The Astrophysical Journal Supplement Series, 273:37 (16pp), 2024 August https://doi.org/10.3847/1538-4365/ad5af3

PADÉ: A Code for Protoplanetary Disk Turbulence Based on Padé Differencing

Karim Shariff
NASA Ames Research Center, Moffett Field, CA 94035, USA; karim.shariff@nasa.gov
Received 2024 February 22; revised 2024 June 19; accepted 2024 June 20; published 2024 August 8

Abstract

The PADÉ code has been developed to treat hydrodynamic turbulence in protoplanetary disks. It solves the compressible equations of motion in cylindrical coordinates. Derivatives are computed using nondiffusive and conservative fourth-order Padé differencing, which has higher resolving power compared to both dissipative shock-capturing schemes used in most astrophysics codes, as well as nondiffusive central finite-difference schemes of the same order. The fourth-order Runge–Kutta method is used for time stepping. A previously reported error-corrected Fargo approach is used to reduce the time step constraint imposed by rapid Keplerian advection. Artificial bulk viscosity is used when shock capturing is required. Tests for correctness and scaling with respect to the number of processors are presented. Finally, efforts to improve efficiency and accuracy are suggested.

Unified Astronomy Thesaurus concepts: Protoplanetary disks (1300); Computational methods (1965)

1. Introduction

This paper presents a code called PADÉ, developed to simulate hydrodynamic turbulence in protoplanetary disks. The code uses a Padé/compact finite-difference scheme (Lele 1992). Such schemes have spectral-like resolving power. This means that they approach the ability of a spectral method to compute derivatives exactly for all wavenumbers that the mesh can support. In particular, for advection problems they have zero diffusive errors (as do all central finite-difference schemes) and they have small dispersion errors across the wavenumber range. Thus they have the ability to more accurately treat the dynamics of the small scales supported by the mesh and produce energy spectra that have a wider inertial (power-law) range for turbulent flows.

On the other hand, most astrophysical codes employ Godunov-type schemes that were an elegant and mathematically supported breakthrough for capturing shock waves. Such schemes employ noncentral upwinded finite-difference or flux reconstruction schemes with a smart nonlinear dissipation that is necessary for capturing shocks, but which leads to excessive dissipation of vortical and other smooth features such as density waves. To fix this issue there have been attempts along many directions, which include hybrid methods (Adams & Shariff 1996; Pirozzoli 2002), nonlinear filtering (Yee & Sjögreen 2018), and vorticity-preserving schemes (Lerat et al. 2007; Seligman & Laughlin 2017; Seligman & Shariff 2019). Since shocks have not been observed in protoplanetary disk turbulence to date, we can sidestep the issue. For treating shock waves that are not too strong, the PADÉ code provides an optional artificial bulk viscosity treatment (Cook & Cabot 2005; Mani et al. 2009). This is not as elegant or oscillation free as Godunov methods, but is designed to apply a diffusivity only where the divergence $\nabla \cdot \boldsymbol{u}$ is strong.

The development of the PADÉ code was motivated by the discovery in the last two decades of a number of mechanisms for the generation of hydrodynamic turbulence in protoplanetary disks (see Lesur et al. 2022 for a comprehensive review). These include the vertical shear instability (VSI), convective overstability (COS), and the zombie vortex instability (ZVI). Each is most strongly amplified for a different range of $\Omega t_{\rm thermal}$, the timescale for radiative relaxation of temperature fluctuations back to the background (normalized by the orbital frequency, Ω). VSI is most strongly amplified when $\Omega t_{\rm thermal} \ll 1$, COS when $\Omega t_{\rm thermal} \sim 1$, and ZVI when $\Omega t_{\rm thermal} \gg 1$. Turbulence can also be driven by the magneto-rotational instability in the radially inner and outer regions of protoplanetary disks where ionization is sufficient.

As mentioned earlier, most astrophysical codes that are applied to protoplanetary disks use dissipative shock-capturing methods. These codes include ATHENA (Stone et al. 2008), ATHENA++ (Stone et al. 2020), PLUTO (Mignone et al. 2007), and FARGO3D (Benítez-Llambay & Masset 2016). The highest-order scheme provided in ATHENA and ATHENA++ is the third-order piecewise parabolic method. The highest-order scheme in PLUTO is a fifth-order weighted essentially nonoscillatory (WENO) finite-difference shock-capturing scheme denoted WENOZ_FD in the PLUTO manual (page 93).[1] FARGO3D employs a staggered mesh such that the scalar variables, density and internal energy per unit volume, are cell centered while vector quantities, velocity and magnetic field, are located at the centers of cell faces. The computation of fluxes at the cell faces employs upwinded interpolation, which is necessarily dissipative. Time advancement uses operator splitting whereby different sets of terms contributing to the time rate of change of flow quantities are time advanced separately, one after the other.

The one exception to the use of dissipative shock-capturing schemes is the PENCIL code (Brandenburg et al. 2021), which uses sixth-order central differencing. Central schemes, including the Padé scheme, produce oscillations at the Nyquist wavenumber of the mesh. To overcome this, PENCIL uses a dissipative fifth-order upwind biased scheme, while PADÉ uses a filter with a sharp cutoff (Section 3.3).

The rest of the paper presents the equations solved (Section 2), computational schemes (Section 3), various tests

[1] http://plutocode.ph.unito.it/userguide.pdf

(Section 4), and closing remarks (Section 5). Code availability is described after the Acknowledgments.

2. Discretized Equations

2.1. Transport Equations

The code solves the equations for mass and momentum (radial, angular, and vertical) transport written in as close to flux-divergence form as possible in cylindrical coordinates

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial z}(\rho u_z) + \frac{1}{r}\frac{\partial}{\partial r}(r\rho u_r) + \frac{1}{r}\frac{\partial}{\partial \phi}(\rho u_\phi) = 0, \tag{1}$$

$$\frac{\partial}{\partial t}(\rho u_r) + \frac{\partial}{\partial z}(\rho u_r u_z) + \frac{1}{r}\frac{\partial}{\partial r}[r(p + \rho u_r^2)] + \frac{1}{r}\frac{\partial}{\partial \phi}(\rho u_\phi u_r) - \frac{\rho u_\phi^2}{r} = \frac{p}{r} + \rho g_r + F_r^{\rm v}, \tag{2}$$

$$\frac{\partial}{\partial t}(\rho u_\phi r) + \frac{\partial}{\partial z}(\rho u_\phi r u_z) + \frac{1}{r}\frac{\partial}{\partial r}(r\rho u_\phi r u_r) + \frac{\partial}{\partial \phi}(p + \rho u_\phi^2) = r\boldsymbol{F}_\phi^{\rm v}, \tag{3}$$

$$\frac{\partial}{\partial r}(\rho u_z) + \frac{\partial}{\partial z}(p + \rho u_z^2) + \frac{1}{r}\frac{\partial}{\partial r}(r\rho u_z u_r) + \frac{1}{r}\frac{\partial}{\partial \phi}(\rho u_z u_\phi) = \rho g_z + F_z^{\rm v}. \tag{4}$$

Here $\boldsymbol{F}^{\rm v}$ represents an optional viscous force, given in Appendix A. Implementation of characteristic boundary conditions requires calculation of the flux Jacobian, and to make the flux look similar to that in Cartesian coordinates, the term $-\partial p/\partial r$ that normally appears on the right-hand side of the radial momentum equation (Equation (2)) was moved into the radial advective flux on the left-hand side. This is accomplished by adding the quantity

$$\frac{1}{r}\frac{\partial}{\partial r}(rp) = \frac{1}{r}\left(r\frac{\partial p}{\partial r} + p\right) = \frac{\partial p}{\partial r} + \frac{p}{r}, \tag{5}$$

to both sides of Equation (2), which results in the source term p/r on the right-hand side.

For the locally isothermal option, the pressure is computed as $p = \rho c_{\rm i}^2(r)$, where $c_{\rm i}(r, z)$ is the local isothermal sound speed. For the nonisothermal option we have

$$p = (\gamma - 1)e_{\rm int}, \tag{6}$$

where $e_{\rm int}$ is the internal energy (per unit volume). The rationale for using the internal energy instead of its sum with the kinetic energy is given in our work on the Fargo method (Shariff & Wray 2018). The transport equation for $e_{\rm int}$ is

$$\frac{\partial e_{\rm int}}{\partial t} + \frac{\partial}{\partial z}(e_{\rm int}u_z) + \frac{1}{r}\frac{\partial}{\partial r}(re_{\rm int}u_r) + \frac{1}{r}\frac{\partial}{\partial \phi}(e_{\rm int}u_\phi) = -p\nabla\cdot\vec{u} + Q^{\rm v} - \nabla\cdot\vec{q}_{\rm cond}. \tag{7}$$

The first term on the right-hand side of Equation (7) is the pressure-dilatation term, which causes heating under compression. The dilatation is given by

$$\nabla\cdot\vec{u} = \frac{\partial u_z}{\partial z} + \frac{1}{r}\frac{\partial}{\partial r}(ru_r) + \frac{1}{r}\frac{\partial u_\phi}{\partial \phi}. \tag{8}$$

The last two terms $Q^{\rm v} - \nabla\cdot\vec{q}_{\rm cond}$ on the right-hand side of Equation (7) represent viscous heating and conductive heat transfer, respectively. They are activated only if viscous terms are activated; their form is given in Appendix A.

The variables evolved at each grid point are $\vec{q} = (\rho, \rho u_r, \rho u_\phi r, \rho u_z, e_{\rm int})$. When spatial derivatives are approximated by Padé finite differences (discussed below), one obtains a system of ordinary differential equations for the time rate of change, $\partial_t\vec{q}$ of $\vec{q}$ at each grid point. This system is evolved using the fourth-order Runge–Kutta method. Note that we do *not* employ operator splitting, whereby different sets of terms in $\partial_t\vec{q}$ are time evolved separately one after another; this would produce a splitting error, which we do not incur.

Currently, the main source of gravity in the code is from a mass M at the origin so that

$$g_r = -\frac{GM}{R}\frac{r}{R},\ g_z = -\frac{GM}{R}\frac{z}{R},\ R \equiv (r^2 + z^2)^{1/2}, \tag{9}$$

which are precomputed. The code also allows other simple choices such as uniform g_z and the thin disk version of Equation (9).

2.2. Fargo

The Fargo method was introduced by Masset (2000) to alleviate the time step restriction resulting from fast Keplerian advection. Shariff & Wray (2018) improved its accuracy by first noting that underlying the method is a transformation of the azimuthal coordinate ϕ

$$\phi' = \phi - \overline{\Omega}(r)(t - t_0), \tag{10}$$

$$r' = r, \tag{11}$$

$$t' = t, \tag{12}$$

for the duration of a time step, $t_0 < t \leqslant t_0 + \Delta t$. Here $\overline{\Omega}(r)$ is a prescribed rotation rate that one wishes to subtract from determining the time step. At the end of the time step, one brings the flow field back to original coordinates by performing a shift using a fast Fourier transform (FFT). The chain rule for differentiation then implies that every t and r derivative in the transport equations carries an additional term

$$\frac{\partial}{\partial t} = \frac{\partial}{\partial t'} - \frac{\partial \phi'}{\partial t}\frac{\partial}{\partial \phi'} = \frac{\partial}{\partial t'} - \overline{\Omega}(r)\frac{\partial}{\partial \phi'}, \tag{13}$$

$$\frac{\partial}{\partial r} = \frac{\partial}{\partial r'} - \frac{\partial \phi'}{\partial r}\frac{\partial}{\partial \phi'} = \frac{\partial}{\partial r'} - (t - t_0)\frac{\partial \overline{\Omega}}{\partial r}\frac{\partial}{\partial \phi'}, \tag{14}$$

$$\frac{\partial}{\partial \phi} = \frac{\partial}{\partial \phi'}. \tag{15}$$

The second term in $\partial/\partial t$ serves to remove $\overline{\Omega}(r)$ from the azimuthal advection velocity, therefore, it no longer influences the time step. To see how this works, consider the transformed mass transport Equation (1)

$$\frac{\partial \rho}{\partial t'} + \frac{1}{r}\frac{\partial}{\partial \phi}(\rho u'_\phi) + \frac{1}{r}\left[\frac{\partial}{\partial r'} + \chi\right](r\rho u_r) + \frac{\partial}{\partial z}(\rho u_z) = 0, \tag{16}$$

where

$$u'_\phi = u_\phi - \bar{\Omega}(r)r, \tag{17}$$

is a shifted velocity that results in a less restrictive time step. In Equation (16), the symbol χ denotes the operator

$$\chi = -(t - t_0)\frac{\partial \overline{\Omega}}{\partial r}\frac{\partial}{\partial \phi}. \tag{18}$$

It arises from the second term on the right-hand side of the last member of Equation (14), and must be added to every r derivative in the transport equations; the code refers to it as the "extra operator." The extra operator χ is missing in the original implementation (Masset 2000); its neglect results in an error of $\mathcal{O}(\Delta t)$. The original implementation also "integerizes" the prescribed $\overline{\Omega}(r)$ to allow a shift of the flow field by an integer number of grid intervals in ϕ at the end of a time step. The integer shift jumps at certain radial locations and this results in additional error at those locations.

