



Orbital Advection with Magnetohydrodynamics and Vector Potential

Wladimir Lyra^{1,2,3,7} , Colin P. McNally^{4,3,7} , Tobias Heinemann⁵ , and Frédéric Masset⁶

¹ Department of Physics and Astronomy, California State University Northridge, 18111 Nordhoff Street, Northridge CA 91130, USA; wlyra@csun.edu

² Jet Propulsion Laboratory, California Institute of Technology, 4800 Oak Grove Drive, Pasadena, CA, 91109, USA

³ Kavli Institute for Theoretical Physics, Kohn Hall, University of California, Santa Barbara, CA 93106-4030, USA

⁴ Astronomy Unit, School of Physics and Astronomy, Queen Mary University of London, Mile End Road, London E1 4NS, UK

⁵ Niels Bohr International Academy, The Niels Bohr Institute, Blegdamsvej 17, DK-2100, Copenhagen Ø, Denmark

⁶ Instituto de Ciencias Físicas, Universidad Nacional Autónoma de México, Av. Universidad s/n, 62210 Cuernavaca, Mor., Mexico

Received 2017 July 10; revised 2017 August 16; accepted 2017 August 21; published 2017 September 15

Abstract

Orbital advection is a significant bottleneck in disk simulations, and a particularly tricky one when used in connection with magnetohydrodynamics. We have developed an orbital advection algorithm suitable for the induction equation with magnetic potential. The electromotive force is split into advection and shear terms, and we find that we do not need an advective gauge since solving the orbital advection implicitly precludes the shear term from canceling the advection term. We prove and demonstrate the third order in time accuracy of the scheme. The algorithm is also suited to non-magnetic problems. Benchmarked results of (hydrodynamical) planet–disk interaction and of the magnetorotational instability are reproduced. We include detailed descriptions of the construction and selection of stabilizing dissipations (or high-frequency filters) needed to generate practical results. The scheme is self-consistent, accurate, and elegant in its simplicity, making it particularly efficient for straightforward finite-difference methods. As a result of the work, the algorithm is incorporated in the public version of the PENCIL CODE, where it can be used by the community.

Key words: hydrodynamics – magnetic fields – methods: analytical – methods: numerical – plasmas – protoplanetary disks

1. Introduction

In numerical models of gaseous disks in astrophysical systems, a common challenge is mitigating the difficulties caused by the superposition of the dynamics of interest on a fast azimuthal flow. In particular, this applies in accretion disks such as protoplanetary disks, cataclysmic variables, and X-ray binaries. In disks in hydrostatic equilibrium, the Mach number of the azimuthal flow with respect to the sound speed is on the order of h^{-1} , where $h \equiv H/R$ is the disk aspect ratio, with H the scale height and R the distance to the central object. As these disks are thin, the Mach number is usually substantial.

Additionally, if the disk is magnetized but dominated by thermal pressure, then other relevant speeds of information propagation such as the Alfvén and fast magnetosonic speeds will be on the order of the sound speed or smaller. Thus, the orbital advection will not just dominate the advection, but also dominate over the propagation of signals in the flow. As a result, special techniques to deal with the orbital advection in such disks have proved very useful for their simulation, both for ameliorating the negative effects of numerical diffusion during the fast azimuthal advection, and as a way of removing the time step constraints imposed on time-explicit integration by the Courant–Fredrichs–Lewy (CFL) condition.

The FARGO code (Fast Advection in Rotating Gaseous Objects; Masset 2000) introduced a split azimuthal advection, giving rise to the popular use of the term “fargo” to denote this feature in other schemes. However, split azimuthal advection has been implemented in a number of different ways, which can nonetheless be broadly summed up in three categories. First, there are schemes which directly solve the full equations

in terms of the total velocity, including the azimuthal advection velocity of the gas on a fixed mesh. FARGO (Masset 2000) is one of these schemes. LA-COMPASS uses a dimensionally split Lagrangian remap piecewise parabolic method, with the advection step arranged as a separate shift by a integer number of cells in the azimuthal sweep (Li & Li 2012) and allows refined meshes to move across coarse parent meshes.

Second, there are schemes which separate the velocity field into two parts, and dynamically solve only for perturbations to a background flow. This is most common in local shearing-box models as opposed to full cylindrical disks, including the implementation where the term orbital advection appears to first originate (Johnson et al. 2008). The method implemented for shearing box simulations in the PENCIL CODE (Brandenburg & Dobler 2002, 2010) is the shearing box orbital advection scheme Shear Advection by Fourier Interpolation (SAFI), introduced by Johansen et al. (2009). The method of Li et al. (2001) uses a hybrid approach, enabled by virtue of the directional splitting of the equations. In the angular sweep the equation solved is for a fluctuation from the background azimuthal flow, and this sweep is sub-cycled in time so that the global time step is only determined by the radial CFL condition. In ATHENA (Stone & Gardiner 2010) the orbital advection implemented is of the type which solves for fluctuation velocities and is the first introduction of the use of a constrained transport algorithm to solve for the magnetic field in the azimuthal interpolation step. For PLUTO (Mignone et al. 2010) the formulation introduced in Mignone et al. (2012) gives a total angular momentum-conserving formulation of the procedure, solving for velocity fluctuations on a cylindrical grid in a Godunov-type method.

In the third category we place methods which use a moving or variable mesh, or no mesh at all, to achieve Lagrangian or partially Lagrangian properties of the method. DISCO (Duffell 2016a, 2016b) uses a Godunov method implemented on a

⁷ Both authors contributed equally to this work.

moving mesh, arranged as a series of concentric sliding rings which follow the orbital motion of the disk. A novel constrained transport formulation adapted to this moving mesh is used for magnetohydrodynamics (MHD). Alternatively, more general approaches can be used, either with moving Voronoi meshes (Springel 2010; Muñoz et al. 2014) or meshless methods (e.g., Maron et al. 2012; McNally et al. 2012) which, by allowing the discretization elements to follow the flow in general, also allows them to follow the orbital motion. Additionally, one might achieve a partly Lagrangian method by allowing the mesh to shear with the background flow, and remap to the original coordinates not within each time step, but after a significant number and before the mesh becomes too distorted. This approach is used in some pseudospectral codes (Umurhan & Regev 2004; Lesur & Longaretti 2005).

In all methods, the treatment of the centrifugal, Coriolis, and gravitational source terms in the radial direction can be critical. In a flow in steady-state, the orbital motion of the disk results in the partial cancellation of the radial component of the gravitational force and (depending on the frame and the full or fluctuation velocity) the centrifugal and Coriolis force terms; the remainder is balanced against the radial pressure gradient. In many studies the ability to hold the physical equilibrium state numerically steady for many dynamical times is important. Thus many methods with lower spatial accuracy need to either analytically cancel radial force terms, or at least ensure the mentioned forces are treated in a similar enough way (e.g., on the same substep of an operator split method) that they cancel with sufficient accuracy.

It is worth emphasizing that, across the range of methods mentioned above, the problem of advancing the magnetic field with the disk motion has taken three distinct approaches. Solving directly for the magnetic field in a ZEUS-family operator-split code, Johnson et al. (2008) constructed a divergence-free slope-limited remap procedure for the azimuthal transport step. This was greatly simplified in the context of a Godunov method in Stone & Gardiner (2010) by applying a constrained transport method to the azimuthal transport of the magnetic field. DISCO (Duffell 2016a) uses a novel constrained transport formulation adapted to its mesh of sliding rings. Using the magnetic vector potential in the PENCIL CODE, Johansen et al. (2009) took advantage of the Keplerian gauge mandatory for shearing boxes so that the vector potential field could be simply shifted and interpolated along the grid.

In this paper, we introduce a novel method that we implemented for the purpose of this work in the PENCIL CODE. PENCIL is a high-order finite-difference method code using point collocated values, here on a cylindrical mesh, with a method of lines approach using sixth-order finite-difference discretization in space, to which a third-order time integration scheme is then applied. Our method is classified in the first category above, solving for the entire flow, but splitting the orbital advection operator, discretizing it in space with a spectral method, and integrating it in time along with the rest of the operators in the original Runge–Kutta time integration. This smoothly inherits the accuracy properties of the base scheme. We introduce an appropriate splitting of the vector potential evolution equation so that this field needs only to be shifted and interpolated across the grid to enable MHD. We also extensively discuss some choices for the stabilizing dissipation operators which must be included in such a scheme for

stability, as the use of a cylindrical mesh means significant attention must be paid to this matter. Our method is closely related to that of Johansen et al. (2009) (SAFI) and, importantly, our proof of third-order accuracy in time applies to it too. However, as our method is developed for cylindrical-global models, the model equations solved are different, using the entire velocity, not simply fluctuations in a shearing box, and the choice of stabilizing dissipation must be different due to the cylindrical-polar grid.

This paper is organized as follows. In Section 2 we describe the model equations. In Section 3 we analytically prove the scheme’s third-order accuracy in time. Tests are given in Section 4 where we also describe and test in detail the dissipation operators needed to stabilize the scheme for these practical problems, finally leading to our summary in Section 5.

2. Model Equations

We first consider the MHD inviscid equations

$$\frac{\partial \rho}{\partial t} = -(\mathbf{u} \cdot \nabla) \rho - \rho \nabla \cdot \mathbf{u}, \quad (1)$$

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \cdot \nabla) \mathbf{u} - \frac{1}{\rho} \nabla p - \nabla \Phi + \frac{\mathbf{J} \times \mathbf{B}}{\rho}, \quad (2)$$

$$\frac{\partial s}{\partial t} = -(\mathbf{u} \cdot \nabla) s + \frac{1}{\rho T} (\mathcal{H} - \mathcal{Q}), \quad (3)$$

$$\frac{\partial \mathbf{B}}{\partial t} = \nabla \times (\mathbf{u} \times \mathbf{B} - \eta \mu_0 \mathbf{J}) \quad (4)$$

$$p = \rho c_s^2 / \gamma. \quad (5)$$

where \mathbf{u} is the velocity, \mathbf{B} is the magnetic field, $\mathbf{J} = \mu_0^{-1} \nabla \times \mathbf{B}$ is the current density, and μ_0 is the magnetic constant; p is the pressure, c_s is the sound speed, and γ is the adiabatic index. The gravitational potential $\Phi = -GM_*/r$ where G is the gravitational constant, and M_* is the stellar mass. We restrict ourselves to the cylindrical approximation where the vertical stratification is ignored, so r is the cylindrical radius. Equation (3) is the energy equation, where s stands for entropy, T is the temperature, \mathcal{H} is a heating term, and \mathcal{Q} is a cooling term.

