Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 227 (2008) 8017–8034
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

A new method for fast transforms in parity-mixed PDEs: Part II. Application to confined rotating convection

Geoffrey M. Vasil [a,*], Nicholas H. Brummell [b,c], Keith Julien [d]

[a] Department of Atmospheric and Oceanic Sciences & JILA, University of Colorado, Boulder, CO 80309, United States
[b] Department of Astrophysical and Planetary Sciences & JILA, University of Colorado, Boulder, CO 80309, United States
[c] Department of Applied Mathematics, University of California, Santa Cruz, CA 95064, United States
[d] Department of Applied Mathematics, University of Colorado, Boulder, CO 80309, United States

Received 16 October 2007; received in revised form 22 April 2008; accepted 22 April 2008
Available online 21 May 2008

Abstract

To demonstrate the utility of the parity filtering methods described by Vasil et al. [G. Vasil, N. Brummell, K. Julien, A new method for fast transforms in parity-mixed PDEs: Part I. Numerical techniques and analysis, J. Comput. Phys. (2008)], we introduce a numerical code designed to solve for Rayleigh–Bénard convection in a confined rotating box using the new methods we have formulated. That is, using a straightforward pseudospectral framework, we incorporate techniques for efficiently computing parity-mixed Coriolis accelerations in a time-dependent numerical solver. The goals of the presented numerical code are to provide a tool to investigate aspects of confined rotating convection experiments with a simple model, and to illustrate the application of parity filtering. In our numerical tests, we find that a correct accounting for parity leads to clear and interesting behavior that has been observed in laboratory experiments but that has not been observed in previous numerical simulations in periodic domains.
Published by Elsevier Inc.

Keywords: Numerical methods; Computational fluid dynamics; Pseudospectral methods; Dealiasing; Parity mixing; Thermal convection; Rotating fluids; Confined fluids

1. Introduction

Pseudospectral methods are often employed in computational fluid dynamics applications, and these methods have been particularly useful in the study of thermal convection dynamics; see [1]. Over the years, numerical exploration of rotating and non-rotating turbulent thermal convection has greatly elucidated the physics of buoyancy in thermally-driven geophysical and astrophysical fluid layers (*e.g.*, see [2,3,5]). Laboratory experiments have also been used to study thermal convection in a controlled environment (*e.g.*, see [6–8]). However, there has been relatively little direct available comparison between geophysically and astrophysically

* Corresponding author. Tel.: +1 303 492 7851; fax: +1 303 492 5235.
E-mail address: geoffrey.vasil@colorado.edu (G.M. Vasil).

0021-9991/$ - see front matter Published by Elsevier Inc.
doi:10.1016/j.jcp.2008.04.040

motivated laboratory experiments and highly supercritical (*i.e.* strongly driven) numerical simulations. There are two main reasons for this gap. First, given the many degrees of freedom present in a strongly driven three-dimensional numerical simulation of thermal convection, an extremely efficient numerical scheme is needed to capture the dynamics accurately and in a reasonable amount of time. Second, from a historical perspective, the efficient numerical algorithm of choice when solving for turbulent convection in the geophysical/astrophysical context has often been some type of Fourier pseudospectral approach in a periodic domain.

When confined domains are considered, traditional pseudospectral methods can become untenable. One of the biggest drawbacks of the pseudospectral approach in connection with laboratory experiments is the difficulty in modeling the influence of realistic boundaries. This is straightforward only in a limited number of situations; even in some Cartesian domains. For a confined and rotating domain, as discussed in [9] (hereinafter Part I), Coriolis accelerations mix trigonometric terms of differing parity (sine versus cosine), and in a confined domain, these differing types of trigonometric bases project onto each other in a manner that can produce serious aliasing errors when spectral coefficients are computed using a discrete Fourier transform. That is, as discussed in Part I, the projection integrals,

$$\int_0^\pi \sin(n_1 z)\cos(n_2 z)\,\mathrm{d}z = \begin{cases} \frac{2n_1}{n_1^2 - n_2^2} & \text{if } n_1 + n_2 \text{ is odd,} \\ 0 & \text{if } n_1 + n_2 \text{ is even,} \end{cases} \tag{1}$$

of an arbitrary sine function onto an arbitrary cosine function do not vanish for general values of n_1 or n_2.

In Part I we devise a new method to overcome the problem of the dense projection of a sine basis onto a cosine basis. We show that if we define the spectral expansion of unity,

$$\mathrm{Id}_m(z) = \frac{4}{\pi}\sum_{n=1}^{m} \frac{\sin((2n-1)z)}{2n-1}, \tag{2}$$

then we can insert Id_m, for $m \geqslant \lceil (n_1 + n_2)/2 \rceil$, into the integrand in Eq. (1) without altering the value of the integral. Since the inclusion of Id_m would flip the parity of either the sine of cosine function in Eq. (1), we can compute an entire collection of integrals of this form via a fast Fourier transform. While Chebyshev methods do provide an alternative for computing in confined domains, these methods are computationally expensive to implement in more than one dimension because of both their high relative cost for obtaining solutions to elliptic equations in multiple dimensions, as well as stringent time-stepping criteria. The solution derived in Part I to the above parity issues can be incorporated easily into many existing Fourier pseudospectral codes. Our methodology therefore allows for the efficient modeling of convection in Cartesian domains that are confined in more that one direction. While more sophisticated methods are required for geometries other than Cartesian boxes or channels, the mere existence of boundaries may impact a system's qualitative dynamics more than the exact shape of those boundaries.

Here in Part II of this series of papers, we discuss the implementation of a numerical code designed to simulate the dynamics of rotating Rayleigh–Bénard convection in a confined stress-free rectilinear box (as in the second example from Part I). Using the parity filtering methods developed in Part I, we present a solution to the problem of aliasing errors that arise from Coriolis accelerations mixing terms of different parity. While we are not able to achieve the exponential convergence enjoyed in periodic domains, we are able to provide a simple method that can achieve high-order algebraic convergence, which is generally difficult in confined (bounded) domains [10]. Furthermore, our parity-filtering methods can be easily incorporated in existing periodic pseudospectral codes for greater flexibility. Specifically, from our numerical tests, we find that a correct accounting for parity in our numerical scheme produces qualitatively interesting dynamics that have been observed in a number of laboratory experiments and are easily missed in numerical solutions where parity mixing is ignored.

2. Dynamical equations and basic assumptions

As a fundamental model we employ the Boussinesq equations for a heated and rotating domain (see [11] Chapter II, Section 8), *viz.*

$$(\partial_t - Pr\Delta)\boldsymbol{u} - Pr\,Ra\,\theta\hat{\boldsymbol{k}} = -\boldsymbol{\nabla}P - (\boldsymbol{\omega} + f\hat{\boldsymbol{k}}) \times \boldsymbol{u}, \tag{3}$$

$$(\partial_t - \Delta)\theta - \hat{\boldsymbol{k}} \cdot \boldsymbol{u} = -\boldsymbol{\nabla} \cdot (\theta \boldsymbol{u}), \tag{4}$$

$$\boldsymbol{\nabla} \cdot \boldsymbol{u} = 0. \tag{5}$$

In Eqs. (3)–(5), $\boldsymbol{u} = (u, v, w)$ is the vector of velocity components in the x, y and z directions, respectively, $\boldsymbol{\omega} = \boldsymbol{\nabla} \times \boldsymbol{u}$ is the relative vorticity (relative to the background vorticity, $f\hat{\boldsymbol{k}}$), and P is the relative kinematic pressure. Eqs. (3)–(5) are valid in a rectilinear Cartesian box that is rotating about the vertical axis, $\hat{\boldsymbol{k}}$. The equations are non-dimensionalized such that the domain has a unit depth in the z-direction. The x and y-directions have aspect ratios of A_x and A_y, respectively.

Regarding thermal forcing, a unit temperature difference is maintained across the z-direction with the bottom boundary being held warmer than the top boundary. The temperature variable, θ, that we employ in Eq. (4), is the thermal perturbation away from the maintained linear background temperature profile. That is, the total temperature is given by

$$T(x, y, z) = 1 - z + \theta(x, y, z). \tag{6}$$

Using θ as a thermal variable is especially convenient since the maintained top and bottom temperature boundary conditions, $T|_{z=0} = 1$ and $T|_{z=1} = 0$, now become homogeneous boundary conditions for the perturbation field, θ. Temperature units are non-dimensionalized by the fixed temperature difference across the depth of the box. Time units are non-dimensionalized according to a thermal diffusion time across the depth of the box. Apart from the aspect ratios, A_x and A_y, the system is described by a set of three dimensionless parameters: the Rayleigh number, the Prandtl number, and the Coriolis parameter. These numbers, denoted $\{Ra, Pr, f\}$, are respectively given by

$$Ra = \frac{g\alpha\delta T d^3}{\nu\kappa}; \quad Pr = \frac{\nu}{\kappa}; \quad f = \frac{2\Omega d^2}{\kappa}. \tag{7}$$

In Eq. (7), d represents the physical depth of the domain, δT is the physical temperature difference across the depth of the domain, κ is the thermal diffusion coefficient, ν is the viscosity, g is the gravitational acceleration, and Ω is the angular rotation frequency. Note that a collection of equivalent dimensionless numbers are also in common use, *e.g.*, the Taylor number, the Ekman number, and the thermal Rossby number, respectively, *viz.*

$$Ta = \frac{f^2}{Pr^2}; \quad Ek = \frac{Pr}{f}; \quad Ro = \frac{\sqrt{Pr\,Ra}}{f}. \tag{8}$$

The particular form of the thermal Rossby number as we have defined it in Eq. (8) was first introduced by Gilman [4], and is discussed in [2], and is useful since it measures the relative influences of rotation and buoyancy in a manner that is independent of molecular diffusion coefficients, ν and κ. As it is defined, Ro gives the ratio of the gravitational free-fall rate, $\sqrt{g\alpha\delta T/d}$, to the basic inertial frequency, 2Ω.