This code provides options to completely or partially revert to Masset's original algorithm if desired. In particular, integer shifting can be selected as opposed to the more accurate real-valued shifting. Also, inclusion of the extra operator χ can be suppressed. At the end of each time step, the flow field is shifted back to original coordinates. Real-valued shifting is performed using an FFT from the FFTW library (Frigo & Johnson 2005).

Following publication of Shariff & Wray (2018), we were contacted by Prof. Pablo Benítez-Llambay (Universidad Adolfo Ibáñez, Chile) who is a main developer of FARGO3D. He pointed out that FARGO3D performs advection by operator splitting, and therefore there is no coordinate transformation requiring chain-rule terms. For instance, consider the continuity equation with only ϕ advection for simplicity

$$\frac{\partial \rho}{\partial t} + \frac{1}{r}\frac{\partial}{\partial \phi}(\rho u_\phi) = 0, \tag{19}$$

with the decomposition

$$u_\phi = \overline{\Omega}(r)r + u'_\phi. \tag{20}$$

In the operator splitting approach, ρ is first partially evolved according to advection by the residual velocity u'_ϕ

$$\frac{1}{r}\frac{\partial}{\partial \phi}(\rho u'_\phi), \tag{21}$$

whose time step restriction is not severe. Next, the term $\overline{\Omega}(r)\partial_\phi \rho$, representing advection by Keplerian flow, is treated in FARGO3D by real-valued shifting (C. McNally, Queen Mary University of London, private communication). Thus, it is true that no coordinate transformation is implied. However, operator splitting has an $\mathcal{O}(\Delta t^2)$ error which is of the same order as the error made by dropping the extra chain-rule terms. Thus, it would appear that for higher-order time integration schemes, chain-rule terms are necessary for consistency.

2.3. Artificial Pressure for Shock Capturing

Padé schemes, like spectral schemes, are not designed to capture shocks. However, one may encounter shocks in protoplanetary disks. Examples include the infall accretion shock (Neufeld & Hollenbach 1994), and bow shocks due to solid bodies moving supersonically relative to the gas. For these reasons, the code allows for an optional treatment of shocks using artificial bulk viscosity (Cook & Cabot 2005; Mani et al. 2009). This method results in an artificial pressure, $p_{\rm art}$, which is then added to the physical pressure. The actual calculation of $p_{\rm art}$ is

$$p_{\rm art} = -\beta_\Delta \nabla \cdot \vec{u}, \tag{22}$$

where β_Δ is the artificial bulk viscosity. Note that the artificial pressure is positive in regions of compression (dilatation $\nabla \cdot \vec{u} < 0$) and negative in regions of expansion. The artificial bulk viscosity is made sensitive to the dilatation as follows

$$\beta_\Delta = C_{\rm ap}\rho \ell^2_{\rm grid}|\nabla \cdot \vec{u}|, \tag{23}$$

where $C_{\rm ap}$ is a user-specified coefficient and the grid size squared is

$$\ell^2_{\rm grid} = \begin{cases} \Delta z^2, & \text{in 1D;} \\ r\Delta\phi\Delta r, & \text{in the planar case;} \\ (r\Delta\phi\Delta r\Delta z)^{2/3}, & \text{in 3D.} \end{cases} \tag{24}$$

Since the absolute value function is not smooth, the Padé filter (Section 3.3) with $\epsilon_{\rm filter} = 0.2$ is applied to β_Δ.

The artificial pressure term imposes a time step constraint appropriate for a viscous (second-derivative) term. The time step must satisfy

$$(\lambda_{\rm ap})_{\rm max}\,\Delta t < {\rm CFL}, \tag{25}$$

where $(\lambda_{\rm ap})_{\rm max}$ is the maximum eigenvalue of the bulk viscosity operator and CFL is the Courant–Friedrichs–Lewy limit specified by the user. For the fourth-order Runge–Kutta method, ${\rm CFL} < \sqrt{8}$ for stability. The maximum eigenvalue is estimated as

$$(\lambda_{\rm ap})_{\rm max} = \left(\frac{\pi^2\beta_\Delta}{\rho\ell^2_{\rm grid}}\right)_{\rm max}, \tag{26}$$

where a factor of π comes from assuming that each numerical derivative has spectral accuracy. Recall that $\beta_\Delta \propto \ell^2_{\rm grid}$ so the grid size actually drops out in Equation (26). For the shock tube and density wave test cases we report below, this eigenvalue was a factor of 1.08 and 1.61 larger, respectively, than the eigenvalue for the Euler terms.

3. Numerics

3.1. Padé Differentiation

The motivation for Padé differencing (also referred to as compact differencing) is its high resolving power (Section 3.2); see Lele (1992) for a comprehensive presentation and various extensions. The idea and initial development of such schemes is due to the Czech-born astronomer and numerical analyst Zdeněk Kopal around 1959; see the bibliographical notes (item IX-C) in his book (Kopal 1955). The basic idea, developed in Chapter 9 of the book, is to write the exact first-derivative operator as an exact function of the central difference operator. This operator function is first expanded in a truncated Taylor polynomial, which results in a conventional difference scheme. The key idea, however, is to next obtain a rational polynomial approximation (known as a Padé approximant) to the Taylor series. It is known that Padé[2] approximants to ordinary functions have a greater range of accuracy and radius of

[2] Henri Padé was a French mathematician who, for his doctoral thesis, studied (c. 1890) the approximation of functions by rational polynomials, now known as Padé approximants.

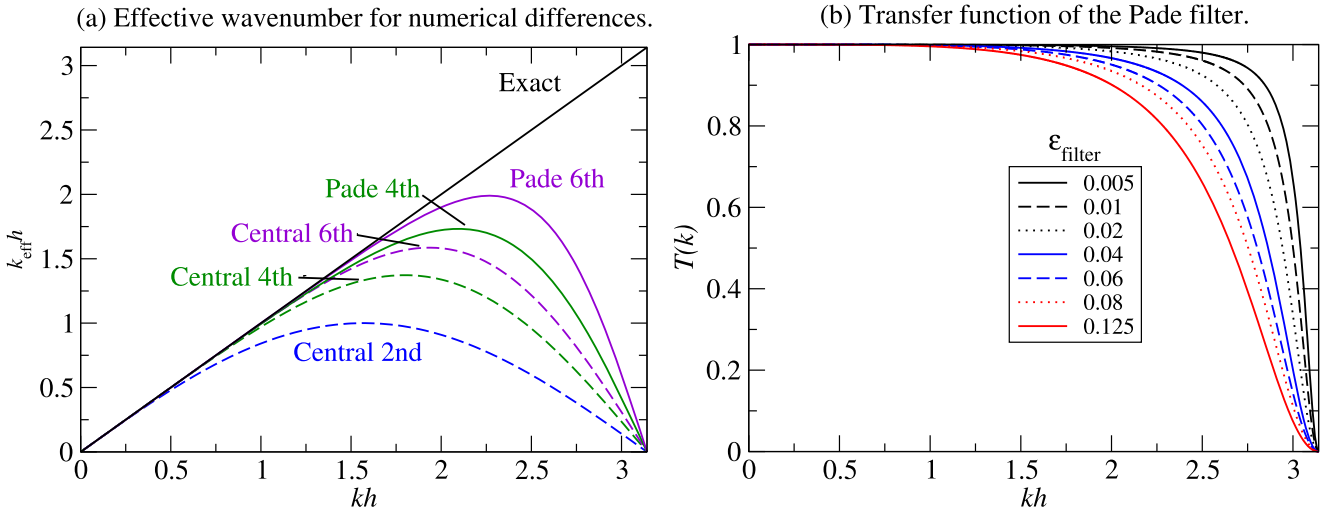

Figure 1. (a) Effective wavenumber for various difference schemes. (b) Transfer function, $T(k)$, for the Padé filter. The grid spacing is h. The limit $kh = \pi$ corresponds to the Nyquist mode, which has wavelength $2h$.

convergence than a Taylor series. In the present case, one obtains difference schemes with better resolving power. The simplest scheme, and the one we use both in the interior of nonperiodic domains and for periodic domains, is

$$\alpha f'_{j-1} + f'_j + \alpha f'_{j+1} = \frac{a}{2h}(f_{j+1} - f_{j-1}), \tag{27}$$

where h is the uniform grid spacing, $\alpha = 1/4$, and $a = 3/2$. Equation (27) constitutes a tridiagonal system of equations along each line of data in the mesh. The system is solved efficiently using the Thomas algorithm, which is simply Gaussian elimination applied to a tridiagonal matrix. Since Gaussian elimination is recursive, namely, operating on row j depends on the result of row $j-1$, the memory cache cannot be preloaded with the required data. Similarly, any available vectorization hardware cannot be engaged by the compiler. To overcome this, we follow the standard practice of having each step of the Thomas algorithm inner loop over a bundle of independent inversions for different grid lines of the mesh. Equation (27) is fourth-order accurate, i.e., its truncation error is $\mathcal{O}(h^4)$. To allow for nonuniform meshes, numerical differentiation is performed with respect to the continuous grid index variable ξ (such that $\xi_j = j$ and $h = 1$), and the chain rule is used, e.g.,

$$\frac{\partial f}{\partial z} = \frac{\partial f}{\partial \xi}\left(\frac{dz}{d\xi}\right)^{-1}. \tag{28}$$

Kopal's operator calculus is abstruse and unwieldy, but once the basic form of Padé schemes is recognized, an easier approach to develop them is to simply write down a specific form with a desired grid-point stencil, and obtain the unknown coefficients by setting the Taylor series error to 0 at various orders. This is the approach followed in Lele (1992), who developed a number of extensions, including resolving power optimization, boundary schemes, higher derivatives, conservation, and filtering.

For robustness, it is desirable that a code maintain positivity of density and internal energy. Negative values can arise in strongly evacuated regions for low-order schemes, and near very strong discontinuities for higher-order schemes; see, for example, Hu et al. (2013), who present a simple method for ensuring positivity for finite-difference schemes that can be written as a difference of numerical fluxes. The present method does not guarantee positivity; however, the code has not encountered difficulties for problems of subsonic turbulence for which it is intended. In the future, it may be possible to implement the method of Hu et al. (2013) using the "reconstruction by primitive function" trick (Harten et al. 1987; Shu & Osher 1989), which is also discussed in Merriman (2003). In this method, one obtains numerical fluxes by differentiating the primitive function; this differentiation would be performed using the Padé scheme.

3.2. Resolving Power of Padé Differencing

The advantage of Padé schemes is, first of all, their compactness: for Equation (27), for instance, fourth-order accuracy is obtained with a stencil width of 3 rather than 4 in the case of standard central differencing. More importantly, they have better resolving power than standard central differencing. This means that for the same formal order of accuracy, they provide an accurate derivative up to higher wavenumbers. This is quantified by the so-called effective wavenumber analysis, where one substitutes

$$f_j = \exp(\mathrm{i}kx_j) \text{ and } f'_j = \mathrm{i}k_{\rm eff}(k)\exp(\mathrm{i}kx_j),\ 0 \leqslant k \leqslant \pi/h, \tag{29}$$

into Equation (27) to obtain the effective wavenumber $k_{\rm eff}(k)$. Figure 1(a) displays the effective wavenumber for various schemes and shows that the fourth-order Padé scheme has better resolving power than a conventional sixth-order scheme. In general, $k_{\rm eff}(k)$ is complex and its imaginary part represents numerical dissipation of the scheme. The fact that $k_{\rm eff}(k)$ is real for central schemes means that they have a dispersion error but no dissipation error (for periodic boundary conditions). As as example of how to use the effective wavenumber diagram, we estimate by eye that the highest wavenumber for which we can trust the fourth-order Padé scheme is $kh = 1.5$, which implies that the smallest number of grid points per wavelength for which the scheme is accurate is about 4.