Instead of solving for the magnetic field, in the PENCIL CODE we write the induction equation in terms of the magnetic potential A , where $\mathbf{B} = \nabla \times \mathbf{A}$. Removing the curl on both sides, the induction equation for A is

$$\frac{\partial A}{\partial t} = \mathbf{u} \times \mathbf{B} - \eta \mu_0 \mathbf{J} \quad (6)$$

This formulation is not adequate for use with an orbital advection algorithm since it lacks an explicit advection term. Instead, we expand the electromotive force

$$\begin{aligned} \mathbf{u} \times \mathbf{B} &= \xi_{ijk} u_j B_k \\ &= \xi_{ijk} \xi_{kpm} u_j \partial_p A_m \\ &= (\delta_{ip} \delta_{jm} - \delta_{im} \delta_{jp}) u_j \partial_p A_m \\ &= -u_j (\partial_j A_i - \partial_i A_j) \\ &= -(\mathbf{u} \cdot \nabla) \mathbf{A} + \mathbf{u} \cdot (\nabla \mathbf{A})^T \end{aligned} \quad (7)$$

to find the advective term $u_j \partial_j A_i$. We use this formulation to apply the orbital advection algorithm to the magnetic potential.

The remainder of the algorithm is as the original FARGO, with the care of performing it as part of a third-order Runge–Kutta timestepping, that has three stages.

At the beginning of the first stage the average azimuthal velocity is defined as a 2D array (in radial and vertical coordinates), $\bar{u}_\phi(r, z)$.

At every stage the residual velocity is computed

$$\mathbf{u}' = \mathbf{u} - \bar{u}_\phi(r, z) \hat{\phi} \quad (8)$$

and that becomes the advection azimuthal velocity for the Courant condition. The modified equations of motion are

$$\left(\frac{\partial}{\partial t} + \mathcal{L}\right)\rho = -(\mathbf{u}' \cdot \nabla)\rho - \rho \nabla \cdot \mathbf{u}, \quad (9)$$

$$\left(\frac{\partial}{\partial t} + \mathcal{L}\right)\mathbf{u} = -(\mathbf{u}' \cdot \nabla)\mathbf{u} - \frac{1}{\rho} \nabla p - \nabla \Phi + \frac{\mathbf{J} \times \mathbf{B}}{\rho}, \quad (10)$$

$$\left(\frac{\partial}{\partial t} + \mathcal{L}\right)s = -(\mathbf{u}' \cdot \nabla)s + \frac{1}{\rho T}(\mathcal{H} - \mathcal{Q}), \quad (11)$$

$$\left(\frac{\partial}{\partial t} + \mathcal{L}\right)\mathbf{A} = -(\mathbf{u}' \cdot \nabla)\mathbf{A} + \mathbf{u} \cdot (\nabla \mathbf{A})^T - \eta \mu_0 \mathbf{J} \quad (12)$$

with

$$\mathcal{L} = \frac{\bar{u}_\phi(r, z)}{r} \frac{\partial}{\partial \phi} \quad (13)$$

This \mathcal{L} term is interpolated in Fourier space as the SAFI algorithm does with the Keplerian advection in the shearing box (Johansen et al. 2009).

2.1. Inertial Terms

The \mathcal{L} operator acting on a vector ψ produces inertia terms of the form $r^{-1} \bar{u}_\phi \psi_i \hat{e}_j$. These combine with similar inertial terms $r^{-1} u'_\phi \psi_i \hat{e}_j$ produced by $(\mathbf{u}' \cdot \nabla)\psi$ so that the inertia terms on \mathbf{u} and \mathbf{A} are unaltered with respect to the implementation without orbital advection, i.e., they carry the full velocity.

The term $\mathbf{u} \cdot (\nabla \mathbf{A})^T$ produces the inertia terms $r^{-1}(A_r u_\phi - A_\phi u_r) \hat{\phi}$.

2.2. The Pseudo-advective Gauge

In Equation (7), the term $u_j \partial_i A_j$, if implemented in connection with the full advection term, would contribute terms that simply cancel the advection term numerically. The reason why Equation (7) works with orbital advection is that we split the velocity in the advection term

$$\partial_i A_i = -\bar{u}_j \partial_j A_i - u'_j \partial_j A_i + u_j \partial_i A_j \quad (14)$$

and implement the first term in the right-hand side implicitly. Other formulations (Brandenburg et al. 1995; Brandenburg 2010; Candelaresi et al. 2011), without orbital advection, explicitly summon an advective gauge, transforming the $u_j \partial_i A_j$ term according to

$$u_j \partial_i A_j = \nabla(\mathbf{u} \cdot \mathbf{A}) - A_j \partial_i u_j \quad (15)$$

and because the curl of a gradient is zero, the gradient term has no bearing in the magnetic field. It can be removed by a gauge transformation, allowing one to write

$$\partial_i A_i = -u_j \partial_j A_i - A_j \partial_i u_j \quad (16)$$

which is the advective gauge. Notice that, essentially, the advective gauge exchanges gradients of \mathbf{A} for gradients of \mathbf{u} . Unfortunately, although analytically elegant, this formulation is not numerically stable. This is because, although the gradient term is analytically irrotational, in practice the numerical cancelation is not perfect. The unphysical residual of the gradient term, although negligibly small at first, grows unboundedly in time, to the point that it eventually dominates over the (physical) solenoidal component, leading to a spurious accumulation of power in small scales of the magnetic potential (Candelaresi et al. 2011). These in turn lead to inaccurate magnetic fields and numerical instabilities at the grid scale. For this reason we work with Equation (14), instead of the advective gauge. This choice does not have the properties of an advective gauge, but it has proved well suited to the orbital advection algorithm. For this reason we call the evolution equation of the vector potential in Equations (12) and (14) a pseudo-advective gauge.

3. Third-order Accuracy in Time

In this section we prove analytically that the scheme of Equations (9)–(12) retains the third-order accuracy of the original formulation without orbital advection. We consider a partial differential equation of the form

$$\left(\frac{\partial}{\partial t} + \mathcal{L}\right)u = f \quad (17)$$

where $u(x, y, t)$ and $f(x, y, t)$ are functions of space and time and \mathcal{L} is a linear differential operator that we assume not to depend explicitly on time.

The solution to the homogeneous part of Equation (17) may be written as

$$u(x, y, t) = \mathcal{Q}(t)u(x, y, 0) \quad (18)$$

where

$$\mathcal{Q}(t) = \exp(-t\mathcal{L}) \quad (19)$$

is a linear operator. We now introduce new dependent variables \tilde{u} and \tilde{f} through

$$\begin{aligned} u(x, y, t) &= \mathcal{Q}(t)\tilde{u}(x, y, t) \\ f(x, y, t) &= \mathcal{Q}(t)\tilde{f}(x, y, t) \end{aligned} \quad (20)$$

Plugging Equation (20) in Equation (17) we have

$$\frac{\partial}{\partial t}[\mathcal{Q}(t)\tilde{u}] + \mathcal{L}\mathcal{Q}(t)\tilde{u} = \mathcal{Q}(t)\tilde{f} \quad (21)$$

and given Equation (19) we have

$$\left[\mathcal{Q}(t) \frac{\partial \tilde{u}}{\partial t} - \mathcal{L}\mathcal{Q}(t)\tilde{u}\right] + \mathcal{L}\mathcal{Q}(t)\tilde{u} = \mathcal{Q}(t)\tilde{f} \quad (22)$$

which eliminates \mathcal{L} , leaving only

$$\frac{\partial \tilde{u}}{\partial t} = \tilde{f} \quad (23)$$

The effect of the transformation Equation (20) is to formally eliminate the linear operator term in Equation (17). With $\mathcal{Q}(t)$ being the advection by $\bar{u}_\phi(r, z)$, this translation may conveniently be carried out in Fourier space as in the SAFI algorithm.

3.1. The RK3-2N Time Integration Scheme

The PENCIL CODE uses a time-stepping scheme that, when applied to Equation (23), reads

$$\tilde{\omega}_i = \alpha_i \tilde{\omega}_{i-1} + \tilde{f}_i \quad (24)$$

$$\tilde{u}_{i+1} = \tilde{u}_i + \beta_i \tilde{\omega}_i \delta t \quad (25)$$

where i runs from 0 to 2 and $u_3 = u(t + \delta t)$. The parameters α_i and β_i are constant coefficients that depend on the integration scheme. We use the values of Williamson (1980), $\alpha = [0, -5/9, -153/128]$ and $\beta = [1/3, 15/16, 8/15]$.

Given Equation (20) and also applying the operator $\mathcal{Q}(t)$ to ω

$$\omega(x, y, t) = \mathcal{Q}(t) \tilde{\omega}(x, y, t) \quad (26)$$

we have

$$\mathcal{Q}(-t_i) \omega_i = \alpha_i \mathcal{Q}(-t_{i-1}) \omega_{i-1} + \mathcal{Q}(-t_i) f_i \quad (27)$$

$$\mathcal{Q}(-t_{i+1}) u_{i+1} = \mathcal{Q}(-t_i) u_i + \beta_i \mathcal{Q}(-t_i) \omega_i \delta t \quad (28)$$

Since β and δt are not affected by \mathcal{Q} , we can group the last equation as

$$\mathcal{Q}(-t_{i+1}) u_{i+1} = \mathcal{Q}(-t_i) [u_i + \beta_i \omega_i \delta t] \quad (29)$$

Applying $\mathcal{Q}(t_i)$ to the former and $\mathcal{Q}(t_{i+1})$ to the latter equation, we have

$$\omega_i = \alpha_i \mathcal{Q}(\delta t_i) \omega_{i-1} + f_i \quad (30)$$

$$u_{i+1} = \mathcal{Q}(\delta t_{i+1}) [u_i + \beta_i \omega_i \delta t] \quad (31)$$

where $\delta t_i = t_i - t_{i-1}$.