To enforce the divergence-free condition, Eq. (5), we employ a poloidal/toroidal formulation, *i.e.*, the velocity is decomposed such that

$$\boldsymbol{u} = \boldsymbol{\nabla} \times (\Psi\hat{\boldsymbol{k}}) + \boldsymbol{\nabla} \times \boldsymbol{\nabla} \times (\Phi\hat{\boldsymbol{k}}), \tag{9}$$

where Ψ is the toroidal function and Φ is the poloidal function. In component form, this decomposition is given by

$$u = \partial_x\partial_z\Phi + \partial_y\Psi, \tag{10}$$

$$v = \partial_y\partial_z\Phi - \partial_x\Psi, \tag{11}$$

$$w = -(\partial_x^2 + \partial_y^2)\Phi. \tag{12}$$

If we use Eqs. (10) and (11) and the definition of the curl, we find that the vertical component of the vorticity is given by

$$\omega_z = -(\partial_x^2 + \partial_y^2)\Psi. \tag{13}$$

Therefore, if we take the vertical velocity, w, and vertical vorticity, ω_z, as our primary dynamical variables (along with θ), then we can solve for the remaining velocity components (u, v), through the diagnostic relations,

$$-(\partial_x^2 + \partial_y^2)u = \partial_x\partial_z w + \partial_y\omega_z, \tag{14}$$

$$-(\partial_x^2 + \partial_y^2)v = \partial_y\partial_z w - \partial_x\omega_z. \tag{15}$$

The horizontal vorticity components follow by taking the curl of the full velocity field.

Using w and ω_z as dynamical variables means that we can eliminate the pressure in the Boussinesq equations by taking their curl and double curl and projecting onto the vertical direction. This produces coupled dynamical equations for the vertical vorticity and vertical velocity. That is,

$$(\partial_t - Pr\Delta)\omega_z = -\hat{\boldsymbol{k}}\cdot\boldsymbol{\nabla}\times((\boldsymbol{\omega} + f\hat{\boldsymbol{k}})\times\boldsymbol{u}), \tag{16}$$

$$-(\partial_t - Pr\Delta)\Delta w + Pr\,Ra\,\Delta_\perp\theta = -\hat{\boldsymbol{k}}\cdot\boldsymbol{\nabla}\times\boldsymbol{\nabla}\times((\boldsymbol{\omega} + f\hat{\boldsymbol{k}})\times\boldsymbol{u}), \tag{17}$$

where $\Delta_\perp = \partial_x^2 + \partial_y^2$ is the horizontal Laplacian.

The evolution equations, Eqs. (16), (17) and (4), along with the diagnostic relations, Eqs. (14) and (15), form the closed system that we solve in our numerical scheme. However, before we begin a detailed discussion of numerics, we must say something about boundary conditions on the collection of field variables.

For our study of convection in a confined domain, we restrict the type of velocity boundary conditions to impenetrable and stress-free on all of the domain boundaries. We further use perfectly conducting thermal boundary conditions on the top and bottom of the box and perfectly insulating on the sides of the box. We summarize the velocity and thermal boundary conditions as

$$u = \partial_x v = \partial_x w = \partial_x\theta = 0 \quad \text{for } x = 0, A_x, \tag{18}$$

$$\partial_y u = v = \partial_y w = \partial_y\theta = 0 \quad \text{for } y = 0, A_y, \tag{19}$$

$$\partial_z u = \partial_z v = w = \theta = 0 \quad \text{for } z = 0, 1. \tag{20}$$

Furthermore, $\omega_z = \partial_x v - \partial_y u$. By inspecting Eqs. (14) and (15) (which are equivalent to the divergence-free condition), Eqs. (18)–(20) lead to the implied set of conditions

$$\omega_z = \partial_x^2 u = 0 \quad \text{for } x = 0, A_x, \tag{21}$$

$$\omega_z = \partial_y^2 v = 0 \quad \text{for } y = 0, A_y, \tag{22}$$

$$\partial_z\omega_z = \partial_z^2 w = 0 \quad \text{for } z = 0, 1. \tag{23}$$

The fact that ω_z vanishes on the horizontal boundaries, follows directly from its definition and stress-free conditions. The fact that $\partial_z\omega_z$ vanishes on vertical boundaries follows in the same manner. The fact that some of the second derivatives vanish on some of the boundaries results from differentiating the divergence-free condition and applying it at the boundaries. For example, in the x direction,

$$\partial_x^2 u + \partial_{x,y} v + \partial_{x,z} w = 0. \tag{24}$$

Stress-free conditions imply that the second two terms in Eq. (24) vanish at $x = 0, A_x$ and hence $\partial_x^2 u = 0$ at $x = 0, A_x$.

With a modest amount of effort, we could include other types of boundary conditions. The most notable alternatives would be to implements no-slip boundaries for modeling rigid no-slip walls of confined domains (see Part I), and/or to allow one horizontal direction to be periodic to model flow in a rotating channel. While we expect to include these options in future work, we are currently interested in highlighting the numerical application of parity filtering with a simple example. Furthermore, we believe that simple stress-free walls provide a great deal of interesting and, thus far, unexplored dynamical behavior.

Before we begin a discussion of our numerical scheme, we mention kinetic energy balance. To produce an equation for kinetic energy, we contract Eq. (3) with the vector velocity, $\boldsymbol{u}$, and integrate over the entire domain volume to obtain

$$\frac{\mathrm{d}}{\mathrm{d}t}\frac{\|\boldsymbol{u}\|^2}{2} = Pr(Ra\langle\theta, w\rangle - \|\boldsymbol{\omega}\|^2), \tag{25}$$

where the angle-bracket notation is used to denote $\langle\theta, w\rangle = \int \theta w \mathrm{d}^3\boldsymbol{x}$. Eq. (25) balances the time rate of change of kinetic energy on the left hand side with buoyancy work and dissipation on the right hand side respectively. Since the system is externally forced and dissipative there is no conserved (time-independent) energy functional. However, maintaining the balance in Eq. (25) is a helpful validity check when we solve our system numerically.

A notable feature of Eq. (25) is that there are no contributions to kinetic energy from fluid advection nor from Coriolis accelerations. In particular, Coriolis accelerations can only act to stir and deflect the fluid, not to speed it up or slow it down on average. That is, they can transfer energy between velocity components (and scales in cases with parity mixing), but they can do no work. This is important to keep in mind when we analyze our numerical solutions. It turns out that if we do not properly account for parity mixing, then we may obtain numerical solutions where Coriolis accelerations spuriously contribute to the work-energy budget.

3. Numerical scheme

In general, stress-free boundaries are particularly convenient since, for a given dynamic variable, each direction can be represented by either a single sine series or a single cosine series. Therefore, taken together, Eqs. (18)–(23) suggest a trigonometric expansion of the following form for our dynamical variables:

$$w = \sum_{l=0}^{N_x}{}' \sum_{m=0}^{N_y}{}' \sum_{n=1}^{N_z} W_{l,m,n}(t)\cos(l\pi x/A_x)\cos(m\pi y/A_y)\sin(n\pi z), \tag{26}$$

$$\omega_z = \sum_{l=1}^{N_x} \sum_{m=1}^{N_y} \sum_{n=0}^{N_z}{}' Z_{l,m,n}(t)\sin(l\pi x/A_x)\sin(m\pi y/A_y)\cos(n\pi z), \tag{27}$$

$$\theta = \sum_{l=0}^{N_x}{}' \sum_{m=0}^{N_y}{}' \sum_{n=1}^{N_z} \Theta_{l,m,n}(t)\cos(l\pi x/A_x)\cos(m\pi y/A_y)\sin(n\pi z). \tag{28}$$

For notational convenience, we employ the prime-summation notation throughout the remainder of this paper, *i.e.*, for any sequence, c_l with $l \geqslant 0$

$$\sum_{l=0}^{N}{}' c_l = \frac{c_0}{2} + \sum_{l=1}^{N} c_l. \tag{29}$$

Furthermore, the horizontal velocities are expanded such that

$$u = \sum_{l=1}^{N_x} \sum_{m=0}^{N_y}{}' \sum_{n=0}^{N_z}{}' U_{l,m,n}(t)\sin(l\pi x/A_x)\cos(m\pi y/A_y)\cos(n\pi z), \tag{30}$$

$$v = \sum_{l=0}^{N_x}{}' \sum_{m=1}^{N_y} \sum_{n=0}^{N_z}{}' V_{l,m,n}(t)\cos(l\pi x/A_x)\sin(m\pi y/A_y)\cos(n\pi z), \tag{31}$$

where Eqs. (14) and (15) give the coefficients $U_{l,m,n}$ and $V_{l,m,n}$ in terms of linear combinations of $W_{l,m,n}$ and $Z_{l,m,n}$, *i.e.*,

$$U_{l,m,n} = -\frac{k_x k_z W_{l,m,n} - k_y Z_{l,m,n}}{k_x^2 + k_y^2}, \tag{32}$$

$$V_{l,m,n} = -\frac{k_y k_z W_{l,m,n} + k_x Z_{l,m,n}}{k_x^2 + k_y^2}, \tag{33}$$

and where $k_x = l\pi/A_x, k_y = m\pi/A_y$ and, $k_z = n\pi$.