3.3. Padé Filtering

It is known that in the presence of nonlinearity or nonuniform grids, Padé schemes produce small 2Δ waves at every time step, which can grow if not controlled; this is true for conventional central schemes as well. Our remedy is to apply a minimum amount of Padé filtering (Lele 1992, Section C.2), which has a sharp cutoff and targets the very highest wavenumbers. We also use Padé filtering as an implicit subgrid treatment; this is discussed in Section 4 where we use it for this purpose in an axisymmetric simulation of VSI.

We use the fourth-order filter of Lele (1992), which for periodic boundary conditions, or in the interior of nonperiodic domains, has the form

$$aU_{j-1} + U_j + aU_{j+1} = P(u_{j-2} + u_{j+2}) + Q(u_{j-1} + u_{j+1}) + Ru_j, \tag{30}$$

where U_j are the filtered u_j. The conditions for fourth-order accuracy are

$$a = -\frac{1}{2} + 2Q, \quad R = \frac{1}{2}(2 + 3a - 3Q), \quad P = \frac{1}{4}(a - Q). \tag{31}$$

The reader can verify that there is no filtering when $Q = 1/2$. To specify the strength of the filter, the code uses the parameter $\epsilon_{\rm filter}$ such that

$$Q = \frac{1}{2} - \frac{1}{4}\epsilon_{\rm filter}. \tag{32}$$

The transfer function $T(k)$ of the filter versus wavenumber can be obtained by substituting $u_j = e^{ikx_j}$ and $U_j = T(k)e^{ikx_j}$ (with $x_j = jh$) into Equation (30). Figure 1(b) shows $T(k)$ for various values of $\epsilon_{\rm filter}$. Axisymmetric simulations for $\epsilon_{\rm filter}$ values ranging from 0.01 to 0.125 will be presented in Section 4.5.

For nonperiodic boundaries, we use the boundary treatment developed by A. Wray (private communication) that is conservative, i.e., it preserves

$$\sum_{j=1}^{N} u_j h_j. \tag{33}$$

The formulae applied at $j = 2$ and $N - 1$ are

$$BU_1 + CU_2 + DU_3 = Eu_1 + Fu_2 + Gu_3 + Hu_4, \tag{34}$$

$$BU_N + CU_{N-1} + DU_{N-2} = Eu_N + Fu_{N-1} + Gu_{N-2} + Hu_{N-3}, \tag{35}$$

with

$$B = 2a, \quad C = 1 + a, \quad D = a, \quad E = \frac{1}{4}(7a + Q), \quad F = \frac{1}{4}(4 + 7a - 3Q), \quad G = \frac{1}{4}(a + 3Q), \quad H = \frac{1}{4}(a - Q). \tag{36}$$

The boundary values u_1 and u_N are unchanged.

The first item of Table 9 in Lele (1992) lists the the leading-order truncation error for the filter (Equation (30))

$$U(x) = \left[1 + \frac{1}{16}(1 - 2a)h^4\partial^4/\partial x^4 + \cdots\right]u(x). \tag{37}$$

Next consider the expression

$$u(x, t + \Delta t) = (1 + \nu_4 \Delta t \partial^4/\partial x^4)u(x), \tag{38}$$

which represents application of an Euler step to

$$\frac{\partial u}{\partial t} = \nu_4 \frac{\partial^4 u}{\partial x^4}. \tag{39}$$

Comparing Equations (37) and (38) gives

$$\nu_4 = \frac{1}{16}(1 - 2a)\frac{h^4}{\Delta t}. \tag{40}$$

Hence, to leading order, Padé filtering corresponds to fourth-order hyperviscosity. Substituting the first member of Equations (31) and (32) into Equation (40) gives

$$\nu_4 = \frac{1}{16}\frac{\epsilon_{\rm filter}}{\Delta t}h^4. \tag{41}$$

Equation (40) implies that as the time step is reduced $\epsilon_{\rm filter}$ should be reduced. It should be noted that the matrix associated with the Padé filter becomes ill-conditioned for very small values of $\epsilon_{\rm filter}$ and a sufficiently large number of grid points. To alleviate this, the filter is applied every $N_{\rm filter}$ time steps (rather than after every step) and with $\epsilon_{\rm filter}$ increased by $N_{\rm filter}$. In that case one should replace Δt with $\Delta t N_{\rm filter}$ in Equation (41).

3.4. Boundary Schemes and Global Conservation

At the end points of a nonperiodic direction ($j = 1$ and N), the scheme in Equation (27) involves values and derivatives outside the domain. Therefore, at $j = 1$ and N we use the third-order one-sided scheme from Equation (4.1.1) in Lele (1992)

$$f_1' + \alpha_1 f_2' = h^{-1}(a_1 f_1 + b_1 f_2 + c_1 f_3), \tag{42}$$

$$f_N' + \alpha_1 f_{N-1}' = -h^{-1}(a_1 f_N + b_1 f_{N-1} + c_1 f_{N-2}), \tag{43}$$

with

$$\alpha_1 = 2,\ a_1 = -15/16,\ b_1 = 2, \text{ and } c_1 = 1/2. \tag{44}$$

According to a theorem for hyperbolic initial boundary value problems, one can reduce the order of accuracy at the boundary by 1 without affecting the global order of accuracy (Gustafsson 1981). The scheme used at the interior points $j \in [2, N-1]$ is Equation (27).

We wish the finite-difference scheme to possess a discrete conservation property. Specifically, we require that a discrete version of the Leibniz integral rule be satisfied: a numerical

integral of the numerical derivative should reduce to a difference of boundary values. This achieved for the Padé scheme as described in Lele (1992, Section 4.2) and Brady & Livescu (2019). Its application to the present scheme is described in Appendix B. Briefly, the condition to be satisfied is that the row entries in columns 2 to $N-1$ of matrix B must have a weighted sum of 0. Here B is the matrix representation of the right-hand side of Equation (27)

$$A\boldsymbol{f}' = \frac{1}{h}B\boldsymbol{f}. \tag{45}$$

Appendix B shows that this condition is satisfied for the present boundary scheme. An alternate way to obtain conservation is the "reconstruction by primitive function" trick referred to above.

3.5. Data Partitioning for Parallelization

Since an entire line of data along x is needed to compute a Padé derivative along x, for any direction x, we employ the pencil data structure. Each processor is assigned a pencil of data that can be thought of as a long brick with the long side along the direction of differentiation. Most of the work is done with z pencils (i.e., with z as the long direction). This work includes initialization, output, and time integration substeps. To compute r derivatives, we perform a so-called "transpose" such that each processor also has r pencils, and similarly for the ϕ derivatives. The flow-field arrays with z, r, and ϕ penciling, respectively, are dimensioned as follows

```
q  (sr:er, sphi:ephi, nz, ndof)
! pencil along z
q_r_space     (sphi:ephi,  sz_r:ez_r,  ndof,
nr)   ! pencil along r
q_phi_space(sr:er, sz_phi:ez_phi, ndof, nphi)
! pencil along phi
```

Here `ndof` refers to the number of degrees of freedom (number of flow variables) at each grid point and the other dimensions can be read, for example, as follows: `sz_r:ez_r` —starting z index to ending z index for an r pencil. Partitioning and transpose routines were taken from Alan Wray's STELLARBOX code.

4. Tests

4.1. Convergence for Two-dimensional Advection

Here we solve the equation for a scalar, $f(z, \phi, t)$, uniformly advecting in the z and ϕ directions

$$\frac{\partial f}{\partial t} + \frac{\partial f}{\partial z} + \frac{\partial f}{\partial \phi} = 0,\ z \in [0, 2\pi),\ \phi \in [0, 2\pi), \tag{46}$$

with periodic boundary conditions and the initial condition

$$f(z, \phi, 0) = 1 + \frac{1}{2}\sin 2z \sin 2\phi. \tag{47}$$

The CFL is fixed at unity and the number of grid points n_z and n_ϕ is varied.

Figure 2 plots the rms error at $t = 6\pi$ compared with the exact solution at $t = 6\pi$. The rate of convergence is seen to be fourth order.

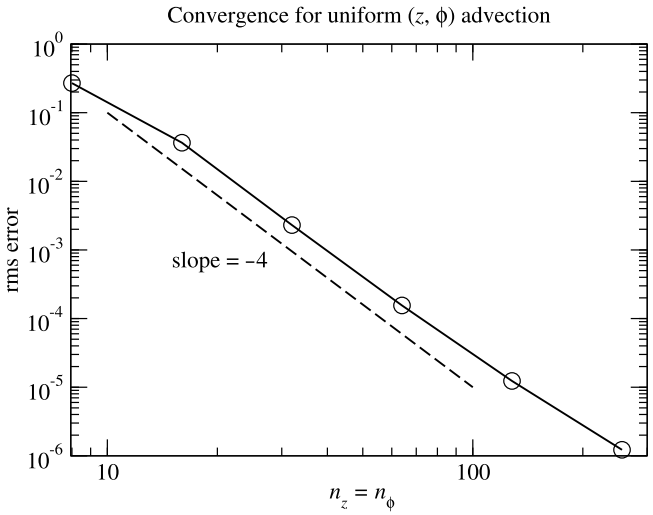

Figure 2. Convergence for 2D advection.

4.2. One-dimensional Euler Equations with Shocks

Figure 3 presents two test cases for the 1D Euler equations, which were run in the code's z direction by suppressing the other two. Both tests employed $n_z = 512$ grid points and artificial pressure (with $C_{\rm ap} = 2$) to capture shocks. Figure 3(a) shows the solution (solid line) to the 1D shock-tube problem with initial conditions to the left and right of the diaphragm (located at $x = 0.8$) as follows

$$\begin{aligned}(\rho_{\rm L}, u_{\rm L}, p_{\rm L}) &= (8, 0, 10/\gamma),\\ (\rho_{\rm R}, u_{\rm R}, p_{\rm R}) &= (1, 0, 1/\gamma),\end{aligned} \tag{48}$$

with 0 velocity everywhere and $\gamma = 1.4$. The computed solution is accurate; however, small oscillations are present in the postshock region

Shu & Osher (1989) introduced the problem of a Mach 3 shock propagating through density waves as way of testing a method's ability to both capture shocks and resolve nonshock wavy features without excessive dissipation. The initial condition is

$$(\rho, u, p) = \begin{cases}(3.857143, 2.629369, 10.33333), & x < 4;\\ (1 + 0.2\sin 5z, 0, 1), & x \geqslant 4.\end{cases} \tag{49}$$

The result of this test is shown in Figure 3 where the baseline comparison (dashed line) was obtained using a fifth-order WENO scheme with $n_z = 1600$ points, the Roe flux, and reconstruction along characteristics. The present code performs very well.