Equation (31) states that in order to evolve the system in time, one can use the RK3-2N scheme unmodified for everything but the shear advection, and then shift both u and ω (the variable and derivative arrays) in the ϕ direction by an amount $\Delta\phi = -\tilde{u}_\phi \delta t_i / r$ at the end of the i th stage. The procedure to translate and interpolate the variables and derivative arrays for each stage to and from Fourier space is a costly one, but this cost is compensated for by the increase in timestep. The increase in time step in general, for disk advection problems, is about a factor of 10. The FFT operations lead to a factor of 3 overhead. The overall gain is thus about a factor of 3 in performance.

The same operations are performed in the shearing sheet in the SAFI algorithm. Notice that this proves that the SAFI algorithm is third-order accurate, something that was not clear from Johansen et al. (2009). Therefore, our algorithm can be seen as the polar coordinate generalization of SAFI for global disks.

3.2. Demonstration of Third-order Accuracy in Time

To demonstrate the third-order accuracy of our modified time integration scheme as proved in Section 3 we present a simple test. A one-dimensional problem in ϕ is solved with a homogeneous fluid accelerated in ϕ by a force $F_\phi(t) = \rho t^4$ with time t . The exact solution for the fluid velocity is then $u_\phi(t) = t^5/5$. Running this test to $t = 1$ where $u_\phi(t) = 1/5$, we plot the relative error in the solution for the fluid velocity and find third-order convergence, as shown in Figure 1. This shows that our algorithm for splitting the ϕ advection does indeed preserve the third-order accuracy of the PENCIL CODE time integration scheme. We remind the reader that this test isolates the time discretization, and that the spatial discretization in the PENCIL CODE is sixth-order accurate in space in most modules.

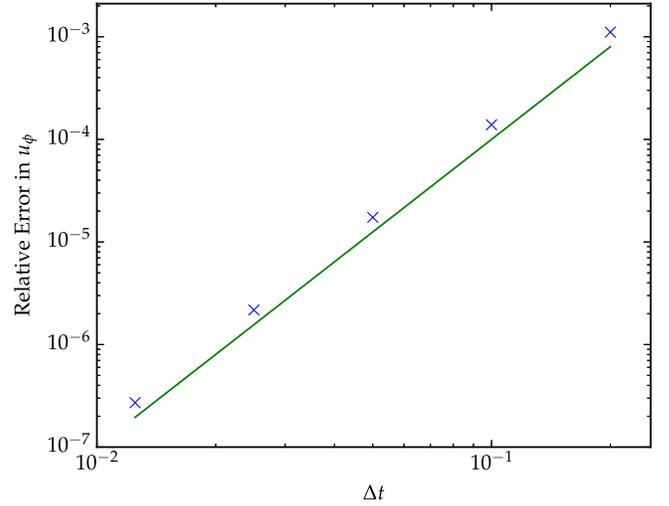


Figure 1. Demonstration of the third-order accuracy of the time integration algorithm. Relative errors of the solution for each time step size are drawn as blue crosses, and a Δt^3 reference slope is shown demonstrating that the numerical solution converges at third order with respect to the time discretization.

4. Tests

We present in this section a series of tests aimed at describing the practical application of the scheme. By design, the PENCIL CODE relies on some high-frequency filter, in the form of a dissipation or diffusion operator, to stabilize the scheme by eliminating structures close to the grid scale. In our first test, a hydrodynamic planet–disk interaction problem, significant attention will be paid to this selection, which differs from that needed for the same problem on a Cartesian grid, and may vary from that needed on a cylindrical grid without the use of or orbital advection scheme. We then proceed to show a classic magnetic field advection test, the field loop advection problem (modified for cylindrical coordinates), and demonstrate the use of orbital advection in a global disk simulation of magnetorotational instability (MRI) driven turbulence.

4.1. Planet–Disk Problem

We now address the planet–disk interaction benchmark problem described in de Val-Borro et al. (2006). In that paper Pencil presented only the viscous runs, for Neptune and Jupiter mass (mass ratios $q = 10^{-4}$ and $q = 10^{-3}$, respectively). Also, the runs were performed in Cartesian coordinates; the cylindrical version of Pencil was coded at a later date (Lyra et al. 2009)

In a single processor at resolution $N_r \times N_\phi = 128 \times 384$, we perform one orbit in 90 s, or 2.5×10^{-2} hr. The 100 orbits of the benchmark thus take 2.5 hr. Even considering the factor ≈ 2 in resolution (320^2 Cartesian versus 128×384 in cylindrical), this is a significant speed-up compared to the original 36 hr that Pencil took in the original code comparison paper. The test was done with a 1.6 GHz Intel Core i5 processor.

Figure 2 shows a comparison between two simulations differing only in the use of the orbital advection algorithm. The upper plots show the state of the flow at 10 orbits, whereas the lower plots show the state of the flow at 100 orbits. The resolution was 256×768 in cylindrical coordinates. The simulation was done in a frame centered at the star and

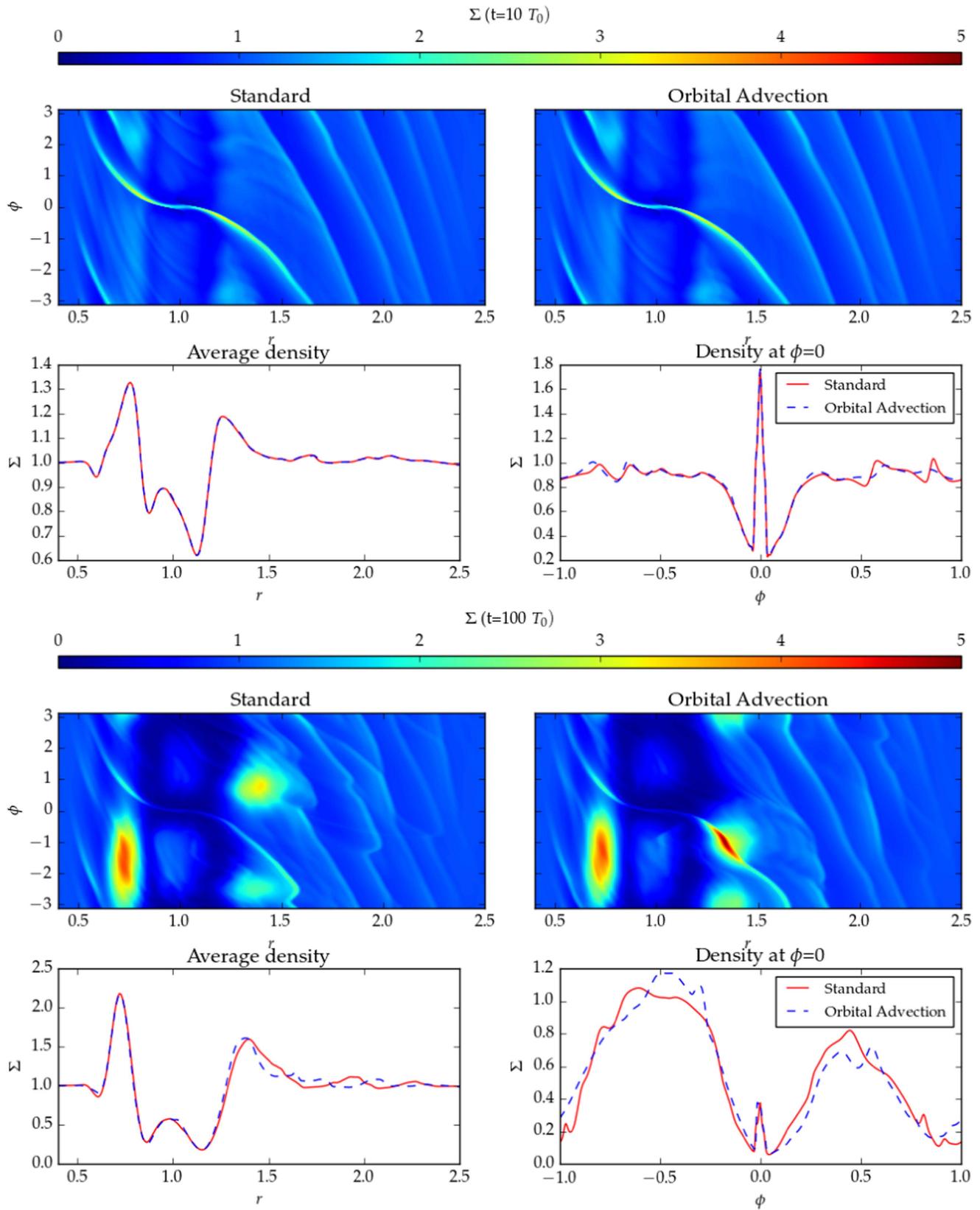


Figure 2. Comparison between runs with and without the orbital advection algorithm. Clockwise from the upper left: the run without the algorithm, the run with the algorithm, an azimuthal slice at the radial position of the planet, and the azimuthally averaged density as a function of radius. The upper plots show the state of the flow at 10 orbits, the lower ones at 100 orbits. The flow at 10 orbits is very similar in both cases. At 100 orbits the higher numerical dissipation of the scheme without orbital advection has made the excitation of the Rossby wave instability at the outer edge slightly different, obviated by the different azimuthal position of the vortices, which hint at a different time of excitation. The flow is otherwise similar in both cases.