The Coriolis terms in Eq. (3) (representing rotation about a vertical axis) mix the horizontal velocities such that u and v are directly coupled, *i.e.*,

$$\partial_t u - fv = -\partial_x P \cdots \tag{34}$$

$$\partial_t v + fu = -\partial_y P \cdots . \tag{35}$$

Also, see Example 2 from Part I for further details. By examining the parity of each term in Eqs. (34) and (35), we see that the Coriolis terms mix parity in the horizontal directions. This parity mixing is carried through to the Coriolis terms in Eqs. (16) and (17). As discussed in Part I, the judicious placement of the spectral identity function, Id, (defined in one dimension in Eq. (2)), can change the parity of the mixed Coriolis terms without changing their projection on the "parity-flipped" basis. This then allows for the use of a fast pseudospectral method to compute the right hand sides of our dynamical equations.

However, if we examine the vertical direction, we see that all the terms in Eqs. (34) and (35) are naturally represented by a cosine series in the z-direction. In general, given stress-free boundaries, all of the terms in the dynamical equations, except the Coriolis terms, preserve parity. Specifically, under the given boundary conditions, all of the nonlinear terms preserve the parity of each dynamical variable. For example, consider the x-direction advection term, $(u\partial_x + v\partial_y + w\partial_z)u$. In the first piece, $u\partial_x u$, the x-derivative renders $\partial_x u$ as a triple cosine series and therefore $u\partial_x u$ has the same parity as $u(x,y,z)$ itself. For the second term, $v\partial_y u$, the y-derivative renders $\partial_y u$ as a sine series in the x-derivative, a sine series in the y-derivative and a cosine series in the z-direction. After differentiation, the multiplication by $v(x,y,z)$ results in $v\partial_y u$ being a sine series in the x-direction and a cosine series in the other two directions. This is the same parity as $u(x,y,z)$. The parity of the third term, $w\partial_z u$ follows from a similar argument as above.

In matrix form, the full system to solve is given by

$$\mathbf{M}.\dot{\boldsymbol{X}} + \mathbf{A}\cdot\boldsymbol{X} = \boldsymbol{F}(\boldsymbol{X}). \tag{36}$$

The array of dynamical variables is given by

$$\boldsymbol{X} = (\omega_z, w, \theta)^{\mathrm{T}}. \tag{37}$$

The differential matrix operators in Eq. (36) are given by

$$\mathbf{M} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -\Delta & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad \mathbf{A} = \begin{pmatrix} -Pr\Delta & 0 & 0 \\ 0 & Pr\Delta^2 & Pr\,Ra\,\Delta_\perp \\ 0 & -1 & -\Delta \end{pmatrix}. \tag{38}$$

The quadratic right hand side of Eq. (36) is given by

$$\boldsymbol{F}(\boldsymbol{X}) = -\begin{pmatrix} \hat{\boldsymbol{k}}\cdot\boldsymbol{\nabla}\times(\boldsymbol{\omega}_{\mathrm{total}}\times\boldsymbol{u}) \\ \hat{\boldsymbol{k}}\cdot\boldsymbol{\nabla}\times\boldsymbol{\nabla}\times(\boldsymbol{\omega}_{\mathrm{total}}\times\boldsymbol{u}) \\ \boldsymbol{\nabla}\cdot(\theta\boldsymbol{u}) \end{pmatrix} \tag{39}$$

where the total vorticity is computed via

$$\boldsymbol{\omega}_{\mathrm{total}} = \boldsymbol{\nabla}\times\boldsymbol{u} + f\,\mathrm{Id}\hat{\boldsymbol{k}}. \tag{40}$$

Explicitly, the identity function in the current setup is given by

$$\mathrm{Id} = \frac{16}{\pi^2}\sum_{l=1}^{N_x}\sum_{m=1}^{N_y}\frac{\sin((2l-1)\pi x/A_x)\sin((2m-1)\pi y/A_y)}{(2l-1)(2m-1)}. \tag{41}$$

Given the above boundary conditions and parities, Eq. (40) is the only place in our numerical scheme where we need to use the identity function given by Eq. (41). Furthermore, given that the vertical direction possesses consistent parity throughout all of the dynamical equations, we only need to dealias that direction with the standard 2/3 rule. The horizontal directions need to be dealiased with a 1/2 rule as described in Part I.

The implementation of our parity-filtering methods typically amounts to a few simple steps. Firstly, we calculate Id in grid space. Given that Id is unchanging, we can compute it from Eq. (41) as a preprocessing step by either explicitly computing the sums, or by feeding its simple Fourier spectrum into a FFT. Secondly, we compute the offending parity-mixing terms form our dynamical equations in grid space and multiply them by

Id. Thirdly, given that multiplication by Id will additionally broaden the spectrum of our dynamical variables, we, therefore, must use a more conservative dealiasing rule than we would if parity filtering were not required. With these three alterations, we proceed just as we would if there were no parity mixing in the problem.

For a time-stepping scheme, we use a semi-implicit Crank–Nicolson method for the time evolution, buoyancy forcing, and diffusion terms and an explicit 3-level Adams–Bashforth scheme for the Coriolis and nonlinear advection terms. Thus,

$$\left(\mathbf{M}+\frac{\delta t_0}{2}\mathbf{A}\right)\cdot \boldsymbol{X}^{(n+1)} = \left(\mathbf{M}-\frac{\delta t_0}{2}\mathbf{A}\right)\cdot \boldsymbol{X}^{(n)} + \delta t_0(\alpha_0 \boldsymbol{F}^{(n)} - \alpha_1 \boldsymbol{F}^{(n-1)} + \alpha_2 \boldsymbol{F}^{(n-2)}). \tag{42}$$

We employ a variable time step, therefore the Adams–Bashforth coefficients are given by

$$\begin{aligned}
\alpha_0 &= 1+\alpha_1-\alpha_2,\\
\alpha_1 &= \frac{\delta t_0 + 2\alpha_2(\delta t_1+\delta t_2)}{2\delta t_1},\\
\alpha_2 &= \frac{\delta t_0(2\delta t_0+3\delta t_1)}{6\delta t_2(\delta t_1+\delta t_2)},
\end{aligned} \tag{43}$$

where δt_0 is the current time step, δt_1 is the previous time step, and δt_2 is the time step prior to that. The only exceptions to the Adams–Bashforth 3-level weights are for the first and second time step. On the first time step, we take a very small Euler step with $\alpha_1=\alpha_2=0$ and $\alpha_0=1$. The second time step is Adams–Bashforth 2-level, *i.e.*, we take a step with $\alpha_2=0$ and (α_0,α_1) following from Eq. (43). In Eq. (42), the superscripts on $\boldsymbol{X}$ represent the following:

$$\begin{aligned}
\boldsymbol{X}^{(n+1)} &= \boldsymbol{X}(t+\delta t_0), \quad \boldsymbol{X}^{(n)} = \boldsymbol{X}(t),\\
\boldsymbol{X}^{(n-1)} &= \boldsymbol{X}(t-\delta t_1), \quad \boldsymbol{X}^{(n-2)} = \boldsymbol{X}(t-\delta t_1-\delta t_2)
\end{aligned} \tag{44}$$

and the superscripts on the right hand side simply imply $\boldsymbol{F}^{(i)} = \boldsymbol{F}(\boldsymbol{X}^{(i)})$.

Mixed Crank–Nicolson–Adams–Bashforth schemes are often successfully employed in many large three-dimensional simulations; see [12]. The time-stepping is second order in the latest time step, δt_0, as long as the previous time steps, δt_1 and δt_2, are of the same general magnitude as δt_0. That is, the various error terms in Eq. (42) are all given by $\mathcal{O}(\delta t_0^{i_0}\delta t_1^{i_1}\delta t_2^{i_2})$ with $i_0+i_1+i_2 \geqslant 2$. However, i_0 by itself is not generally greater than 2. Nevertheless, the particular scheme is chosen for its good stability and convergence properties, and its ease of implementation; see [13] for a recent review.

Since the Crank-Nicolson part of our scheme includes buoyancy forcing, it is not unconditionally stable. However, one can determine that if $\delta t_0 < 2(PrRa)^{-1/2}$, then stability is ensured. This by itself is not extremely restrictive, and in practice it is quite a conservative estimate. For large Rayleigh numbers, one can obtain a much better asymptotic stability condition. Nevertheless, the specific form of the asymptotic condition is not important since we find that our time step is mostly limited by either the explicit CFL condition or by a diffusive accuracy condition.

With regard to calculating the various terms in Eq. (42), all of the derivatives are computed in spectral space, and therefore the solution is updated in spectral space. After the current vector velocity and relative vorticity are known in spectral space, we transform (inverse FFT) both of these quantities to grid space. To insure that variables represented by both sine or cosine series can be multiplied on the same grid, we use transforms with the collocation points given by

$$\frac{x_i}{A_x} = \frac{2i-1}{2N_x}, \quad \frac{y_j}{A_y} = \frac{2j-1}{2N_y}, \quad z_k = \frac{2k-1}{2N_z}. \tag{45}$$

After we know the relative vorticity in grid space we compute the total vorticity from Eq. (40) and compute the vector product, $\boldsymbol{\omega}_{\text{total}} \times \boldsymbol{u}$, in grid space. The heat flux, $\theta \boldsymbol{u}$, is also computed in grid space. After these multiplications, both resulting vectors are brought to spectral space with a forward FFT, and the remaining derivative operations are calculated, yielding the current value of $\boldsymbol{F}^{(n)}$. The horizontal directions are dealiased according to the methods described in Part I, and the vertical direction is dealiased according to the standard 2/3 rule. Computing the current nonlinear forcing allows the calculation of the complete right hand side of Eq.