4.3. Kelvin–Helmholtz Instability

Here we consider the benchmark for viscous Kelvin–Helmholtz (KH) instability with a density gradient starting with smooth initial conditions constructed by Lecoanet et al. (2016, hereafter L2016). The full domain size is $L_x \times L_z = 1 \times 2$, however, only the lower half of the z domain will be shown, since the rest is shift symmetric. The resolution is $n_x \times n_z$ with $n_z = 2n_x$. It should be noted that L2016 write the heat conductivity as $k = \rho\chi$, where χ is the heat diffusivity. The actual definition (which PADÉ uses) is $k = \rho\chi c_p$. Therefore, to match L2016 we needed to divide our k by c_p, which equals

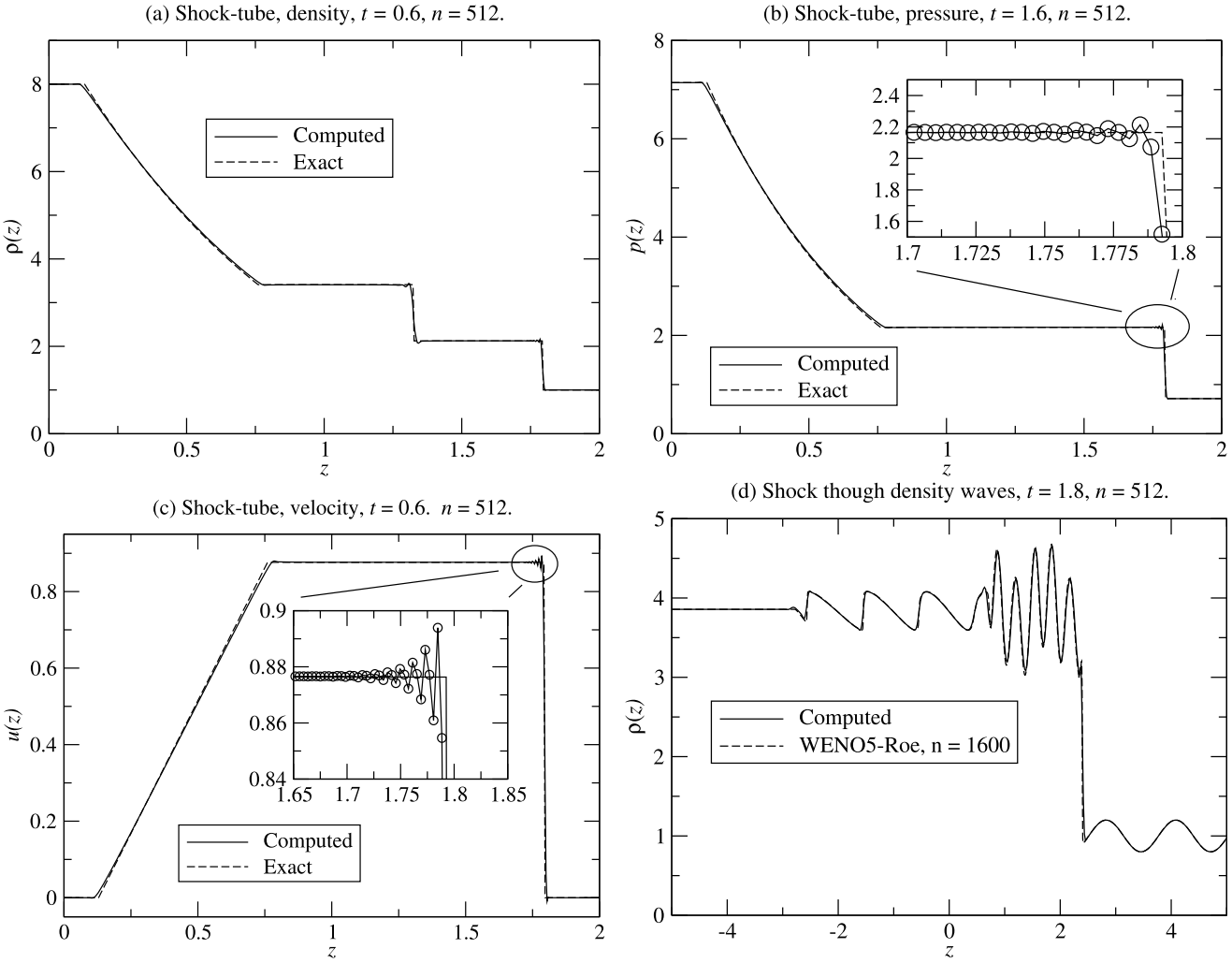

Figure 3. 1D Euler equation tests. (a)–(c) Shock tube. (a) Density. (b) Pressure. (c) Velocity. (d) A Mach 3 shock propagating into density waves (Shu & Osher 1989). The result of the present code is compared with the WENO5–Roe method. For both tests the coefficient of artificial pressure $C_{\rm ap} = 2$ and Pade/compact filter strength $\epsilon_{\rm filter} = 0.05$ (applied at the end of every step). CFL = 1.

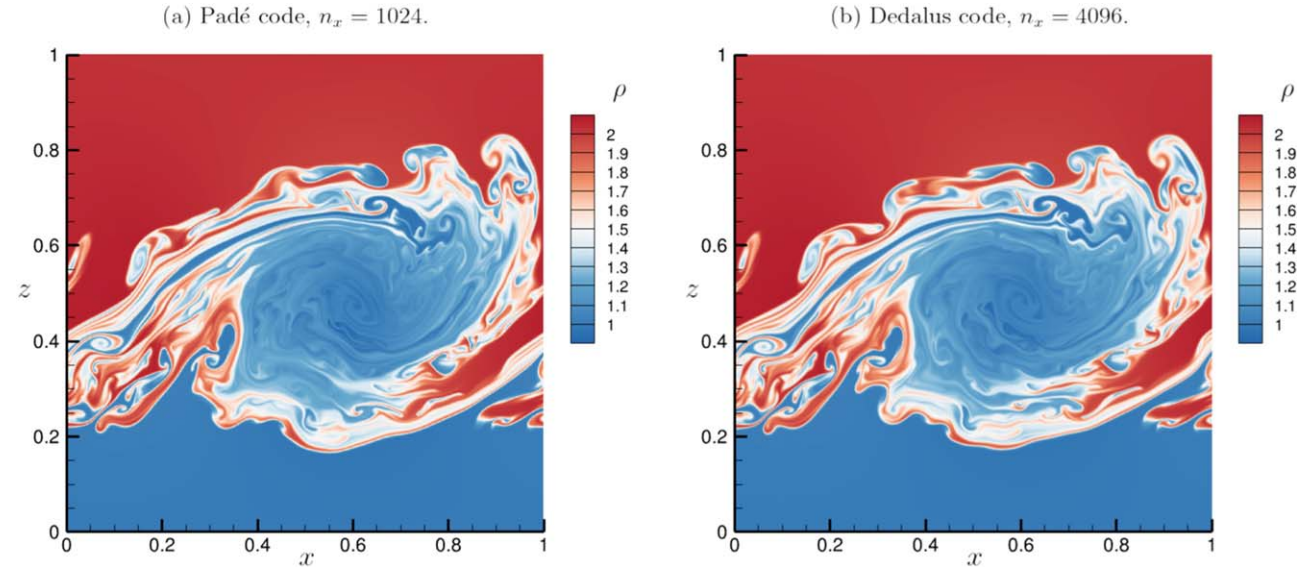

Figure 4. Density field at $t = 6$ for the KH instability benchmark of L2016. (a) PADÉ code with $n_x = 1024$. (b) Pseudospectral DEDALUS code with $n_x = 4096$. Data provided by D. Lecoanet.

Table 1
Computed Torques G_i and G_o per Unit Axial Length Exerted on the Fluid by the Inner and Outer Cylinders, Respectively, for a Steady Axisymmetric Taylor–Couette Flow

$\eta \equiv r_i/r_o$	L_z/d	Re_i	G_i	G_o	Published G_i	References
0.875	2.5	139.22	3.3485	−3.3482	3.3539	Marcus (1984)
0.50	1.988	78.8	1485	−1485	1487	Moser et al. (1983)

Note. These values are compared with values obtained in the cited references. The values are normalized differently in the two references as described in the text.

5/2 for the setup of L1016, which assumes that the gas constant $R = c_p - c_v = 1$ and $\gamma \equiv c_p/c_v = 5/3$. We chose time step Δt so that the CFL number was 1.5. We initially applied the Padé filter after every step with $\epsilon_{\mathrm{filter}} = 0.0025$, however, the associated matrix becomes ill-conditioned for very small values of $\epsilon_{\mathrm{filter}}$ when the number of grid points is sufficiently large. To alleviate this, the strength of the Padé filter was applied every 40 time steps with $\epsilon_{\mathrm{filter}} = 0.10$. The simulation was run on a laptop with an Apple M2 Pro chip using eight CPUs. The CPU time per step was 0.3 s.

Figure 4(a) shows the density field at $t = 6$ obtained from PADÉ with $n_x = 1024$. It is compared with the result of L2016 obtained using their pseudospectral code DEDALUS with $n_x = 4096$; we thank Prof. D. Lecoanet (Northwestern University) for sending us the data. The agreement is very good.

4.4. Taylor–Couette Flow

A natural test case in cylindrical coordinates is a Taylor–Couette flow, i.e., a viscous flow driven by rotating inner and outer cylinders with radii r_i and r_o, respectively. The corresponding rotation rates are Ω_i and Ω_o, giving corresponding rotational speeds $U_i \equiv \Omega_i r_i$ and $U_o \equiv \Omega_o r_o$. Two of the four nondimensional parameters are the inner and outer Reynolds numbers $\mathrm{Re}_i \equiv U_i d/\nu$ and $\mathrm{Re}_o \equiv U_o r_i d/\nu$, where $d = r_o - r_i$ is the gap width. The third and fourth nondimensional parameters are the ratio $\eta \equiv r_i/r_o$, and the nondimensional vertical domain size $\lambda \equiv L_z/d$.

The present code solves the compressible equations while most simulations reported in the literature are for an incompressible flow. To approximate incompressible simulations the rotational Mach number of the inner cylinder is set to 0.1, the equation of state is isothermal, and isothermal boundary conditions are applied at the two walls. The initial density is set to a uniform value ρ_0. Code units are such that $U_i = d = \rho_0 = 1$.

Torques, G_i and G_o, per unit axial length exerted on the fluid by the inner and outer cylinders, respectively, are computed as diagnostics:

$$\text{torque} = \text{area-averaged shear stress} \times r \times \text{area}. \tag{50}$$

The area-averaged shear stress is given by

$$\tau_{r\phi} = \left\langle \mu r \frac{\partial}{\partial r}\left(\frac{u_\phi}{r}\right)\right\rangle, \tag{51}$$

where μ is the dynamic viscosity and $\langle . \rangle$ denotes an average over the surface. Therefore

$$G_i = -2\pi r_i^3 \left\langle \frac{\partial}{\partial r}\left(\mu \frac{u_\theta}{r}\right)\right\rangle_{r=r_i}, \tag{52}$$

$$G_o = +2\pi r_o^3 \left\langle \frac{\partial}{\partial r}\left(\mu \frac{u_\theta}{r}\right)\right\rangle_{r=r_o}. \tag{53}$$

The negative sign in Equation (52) arises from the fact that at $r = r_i$, the normal to the fluid surface is in the $-r$ direction. For a steady flow, conservation of angular momentum implies that G_i and G_o must be equal and opposite in sign.

We first consider two axisymmetric cases that produce a steady flow with counterrotating vortices. The inner cylinder rotates at angular velocity Ω_i while the outer cylinder is fixed. Each case was run using a 32×32 grid ($n_r \times n_z$) with $\epsilon_{\mathrm{filter}} = 0.005$. The first two entries of Table 1 shows that the computed torque per unit length agrees with previous published results. Marcus (1984) uses units in which $\rho_0 = d = \Omega r_i = 1$. Moser et al. (1983) use the same units but normalize G by $\rho_0 \nu^2$ where $\nu = \mu/\rho_0$. The values in Table 1 use the same conventions.

Finally, a case of a 3D unsteady counterrotating Taylor–Couette flow is considered following Dong (2008). The inner and outer cylinders rotate counterclockwise and clockwise, respectively, with $\mathrm{Re}_i = -\mathrm{Re}_o = 500$, $\eta = 0.5$ and $L_z/d = 2\pi$. Dong (2008) defines the nondimensional torque coefficient for the inner cylinder as

$$(C_T)_i = \frac{G_i}{\frac{1}{2}\pi \rho_0 U_i^2 r_i^2 L_z}, \tag{54}$$

and similarly for the outer cylinder. The number of grid points is 48×96^2 (n_r, n_z, n_ϕ) and the strength of the Padé filter was set to $\epsilon_{\mathrm{filter}} = 0.005$. Figures 5(a) and (b) show the radial velocity in a meridional and horizontal plane and reveals the three dimensionality of the flow. Figure 5(c) shows the torque coefficients for the inner and outer cylinders after the flow has reached statistical stationarity. The values agree with those shown in Figure 3 of Dong (2008).

4.5. Vertical Shear Instability and the Effect of Varying the Strength of the Padé filter

Here we present results for axisymmetric VSI at an early stage of evolution following the setup of Nelson et al. (2013). Detailed results will be presented in a forthcoming publication, which will also include 3D results.