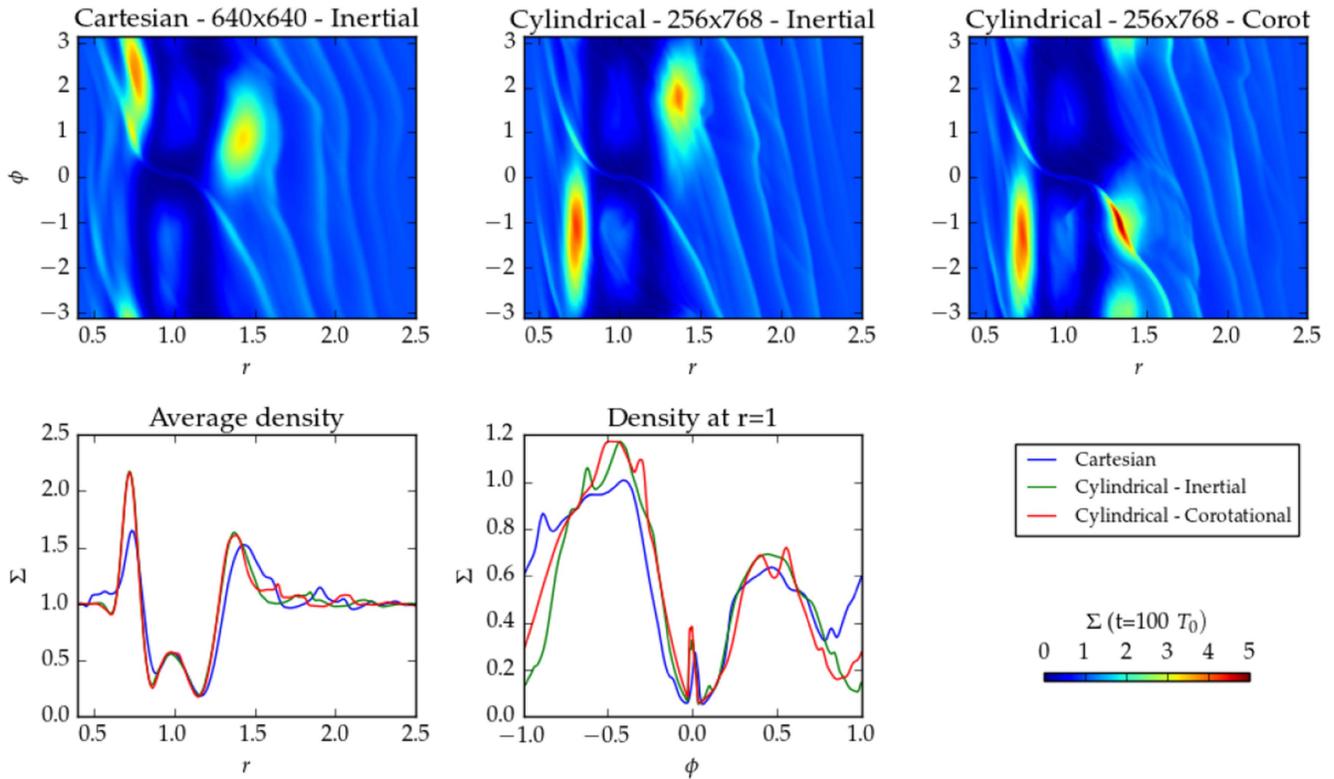


Figure 3. State of the flow at 100 orbits of a Jupiter-mass planet. The left panels show a Cartesian calculation at resolution 640×640 . The cylindrical calculations are done in the inertial and corotational frame, which should be identical except for the timestep. Both use the orbital advection algorithm. The resolution in the Cartesian run is similar to the cylindrical ones. The evolution of outer vortices is different in the three realizations, which hints at differences in the excitation of the RWI due to different numerical dissipation in the different methods.

corotating with the planet. Coriolis and centrifugal forces, as well as the indirect term, were added to the simulation accordingly (e.g., Lyra et al. 2016). The velocity that enters in the Coriolis force is the full velocity.

It is seen that at 10 orbits the flows with and without the orbital advection algorithm are very similar. Differences are expected as the flow with orbital advection executes fewer timesteps and therefore has less numerical diffusion. Indeed, at 100 orbits the flows already show slight differences in the excitation of the Rossby vortices at the outer gap edge. These vortices result from Rossby wave instability (RWI, Lovelace et al. 1999) and are an expected result of disk–planet interaction (Koller et al. 2003).

Although the overall shape of the gap is very similar in both runs, numerical dissipation is indeed an issue that affects the excitation of the RWI, as also shown in Figure 3. In this figure, we test the different methods we use to model planet–disk interaction: a Cartesian grid (inherently without the orbital advection algorithm), a cylindrical calculation in the inertial frame, where the bodies are evolved with a built-in N -body code, and the cylindrical corotational frame as used in Figure 2. The resolution element in the Cartesian run at 640×640 resolution is approximately the same as in the cylindrical run at 256×768 . We see that numerical dissipation led to markedly different evolutions of the vortices, the shape of the gap, and the Lagrangian clouds as well.

If the numerical dissipation we do not control modifies the results of the simulation, then the dissipation we do control should also have an effect. Prompted by this thought, we investigate the results of planet–disk interaction we get under different explicit dissipation. Explicit dissipation terms appear

in the code as hyperviscosity and hyperdiffusion, functioning as high-frequency filters. We also employ a shock-capturing dissipation in the form of a bulk (artificial) viscosity term.

4.1.1. High-frequency Filter

In the PENCIL CODE we use three different varieties of high-frequency filters, that we call *strict*, since it strictly obeys a conservation law, *polar* hyperdiffusion and *mesh* hyperdiffusion. Polar and mesh hyperdiffusions are not conservative, and differ in the way they scale with the grid element, mesh hyperviscosity being independent of resolution. The Appendix details their implementation.

We check the different hyperviscosity formulations in the panels of Figure 4. The chosen problem is the benchmarked de Val-Borro planet–disk interaction (see Section 4.1), a classical fargo problem. For the polar, strict, and mesh formulations, the coefficients used were $5e-4$, $3e-14$, and 20, respectively, which should give roughly the same amount of dissipation at $r = 1$.⁸

The figures show that the mesh hyperdiffusion formulation is superior, with shock viscosity $\nu_{\text{shock}} = 4$ best for all cases, allowing for better excitation of the gap edge vortices. That

⁸ For reproducibility, we quote the exact code options. For the mesh formulation, the options are `idiff='hyper3-mesh'`, `diffrho_hyper3_mesh=20` and `ivisc='hyper3-mesh'`, `nu_hyper3=20`. For the strict formulation, the options are `idiff='hyper3-strict'`, `diffrho_hyper3=3e-14` and `ivisc='hyper3-mu-strict-onthefly'`, `nu_hyper3=3e-14`. For the polar formulation, the options are `idiff='hyper3-cyl'`, `diffrho_hyper3=5e-4` and `ivisc='hyper3-cyl'`, `nu_hyper3=5e-4`. The version of the code used was #87ac0e5 on github (<https://github.com/pencil-code>).

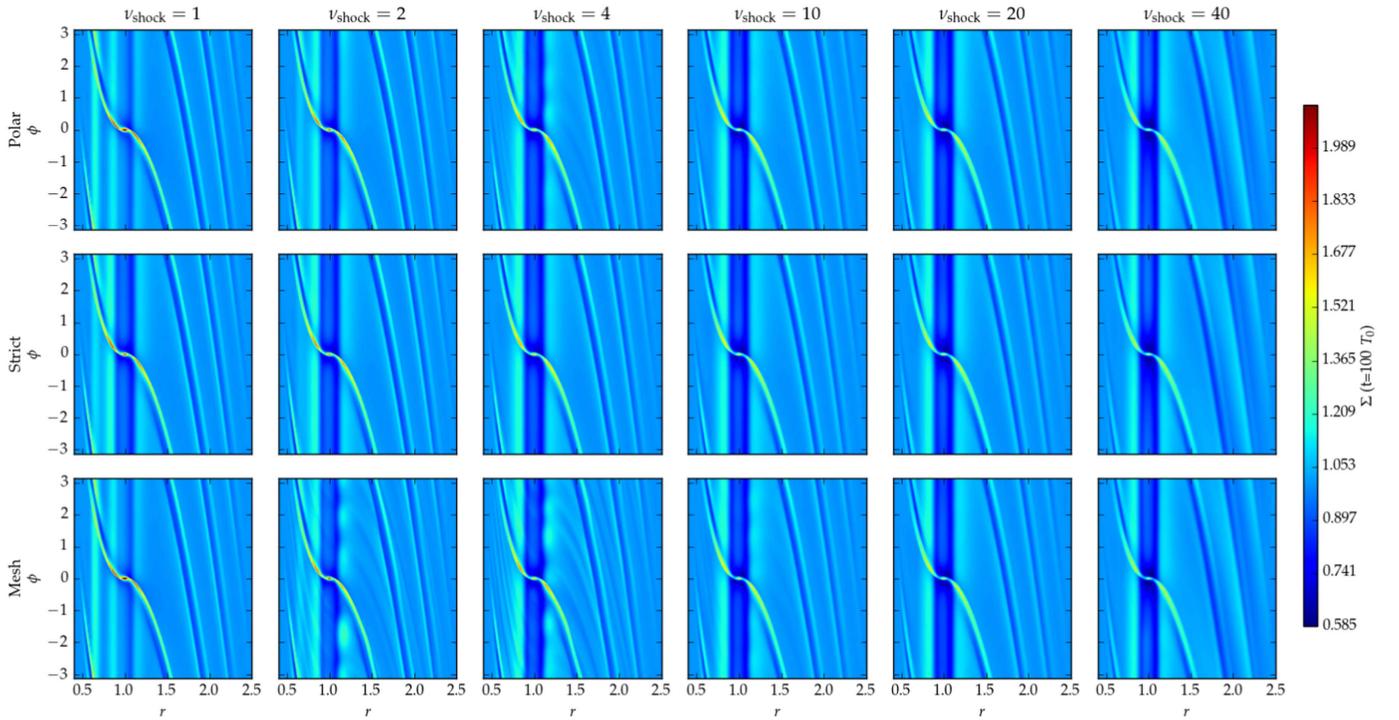


Figure 4. Testing the different hyperdiffusion schemes. The rows test different shock viscosity coefficients. The columns test different schemes: *polar* hyperdiffusion and *mesh* hyperdiffusion consider only the ∇^6 term. Polar scales as δx^4 , whereas mesh is resolution independent, scaling as δx^5 . *Strict* solves the $\nabla^2 \nabla^2 \nabla^2$ formulation. Judging from the excitation of the RWI at the outer gap edge, a shock viscosity of 4 is the best choice, working on both polar and mesh. Mesh also works well at other values of shock viscosity (2 and 10, respectively).