(42). The last step is to update the solution by inverting the Crank-Nicolson matrix on the left hand side of Eq. (42). This is easily carried out in spectral space.

With regard to checking energy balance, we compute both sides of Eq. (25) independently and compare the values. We can easily compute the right hand side from available data at each time step. The left hand side requires computing the second-order-finite-difference derivative of the kinetic energy at each time step. That is, if $K = \|\boldsymbol{u}\|^2/2$, then

$$\frac{\mathrm{d}K(t)}{\mathrm{d}t} = c_1 K(t) - c_2 K(t-\delta t_1) + c_3 K(t-(\delta t_1+\delta t_2)) + \mathcal{O}(\delta t_1(\delta t_1+\delta t_2)), \tag{46}$$

where

$$c_1 = \frac{1}{\delta t_1} + \frac{1}{\delta t_1+\delta t_2}, \quad c_2 = \frac{1}{\delta t_1} + \frac{1}{\delta t_2}, \quad c_3 = \frac{1}{\delta t_2} - \frac{1}{\delta t_1+\delta t_2} \tag{47}$$

and where δt_1 and δt_2 are the past two time steps.

Our numerical code is written in Fortran 90/95 and we take advantage of the language's object-oriented features. We compute our Fourier transforms using the portable *FFTW* library [14]. While our numerical code in not massively parallelized, we have optimized it to run efficiently on current shared-memory 2-core architectures. Using a relatively fast single processor machine, we are able to achieve moderate resolutions for our three-dimensional computations.

4. Numerical tests

To validate our numerical code, we run a set of test cases. The ultimate goals are to highlight the numerical consequences of using the parity filtering techniques developed in Part I and to introduce a code for future studies of confined rotating convection. There is a slight difficulty in that there are no known exact analytical solutions for direct comparison. However, we do have a large amount of information about what kinds of general properties we expect physical solutions to possess. Therefore, through physical and mathematical reasoning and some simple related analytical results we are able to determine that our parity filtering techniques are both valid and necessary in the current context.

In all of our rotating tests, we set the box aspect ratios equal to unity, $A_x = A_y = 1$. For all of our rotating and non-rotating numerical tests, we set the Prandtl number, $Pr = 8.4$. For this particular example, parity mixing (and therefore parity filtering) is entirely related to Coriolis accelerations. However, as a useful check of the non-rotational parts of our code, we can consider non-rotating examples, *i.e.*, with $f = 0$.

4.1. Non-rotating tests

If we do not include rotational effects, then we can compare our numerical results to standard analytical stability/instability results. In particular, if $A_x = A_y = 2$, then we know that convection will arise spontaneously for Rayleigh numbers that are greater than $Ra_{\text{critical}} = 27\pi^4/4 \simeq 657.511$ (see [11], Chapter II, Section 15). The aspect ratios are set such that the most unstable mode exactly fits inside the box. Below this critical value for Ra, the quiescent state is globally attracting.

We run two tests with rotation switched off. For these tests we use a resolution of $32 \times 32 \times 32$ (higher than actually needed, the same results hold true for lower resolution) and $Ra = 650 < Ra_{\text{critical}}$ and $Ra = 665 > Ra_{\text{critical}}$. Both runs start from the same small random initial condition. Just as expected, the supercritical case grows at a slow exponential rate until it reaches a saturated state, and the subcritical case simply decays at a similarly slow exponential rate toward the rest state. Furthermore, after some fast transients die away, the growth rates obtained from our numerical tests match expected theoretical growth rates quite well. These rates are obtained by determining the slope of the lines in the linear portion of the plots from Fig. 1a. We measure these slopes by performing a least-squares fit to a middle section of each time series after transients die away but before nonlinear saturation becomes too important.

For the $Ra = 650$ case, theory predicts that the time for the energy to decrease by a factor of 10 is given by $t_{10} \simeq 7.6094849$. We match this value extremely well with a measured value of $t_{10} \simeq 7.6094823$ (see [11],

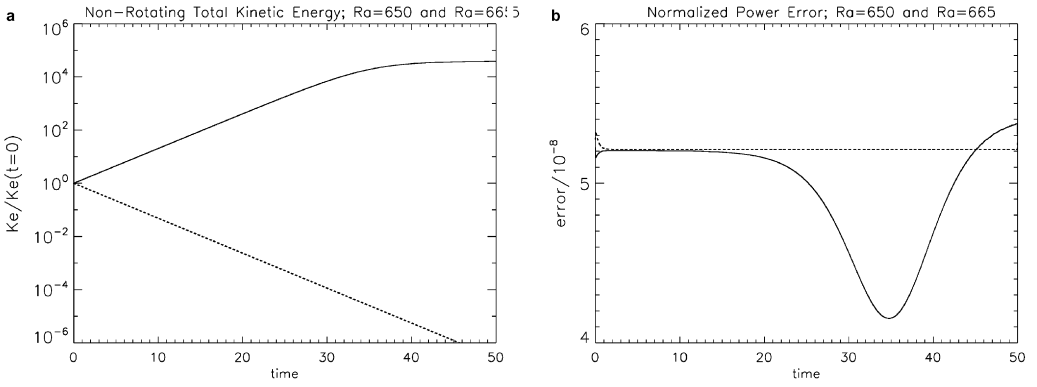

Fig. 1. (a) Kinetic energy versus time for the non-rotating tests. The solid (growing) line corresponds to $Ra = 665$ and, the dashed (decaying) line corresponds to $Ra = 650$. After some fast transients die away the solution either grows or decays exponentially until, in the supercritical case, the energy saturates at a finite value. (b) A time series showing the normalized difference between the time rate of change of kinetic energy and the power generation; $|dKe/dt - \text{Power}|/|\text{Power}|$. We are able to meaningfully normalize by the absolute value of the actual computed power since the solution is either growing or decaying; not oscillating. This is not possible in general. Note that the error is shown increased by a factor of 10^8. The dashed line corresponds to $Ra = 650$ and the solid line corresponds to $Ra = 665$. In either case the error is smooth and never more than 5–6 parts in 10^8.

Chapter II, Section 10). The difference between these two values is easily within the error of our time-stepping scheme.

For the $Ra = 665$ case, theory predicts that the time for the energy to increase by a factor of 10 is given by $t_{10} \simeq 7.641$. We match this value less well than for the decaying solution with a measured value of $t_{10} \simeq 7.650$. The values are still reasonably close (within about a 0.1%), given that the relatively larger difference should be anticipated. For a supercritical Rayleigh number, the solution is growing and hence nonlinearities become increasingly important as time progresses. This saturation mechanism acts to slow down the growth of the supercritical solution, and hence it takes a somewhat longer time to grow by a factor of 10.

The non-rotating tests provide validation that the parts of the code that do not pertain to parity filtering are operating correctly. We now proceed to examine the behavior of our code with rotational effects.

4.2. Rotating tests: non-parity-filtered

In this section, we demonstrate the imprudence of not properly accounting for parity mixing in the Coriolis accelerations. We run a calculation with the parameters $Ra = 6000, f = 350, Pr = 8.4$, and $A_x = A_y = 1$. These parameters correspond to a thermal Rossby number of $Ro = 0.64$, a viscous Ekman number of $Ek = 0.024$, and a Taylor number of $Ta = 1,736$. We use a resolution of $32 \times 32 \times 32$. These parameters are also the values we use in the following tests with parity filtering included. The parameters are convenient since they allow us to run reasonable computations at moderate-to-low resolutions for the purposes of testing. The values also lie in a simple, but non-trivially interesting, region of parameter space.

When we turn parity filtering off, we simply do not employ the spectral expansion of unity, Id, in the calculation of the Coriolis terms on the right hand side of Eq. (40). When we do this, we obtain a solution that does not outwardly appear implausible. However, upon closer examination the solution lacks some important properties that we should expect.

We begin computation from a small random initial condition. The solution begins to grow exponentially and eventually saturates at some steady state solution; see Fig. 2a. This solution seems physically plausible but there are a number of reasons to be unsatisfied with a steady solution. Symmetry arguments, stability calculations, and a number of recent experiments imply that the onset of convection in a rotating and confined system should generally be oscillatory rather than steady; see [15,6].

Furthermore, when we consider the energy budget as defined in Eq. (25), we find that there exists spurious power generation. Since the solution eventually goes to a steady state, the time rate of change of kinetic energy goes to zero. However, the rate of energy generation as given by the right hand side of Eq. (25) clearly tends

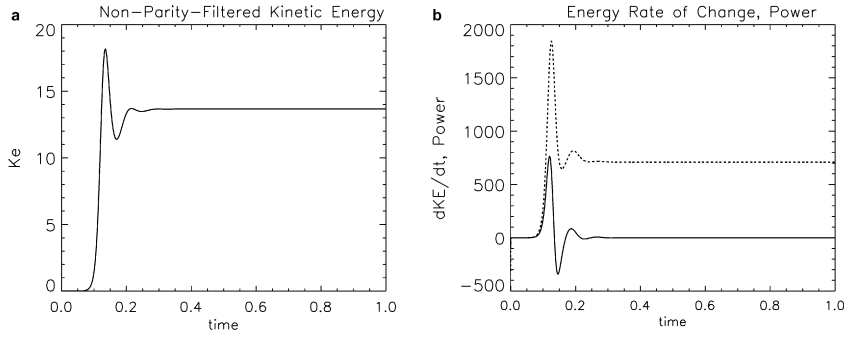

Fig. 2. (a) A time series of kinetic energy for the non-parity-filtered calculation. (b) The solid line shows a time series of the time rate of change of kinetic energy. The dashed line shows the sources of power generation as computed from the right hand side of Eq. (25). While the two time series should coincide, they clearly do not.

toward a finite value; see Fig. 2b. A constant rate of power generation should imply a perpetual linear increase in the kinetic energy, and this is clearly unphysical. It is apparent that the system is leaking energy via some additional power sink. Since we account for energy balance in the non-rotating case, this sink of energy can only come from an incorrect treatment of the Coriolis terms. We find in the next section that parity filtering solves this incorrect treatment.