The parameters of the setup are given in Table 2. A locally isothermal equation of state

$$p = \rho c_i^2(r), \tag{55}$$

is used, which represents the case of infinitely rapid relaxation of temperature to the basic state. The square of the sound speed, which is proportional to the temperature, is specified as a power law

$$c_i^2(r) = c_0^2 (r/r_0)^q, \tag{56}$$

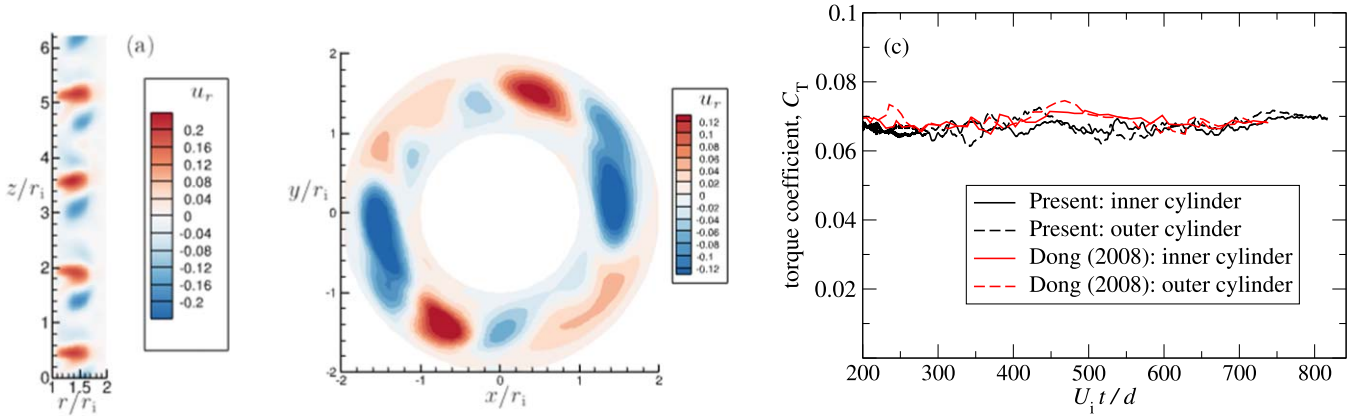

Figure 5. Counterrotating Taylor–Couette flow at $\mathrm{Re_i} = -\mathrm{Re_o} = 500$ following Dong (2008). The mesh is $48 \times 96^2 (n_r \times n_z \times n_\phi)$. (a) and (b) radial velocity contours at $U_i t/d = 798.8$. (a) The meridional plane $\phi = 0$. (b) Horizontal plane at midheight ($z = \pi$) to show that the flow is three dimensional. (c) Torque coefficients (black) compared with curves (red) digitized from Dong (2008, Figure 3) and shifted backward in time by $U_i \Delta t/d = 802$. The negative of the torque coefficient is plotted for the outer cylinder.

Table 2
Parameters for the Axisymmetric Vertical Shear Instability Run

Parameter	Value
Orbital period, T_0, at r_0	1
Density, ρ_0, at r_0	1
Scale height, H_0, at r_0	1
Density exponent, p	$-3/2$
Temperature exponent, q	-1
Disk aspect ratio, H_0/r_0	0.10
Radial domain (including sponge), $[r_{\min}/H_0, r_{\max}/H_0]$	[6.5, 13.5]
Vertical domain (including sponge), $[z_{\min}/H_0, z_{\max}/H_0]$	$[-3.5, 3.5]$
Width of the sponge at the domain border, $\delta_{\mathrm{sponge}}/H_0$	0.5
Decay period, t_{sponge}, for the sponge,	20 time steps
Number of grid points, $n_r \times n_z$	512^2
Strength of the Padé filter, $\epsilon_{\mathrm{filter}}$	0.01–0.125

Note. H_0 is the disk scale height at midradius and was set to unity. Subscript "0" refers to a quantity evaluated at midradius.

where the temperature exponent $q = -1$, r_0 is the midradius of the computational domain, and c_0 is chosen to make the scale height $H_0 \equiv H(r_0) = 1$.

Zero normal velocity boundary conditions are applied at all four domain edges. Each edge abuts a sponge region of width $\delta_{\mathrm{sponge}} = 0.5 H_0$ where the flow relaxes back to the basic state with a characteristic period t_{sponge} given in the table.

Figure 6 shows the azimuthal vorticity, ω_ϕ at $t/T_0 = 37$, which is just after saturation of the linear phase of the instability. We caution the reader that the statistically stationary state is quite different and reached much later at $t/T_0 \approx 350$; this will be reported in a later publication. The result of using three different values for the strength, $\epsilon_{\mathrm{filter}}$, of the Padé filter is shown.

The azimuthal vorticity consists of pairs of shear layers of opposite sign which induce up and down jets of vertical velocity (Figure 7(a)) that are characteristic of VSI (Nelson et al. 2013). The shear layers roll up into discrete eddies via the KH instability. It is important to note that VSI also produces shear layers with $\partial \delta u_\phi / \partial r$ shear, i.e., radial gradients of perturbation angular velocity. These are shown in Figure 7(b), which plots the azimuthal velocity perturbation, δu_ϕ, and shows that is it about two-thirds of the value of the vertical velocity perturbation. This shear will also be subject to the KH instability, however, it will be modified by the presence of mean Keplerian shear.

We now discuss how a user should choose the filter strength, $\epsilon_{\mathrm{filter}}$. Direct numerical simulations (DNSs) of turbulent flow are performed with molecular viscosity and all the scales of the turbulence down to the dissipation scale are well resolved. In this case, the very minimum value of $\epsilon_{\mathrm{filter}}$ should be chosen. For instance in the Taylor–Couette simulations we chose $\epsilon_{\mathrm{filter}} = 0.005$.

However, the Reynolds number in protoplanetary disks is too large for numerical simulations to be able to resolve all the turbulent scales. Therefore, some treatment of the unresolved scales is required. For engineering and geophysical flows, the most common practice is to use an explicit model for the subgrid stresses, the simplest being the Smagorinsky model. Another approach, followed for all protoplanetary disk simulations to date, is to simply let the dissipation inherent in the numerical method damp scales near the grid cutoff. This procedure is referred to as implicit large-eddy simulation (ILES) and was first articulated by Boris et al. (1992). Comparison with DNSs, in which all scales are resolved, have since shown that it is accurate (Ritos et al. 2018). In our approach, we use the dissipation provided by the Padé filter as an implicit subgrid treatment. To leading order, the Padé filter corresponds to a fourth-order hyperviscosity; see Equation (30) in Shariff & Wray (2018) and the discussion following it.

When the Padé filter is used as an ILES treatment, $\epsilon_{\mathrm{filter}}$ should be chosen to balance the desire to capture as wide a range of small scales as possible (with a fixed grid size) by lowering $\epsilon_{\mathrm{filter}}$, while at the same not allowing to much energy to to pileup in 2Δ waves. For illustration, Figure 6 shows the result of varying the strength, $\epsilon_{\mathrm{filter}}$, of the filter on the vorticity field. The reader may refer back to Figure 1(b), which shows the filter transfer function for different $\epsilon_{\mathrm{filter}}$ choices. The left-hand column of plots in Figure 6 shows that with reduced $\epsilon_{\mathrm{filter}}$ more finer scales of the flow are resolved. The right-hand column of the plots zoom in to a square region with sides equal to H. For the lowest filter strength ($\epsilon_{\mathrm{filter}} = 0.01$), 2Δ (sawtooth) oscillations can be observed in thin vortex layers oriented at 45° to the mesh. For $\epsilon_{\mathrm{filter}} = 0.04$ and 0.125, smaller amplitude oscillations are present in one vorticity layer whose

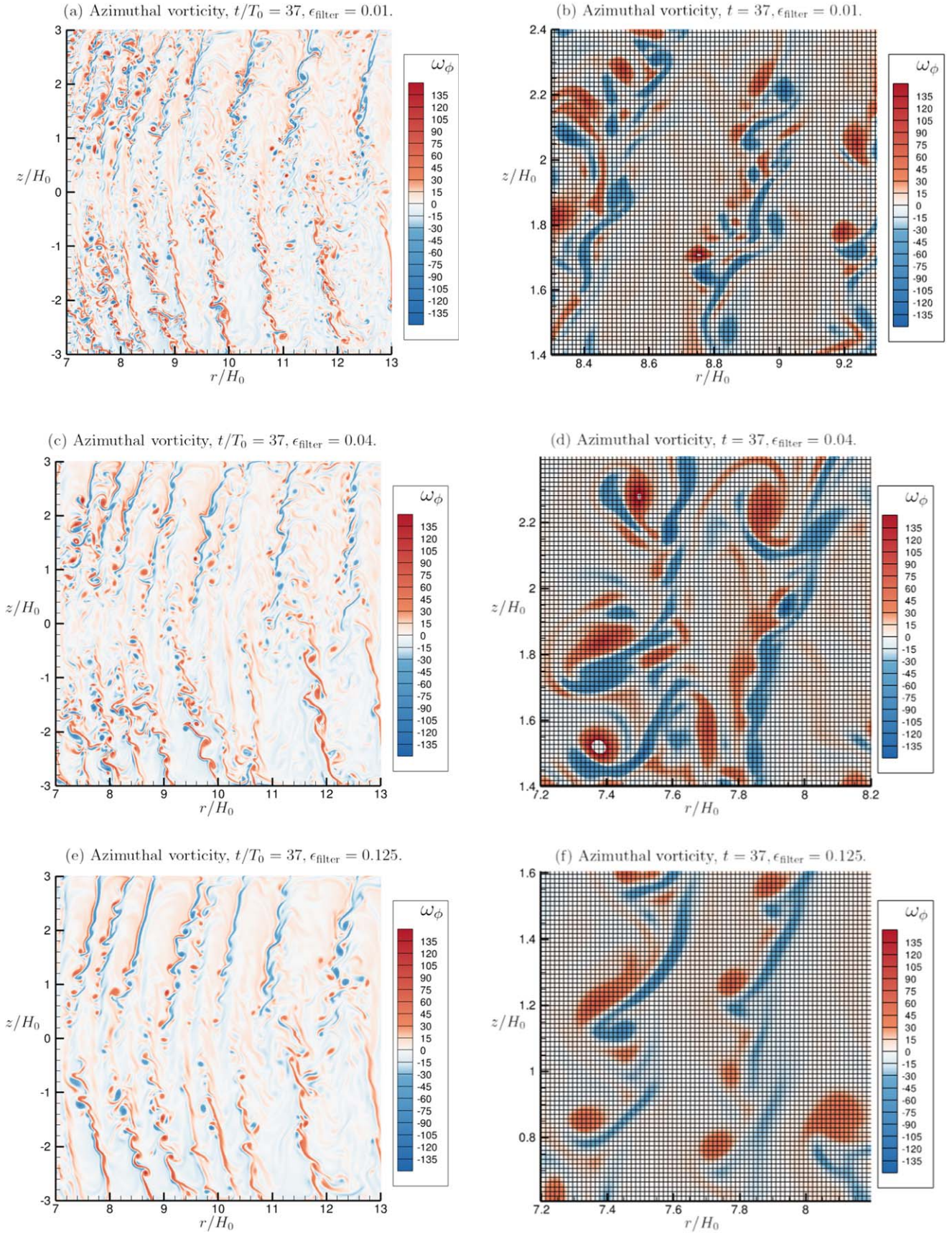

Figure 6. Effect of varying the strength of the Padé filter, $\epsilon_{\rm filter}$ when the Padé filter is used as an implicit subgrid scale treatment for an axisymmetric VSI simulation. Completely white pixels are where ω_ϕ exceeds the range of the legend. The mesh size is 512^2. The plots in the right-hand column are intended to show the level of numerical 2Δ oscillations, which appear as sawtooth features.

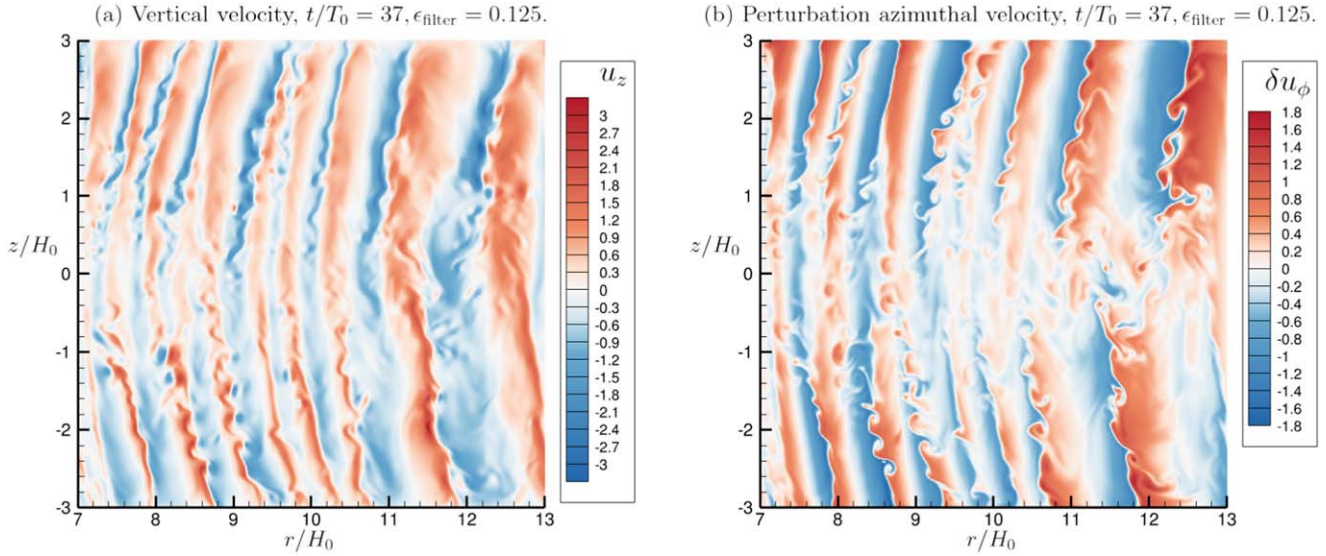

Figure 7. (a) Vertical velocity for the axisymmetric VSI simulation using a 512^2 mesh. The units of velocity are scale heights (H_0) per orbit time (T_0). (b) Similarly, the perturbation azimuthal velocity.