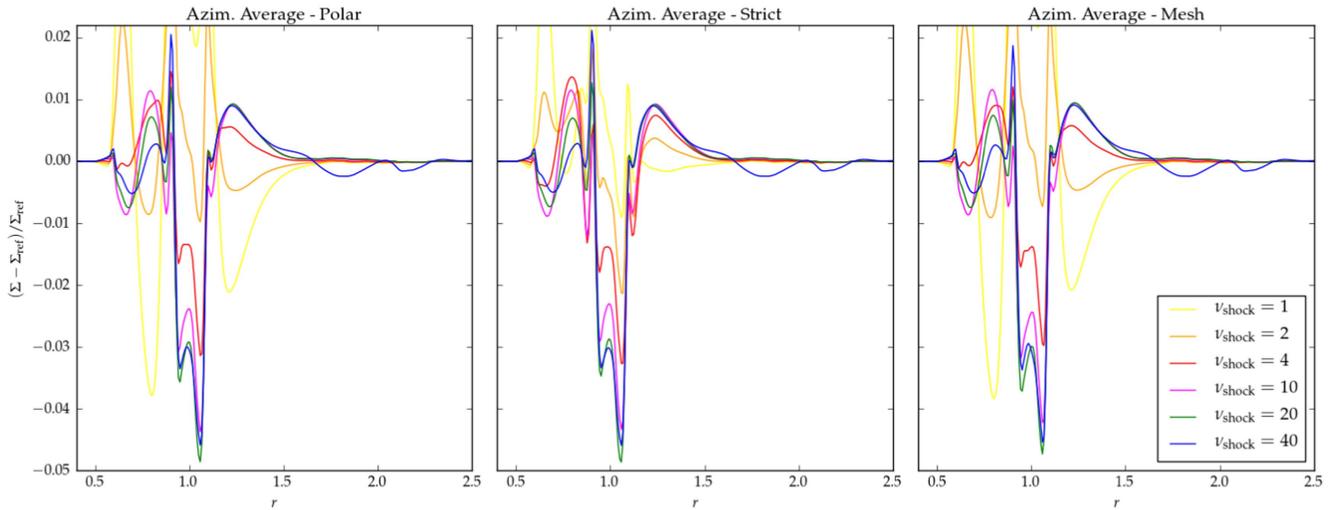


Figure 5. Comparison between the different hyperdiffusion methods at lower resolution (256×768) against a reference solution at higher resolution (1024×3072). Shock viscosities of 1 and 2 show large deviations from the reference solution not only at the gap but also at the edges, with mass concentration in the inner edge. The gap gets progressively shallower as the shock viscosity increases, until converging at $\nu_{\text{shock}} = 20$. This high value of artificial viscosity is undesirable, and we use $\nu_{\text{shock}} = 4$ instead.

level of shock viscosity is needed because of the high compressibility due to gap carving.

We construct now a high-resolution simulation where both numerical and artificial dissipation are minimized, and use it as a reference to benchmark the solutions at lower resolution that are more affected by high-frequency filters. In Figure 5 we show the azimuthal average of each of the runs, plotted against this reference solution. In each plot the average density for different shock viscosities is shown, for each hyperviscosity method. The reference solution is computed at resolution

1024×3072 , 4 times the usual resolution, so that the effect of hyperviscosity is diminished by a factor of 1000. The different runs show that increasing the shock viscosity has a non-monotonic effect on the gap structure. Shock viscosities of 1 and 2 (yellow and orange lines) show a large departure from the gap shape, as well as an accumulation next to the inner boundary. As the shock viscosity is increased, the gap profile becomes increasingly shallower than the reference value. Yet, when the shock viscosity is increased from 20 to 40, the behavior stops being monotonic, as the gap either converges at

the shock 20 level or gets slightly less shallow than it. However, such high values of an artificial coefficient such as shock viscosity should be avoided; it is seen in Figure 4 that the vortices at gap edges are quenched at the polar and mesh formulation at this high level of shock viscosity. We thus decide on $\nu_{\text{shock}} = 4$ as a good compromise between stabilizing the numerics and getting the physics correct.

4.2. Field Loop Advection

We test the advection algorithm with a field loop advection similar to that in Gardiner & Stone (2005), adapted to cylindrical coordinates. Formally, this is an MHD test, and is commonly used to test constrained transport schemes for the magnetic induction equation when written in terms of the magnetic field. However, it is a significantly simpler problem for the vector potential formulation of the induction equation, as a constrained transport scheme is not needed to ensure a divergence-free magnetic field in this case. Here, this test essentially demonstrates the advection errors generated by the scheme.

We perform the test in an $N_r \times N_\phi = 32 \times 64$ grid. The range in radius is from 1 to 2, and the range in azimuth from -0.5 to 0.5 . The radial boundary is non-periodic and “frozen,” i.e., the boundary values are fixed at the initial condition. The velocity is initialized in rigid rotation in the azimuthal direction, with $\Omega = 1$. An acceleration $\mathbf{g} = -\Omega^2 \mathbf{r}$ is added to maintain centrifugal balance. The sound speed $c_s = 0.01$, and the adiabatic index $\gamma = 1$.

The field loop is a patch of constant magnetic energy with sharp edges. In terms of the magnetic potential, it is

$$\mathbf{A} = \max[A_0(0.3 - d), 0] \hat{\mathbf{z}} \quad (32)$$

where $d = |\mathbf{r} - \mathbf{r}_0|$, with $|\mathbf{r}_0| = 1.5$ the location of the center of the loop. We use $A_0 = 10^{-3}$.

The left panels of Figure 6 show the initial condition for the magnetic potential (upper) and magnetic energy (lower). The middle panels show the result after 20 periods of evolution with (middle left) and without (middle right) the orbital advection algorithm. For the simulation with the orbital advection algorithm, at this resolution one period takes only two timesteps. With standard advection, one period takes about 200 timesteps, which adds considerably more diffusion. The difference between the cases with and without the algorithm is shown in the rightmost panels. The continuity, momentum, and induction equations are solved. No explicit hyperdiffusion is added.

4.3. MRI in a Cylindrical Domain

To demonstrate the use of the induction equation, we show here a simulation that produces MRI in a cylindrical geometry. This configuration is a derivative of that used in Lyra & Mac Low (2012), solving the same MHD equations in a similar domain. The cylindrical grid has a radial domain $r = [0.4, 2.0]$, azimuthal domain $\phi = [-\pi/2, \pi/2]$, vertical domain $z = [-0.1, 0.1]$ and resolution $N_r \times N_\phi \times N_z = 384 \times 192 \times 64$. A net vertical field is imposed, with a radial profile set so that three λ_{max} fit in the domain vertically at each radius, where λ_{max} is the most unstable wavelength of the linear MRI. The density is constant $\rho = 1$ and the gas is isothermal with constant sound speed, $c_s = 0.1$, and initially rotates in Keplerian equilibrium with angular velocity $\Omega = r^{-1.5}$. The inner and outer boundary

conditions are antisymmetric zero value for the radial velocity, zero second derivative for the azimuthal velocity, symmetrical for the vertical velocity, symmetrical for the density, and zero second derivative for all components of the magnetic vector potential. Noise at the level 10^{-4} is added in the radial region $[0.6, 1.8]$, and to stabilize the scheme, `hyper3-mesh` type diffusivities on all fields set with the coefficient 40 and $\nu_{\text{shock}} = 5$. We employ de Val-Borro et al. (2006) buffer zones with radial width 0.1 and driving timescale 0.1 local orbital periods on both radial boundaries. In Figure 7 we show the evolution of the ϕ , z -averaged azimuthal magnetic field, normalized by the ϕ , z -averaged pressure. The two panels show that the initial growth and saturation behavior are equivalent in the two formulations. Additionally, Figure 8 shows the gas density ρ at $T = 8$ orbits for both runs.

5. Conclusions

We have constructed an orbital advection algorithm for global MHD simulations using the vector potential, suitable for high-order time discretization. The algorithm relies on Fourier interpolation of the Keplerian advection term, as in the SAFI algorithm. The main differences between our algorithm and SAFI are the treatment of the induction equation, and that ours is generalized to polar coordinates. The magnetic potential evolution relies on expanding the electromotive force and split the resulting advection term into average orbital velocity and perturbation, the former solved implicitly as with the other variables. Although an advection term is required, this formulation does not require a gauge transformation to an advective gauge since the remainder of the induction equation, a shear term $u_j \partial_i A_j$, can be kept as it is in the right-hand side. Without orbital advection, this term would contribute terms that cancel the advection, but with orbital advection solved implicitly, the cancellation is avoided. This circumvents the presence of irrotational terms in the induction equation that spoil the quality of the numerical solution in the advective gauge (Candelaresi et al. 2011).

We prove analytically that the scheme is third-order accurate in time, as is the rest of the code, also showing that this accuracy is achieved numerically. As a consequence, we prove that SAFI is third-order accurate as well. We test the implementation with a standard field loop advection test (Gardiner & Stone 2005) modified for the cylindrical grid geometry, planet-disk interaction as benchmarked in de Val-Borro et al. (2006) and, finally, the excitation of the MRI, one of the main applications of MHD in disk physics. The de Val-Borro et al. (2006) test is of particular importance as it allowed us to test different high-frequency filters and artificial viscosity schemes in order to correctly reproduce the gap shape, as documented in the Appendix. In the future we will apply the algorithm to 3D spherical coordinates with stratification, as in Lyra et al. (2016).

This scheme is self-consistent and elegant in its simplicity, allowing for implementation without elaborate numerical tricks. Its relatively effortless treatment of magnetic field and accuracy makes it particularly efficient for straightforward finite-difference methods.