4.3. Rotating tests: parity-filtered

In this section, we re-examine the previous numerical tests with properly parity-filtered Coriolis accelerations, *i.e.*, with Id included in the Coriolis terms in Eq. (40). As in the previous section, we run calculations

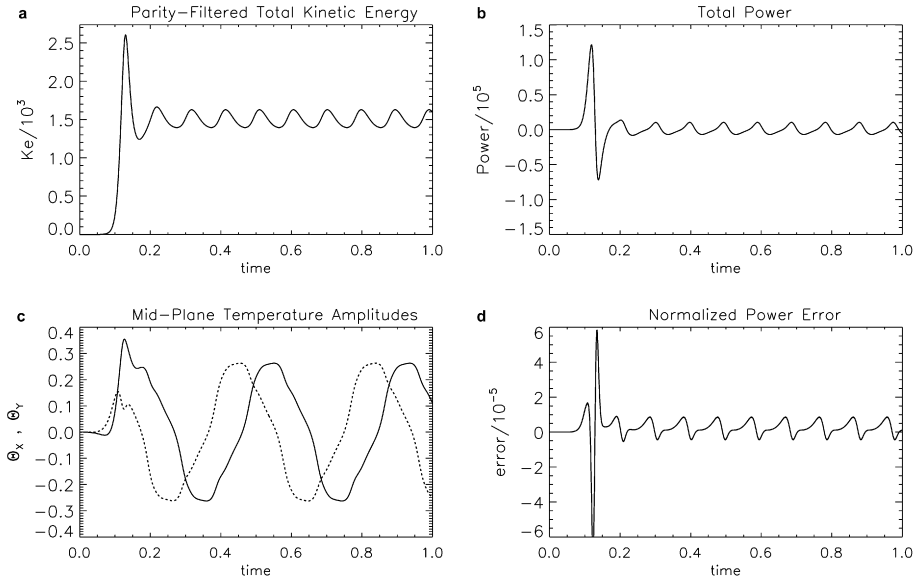

Fig. 3. (a) A time series of kinetic energy for the parity-filtered calculation. The oscillations in the energy are from higher-order harmonics rather than the dominant precessing mode. The energy amplitude scale is shown reduced by a factor of 10^3. (b) A time series of power generation as computed from the right hand side of Eq. (25). The power amplitude scale is shown reduced by a factor of 10^5. (c) Two time series showing the precession of the dominant mode amplitudes. The solid line, Θ_x, shows the temperature at the mid-plane, $z = 1/2$, projected onto the function $\cos(\pi x)$. The dashed line, Θ_y, shows the temperature at the mid-plane, $z = 1/2$, projected onto the function $\cos(\pi y)$. After an initial transient, the two time series clearly oscillate with a well-defined dominant frequency and are a quarter period out of phase. (d) A time series showing the normalized difference between the time rate of change of kinetic energy and the power generation; $(\mathrm{d}Ke/\mathrm{d}t - \mathrm{Power})/\langle|\mathrm{Power}|\rangle$, where $\langle|\mathrm{Power}|\rangle \simeq 8200$. The error amplitude scale is shown increased by a factor of 10^5. After the initial transient, the normalized error is never more than about 1 part in 10^5. This is consistent with the error in our time-stepping scheme.

with the parameters $Ra = 6000, f = 350, Pr = 8.4$, and $A_x = A_y = 1$. Except when we perform convergence tests, we use a resolution of $32 \times 32 \times 32$. As before, we begin a simulation from a small random initial condition and allow the initial instability to saturate nonlinearly at a finite solution.

The first two notable features in the new parity-filtered results are that the kinetic energy and power settle into an oscillatory state and that the average value of the energy is significantly larger than in the non-parity-filtered example; see Figs. 3a and b. The significant difference between the two energy amplitudes is readily explained by the elimination of the spurious energy sink in the previous tests. The most intriguing aspect of the current solution is the presence of a large-scale precessing meridional cell. This precession is shown in the temperature amplitude traces in Fig. 3c. We calculated these amplitudes by projecting the mid-plane temperature perturbations onto $\cos(\pi x/A_x)$ and $\cos(\pi y/A_y)$, respectively, *i.e.*,

$$\Theta_x = 2\int_0^{A_x}\int_0^{A_y} \cos(\pi x/A_x)\theta(x,y,z=1/2)\,\mathrm{d}x\,\mathrm{d}y, \tag{48}$$

$$\Theta_y = 2\int_0^{A_x}\int_0^{A_y} \cos(\pi y/A_y)\theta(x,y,z=1/2)\,\mathrm{d}x\,\mathrm{d}y. \tag{49}$$

These amplitude traces are reminiscent of temperature traces seen in highly turbulent laboratory experiments, in particular, those of Hart and colleagues (see [7]). Among other interesting effects, these experiments found a prominent large-scale meridional circulation, the plane of which precessed counter to the basic rotation direction. While these experiments were carried out in a cylinder, the same behavior was also observed in a rotating box; see [16]. In our numerical experiments, we can clearly see a prominent mode that precesses in a retrograde direction in the mid-plane temperature perturbations seen in Fig. 4. Furthermore, these temperature pertur-

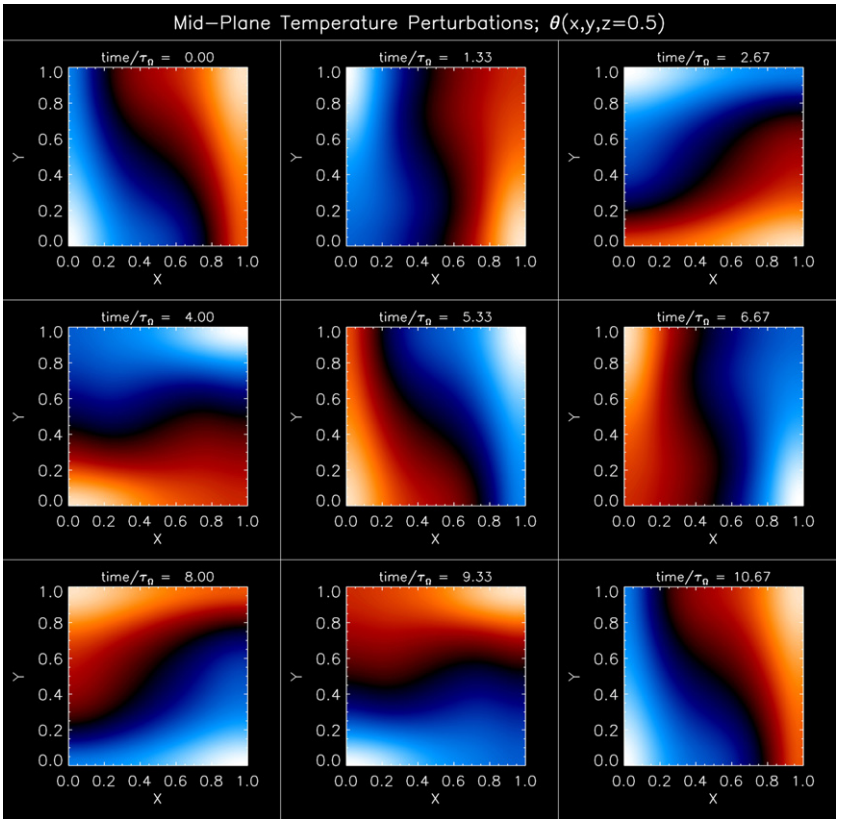

Fig. 4. A time sequence showing the precession of temperature perturbations in the horizontal mid-plane, $z = 1/2$. Relatively warm regions are shown with red tones and relative cool regions are shown with blue tones. The sense of rotation is in a counterclockwise direction, and the precession is in a clockwise (retrograde) sense. The relative time is given in units of rotation period, where $\tau_\Omega = 4\pi/f \simeq 0.036$ in the thermal time units defined in Section 2.

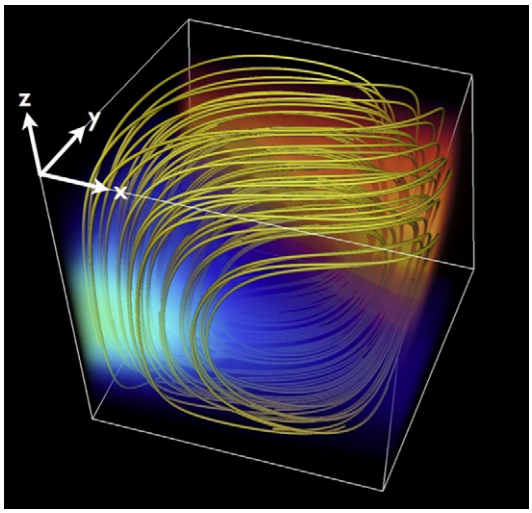

Fig. 5. A volume rendering (see [22] for a review of methods) showing temperature perturbations, $\theta(x, y, z)$, for a typical time snapshot. Relatively warm regions are shown with red tones and relative cool regions are shown with blue tones. The yellow lines trace a number of instantaneous flow stream lines. The warm regions generally correspond to rising flow, and the cool regions generally correspond to downward flow. The rotation axis is aligned with the z-direction and is in a right-handed orientation.

bations are accompanied by a large-scale meridional flow that can be seen in the volume visualization in Fig. 5.

