnetic field.

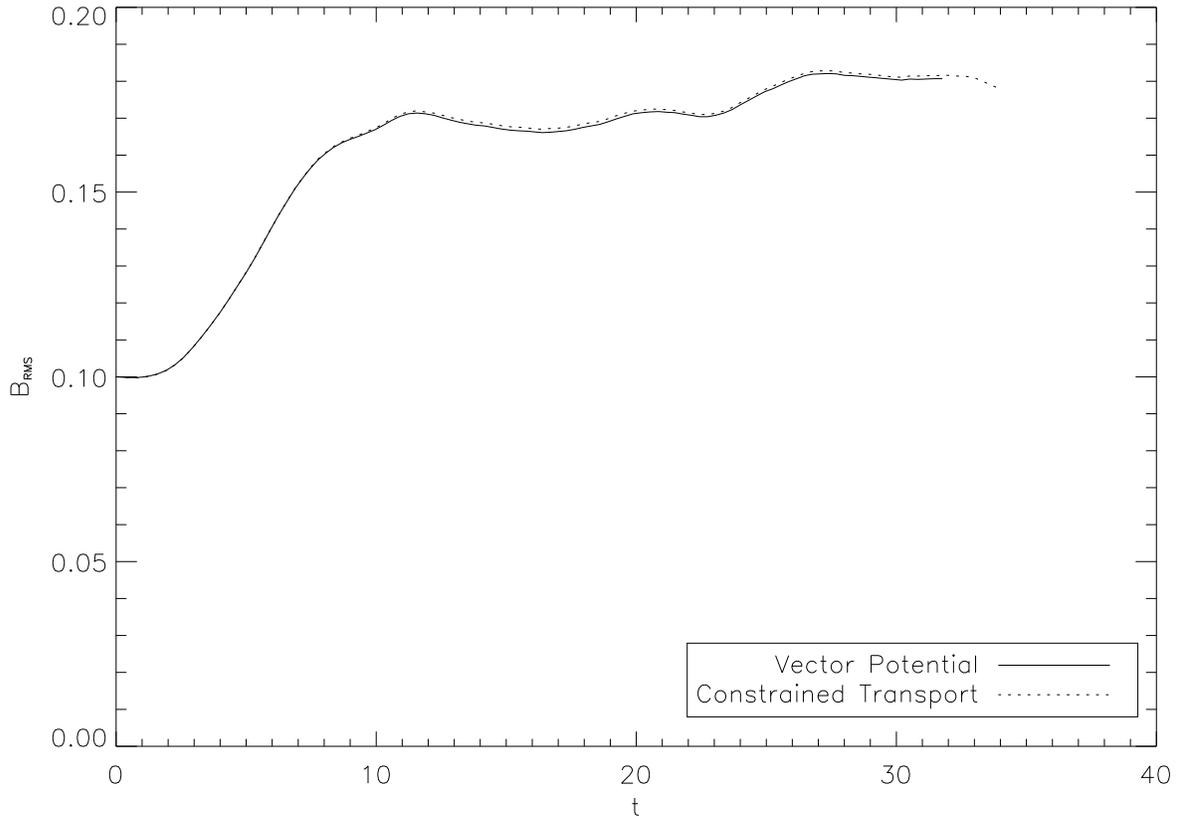


Fig. 4.— We plot the RMS magnetic field strength for a turbulent dynamo simulation using the Pencil code in vector potential and CT mode (§ 3.2). Both techniques agree to 1 percent after 30 crossing times.