W.L. acknowledges support of Space Telescope Science Institute through grant HST-AR-14572 and the NASA Exoplanet Research Program through grant 16-XRP16_2-0065. C.P.M.

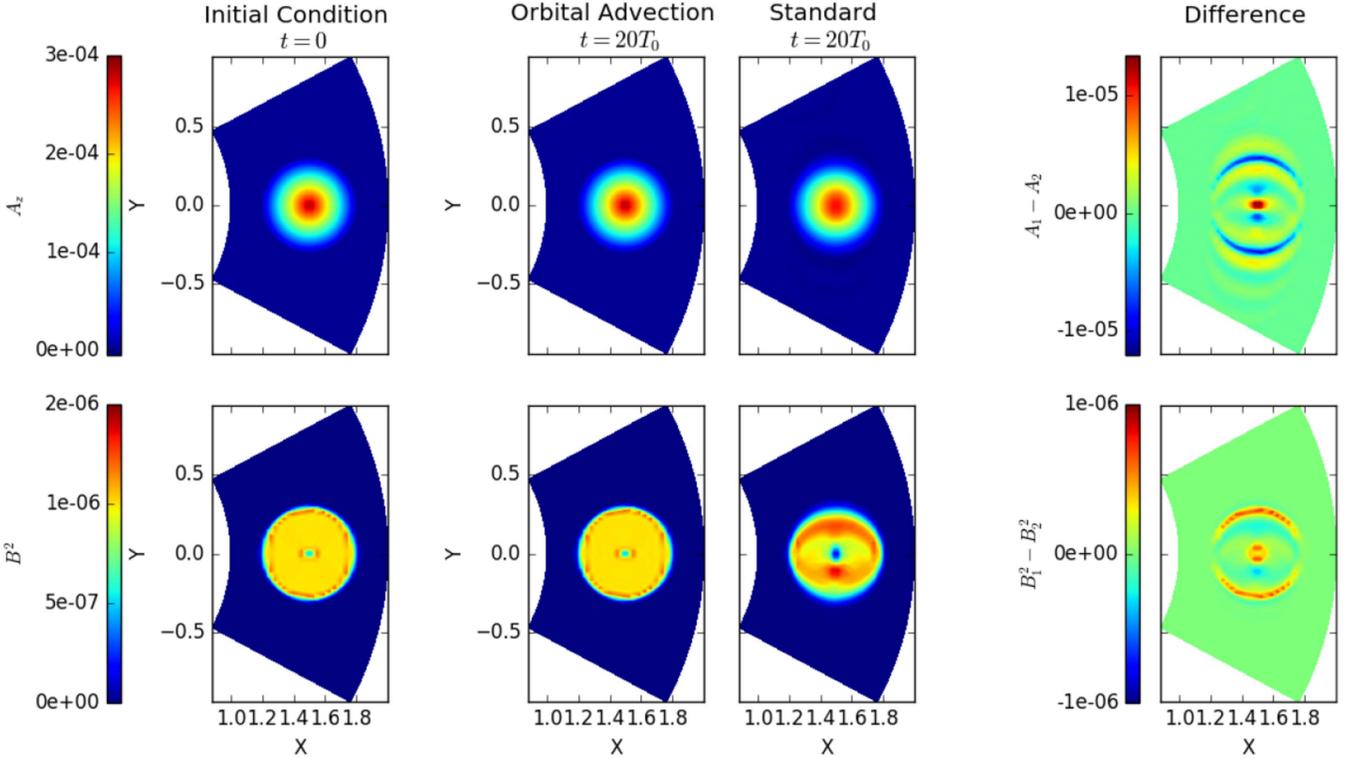


Figure 6. Test of the MHD orbital algorithm by advection of a field loop, a cylindrical coordinate adaptation of the test of Gardiner & Stone (2005). The upper panels show the magnetic potential, the lower ones the magnetic energy. The leftmost panels show the initial condition. The middle panels show the results after 20 revolutions of the field loop. The center-left panels are calculated with the orbital advection algorithm, the center-right ones without it. The rightmost panels show the difference between the results. The calculation without orbital advection results in significantly more numerical diffusion: one revolution takes about two timesteps with the algorithm, whereas the same time corresponds to over 200 timesteps otherwise.

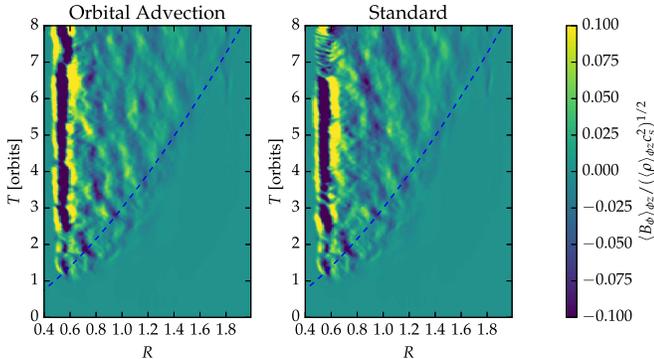


Figure 7. Comparison of the initial evolution of the MRI in a cylindrical domain. Shown are runs with our new azimuthal advection (left) and without (right). The blue dashed line denotes three local orbital periods at each radius.

acknowledges support of an STFC Consolidated grant awarded to the QMUL Astronomy Unit 2012–2016. This research was supported in part by the National Science Foundation under Grant No. NSF PHY11-25915. W.L., C.P.M., and T.H. thank the hospitality of the Kavli Institute for Theoretical Physics in Santa Barbara, CA, where this work was developed. The simulations presented in this paper utilized the Stampede cluster of the Texas Advanced Computing Center (TACC) at The University of Texas at Austin, and the Comet cluster at the University of California at San Diego, both through XSEDE grant TG-AST140014. Simulations were also run at the Queen Mary’s MidPlus computational facilities, supported by QMUL Research-IT and funded by EPSRC grant

EP/K000128/1. This work was performed in part at the Jet Propulsion Laboratory, California Institute of Technology.

Appendix

We document here the formulation of high-frequency filters as used in the PENCIL CODE in polar coordinates. The *strict* formulation is new and coded for the purpose of this work. The *polar* formulation is recapitulated from the discussion of hyperviscosity given in Lyra (2009). The *mesh* formulation was implemented by A. Brandenburg (2017, private communication).

Because of the relatively high-order discretization employed, Pencil has much reduced numerical dissipation compared to an analogous low-order scheme. In order to perform inviscid simulations, high-frequency filters can be used to provide extra dissipation for modes approaching the Nyquist frequency, as required to stabilize the method. The usual Laplacian viscosity $\nu \nabla^2 \mathbf{u}$ is equivalent to a multiplication by k^2 in Fourier space, where k is the wavenumber. Another tool is hyperviscosity, which replaces the k^2 dependency by a higher power law, k^n , $n > 2$. The idea behind it is to provide large dissipation only where it is needed, at the grid scale (high k), while minimizing it at the largest scales of the box (small k). In principle, one can use as high n as desired, but in practice we are limited by the stencil size used in the rest of the code. A multiplication by k^n is equivalent to an operator ∇^n in real space. As Pencil is designed to use sixth-order finite-difference stencils for the first and second derivatives, three ghost cells are available in each direction; thus the sixth-order derivative is the highest that can be conveniently computed. The hyperdissipation we use is

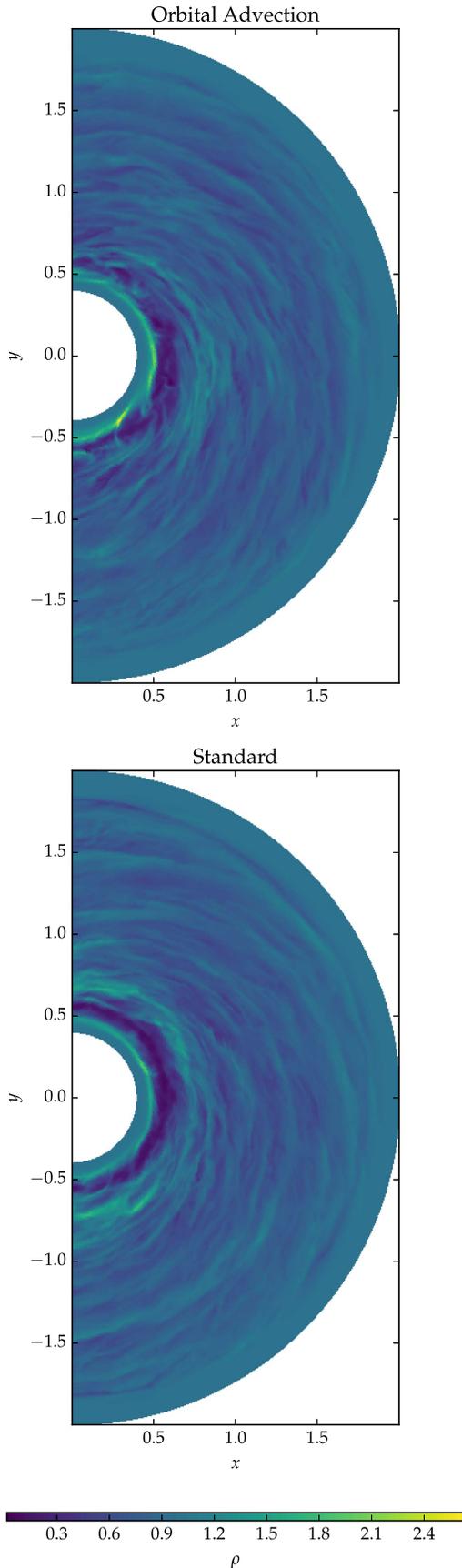


Figure 8. Comparison of the evolved state of the MRI in a cylindrical domain, being a cut of the gas density field ρ at $T = 8$ orbits. Shown are runs with our new azimuthal advection (top) and without (bottom).