Our laminar low-resolution numerical tests do not nearly approach the level of nonlinearity present in most laboratory experiments. Nevertheless, we believe that the large meridional circulation is a thermally-excited inertial wave that can exist in spite of background turbulence as opposed to resulting from background turbulence. Furthermore, a great deal of theoretical work and other experiments imply that precessing modes are favorable in rotating systems; see again [15,6]. Hence, the qualitative agreement between our numerical work and laboratory experiments provides good verification that our code is producing physical results. Alternatively, in our non-parity-filtered tests discussed in Section 4.2, there is no vestige of a precessing signal. This provides further evidence that our parity-filtering methods are essential in obtaining the proper qualitative behavior from our given system. We will discuss the detailed physics of these precessing circulations in an upcoming publication.

We must also check the energy consistency of the parity-filtered solutions. Recall that in Section 4.2, when we computed the power from the right hand side of Eq. (25), we obtained a finite value at late times even though there was no change in the value of the energy for these times. When we make the same comparison in the current case, we achieve very good agreement. Fig. 3d shows that the difference between the power sources and the time rate of change of kinetic energy is typically around 1 part in 10^5.

Finally, we perform a series of calculations for the purpose of testing convergence. For the convergence test, we continue to use $Ra = 6000, f = 350$, and $Pr = 8.4$. Since parity filtering in the current example is only applied to the horizontal directions, we will mainly focus on the resolution scaling in these directions. We feature a series of calculations with a fixed vertical resolution of $n_z = 36$. This resolution is sufficient to capture the boundary layers accurately that develop at the top and bottom of the domain. With this choice of vertical resolution, we run a calculation for each horizontal resolution $n_x, n_y = 8, 12, 16, 24, 32, 64, 96$.

4.4. Parity-filtered rotating tests: convergence results

Given that we do not have any exact analytical result to compare with each approximate solution, we choose to show the Cauchy series of the difference between each successive resolution doubling. That is, we choose one or more norms, $\|\circ\|$, and show the series, $\|\boldsymbol{u}_n - \boldsymbol{u}_{2n}\|$ converges over our range of resolutions. If we choose a norm that is equivalent to an L^p-norm for $1 \leqslant p \leqslant \infty$, then we can use the completeness of

L^p-spaces to argue for strong convergence; see [17] Section 2.7. We use two different norms that can easily be computed from Fourier amplitudes. This is most convenient since calculations of different resolutions do not share the same grid points (see Eq. (45)), but their Fourier amplitudes are still easily combined. The choice of norms avoids performing some form of trigonometric interpolation for each calculation. The first norm we choose is the L^2-norm, *i.e.*, the energy-norm. This is computed using Parseval's identity via,

$$\|\boldsymbol{u}\|_{L^2}^2 = 2\sum_{l=0}^{N_x}{}'\sum_{m=0}^{N_y}{}'\sum_{n=0}^{N_z}{}'(U_{l,m,m}^2 + V_{l,m,m}^2 + W_{l,m,m}^2). \tag{50}$$

We are also interested in the L^∞-norm, *i.e.* the sup-norm. This norm must be computed in grid space. However, we can bound the L^∞-norm by the l^1-norm of the Fourier amplitudes. That is, if,

$$\|\hat{\boldsymbol{u}}\|_{l^1} = \sqrt{3}\sum_{l=0}^{N_x}{}'\sum_{m=0}^{N_y}{}'\sum_{n=0}^{N_z}{}'(U_{l,m,m}^2 + V_{l,m,m}^2 + W_{l,m,m}^2)^{1/2}, \tag{51}$$

then

$$(u^2 + v^2 + w^2)^{1/2} \leqslant \|\hat{\boldsymbol{u}}\|_{l^1}. \tag{52}$$

Also, using the Hölder Inequality, we have the ordering, $\|\boldsymbol{u}\|_{L^2} \leqslant \|\boldsymbol{u}\|_{L^\infty} \leqslant \|\hat{\boldsymbol{u}}\|_{l^1}$.

Fig. 6 shows the Cauchy error convergence sequences for both norms given in Eqs. (50) and (51). The solutions are calculated over a complete precession period, and the norms are averaged over this time span. In both cases, the sequences are very close to linear on a log–log scale. This linearity implies a power-law form of convergence. Also, both scaling exponents are consistent with each other and with a *velocity-spectrum* scaling exponent of $2.81 + 1 \approx 3.35 + 1/2 = 3.85$. This is reasonably close to the value of 4 that one might obtain from a somewhat naive analysis of Eq. (3) and its derivatives at the domain boundary. However, a subtle analysis reveals that weak corner singularities likely do imply a fractional scaling exponent that is greater than three, but still somewhat less than four.

Finding the (approximate) asymptotic convergence rate a requires an analysis of each component of the momentum equations and derivatives thereof evaluated at the horizontal boundaries as well as the application of an "*Integration By-Parts Coefficient Bound Theorem*" (see [10], Section 2.9). That is, for each dynamic variable, we must determine the highest-order derivatives that we are able to represent with a pointwise-convergent Fourier series (in the horizontal directions). We already know from impenetrability that $u = 0$ for $x = 0, A_x$ and $v = 0$ for $y = 0, A_y$. Also, Eqs. (18)–(22) summarize higher-order boundary conditions arising

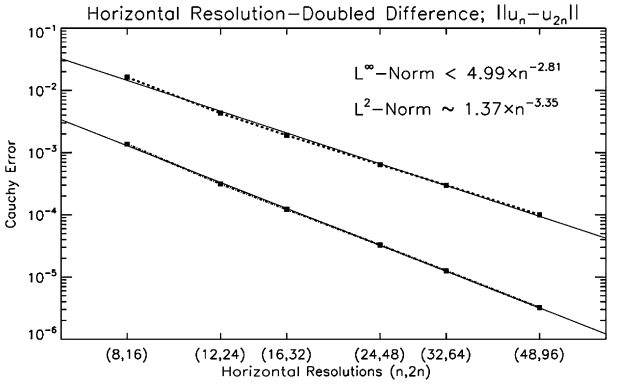

Fig. 6. Log–log plots of horizontal convergence tests with two different norms. Since we do not have an exact solution to compare against, we show the Cauchy sequence, $\|\boldsymbol{u}_n - \boldsymbol{u}_{2n}\|$, where n gives both horizontal resolutions, *i.e.*, $n_x = n_y = n$. The vertical resolution is held fixed at $n_z = 32$. The dashed lines show the computed differences, while the solid lines show the least-squares fit. The top line uses a norm that bounds the L^∞-norm, *i.e.*, the l^1-norm of the Fourier spectral amplitudes. The bottom line uses the L^2-norm, which is equivalent to the l^2-norm of the Fourier spectral amplitudes. In both cases, power-law scalings of the best-fit lines are given in the upper right-hand corner. In all cases, the normed difference was averaged over a complete period of the precessing solution; see Fig. 4.

from stress-free dynamic boundary conditions. Therefore, integration-by-parts for the spectral coefficients of, say, $u(x,y,z)$ is allowed to proceed until *at least* fourth order in the x-direction and at third order in the y-direction. That is, if we (temporarily) assume for simplicity that $A_x = A_y = \pi$, then given

$$U_{l,m} = \frac{4}{\pi^2}\int_0^\pi \int_0^\pi u(x,y,z)\sin(lx)\cos(my)\,\mathrm{d}x\,\mathrm{d}y, \tag{53}$$

$$V_{l,m} = \frac{4}{\pi^2}\int_0^\pi \int_0^\pi v(x,y,z)\cos(lx)\sin(my)\,\mathrm{d}x\,\mathrm{d}y, \tag{54}$$

$$W_{l,m} = \frac{4}{\pi^2}\int_0^\pi \int_0^\pi w(x,y,z)\cos(lx)\cos(my)\,\mathrm{d}x\,\mathrm{d}y, \tag{55}$$

after a number of integration-by-parts and application of boundary conditions we have,

$$= \frac{4\pi^{-2}}{l^4 m^3}\int_0^\pi \int_0^\pi \partial_x^4\partial_y^3 u(x,y,z)\sin(lx)\sin(my)\,\mathrm{d}x\,\mathrm{d}y, U_{l,m} \tag{56}$$

$$= \frac{4\pi^{-2}}{l^3 m^4}\int_0^\pi \int_0^\pi \partial_x^3\partial_y^4 v(x,y,z)\sin(lx)\sin(my)\,\mathrm{d}x\,\mathrm{d}y, V_{l,m} \tag{57}$$

$$W_{l,m} = \frac{4\pi^{-2}}{l^3 m^3}\int_0^\pi \int_0^\pi \partial_x^3\partial_y^3 w(x,y,z)\sin(lx)\sin(my)\,\mathrm{d}x\,\mathrm{d}y. \tag{58}$$

In Eqs. (56)–(58), we stop integrating by-parts at the highest order for which we can *guarantee* vanishing boundary contributions. Eqs. (56)–(58) imply that the dynamic variables conform to their respective trigonometric forms up to third or fourth order (depending on the direction). However, the relevant next-order derivatives are given by the following set of relations,

$$\begin{aligned} &\partial_x^3 v = -Ek^{-1}\partial_z w, \quad \partial_x^3 w = Ek^{-1}\partial_z v, \quad \text{at } x = 0, A_x, \\ &\partial_y^3 u = Ek^{-1}\partial_z w, \quad \partial_y^3 w = -Ek^{-1}\partial_z u, \quad \text{at } y = 0, A_y, \end{aligned} \tag{59}$$

where Ek is the Ekman number as defined in Eq. (8). The fourth-order boundary conditions follow in a similar manner. Normally, we would expect to be able to integrate by-parts one more time for each direction in each of Eqs. (56)–(58) and thereby obtain $U_{l,m} = \mathcal{O}(l^{-5}m^{-4})$, $V_{l,m} = \mathcal{O}(l^{-4}m^{-5})$, and $W_{l,m} = \mathcal{O}(l^{-4}m^{-4})$, rather than the current asymptotic rates of $\mathcal{O}(l^{-4}m^{-3})$, $\mathcal{O}(l^{-3}m^{-4})$, and $\mathcal{O}(l^{-3}m^{-3})$ respectively. However, Fig. 6 seems to imply that we cannot integrate by-parts one more full order. However, Fig. 6 also seems to indicate that we should not be content with the current degree of convergence.