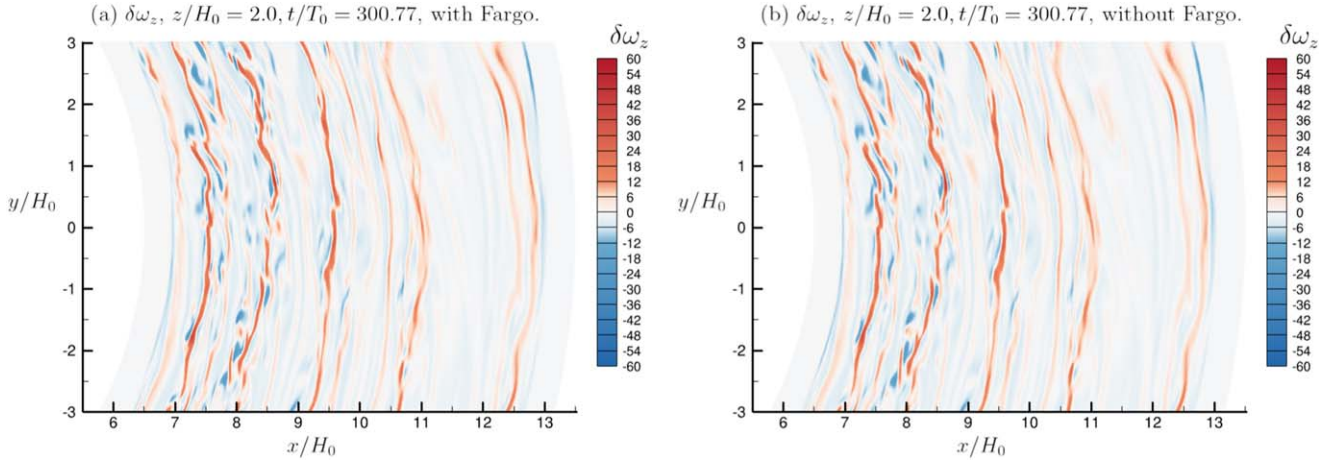

Figure 8. A portion of the horizontal plane (r, ϕ) cutting through the disk at $z/H_0 = 2$ comparing the perturbation vertical vorticity, $\delta\omega_z$, for a 3D VSI run with and without Fargo. Each run was started at $t/T_0 = 300.27$ and run for $0.5T_0$ up to $t/T_0 = 300.77$. (a) With Fargo. (b) Without Fargo.

width is about one grid diagonal, which is very thin indeed. These oscillations are not visually detectable in plots of the velocity field even for $\epsilon_{\rm filter} = 0.01$. We conclude that 0.125 would be a conservative choice for $\epsilon_{\rm filter}$.

Finally, we would like to discuss some physical effects that manifest as diffusivity when the filter is reduced and the effective resolution is increased. (a) The rolled-up vortices are smaller and there are more of them. This is explained as follows. The shear layer thickness, δ, is reduced with smaller diffusivity. The most amplified KH wavelength $\lambda \approx 2\pi\delta$ is therefore also reduced, and with it the size of the vortices. The number of vortices increases because there are more waves per unit length. (b) The shear layer vorticity increases. We have that $\omega_\phi \sim \Delta U/\delta$ where ΔU, the jump in velocity across each shear layer, is independent of diffusivity. Therefore a reduction in δ with diffusivity leads to increased ω_ϕ. (c) The rolled-up vortices appear earlier. The KH growth-rate $\propto \Delta U k_{\rm max}$, where $k_{\rm max}$ is the most amplified wavenumber. Since $k_{\rm max}$ increases with reduced thickness, the KH vortices develop earlier with reduced diffusivity. (d) The vorticity in the the rolled-up vortex cores increases. For each vortex we have $\omega_\phi \sim \Gamma/\text{area}$, where $\Gamma \sim \Delta U \lambda$ is its circulation. Since the vortex area $\sim \lambda^2$, we get that $\omega_\phi \sim \Delta U/\lambda$, which increases since λ decreases.

4.6. Three-dimensional Vertical Shear Instability: Comparison of Fargo and Non-Fargo Runs

In Shariff & Wray (2018) a comparison (with plots of error) was made between runs with and without the Fargo treatment for the case of two corotating vortices in a razor-thin disk. Here, we perform a similar comparison for 3D VSI. The simulation was first run until $t/T_0 = 300.27$ with Fargo activated. Next runs were made with and without Fargo for one orbital period (T_0) at midradius. The run parameters are the same as for the axisymmetric run presented in Table 2. The only differences are a resolution of $512 \times 512 \times 1024$ ($n_r \times n_z \times n_\phi$) with an azimuthal domain of $\phi \in [0, 2\pi)$, and a Padé filter strength of $\epsilon_{\rm filter} = 0.125$. For the non-Fargo run, the Padé filter was applied every other time step due to the fact that this run required about twice as many time steps as the run with Fargo.

Figure 8 compares the vertical vorticity ($\delta\omega_z$) perturbation (relative to the basic state) in the $z/H_0 = 2$ plane after a time of $0.5T_0$, i.e., at $t/T_0 = 300.77$. Here T_0 is the orbital period at the

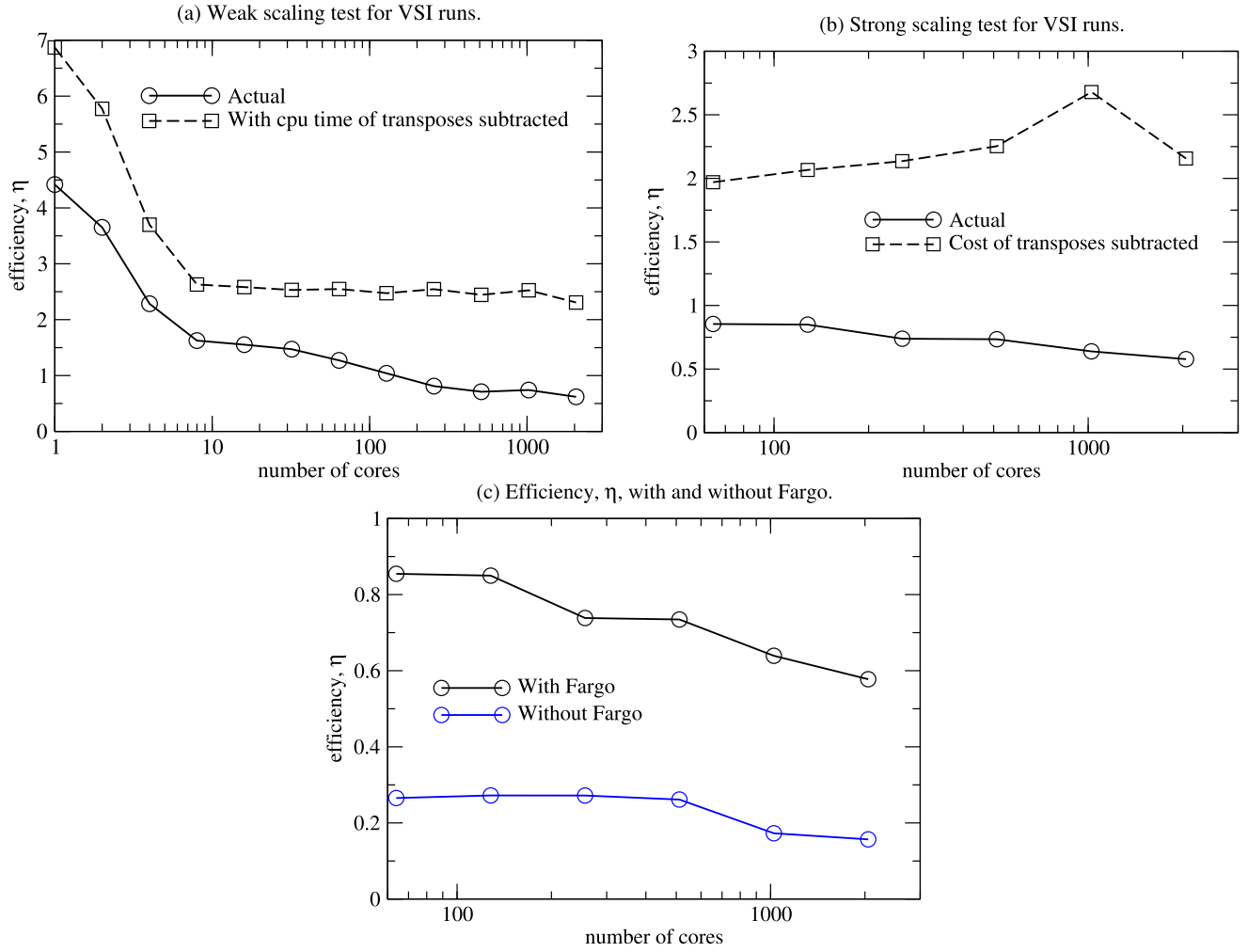

Figure 9. Scaling tests for the 3D VSI setup performed using the Intel Haswell nodes. The environment variable `MPI_IB_RAILS` used by the IB network was set to 2 for all runs. (a) Efficiency, η, for the weak scaling test where the problem size per processor is fixed. (b) Efficiency for the strong scaling test where the total problem size is fixed at $n_r \times n_z \times n_\phi = 512 \times 512 \times 1024$. (c) Efficiency, η, with Fargo deactivated. The measure, η, accounts for the decrease in time step when Fargo is deactivated.

midradius of the computational domain. The difference in the two solutions at this time is very small. Due to the chaotic nature of the flow, the error due to the different time steps, Δt, chosen for the two simulations grows with time and the differences become more apparent. The CFL number was chosen to be 1.5 and based on this, the time step selected by the code for the non-Fargo run was $\Delta t/T_0 = 2.25 \times 10^{-4}$. This choice was constrained by Keplerian advection. For the Fargo run, the code selected $\Delta t/T_0 = 5.49 \times 10^{-4}$ for the same CFL number, which represents a better than factor of 2 improvement. In the run with Fargo, the choice of time step was constrained by the characteristic wave speed and grid size in the radial direction. The CPU time for the non-Fargo and Fargo runs was 1.50 and 1.76 s per step, respectively. These represent a 17.3% overhead for a more than factor of 2 gain in time step. An Intel Broadwell processor was used for these runs.

We close with a brief description of the physics observed in the 3D VSI runs. The Keplerian mean flow is counterclockwise in Figure 8 and the vertical vorticity perturbation consists of layers of cyclonic $\delta\omega_z$ (red bands), which induce across them a positive jump in specific angular momentum $j_\phi = u_\phi r$ as r increases. These layers are formed by the vertical transport of basic state angular momentum by the vertical jets that are the main feature of VSI. This will be discussed in more detail in a forthcoming paper. In between the cyclonic layers, one observes weaker and more diffuse anticyclonic $\delta\omega_z$, which reduces j_ϕ compared to the basic state. In the inner portion of the disk, anticyclonic $\delta\omega_z$ structures take the form of smaller aspect ratio structures.

4.7. Parallel Scaling Tests for Three-dimensional Vertical Shear Instability Runs

Following the suggestion of the referee, the efficiency η of the code is defined as the ratio of the useful work to the resources consumed. The useful work is the number $N_{\rm grid}$ of grid points evolved while the resources consumed is the CPU time $t_{\rm cpu}$ (per time step in seconds) times the number of cores, $N_{\rm cores}$. We also include the ratio $\Delta t_{\rm actual}/\Delta t_{\rm Fargo}$ to account for the reduced time step in non-Fargo runs

$$\eta \equiv \frac{N_{\rm grid}/2^{17}}{t_{\rm cpu} N_{\rm cores}} \frac{\Delta t_{\rm actual}}{\Delta t_{\rm Fargo}}, \tag{57}$$

where we have normalized the number of grid points to 2^{17}. When there is perfect scaling, η should be a constant as $N_{\rm cores}$ is increased.