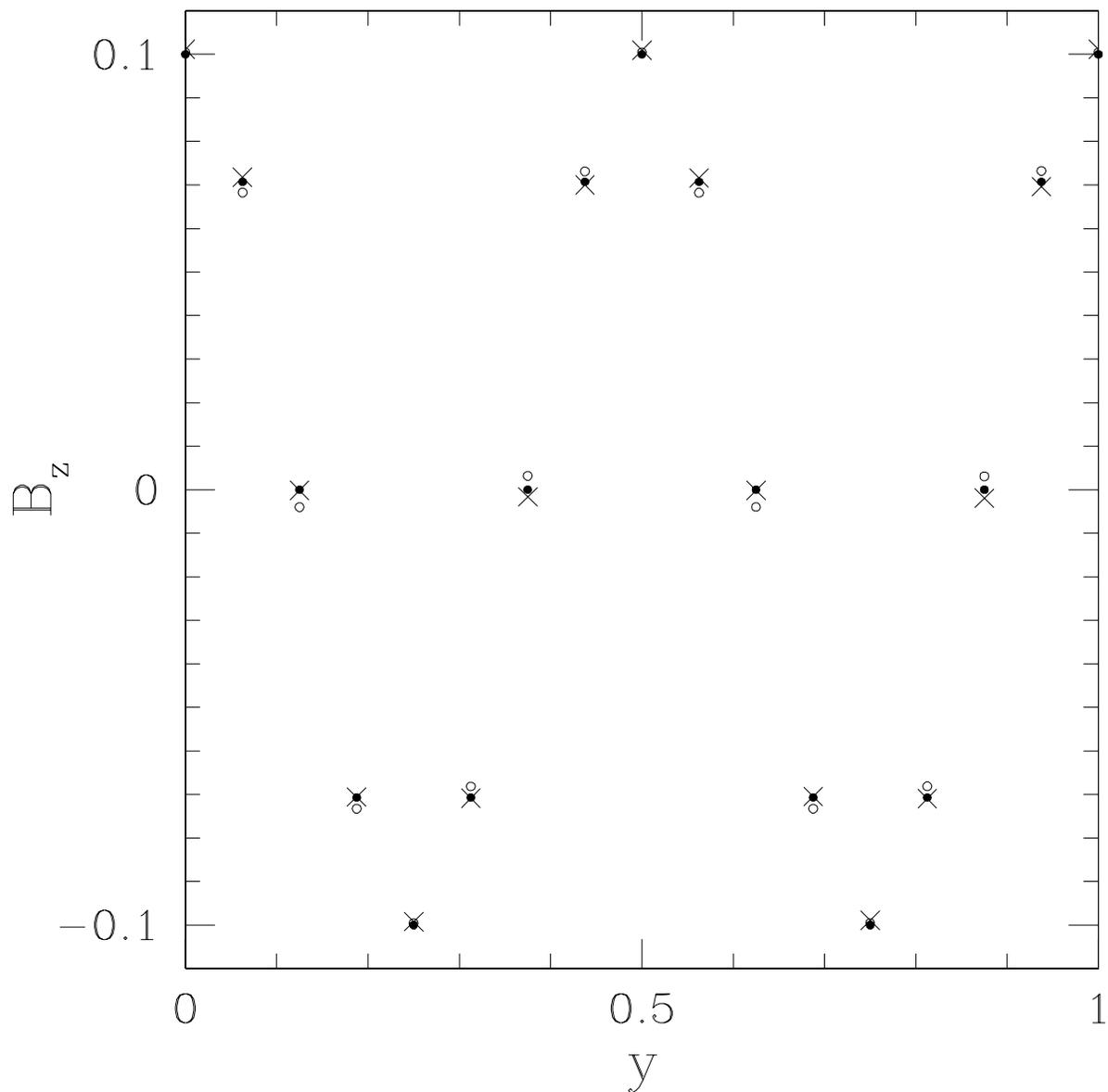


Fig. 5.— The oblique Alfvén wave test from § 3.3. We plot the  $B_z$  field as a function of  $y$ , with  $x$  and  $z$  equal to zero. We show initial conditions (*solid points*), and results from simulations with polynomial finite differences (*open circles*) and tuned finite differences (*crosses*). In each simulation, the grid size is  $16^3$  and the wave has traveled 16 times around the periodic box. The tuned finite differences yield a more precise result.

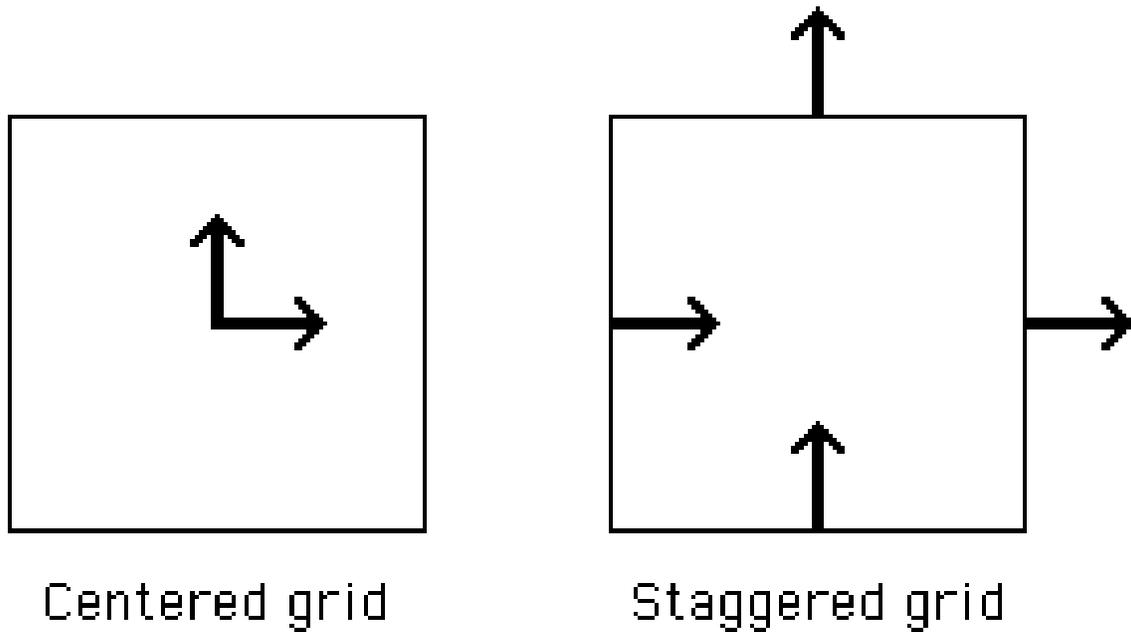


Fig. 6.— We show the configuration for zone-centered and zone-staggered fields in two dimensions, where the generalization to three dimensions is straightforward. (*Left*) A zone-centered vector field, where both the  $\hat{X}$  and  $\hat{Y}$  vectors reside at zone centers. (*Right*) A zone-staggered vector field, where  $\hat{X}$  vectors reside on  $\hat{X}$  faces and  $\hat{Y}$  vectors reside on  $\hat{Y}$  faces.