To uncover the resolution to this problem, we note that integrating by-parts one additional order in Eqs. (56)–(58) would produce integrals of, say, fourth-order derivatives of $w(x,y,z)$, *i.e.*, $\partial_x^4\partial_y^4 w$. However, a close inspection of Eq. (59) suggests that the higher-order boundary conditions can potentially produce weak corner singularities in the higher-order derivatives. These corner singularities can render higher-order derivatives non-integrable and therefore disallow any further by-parts integration [10]. To see how the extra fractional rate of convergence arises, we note that while corner singularities perhaps render higher-order derivatives non-integrable, these types of singularities are usually weak and the corresponding functions are usually "*almost*" differentiable by a full order. When we say "*almost*" differentiable, we mean that the functions are typically at least Lipschitz of degree $0 < \lambda < 1$. That is, a given function, $f(\boldsymbol{x})$, (where $f(\boldsymbol{x})$ can represent any of the given dynamical functions in Eqs. (56)–(58)), is called Lipschitz of degree λ if $|f(\boldsymbol{x}+\boldsymbol{x}_0) - f(\boldsymbol{x})| \leqslant C_L|\boldsymbol{x}_0|^\lambda$, for some constant, C_L. Specifically, if $\partial_x^4\partial_y^4 w(x,y,z)$ is a Lipschitz function of degree $0 < \lambda < 1$, then we can extend the asymptotic order of $W_{l,m}$ to $\mathcal{O}(l^{-(3+\lambda)}m^{-(3+\lambda)})$ (see [18], Section 1.4), and we can produce a similar extension for the amplitudes, $U_{l,m}$, and $V_{l,m}$.

When we now consider the vector magnitude of the velocity-spectrum coefficients, we have

$$|\hat{\boldsymbol{u}}_{\boldsymbol{k}}|^2 = U_{l,m}^2 + V_{l,m}^2 + W_{l,m}^2 = \mathcal{O}(|\boldsymbol{k}|^{-2(3+\lambda)}), \tag{60}$$

where (in this case) $\boldsymbol{k} = (l,m)$. Thus, the squared Cauchy error (*cf.* Fig. 6) is given by,

$$\|\boldsymbol{u}_n - \boldsymbol{u}_{2n}\|_{L^2}^2 = \sum_{|\boldsymbol{k}|=n+1}^{2n} |\hat{\boldsymbol{u}}_{\boldsymbol{k}}|^2 = \mathcal{O}(n^{-(2(3+\lambda)-1)}) \tag{61}$$

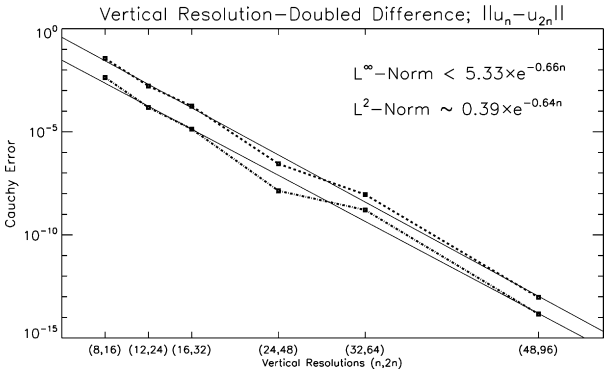

Fig. 7. Log-linear plots of vertical convergence tests with two different norms. The details of this plot are similar to Fig. 6, however, in this case, the horizontal resolution is held fixed at $n_x = n_y = 32$. As in Fig. 6, the dashed lines show the computed differences, while the solid lines show the least-squares fit.

and (after taking the square root) the Cauchy error itself is $\|\boldsymbol{u}_n - \boldsymbol{u}_{2n}\|_{L^2} = \mathcal{O}(n^{-(5/2+\lambda)})$. A similar argument gives $\|\hat{\boldsymbol{u}}\|_{l^1} = \sum_{|\boldsymbol{k}|=n+1}^{2n} |\hat{\boldsymbol{u}}_{\boldsymbol{k}}| = \mathcal{O}(n^{-(2+\lambda)})$.

Our numerical convergence tests yield $\|\boldsymbol{u}_n - \boldsymbol{u}_{2n}\|_{L^2} = \mathcal{O}(n^{-3.35})$ and $\|\hat{\boldsymbol{u}}_n - \hat{\boldsymbol{u}}_{2n}\|_{l^1} = \mathcal{O}(n^{-2.81})$. These scalings appear to imply a Lipschitz degree of $\lambda \simeq 0.81$–0.85; which is not quite unity but is reasonably close as to imply almost one additional degree of differentiability.

For comparison to the horizontal scaling tests, we also perform a resolution scaling test on the vertical resolution. In this case, we fix the horizontal resolution at $n_x = n_y = 32$ and run a calculation for each of the vertical resolutions, $n_z = 8, 12, 16, 24, 32, 64, 96$. The results of these tests are shown in Fig. 7 on a log-linear scale. In this case, the mostly linear trend implies an exponential form of convergence. This exponential convergence results form the form of the dynamical equations in the vertical direction. That is, if we perform integration-by-parts on the spectral coefficients as we do in (56)–(58), we would find that the dynamical equations imply that the boundary contributions vanish for all orders of integration. This implies exponential convergence in the vertical direction even though we are not using a full Fourier series in that direction, *i.e.*, we are using half-period Fourier sine and cosine series in the z-direction.

5. Discussion

At this point, it is worthwhile to highlight how parity mixing arises in the current problem and others. In general, boundary conditions are ultimately the cause of parity mixing. In the current problem, the diffusion operators determine the differential order of the evolution equations and therefore necessitate dynamic boundary conditions such as stress free or no slip. Hypothetically, if we were to consider the Boussinesq equations in the inviscid limit, then we must drop two boundary conditions in each direction. The boundary conditions that we must keep are those of impenetrability, and hence we would drop any extra stress-free or no-slip conditions. Therefore, if we still wish to employ a trigonometric representation, then we can still represent the normal velocity components with sine functions in the normal directions. With this choice, the parity of the remaining directions is automatically set by the dynamical equations, not by boundary conditions. While there still can be parity mixing, the boundary conditions are no longer trying to enforce any *incompatible* parity.

This lack of competition between the boundary conditions and the dynamical equations implies that, in the situation of a small (but nonzero) diffusion, variables of differing parity could potentially take their inviscid forms (relative phases) in the interior of the domain with small boundary layers that allow the higher-order derivatives to match onto their proper values at the boundaries. It is not impossible to solve for the exact form of the boundary layers in the current system. Given that the Ekman number, Ek, is the ratio of viscous to Coriolis forces, the most useful piece of information that comes from such an analysis is that, in the horizontal directions, the minimum boundary layer thickness is $\mathcal{O}(Ek^{1/3})$ which is generally wider than the $\mathcal{O}(Ek^{1/2})$ Ekman layer on the top and bottom boundaries; see [19]. These Ekman number scalings are fortunate since

the vertical direction converges at an exponential rate, and the horizontal directions converge as a power-law. Nevertheless, as we have seen in our numerical tests, seemingly small changes in how boundary conditions are treated can have dramatic consequences for the bulk dynamics of a system, *e.g.*, precession versus not. In general, there really is no sound notion of a system's independence from its boundaries.

Recall that to compute the projection of a cosine-represented function onto a sine basis and vice versa, correctly, we first need to multiply by Id_N, where N is usually a large number. Even though Id_N is a smooth function for any finite value of N, it still has a rapid change near the boundary. Based on the required size of N, this field's transition near the boundary is sharp enough that it would appear poorly resolved (*i.e.*, it would possess noticeable Gibbs phenomena) if it were an actual feature of the flow that had arisen via some hydrodynamic process. Of course, Id_N is a convenient numerical construction (not physical). We only care if it causes ringing in the *actual* flow. Though it may seem unpleasant that Id_N contains Gibbs phenomena, from the discussion in Part I, it is a necessity for a accurate result given our choice of scheme, and the situation would be worse without it (compare Figs. 2b and 3d). Ringing or not, Id_N is a numerical trick that exists to cancel out aliasing errors from an infinite tail and is exactly what is needed to ensure proper energy redistribution from Coriolis accelerations in a pseudospectral framework.