All the tests in this subsection were performed using Intel Haswell nodes, which use an E5-2680V3 (Xeon) processor. The variable `MPI_IB_RAILS` used by the InfiniBand (IB) network was set to 2; this causes two IB fabrics to be used for communication and results in reduced CPU time. Figures 9(a) and (b) plot η for a 3D VSI setup with Fargo activated; therefore the ratio $\Delta t_{\rm actual}/\Delta t_{\rm Fargo}$ in Equation (57) equals unity.

Figure 9(a) is for a weak scaling test in which the grid size per core is kept fixed while $N_{\rm cores}$ is increased. In other words, both $N_{\rm cores}$ and the total grid size increase simultaneously. In the present test, the number of grid points for the one core run is $64\times 32\times 64$ ($n_r\times n_z\times n_\phi$) and each direction is successively doubled in resolution as $N_{\rm cores}$ doubles. Figure 9(a) plots two curves, one which used the actual CPU time (solid) and another (dashed) for which the time taken to perform transposes was subtracted out. Both curves show an initial rapid decrease in η up to $N_{\rm cores}=8$. The main cause of this is likely the fact that each Xeon processor in a Haswell node has 12 cores which share a single memory and level 3 cache. Contention for both resources increases as the number of cores increases from one to 12. Therefore, we focus on the region $N_{\rm cores}\geqslant 12$. At $N_{\rm cores}=2048$, the efficiency has decreased to 38% of its value at $N_{\rm cores}=16$. When the CPU time for transposes is subtracted out, the efficiency remains relatively flat. This indicates that the loss in efficiency is due to communication intensive transposes. Indeed, the fraction of time (not shown) taken for transposes increases from 0.38 to 0.73 as $N_{\rm cores}$ increases from 16 to 2048.

Figure 9(b) is for a strong scaling test in which the total problem size is fixed at $512\times 512\times 1024$ ($n_r\times n_z\times n_\phi$) and the number of cores is varied. At $N_{\rm cores}=2048$, the efficiency has decreased to 67% of its value at $N_{\rm cores}=64$. The CPU fraction taken for transposes increases from 0.57 to 0.73 (not shown) in this range. When the CPU time for transposes is subtracted out, the efficiency increases slowly. This is explained as follows. In the strong test, the total number of memory fetches is constant with increasing $N_{\rm cores}$, however, the number of cache hits (when needed data are found to be already in the cache) is likely to be statistically higher since the number of cache slots per fetch is higher with more cores.

Figure 9(c) compares the efficiency of the Fargo versus the non-Fargo scheme. It shows that there is at least a factor of 3.7 advantage to using Fargo, i.e., the overhead of the Fargo method is more than compensated by an increase in time step.

In conclusion, the all-to-all communication of transposes leads to a significant loss in efficiency as the number of cores is increased. To reduce this communication overhead, an effort is underway to use a parallel Padé algorithm (Kim et al. 2021).

5. Closing Remarks

A code has been developed that uses a fourth-order Padé scheme to simulate hydrodynamic turbulence in protoplanetary disks. Padé schemes are nondissipative and have high resolving power. Thus, with the same grid resolution, they are better able to capture fine scale vortical features compared to the dissipative shock-capturing schemes employed in most astrophysics codes. They also have better resolving power then central finite-difference schemes of the same order.

Suggested improvements are as follows. (i) To eliminate communication intensive transposes, consider using parallel tridiagonal matrix algorithms (Kim et al. 2021, and the references therein), which require much less communication. Kim et al. (2021) demonstrate good scaling as the number of cores is increased. (ii). For simulations that require long radial domains (more than ≈6 scale heights) it would be better to use spherical rather than cylindrical coordinates. An option for this could be provided. (iii) The sixth-order tridiagonal Padé scheme

$$\alpha f'_{j-1}+f'_j+\alpha f'_{j+1}=\frac{a}{2h}(f_{j+1}-f_{j-1})+\frac{b}{4h}(f_{j+2}-f_{j-2}), \tag{58}$$

with $\alpha=1/3$, $a=14/9$, and $b=1/9$ could be implemented. (iv) The capability to track Lagrangian solid particles could be provided. (v) An intercomparison effort with other codes could be performed.

The code is available at https://github.com/NASA-Planetary-Science/Pade-disk-code with a copy deposited to Zenodo (Shariff 2024).

Acknowledgments

I am grateful to Alan Wray (NASA Ames) for providing several routines that have been used in PADÉ, including directional transposes, tridiagonal solvers, and the Padé filter. He also helped to explain features observed in the scaling tests. I am grateful to Jeff Cuzzi (NASA Ames) for performing the internal review and for his encouragement; Debanjan Sengupta (New Mexico State University, Las Cruces) for performing the internal review and suggesting the weak scaling test; Orkan Umurhan (SETI Institute) for his encouragement and advice on the VSI runs; Prof. Ali Mani (Stanford University) for advice on the implementation of artificial bulk viscosity; and Prof. Daniel Lecoanet (Northwestern University) for providing the result of the KH test case run with DEDALUS. Finally, I thank the referee for many useful suggestions that improved the paper. This work was partly supported by NASA's Internal Scientist Funding Model (ISFM) through the "Origins" group in NASA Ames' Space Science Division.

Appendix A
Molecular Viscous Force, Viscous Heating, and Heat Conduction

The code provides the option to add terms that represent molecular viscosity and heat conduction. Similar terms arise in models of subgrid turbulence. For these we have coded the models due to Smagorinsky (1963) and Vreman (2004). However, since we have not tested them, this section describes the implementation for the molecular/laminar case only. If the Fargo option has been activated, the Fargo chain rule is applied wherever needed.

A.1. Viscous Force

The viscous stress tensor is

$$T_{ij}=2\mu S_{ij}+\left(\mu_{\rm b}-\frac{2}{3}\mu\right)S_{kk}\,\delta_{ij}, \tag{A1}$$

where μ and μ_b are the shear and bulk viscosities, respectively, and

$$S_{ij} = \frac{1}{2}(\partial_j u_i + \partial_i u_j), \tag{A2}$$

is the strain rate tensor.

Equation (A1) is implemented in `subroutine laminar_stress_and_heat_flux`. The components of the strain tensor (which is symmetric) are computed in `subroutine strain_tensor` as follows

$$S_{zz} = \partial_z u_z,\ S_{\phi z} = \frac{1}{2}\left(\partial_z u_\phi + \frac{1}{r}\partial_\phi u_z\right),$$
$$S_{zr} = \frac{1}{2}(\partial_r u_z + \partial_z u_r), \tag{A3}$$

$$S_{rr} = \partial_r u_r,\ S_{r\phi} = \frac{1}{2}\left(\frac{1}{r}\partial_\phi u_r + \partial_r u_\phi - \frac{u_\phi}{r}\right),$$
$$S_{\phi\phi} = \frac{1}{r}\partial_\phi u_\phi + \frac{u_r}{r}. \tag{A4}$$

The above expressions are from Aris (Aris 1989, page 181) and agree with Batchelor (1967).

The viscous force is the divergence of the viscous stress tensor and given by Aris (1989, page 179) for orthogonal coordinates as follows (in his notation)

$$F_i^v = T(ij, j) = \frac{h_i}{h_1 h_2 h_3}\frac{\partial}{\partial x_j}\left[\frac{h_1 h_2 h_3}{h_i h_j}T(ij)\right]$$
$$+\frac{h_i}{h_j h_k}\left\{\begin{matrix} i \\ jk \end{matrix}\right\}T(jk),\ \text{(no sum on } i\text{)}. \tag{A5}$$

Here $(x_1, x_2, x_3) = (r, \phi, z)$ and the corresponding scale factors are $h_1 = h_r = 1$, $h_2 = h_\phi = r$, and $h_3 = h_z = 1$. The quantities in braces are Christoffel symbols and the only nonzero ones are

$$\left\{\begin{matrix} 2 \\ 12 \end{matrix}\right\} = \left\{\begin{matrix} 2 \\ 21 \end{matrix}\right\} = r \text{ and } \left\{\begin{matrix} 1 \\ 22 \end{matrix}\right\} = -r. \tag{A6}$$

The symbolic algebra package maxima was used to verify the correctness of Aris' expression (Equation (A5)) by writing Cartesian velocities in terms of cylindrical quantities (velocities and coordinates) and computing Cartesian viscous forces in terms of cylindrical quantities using the chain rule throughout. These can be be rotated to obtain the forces in cylindrical coordinates in terms of cylindrical quantities and compared with Equation (A5). The relevant maxima script can be found in `check_Aris.mac` in the `Symbolic_algebra` subdirectory.

The final expressions for the viscous force area are

$$F_z^v = \frac{1}{r}[\partial_r(rT_{zr}) + \partial_\phi(T_{z\phi}) + \partial_z(rT_{zz})], \tag{A7}$$

$$F_r^v = \frac{1}{r}[\partial_r(rT_{rr}) + \partial_\phi(T_{r\phi}) + \partial_z(rT_{rz})] - \frac{1}{r}T_{\phi\phi}, \tag{A8}$$

$$F_\phi^v = \partial_r(T_{\phi r}) + \frac{1}{r}\partial_\phi(T_{\phi\phi}) + \partial_z(T_{\phi z}) + \frac{2}{r}T_{\phi r}. \tag{A9}$$

These agree with the expressions given in Bird et al. (1960).

A.2. Viscous Heating

The equation for total energy $e = e_{\text{int}} + \rho u_i u_i$ (per unit volume) has a term for the work done by shear stresses (Liepmann & Roshko 1957)

$$W^{\text{shear}} = \frac{\partial}{\partial x_j}(T_{ij}u_i). \tag{A10}$$

For the internal energy equation which we solve, we must subtract the kinetic energy dissipation

$$D = u_i\frac{\partial}{\partial x_j}T_{ij}. \tag{A11}$$

This gives the viscous heating term for the internal energy

$$Q^v = W^{\text{shear}} - D = T_{ij}\frac{\partial u_i}{\partial x_j}. \tag{A12}$$

Now T_{ij} is symmetric so only the symmetric part of $\partial_j u_i$ survives. Then substituting the constitutive Equation (A1) for T_{ij} into Equation (A12) one obtains

$$Q^v = 2\mu S_{ij}S_{ij} + \left(\mu_b - \frac{2}{3}\mu\right)S_{kk}^2. \tag{A13}$$

A.3. Heat Conduction

The flux of internal energy due to molecular conductivity is given by Fourier's law

$$\vec{q}_{\text{cond}} = -\frac{k}{c_v}\nabla(c_v T), \tag{A14}$$

where we have multiplied and divided by c_v inside and outside the gradient, respectively. This assumes that c_v is constant, i.e., that we have a calorically perfect gas. Now $c_v T$ is simply e_{int}/ρ, and using the definition of the Prandtl number $\text{Pr} \equiv \mu c_p/k$ we get

$$\frac{k}{c_v} = \frac{\mu\gamma}{\text{Pr}}. \tag{A15}$$

Appendix B
Discrete Conservation

B.1. Theory

Here we describe how to choose boundary schemes to ensure that the overall scheme possesses a discrete conservation property. We follow Lele (1992) and Brady & Livescu (2019) and offer two clarifications: (1) There is a distinction between provisional and final weights; the weights given in Section 4.2 of Lele (1992) are provisional. (2) The weights cannot be specified a priori and must be determined as part of the solution. Specifically, one needs to verify that the final weights provide a reasonable discrete conservation law. In general, the final weights will not correspond exactly to a quadrature rule.

For a system of partial differential equations in more than one dimension, we compute derivatives of fluxes along each direction separately. Hence, it is sufficient to consider the 1D partial differential equation

$$\frac{\partial u}{\partial t} + \frac{\partial f}{\partial x}, \quad x \in [0, L]. \tag{B1}$$

Upon integration over the domain, Equation (B1) gives the conservation law

$$\frac{d}{dt}\int_0^L u(x,\,t)dx = f(0) - f(L). \tag{B2}$$

Padé differencing applied to Equation (B1) should possess a discrete analog of Equation (B2); in its absence, a long time solution can drift and fail to achieve statistical stationarity. Padé difference schemes have the form

$$A\boldsymbol{f}' = B\boldsymbol{f}, \tag{B3}$$

where A and B are banded matrices and henceforth, lowercase bold letters will be used to denote column vectors.