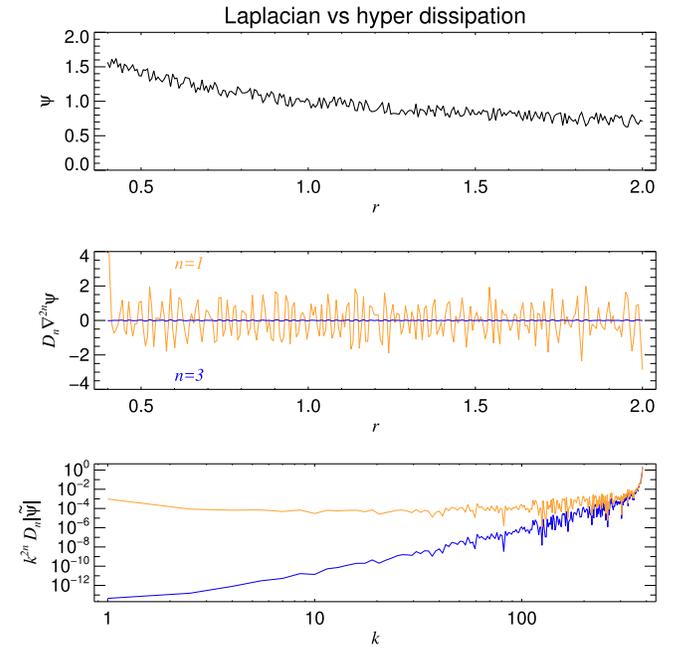


Figure 9. Dissipation acting on a scalar field ψ , for $n = 1$ (Laplacian dissipation) and $n = 3$ (third-order hyperdissipation). The field is initially seeded with noise (upper panel). For $n = 3$ the large scale is not affected as much as in the $n = 1$ case, which is seen by the larger wiggling of the latter in the middle panel. In Fourier space (lower panel) we see that near the grid scale both formulations give strong dissipation. It also illustrates that at the large scales ($k \simeq 1$), the effect of $n = 3$ is indeed negligible. The figure is reproduced from Lyra (2009).

therefore ∇^6 , or k^6 in Fourier space. Figure 9 illustrates how this tool maximizes the inertial range of a simulation. For further and evolving information and wisdom on the use of hyperdissipation, the reader is referred to the manual of the PENCIL CODE.⁹

A.1. Strictly Conservative Hyperdissipation

Hyperdiffusivity is meant purely as a numerical tool to dissipate energy at small scales and comes with no guarantee that results are convergent with regular second-order dissipation (see Haugen & Brandenburg 2004 for a discussion). In fact, large-scale dynamo action is known to be seriously altered in simulations of closed systems where magnetic helicity is conserved: this results in prolonged saturation times and enhanced saturation amplitudes (Brandenburg & Sarson 2002).

It is desirable to have the high-frequency filters obeying the conservation laws. So, for density we want a mass-conserving term, for velocities we want a momentum-conserving term, for magnetic fields we want a term conserving magnetic flux, and for entropy we want an energy-conserving term. These enter as hyperdiffusion, hyperviscosity, hyper-resistivity, and hyper heat conductivity terms in the evolution equations. To ensure conservation under transport, they must take the form of the divergence of the flux \mathcal{J} of the quantity ψ , so that the Gauss theorem applies and we have

$$\frac{\partial \psi}{\partial t} + \nabla \cdot \mathcal{J} = 0. \quad (33)$$

⁹ Some of the discussion in this section is adapted from entries in the manual of the PENCIL CODE, which were added by the authors.

For density, the flow due to mass diffusion is usually taken as the phenomenological Fick's law

$$\mathcal{J} = -D\nabla\rho \quad (34)$$

i.e., proportional to the density gradient, in the opposite direction. This leads to the usual Laplacian diffusion

$$\frac{\partial\rho}{\partial t} = D\nabla^2\rho \quad (35)$$

under the assumption that the diffusion coefficient D is isotropic. Higher-order hyperdiffusion of order $2n$ involves a generalization of Equation (34), to

$$\mathcal{J}^{(n)} = (-1)^n D^{(n)} \nabla^{2n-1}\rho. \quad (36)$$

In our case, we are interested in the case $n = 3$, so that the hyperdiffusion term is

$$\frac{\partial\rho}{\partial t} = D^{(3)}\nabla^6\rho, \quad (37)$$

where the operator

$$\nabla^6 \equiv \nabla^2(\nabla^2(\nabla^2)). \quad (38)$$

In Cartesian coordinates this is expanded into

$$\begin{aligned} \nabla^6 = & \frac{\partial^6}{\partial x^6} + \frac{\partial^6}{\partial y^6} + \frac{\partial^6}{\partial z^6} + 3\left(\frac{\partial^4}{\partial x^4}\frac{\partial^2}{\partial y^2} + \frac{\partial^4}{\partial x^4}\frac{\partial^2}{\partial z^2}\right. \\ & + \frac{\partial^2}{\partial x^2}\frac{\partial^4}{\partial y^4} + \frac{\partial^2}{\partial x^2}\frac{\partial^4}{\partial z^4} + \frac{\partial^4}{\partial y^4}\frac{\partial^2}{\partial z^2} + \left.\frac{\partial^4}{\partial z^4}\frac{\partial^2}{\partial y^2}\right) \\ & + 6\frac{\partial^2}{\partial x^2}\frac{\partial^2}{\partial y^2}\frac{\partial^2}{\partial z^2}. \end{aligned} \quad (39)$$

In cylindrical coordinates curvature terms would appear, involving powers of $1/r$. However, if we keep only the highest-order terms, those of order $\mathcal{O}(\Delta x^{-6})$, the curvature terms can be discarded. The only remaining terms are those in Equation (39), provided we do the trivial substitution $\partial x \rightarrow \partial r$ and $\partial y \rightarrow r\partial\phi$. We call this *strict* hyperdiffusion, since we strictly retain all the highest-order terms.

A.1.1. Hyperviscosity

Viscosity has some caveats where subtleties apply. The difference is that the momentum flux due to viscosity is not proportional to the velocity gradient, but to the rate-of-strain tensor

$$S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{1}{3}\delta_{ij}\nabla\cdot\mathbf{u}\right), \quad (40)$$

which only allows the viscous acceleration to be reduced to the simple formulation $\nu\nabla^2\mathbf{u}$ under the condition of incompressibility and constant dynamical viscosity $\mu = \nu\rho$. Due to this, the general expression for conservative hyperviscosity involves

more terms. In the general case, the viscous acceleration is

$$f_{\text{visc}} = \rho^{-1}\nabla\cdot(2\rho\nu\mathbf{S}). \quad (41)$$

So, for the hyperviscous force, we must replace the rate-of-strain tensor by a high-order version

$$f_{\text{visc}}^{(\text{hyper})} = \rho^{-1}\nabla\cdot(2\rho\nu_n\mathbf{S}^{(n)}) \quad (42)$$

where the n th-order rate-of-strain tensor is

$$\mathbf{S}^{(n)} = (-\nabla^2)^{n-1}\mathbf{S}. \quad (43)$$

For the $n = 3$ case it is

$$S_{ij}^{(3)} = \frac{1}{2}\left(\frac{\partial^5 u_j}{\partial x_i^5} + \frac{\partial^4}{\partial x_i^4}\left(\frac{\partial u_i}{\partial x_j}\right) - \frac{1}{3}\frac{\partial^4}{\partial x_i^4}(\nabla\cdot\mathbf{u})\right). \quad (44)$$

Plugging this into Equation (42), and assuming $\mu_3 = \rho\nu_3 = \text{const}$

$$f_{\text{visc}}^{(\text{hyper})} = \nu_3\left(\nabla^6\mathbf{u} + \frac{1}{3}\nabla^4(\nabla(\nabla\cdot\mathbf{u}))\right). \quad (45)$$

For $\nu_3 = \text{const}$, we have to take derivatives of density as well

$$f_{\text{visc}}^{(\text{hyper})} = \nu_3\left(\nabla^6\mathbf{u} + \frac{1}{3}\nabla^4(\nabla(\nabla\cdot\mathbf{u})) + 2\mathbf{S}^{(3)}\cdot\nabla\ln\rho\right) \quad (46)$$

Here we can again ignore the curvature terms as they will produce terms of lower order than $\mathcal{O}(\Delta x^{-6})$. Therefore, the ∇^6 operator applied to a vector is equal to the sum of ∇^6 on its components, as would ∇^2 in Cartesian coordinates. As for the other term, we can write for the x -component

$$\nabla^4(\nabla(\nabla\cdot\mathbf{u}))_x = f(u_x) + g(u_y) + h(u_z) \quad (47)$$

and retaining only the ∂^6 terms, we have

$$\begin{aligned} f(u_x) = & \left[\frac{\partial^6}{\partial x^6} + 2\left(\frac{\partial^4}{\partial x^4}\frac{\partial^2}{\partial y^2} + \frac{\partial^4}{\partial x^4}\frac{\partial^2}{\partial z^2}\right)\right. \\ & \left.+ \frac{\partial^4}{\partial y^4}\frac{\partial^2}{\partial x^2} + \frac{\partial^4}{\partial z^4}\frac{\partial^2}{\partial x^2}\right]u_x, \end{aligned} \quad (48)$$

$$\begin{aligned} g(u_y) = & \left(\frac{\partial^5}{\partial x^5}\frac{\partial}{\partial y} + \frac{\partial}{\partial x}\frac{\partial^5}{\partial y^5} + 3\frac{\partial^3}{\partial x^3}\frac{\partial^3}{\partial y^3}\right. \\ & \left.+ 2\frac{\partial^3}{\partial x^3}\frac{\partial}{\partial y}\frac{\partial^2}{\partial z^2} + 3\frac{\partial}{\partial x}\frac{\partial^3}{\partial y^3}\frac{\partial^2}{\partial z^2} + \frac{\partial}{\partial x}\frac{\partial}{\partial y}\frac{\partial^4}{\partial z^4}\right)u_y, \end{aligned} \quad (49)$$

and

$$\begin{aligned} h(u_z) = & \left(\frac{\partial^5}{\partial x^5}\frac{\partial}{\partial z} + \frac{\partial}{\partial x}\frac{\partial^5}{\partial z^5} + 3\frac{\partial^3}{\partial x^3}\frac{\partial^3}{\partial z^3}\right. \\ & \left.+ 2\frac{\partial^3}{\partial x^3}\frac{\partial}{\partial z}\frac{\partial^2}{\partial y^2} + 3\frac{\partial}{\partial x}\frac{\partial^3}{\partial z^3}\frac{\partial^2}{\partial y^2} + \frac{\partial}{\partial x}\frac{\partial}{\partial z}\frac{\partial^4}{\partial y^4}\right)u_z. \end{aligned} \quad (50)$$

Per symmetry, the formulation for the y and z components are identical under the permutation $[xyz]$. As a result of the present work, this formulation of hyperviscosity is

implemented in the PENCIL CODE. It is exact for Cartesian coordinates and accurate to first order in polar coordinates.