While parity mixing (or equivalently, the numerical interaction with Id_N) scatters energy to various scales, diffusion withdraws energy at relatively small scales. If diffusion can act at a sufficient rate, the velocity field will remain smooth (or at least as smooth as the true physics dictate). Of course the rate of diffusion is controlled by the Ekman number, *Ek*, *i.e.*, the size of the (previously mentioned) boundary layers (where usually, most dissipation takes place), is controlled by the size of the Ekman number. It, therefore, appears that resolving boundary layers at least partially dictates the success or failure of parity filtering. Of course, for hydrodynamics in general, any numerical scheme's practicality will in some way hinge on its ability to resolve boundary layers properly.

At this point, some may be wondering, if boundary layers are a major issue, then why bother with parity-mixed trigonometric functions in the first place? Why not use some technique like nonuniform finite-differences or Chebyshev polynomials? After all, in a strict sense, parity mixing is not physical, it is just an artifact of applying trigonometric representations to functions whose boundary conditions advocate it. The basic answer to these questions is computational speed. Recall that parity filtering is nothing more than representing a sine function with a cosine series or vice versa. Eq. (1) can be used for this purpose, and this is nothing new. However, what our new method provides is *fast* parity filtering from the utilization of the FFT for trigonometric functions.

Chebyshev polynomials are often good candidates when boundary layers are involved. Furthermore, in one dimension, derivative operations with Chebyshev polynomials or finite-differences are not a serious drawback. There are many fast schemes (*i.e.*, $\mathcal{O}(N)$) for computing forward one-dimensional derivatives and taking inverses. However, if Chebyshev polynomials are simultaneously employed in more than one direction, then derivative operations can become expensive. In one dimension, derivative operations can often utilize $\mathcal{O}(N)$ recursion relations, but in multiple dimensions the $\mathcal{O}(N)$ recursion relations couple across multiple directions and therefore require the inversion of dense matrices (see [20] for a discussion of and partial solution to these issues). In many situations, it is wise to allow only one direction to be represented without trigonometric functions whenever possible. Since, in the current problem, we have parity mixing in both the x and y directions, if we used Chebyshev polynomials in one of these directions, we should probably do the same for the other direction. This would make for an expensive approach. Moreover, in a timestepping solver, employing a Chebyshev basis in the spatial directions results in a timestep constraint that is $\mathcal{O}(N^{-2})$ for each derivative order, rather than $\mathcal{O}(N^{-1})$ in our current situation. That is, even though we are multiplying by a rapidly changing function near the boundaries (*i.e.*, Id), this does not alter the grid spacing and thus does not alter the timestepping criteria.

As a final comment, we must mention the horizontal convergence rate in Fig. 6. Based on theoretical analysis, the scaling exponent for the L^2-norm should be $5/2 + \lambda$, while the exponent for the l^1-norm should be $2 + \lambda$, where λ is the degree of Lipschitz continuity of the high-order derivatives. We measure values that imply $\lambda \simeq 0.85$. Nevertheless, while a third or fourth-order power-law scaling is not exponential (as in the vertical directions), it is still quite adequate for this problem. This scaling is arguably about as good as a fourth-order-finite-difference scheme. Greater than third-order convergence implies that there are no Gibbs phenom-

ena in any of the higher-order terms in our dynamical equations. Furthermore, Gegenbauer techniques are an albeit expensive method for eliminating Gibbs ringing (*i.e.*, globally obtaining an exponentially accurate representation) solely from the Fourier coefficients; see [21]. These techniques ($\mathcal{O}(N^2)$ in each direction) could be used in two ways. The first would be to perform post-processing on the solution amplitudes if we are interested in presenting an accurate representation of higher than third-order derivatives. Secondly, we could filter the solution in grid space at each time step to compute the nonlinear terms more accurately. For the current problem, this type of filtering is both unnecessary and expensive. Away from boundaries, some types of inexpensive Fourier space filters may help convergence; see [21]. Nevertheless, Gegenbauer filtering would provide for an exponentially-accurate calculation *of the Fourier amplitudes* which would decay roughly at fourth order. Further post-processing would then provide an exponentially accurate solution in physical space. In the current context, expensive filtering provides little advantage over Chebyshev methods which are already expensive and highly accurate. Therefore, we believe that high-resolution, parity-filtered trigonometric calculations provide a good balance between high accuracy and high computational speed.

6. Conclusion

We have developed a code that applies the numerical methods from Part I to the problem of rapidly rotating and high-Rayleigh-number convection in a confined box. The above tests demonstrate that we have developed an effective and practical procedure for computing subtle parity-mixed terms in evolution equations that can have dramatic consequences for overall dynamics. Furthermore, we believe that the current methodology allows for a wider range of use of many existing numerical pseudospectral codes. With only a relatively modest amount of modification in Fourier transform types, and to incorporate the proper multiplications by the spectral expansion of unity, a wider range of geometries can be investigated.

While we carried out tests only for limited and moderate parameter values, our current results do elucidate some of the observed differences between past numerical experiments performed in periodic domains and what is seen in many laboratory experiments (compare [7,2]). In an upcoming paper, we discuss detailed physics and mathematics that explain exactly how and why precessing convection arises in confined domains. We also employ our numerical code to survey a much larger and more supercritical region of parameter space of rotating Boussinesq convection.

Acknowledgments

G.M.V. and N.H.B. were supported by NASA Sun-Earth Connections Division Grant No. NNG04GB86G. K.J. was supported by NASA Award NNG05GD37G, and University of Colorado SEED Grant. The volume rendering imagery was produced by VAPOR, a product of the National Center for Atmospheric Research (www.vapor.ucar.edu). G.M.V. thanks John Hart for originally suggesting the study of confined rotating convection and for his general insight. G.M.V. also thanks Michael Proctor for lively discussions regarding rotating fluids and Hilbert space methods. All of the author gratefully acknowledge an anonymous referee for very helpful comments.

References

[1] C. Canuto, M. Hussaini, A. Quateroni, T. Zang, Spectral Methods in Fluid Dynamics, Springer-Verlag, New York, 1988.
[2] K. Julien, S. Legg, J. McWilliams, J. Werne, Rapidly rotating turbulent Rayleigh–Bénard convection, J. Fluid Mech. 322 (1996) 243–273.
[3] N. Brummell, N. Hurlburt, J. Toomre, Turbulent compressible convection with rotation. II. Mean flows and differential rotation, Astrophys. J. 493 (2) (1998) 955–969.
[4] P. Gilman, Nonlinear dynamics of Boussinesq convection in a deep rotating spherical shell, Geophys. Astrophys. Fluid Dyn. 8 (1977) 93–135.
[5] R. Verzicco, R. Camussi, Numerical experiments on strongly turbulent thermal convection in a slender cylindrical cell, J. Fluid Mech. 477 (2003) 19–49.
[6] Y. Liu, R. Ecke, Nonlinear traveling waves in rotating Rayleigh–Bénard convection: stability boundaries and phase diffusion, Phys. Rev. E 59 (4) (1999) 4091–4105.

[7] J. Hart, S. Kittelman, D. Ohlsen, Mean flow precession and temperature probability density functions in turbulent rotating convection, Phys. Fluids 14 (3) (2002) 955–962.
[8] P. Vorobieff, R. Ecke, Turbulent rotating convection: an experimental study, J. Fluid Mech. 322 458 (2002) 191–218.
[9] G. Vasil, N. Brummell, K. Julien, A new method for fast transforms in parity-mixed PDEs: Part I. Numerical techniques and analysis, J. Comput. Phys. 227 (17) (2008) 7999–8016.
[10] J. Boyd, Chebyshev and Fourier Spectral Methods, second ed. (revised)., Dover Publications, New York, 2001.
[11] S. Chandrasekhar, Hydrodynamic and Hydromagnetic Stability, Oxford University Press, London, 1961.
[12] S. Tobias, N. Brummell, T. Clune, J. Toomre, Transport and storage of magnetic field by overshooting turbulent compressible convection, Astrophys. J. 549 (2) (2001) 1183–1203.
[13] P. Livermore, An implementation of the exponential time differencing scheme to the magnetohydrodynamic equations in a spherical shell, J. Comput. Phys. 220 (2) (2007) 824–838.
[14] M.Frigo, Frigo, S.Johnson, Johnson, The design and implementation of FFTW3, Proc.IEEE 92 (2) (2005) 216–231 (Special issue on Program Generation, Optimization, and Platform Adaption).
[15] E. Knobloch, Rotating convection: recent developments, Int. J. Eng. Sci. 36 (1998) 1421–1450.
[16] D. Ohlsen, J. Hart, S. Kittelman, Laboratory experiments on rotating turbulent convection, in: Proc. Am. Meteorolog. Soc. Tenth Conference on Atmospheric and Oceanic Waves and Stability, 1995, p. 255.
[17] E. Lieb, M. Loss, Analysis, American Mathematical Society, Providence, 1997.
[18] Y. Katznelson, An Introduction to Harmonic Analysis, Cambridge University Press, Cambridge, 2004.
[19] H. Greenspan, The Theory of Rotating Fluids, Cambridge, London, 1969.
[20] K. Julien, M. Watson, Efficient multi-dimensional solution of PDEs using Chebyshev spectral methods, J. Comput. Phys., submitted for publication.
[21] D. Gottlieb, C. Shu, On the Gibbs phenomenon and its resolution, SIAM Rev. 39 (4) (1997) 644–668.
[22] J. Clyne, M. Rast, A prototype discovery environment for analyzing and visualizing terascale turbulent fluid flow simulations, Proc. Visual. Data Anal. (2005) 284–294.