We now state a result that was stated by Lele (1992) without a proof, which was later provided by Brady & Livescu (2019).

Proposition 1. *To obtain a discrete analog of Equation (B2), columns 2 through $N-1$ of matrix* B *must have a weighted sum of zero, i.e.,*

$$\boldsymbol{w}^T\boldsymbol{b}_i = 0 \quad \text{for } i \in [2,\, N-1], \tag{B4}$$

where **w** *is a column vector of weights and* $\mathbf{b}_i$ *is the* i*th column of matrix* B. *The weights* **w** *are provisional; final weights will be given below.*

Remark. Not all the weights, $\boldsymbol{w}$, can be specified a priori but must be obtained as part of the process of satisfying Equation (B4).

Proof. We assume that grid points x_i, $i = 1,\ldots,N$ have been laid out according to a smooth analytic mapping $x(\xi)$ such that $\xi_i = i$, i.e., ξ is a smooth grid index variable. In implementations, it is convenient to take derivatives with respect to ξ and then use the chain rule

$$\left(\frac{\partial f}{\partial x}\right)_i = \left(\frac{\partial f}{\partial \xi}\right)_i \left(\frac{d\xi}{dx}\right)_i. \tag{B5}$$

For simplicity we use the notation

$$f_i'(\xi) \equiv \left(\frac{\partial f}{\partial \xi}\right)_i, \quad h_i^{-1} \equiv \left(\frac{d\xi}{dx}\right)_i. \tag{B6}$$

Then the spatially discretized version of Equation (B2) is

$$\frac{du_i}{dt} + \frac{1}{h_i}\sum_{j,k}[A^{-1}]_{ij}B_{jk}f_k = 0. \tag{B7}$$

Defining the vector $\boldsymbol{U} = [h_1u_1,\, h_2u_2,\ldots,h_Nu_N]^T$, we can write Equation (B7) as

$$A\frac{d\boldsymbol{U}}{dt} + B\boldsymbol{f} = 0. \tag{B8}$$

A weighted sum is applied to Equation (B8) to mimic the integration in Equation (B2) as

$$\frac{d}{dt}\boldsymbol{w}^TA\boldsymbol{U} = -\boldsymbol{w}^TB\boldsymbol{f}. \tag{B9}$$

Following Brady & Livescu (2019), let $B = [\boldsymbol{b}_1,\, \boldsymbol{b}_2,\ldots,\boldsymbol{b}_N]$ where $\boldsymbol{b}_i$ denotes the ith column vector of the matrix B. Now

$$\boldsymbol{w}^TB = \boldsymbol{w}^T[\boldsymbol{b}_1,\, \boldsymbol{b}_2,\ldots,\boldsymbol{b}_N], \tag{B10}$$

$$= [\boldsymbol{w}^T\boldsymbol{b}_1,\, \boldsymbol{w}^T\boldsymbol{b}_2,\ldots,\boldsymbol{w}^T\boldsymbol{b}_N], \tag{B11}$$

so that

$$\boldsymbol{w}^TB\boldsymbol{f} = \sum_{i=1}^{N}\boldsymbol{w}^T\boldsymbol{b}_if_i. \tag{B12}$$

Hence Equation (B9) can be written as

$$\frac{d}{dt}\boldsymbol{w}^TA\boldsymbol{U} = -\boldsymbol{w}^T\boldsymbol{b}_1f_1 - \boldsymbol{w}^T\boldsymbol{b}_Nf_N - \sum_{i=2}^{N-1}\boldsymbol{w}^T\boldsymbol{b}_if_i. \tag{B13}$$

In order to arrive at a discrete conservation law analogous to Equation (B2), let us try imposing

$$\sum_{i=2}^{N-1}\boldsymbol{w}^T\boldsymbol{b}_if_i = 0. \tag{B14}$$

For Equation (B14) to be true for arbitrary f_i we must have that

$$\boldsymbol{w}^T\boldsymbol{b}_i = 0 \qquad \text{for } i \in [2,\, N-1]. \tag{B15}$$

In other words we want columns 2 through $N-1$ of the matrix B to have a weighted sum of 0.

We now show, finally, that the trial condition in Equation (B15) does indeed lead to a discrete conservation law. Equation (B15) gives the set of conditions one uses to solve for the provisional weights. Then Equation (B13) becomes

$$\frac{d}{dt}\widetilde{\boldsymbol{w}}^T\boldsymbol{U} = -\boldsymbol{w}^T\boldsymbol{b}_1f_1 - \boldsymbol{w}^T\boldsymbol{b}_Nf_N, \tag{B16}$$

where we have defined a new set of weights

$$\widetilde{\boldsymbol{w}}^T \equiv \boldsymbol{w}^TA. \tag{B17}$$

Now $\boldsymbol{w}^T\boldsymbol{b}_1$ is a number and if we use the same boundary schemes and quadrature weights at the left and right boundaries then $-\boldsymbol{w}^T\boldsymbol{b}_1 = +\boldsymbol{w}^T\boldsymbol{b}_N$ and we can divide Equation (B16) by it. Finally, putting back $U_i = u_ih_i$ (Equation (B16)) we get the desired discrete conservation law

$$\frac{d}{dt}\sum_{i=1}^{N}\widehat{w}_jh_ju_j = f_1 - f_N, \tag{B18}$$

where the final weights are

$$\widehat{\boldsymbol{w}} = -\frac{\boldsymbol{w}^TA}{\boldsymbol{w}^T\boldsymbol{b}_1}. \tag{B19}$$

Note that this expression is homogeneous in the provisional weights $\boldsymbol{w}$. Hence we may scale the provisional weights as we wish when we solve Equations (B15) to determine them. In particular, it it is convenient to choose them to be unity in the interior of the domain.

B.2. Application to the Present Differentiation Scheme

We consider matrix entries near the left end of the domain; the weights near the right end are obtained by symmetry. Referring back to Section 3.4, the first five columns of the

right-hand-side matrix of the present Padé scheme are

$$B = \begin{pmatrix} a_1 & b_1 & c_1 & & \\ -\hat{a} & 0 & \hat{a} & & \\ & -\hat{a} & 0 & \hat{a} & \\ & & -\hat{a} & 0 & \hat{a} \\ & & & -\hat{a} & 0 \\ & & & & -\hat{a} \end{pmatrix}, \tag{B20}$$

where we have defined $\hat{a} = a/2$. The condition that the weighted sum of the fifth column equals 0 implies that $w_4 = w_6$, which we set equal to unity. The same holds true for the rest of the interior weights. The condition on the fourth column implies that $w_3 = w_5$, which we also set equal to unity. Using the fact that $w_3 = 1$, the second column gives the condition

$$w_1 b_1 - \hat{a} = 0, \tag{B21}$$

or

$$w_1 = \hat{a}/b_1 = (3/4)/2 = 3/8. \tag{B22}$$

The third column gives

$$w_1 c_1 + w_2 \hat{a} - w_4 \hat{a} = 0. \tag{B23}$$

Using known information, we get

$$w_2 = 1 - c_1/b_1 = 3/4. \tag{B24}$$

We now have all the provisional weights.

The final weights are obtained from Equation (B19) where the first four columns of the left-hand-side matrix for the present scheme is

$$A = \begin{pmatrix} 1 & \alpha_1 & 0 & \\ \alpha & 1 & \alpha & \\ & \alpha & 1 & \alpha \\ & & \alpha & 1 \\ & & & \alpha \end{pmatrix}. \tag{B25}$$

Hence Equation (B19) gives the final weights as

$$\widehat{w}_1 = -\frac{w_1 + w_2\alpha}{w_1 a_1 - w_2 \hat{a}} = \frac{3}{8}, \tag{B26}$$

$$\widehat{w}_2 = -\frac{w_1\alpha_1 + w_2 + w_3\alpha}{w_1 a_1 - w_2 \hat{a}} = \frac{7}{6}, \tag{B27}$$

$$\widehat{w}_3 = -\frac{w_2\alpha + w_3 + w_4\alpha}{w_1 a_1 - w_2 \hat{a}} = \frac{23}{24}, \tag{B28}$$

$$\widehat{w}_j = -\frac{w_{j-1}\alpha + w_j + w_{j+1}\alpha}{w_1 a_1 - w_2 \hat{a}} = 1, \quad j \in [4, N-3]. \tag{B29}$$

It is interesting that $\widehat{w}_1 + \widehat{w}_2 + \widehat{w}_3 = 5/2$, which would be the case for the trapezoidal rule and indeed the three weights approximate the weights 1/2, 1, and 1 for the trapezoidal rule.

ORCID iDs

Karim Shariff https://orcid.org/0000-0002-7256-2497

References

Adams, N., & Shariff, K. 1996, JCoPh, 127, 27
Aris, R. 1989, Vectors, Tensors, and the Basic Equations of Fluid Mechanics (New York: Dover)
Batchelor, G. 1967, An Introduction to Fluid Dynamics (Cambridge: Cambridge Univ. Press), 602
Benítez-Llambay, P., & Masset, F. S. 2016, ApJS, 223, 11
Bird, R., Stewart, W., & Lightfoot, E. 1960, Transport Phenomena (New York: Wiley), 739
Boris, J. P., Grinstein, F. F., Oran, E. S., & Kolbe, R. L. 1992, FlDyR, 10, 199
Brady, P., & Livescu, D. 2019, CF, 183, 84
Brandenburg, A., Johansen, A., Bourdin, P., et al. 2021, JOSS, 6, 2807
Cook, A. W., & Cabot, W. H. 2005, JCoPh, 203, 379
Dong, S. 2008, JFM, 615, 371
Frigo, M., & Johnson, S. 2005, IEEEP, 93, 216
Gustafsson, B. 1981, SJNA, 18, 179
Harten, A., Engquist, B., Osher, S., & Chakravarthy, S. 1987, JCoPh, 71, 231
Hu, X., Adams, N., & Shu, C.-W. 2013, JCoPh, 242, 169
Kim, K.-H., Kang, J.-H., Pan, X., & Choi, J.-I. 2021, CoPhC, 260, 107722
Kopal, Z. 1955, Numerical Analysis (London: Chapman & Hall)
Lecoanet, D., McCourt, M., Quataert, E., et al. 2016, MNRAS, 455, 4274
Lele, S. 1992, JCoPh, 103, 16
Lerat, A., Falissard, F., & Sidès, J. 2007, JCoPh, 225, 635
Lesur, G., Ercolano, B., Flock, M., et al. 2022, arXiv:2203.09821
Liepmann, H., & Roshko, A. 1957, Elements of Gasdynamics (Chichester: Wiley), 335
Mani, A., Larsson, J., & Moin, P. 2009, JCoPh, 228, 7368
Marcus, P. S. 1984, JFM, 146, 65
Masset, F. 2000, A&AS, 141, 165
Merriman, B. 2003, JSCom, 19, 309
Mignone, A., Bodo, G., Massaglia, S., et al. 2007, ApJS, 170, 228
Moser, R., Moin, P., & Leonard, A. 1983, JCoPh, 52, 524
Nelson, R. P., Gressel, O., & Umurhan, O. M. 2013, MNRAS, 435, 2610
Neufeld, D. A., & Hollenbach, D. J. 1994, ApJ, 428, 170
Pirozzoli, S. 2002, JCoPh, 178, 81
Ritos, K., Kokkinakis, I., & Drikakis, D. 2018, CF, 173, 307
Seligman, D., & Laughlin, G. 2017, ApJ, 848, 54
Seligman, D., & Shariff, K. 2019, ApJ, 877, 113
Shariff, K., 2024 Pade: A code for protoplanetary disk turbulence based on Pade differencing., v1.0, Zenodo, doi: 10.5281/zenodo.11114378
Shariff, K., & Wray, A. 2018, ApJS, 238, 12
Shu, C.-W., & Osher, S. 1989, JCoPh, 83, 32
Smagorinsky, J. 1963, MWRv, 91, 99
Stone, J., Gardiner, T., Teuben, P., Hawley, J., & Simon, J. 2008, ApJS, 178, 137
Stone, J., Tomida, K., White, C., & Felker, K. 2020, ApJS, 249, 4
Vreman, A. W. 2004, PhFl, 16, 3670
Yee, H., & Sjögreen, B. 2018, CF, 169, 331