A.2. Non-conservative Hyperdiffusion

Equations (45) and (46) explicitly conserve linear and angular momentum. Although desirable properties, such expressions are cumbersome due to the mixed derivatives. Yet, the spectral range in which hyperviscosity operates is very limited and, as a numerical tool, only its performance as a high-frequency filter is needed. Since the terms are artificial, it is not clear if enforcing conservation on them is of great utility. We try thus other hyperdiffusion formulations that abandon strict conservation. We apply a simple hyperviscosity

$$f_{\text{visc}} = \begin{cases} \nu_3 \nabla^6 \mathbf{u} & \text{if } \mu = \text{const} \\ \nu_3 (\nabla^6 \mathbf{u} + 2 \mathbf{S}^{(3)} \cdot \nabla \ln \rho) & \text{if } \nu = \text{const}. \end{cases} \quad (51)$$

Notice that this can indeed be expressed as the divergence of a simple rate-of-strain tensor

$$S_{ij}^{(3)} = \frac{\partial^5 u_i}{\partial x_j^5}, \quad (52)$$

so it does conserve linear momentum. It does *not*, however, conserve *angular* momentum, since the symmetry of the rate-of-strain tensor was dropped. Thus, vorticity sinks and sources may be spuriously generated at the grid scale.

A symmetric tensor can be computed that conserves angular momentum and can be easily implemented

$$S_{ij} = \frac{1}{2} \left(\frac{\partial^5 u_i}{\partial x_j^5} + \frac{\partial^5 u_j}{\partial x_i^5} \right). \quad (53)$$

This tensor, however, is not traceless, and therefore accurate only for weak compressibility. It should work well if the turbulence is subsonic. We performed simulations of the MRI with both formulations, not finding significant differences for the turbulent problem studied. This supports the use of the highest-order terms only, since these are the ones that provide quenching at high k . We are thus drawn to a formulation that keeps only the $\partial^6/\partial q^6$ terms.

A.2.1. Polar Hyperdiffusion

The hyperdiffusion coefficient $D^{(3)}$ in Equation (37) can be calculated from D assuming that at the Nyquist frequency the two formulations (35) and (37) yield the same quenching. Considering a wave as a Fourier series of a generalized coordinate (q), one element of the series is expressed as

$$\psi_k = A e^{i(kq - \omega t)}. \quad (54)$$

Plugging it into the second-order diffusion Equation (35) we have the dispersion condition $i\omega = Dk^2$. The sixth-order version (37) yields $i\omega = D^{(3)}k^6$.

Equating both we have $D^{(3)} = Dk^{-4}$. This condition should hold at the grid scale, where $k = \pi/\Delta q$, therefore

$$D^{(3)} = D \left(\frac{\Delta q}{\pi} \right)^4. \quad (55)$$

And the discretized equation of motion is therefore

$$\frac{\Delta \psi}{\Delta t} = D^{(3)} \frac{\Delta^6 \psi}{\Delta q^6} = \frac{D}{\pi^4} \frac{\Delta^6 \psi}{\Delta q^2}. \quad (56)$$

This formulation, being dependent on line element Δq , is appropriate for use with polar coordinates. As such, we call it *polar* hyperdiffusion. The formulation for viscosity, resistivity, and heat conduction follows along the same lines.

A.2.2. Resolution-independent Mesh Hyper-Reynolds Number

The polar hyperdiffusion above scales with resolution in a way that it can be written in terms of a Laplacian second-order coefficient. In this way, the relationship between the second-order coefficient and the Reynolds number scales linearly with resolution.

One can rewrite this in terms of a coefficient that instead keeps the hyper-Reynolds number constant with resolution. The hyper-Reynolds number is

$$\text{Re}_{\text{grid}} = u_{\text{rms}} / (\nu_n k_{\text{Ny}}^{2n-1}) \quad (57)$$

where $k_{\text{Ny}} = \pi/\Delta q$ is the Nyquist wavenumber. The $n = 3$ hyperdiffusion coefficient should thus be $\nu^{(3)} = C \Delta q^5$, with $C = u_{\text{rms}} / (\text{Re } \pi^5)$ as proportionality coefficient. So, the discretized equation, in a way that strictly keeps the Reynolds number constant at the grid scale, is

$$\frac{\Delta \psi}{\Delta t} = D^{(3)} \frac{\Delta^6 \psi}{\Delta q^6} = \frac{u_{\text{rms}}}{\text{Re } \pi^5} \frac{\Delta^6 \psi}{\Delta q}. \quad (58)$$

Because u_{rms} is a dynamical quantity, a strict retention of constant Re would need to have a dynamical $D^{(3)}$ as well. Instead, we use a representative u_{rms} at start time, and write

$$D^{(3)} = \frac{D_{\text{mesh}} \Delta q}{60 \pi^5} = \frac{u_{\text{rms}}}{\text{Re } \pi^5} \frac{\Delta^6 \psi}{\Delta q} \quad (59)$$

so, that way, $D_{\text{mesh}}/60 = u_{\text{rms}}/\text{Re}$ sets the hyperdiffusion.

Like the polar hyperviscosity, this formulation sets the hyperviscosity in a coordinate-independent way. Moreover, it has the advantage that the Reynolds number is kept constant over the mesh as the cells in the polar or non-uniform grid differ in size. We call this formulation *mesh* hyperdiffusion for this reason. A similar choice is implemented in the code for shearing boxes, using the maximum velocity instead of u_{rms} and updated dynamically, described in Yang & Krumholz (2012).

ORCID iDs

Wladimir Lyra  <https://orcid.org/0000-0002-3768-7542>
Colin P. McNally  <https://orcid.org/0000-0002-2565-6626>
Tobias Heinemann  <https://orcid.org/0000-0002-2476-9733>

References

- Brandenburg, A. 2010, *MNRAS*, 401, 347
- Brandenburg, A., & Dobler, W. 2002, *CoPhC*, 147, 471
- Brandenburg, A., & Dobler, W. 2010, Pencil: Finite-difference Code for Compressible Hydrodynamic Flows, Astrophysics Source Code Library, ascl:1010.060
- Brandenburg, A., Nordlund, A., Stein, R. F., & Torkelsson, U. 1995, *ApJ*, 446, 741
- Brandenburg, A., & Sarson, G. 2002, *PhRvL*, 88, 055003

- Candelaresi, S., Hubbard, A., Brandenburg, A., & Mitra, D. 2011, *PhPI*, **18**, 012903
- de Val-Borro, M., Edgar, R. G., Artymowicz, P., et al. 2006, *MNRAS*, **370**, 529
- Duffell, P. C. 2016a, DISCO: 3-D moving-mesh magnetohydrodynamics package, Astrophysics Source Code Library, ascl:1605.011
- Duffell, P. C. 2016b, *ApJS*, **226**, 2
- Gardiner, T., & Stone, J. 2005, *JCoPh*, **205**, 509
- Haugen, N., & Brandenburg, A. 2004, *PhRvE*, **70**, 026405
- Johansen, A., Youdin, A., & Klahr, H. 2009, *ApJ*, **697**, 1269
- Johnson, B. M., Guan, X., & Gammie, C. F. 2008, *ApJS*, **177**, 373
- Koller, J., Li, H., & Lin, D. 2003, *ApJ*, **596**, L91
- Lesur, G., & Longaretti, P.-Y. 2005, *A&A*, **444**, 25
- Li, H., Colgate, S. A., Wendroff, B., & Liska, R. 2001, *ApJ*, **551**, 874
- Li, S., & Li, H. 2012, A Fast Parallel Simulation Code for Interaction between Proto-Planetary Disk and Embedded Proto-Planets: Implementation for 3D Code, Tech. Rep. LA-UR-12-22213, Los Alamos National Lab
- Lovelace, R., Li, H., Colgate, S., & Nelson, A. 1999, *ApJ*, **513**, 805
- Lyra, W. 2009, PhD thesis, Uppsala Univ.
- Lyra, W., Johansen, A., Zsom, A., Klahr, H., & Piskunov, N. 2009, *A&A*, **497**, 869
- Lyra, W., & Mac Low, M.-M. 2012, *ApJ*, **756**, 62
- Lyra, W., Richert, A., Boley, A., et al. 2016, *ApJ*, **817**, 102
- Maron, J. L., McNally, C. P., & Mac Low, M.-M. 2012, *ApJS*, **200**, 6
- Masset, F. 2000, *A&AS*, **141**, 165
- McNally, C. P., Maron, J. L., & Mac Low, M.-M. 2012, *ApJS*, **200**, 7
- Mignone, A., Flock, M., Stute, M., Kolb, S. M., & Muscianisi, G. 2012, *A&A*, **545**, A152
- Mignone, A., Tzeferacos, P., Zanni, C., et al. 2010, PLUTO: A Code for Flows in Multiple Spatial Dimensions, Astrophysics Source Code Library, ascl:1010.045
- Muñoz, D. J., Kratter, K., Springel, V., & Hernquist, L. 2014, *MNRAS*, **445**, 3475
- Springel, V. 2010, *MNRAS*, **401**, 791
- Stone, J. M., & Gardiner, T. A. 2010, *ApJS*, **189**, 142
- Umurhan, O. M., & Regev, O. 2004, *A&A*, **427**, 855
- Williamson, J. H. 1980, *JCoPh*, **35**, 48
- Yang, C.-C., & Krumholz, M. 2012, *ApJ*, **758**, 48