

The EPIC atmospheric model with an isentropic/terrain-following hybrid vertical coordinate

Timothy E. Dowling^{a,*}, Mary E. Bradley^b, Edward Colón^c, John Kramer^a, Raymond P. LeBeau^d, Grace C.H. Lee^a, Timothy I. Mattox^e, Raul Morales-Juberías^f, Csaba J. Palotai^a, Vimal K. Parimi^d, Adam P. Showman^g

^a Comparative Planetology Laboratory, University of Louisville, 211 Sackett Hall, Louisville, KY 40292, USA

^b Department of Mathematics, University of Louisville, Louisville, KY 40292, USA

^c Network Computing Services, Adelphi, MD 20783, USA

^d Department of Mechanical Engineering, University of Kentucky, Lexington, KY 40506, USA

^e Department of Electrical Engineering, University of Kentucky, Lexington, KY 40506, USA

^f Department of Physics, New Mexico Tech., Socorro, NM 87801, USA

^g Lunar & Planetary Laboratory, Department of Planetary Sciences, University of Arizona, Tucson, AZ 85721, USA

Received 13 September 2005; revised 27 December 2005

Available online 17 February 2006

Abstract

The explicit planetary isentropic coordinate (EPIC) atmospheric model has been upgraded to use a hybrid vertical coordinate, ζ , that transitions continuously from potential temperature, θ , aloft to a function of a pressure coordinate, σ , that is terrain following near topography. The result is a model that simulates terrestrial and gas-giant atmospheres equally well. Considering that surface pressure varies by orders of magnitude from one planet to the next whereas topography has a roughly similar geometric scale everywhere, we define σ in terms of $\log p$ rather than the traditional p . We include a pure-sigma region at the bottom that allows for accurate modeling of the planetary boundary layer (PBL) for terrestrial applications and the deep atmosphere for gas-giant applications. We describe the functional form for $\zeta(\theta, \sigma)$, the method used to calculate θ , and the method used to calculate the hybrid vertical velocity, $\dot{\zeta}$, all of which are new. Potential temperature is only predicted in the pure-sigma region while in the hybrid region it is found diagnostically; in a complementary manner, pressure is predicted in the hybrid region and at the surface but found diagnostically in the pure-sigma region. The hybrid vertical velocity, $\dot{\zeta}$, is calculated directly near the beginning of each timestep rather than iteratively at the end. A brief description of the model's new turbulence scheme is included. To compare with previous models and to illustrate the flexibility of the hybrid coordinate, we run the Held–Suarez benchmark for Earth and a published Great Dark Spot simulation for Neptune.

© 2006 Elsevier Inc. All rights reserved.

Keywords: Atmospheres, dynamics; Computer techniques

1. Introduction

If atmospheric dynamics is a difficult subject it is also one for which we have fully one dozen different case studies in our Solar System alone: the Sun, Venus, Earth, Mars, Jupiter, Io, Saturn, Titan, Uranus, Neptune, Triton, and Pluto, not to mention extrasolar planets. One of the notable discoveries of Solar System exploration has been the large variation in weather from one planet to the next. Jet streams—their strength, direction,

shape, and number—are a well-documented feature for which only partial explanations have been found, and the lifecycles of vortices is another. Consider the intricate and pervasive role of water in Earth's meteorology—does water play a critical role in Jupiter's atmospheric dynamics? Does methane in Titan's? How do parameterizations of subgrid-scale processes, which are prone to tuning to a particular planet, usually Earth, affect the simulation of different atmospheres, especially in sensitive regions like planetary boundary layers? We do not yet know the answers to these questions. However, such inquiries help to guide us toward a major goal in comparative planetology: to develop a single model that accurately simulates all known

* Corresponding author. Fax: +1 502 852 6053.

E-mail address: dowling@louisville.edu (T.E. Dowling).

atmospheres. This motivation drives the development of the EPIC atmospheric model and here we describe the latest step in its evolution. The model is now able to treat both terrestrial atmospheres, where the bottom boundary is a solid or liquid surface, and gas-giant atmospheres, where the bottom boundary is a gaseous interior, in an evenhanded manner. The key is the vertical coordinate; the EPIC model now employs a hybrid, ζ , that smoothly combines the advantages of an isentropic coordinate aloft with a pressure coordinate at depth that is terrain following in the presence of topography.

This paper is organized as follows. Section 2 describes the model's new hybrid vertical coordinate, including the functional form for $\zeta(\theta, p)$, the calculation of θ , and the calculation of the hybrid vertical velocity, $\dot{\zeta}$. Section 3 illustrates the flexibility of the new model with two sample simulations, a standard terrestrial benchmark (Held and Suarez, 1994) and a Neptune vortex-drift simulation (LeBeau and Dowling, 1998). Section 4 points out those algorithms that have been retired since the model's first exposition (Dowling et al., 1998), Section 5 introduces significant components that has been added, and Section 6 concludes with a brief discussion about the hybrid-coordinate approach.

2. Design of the hybrid vertical coordinate

Our starting point is the atmospheric model developed by Konor and Arakawa (1997, hereafter KA97), who introduced a hybrid vertical coordinate, ζ , in the form

$$\zeta = \tilde{f}(\sigma) + \tilde{g}(\sigma)\theta, \quad (2.1)$$

where θ is the potential temperature and σ is a terrain-following pressure variable. There are viable alternatives, for example the vertical-coordinate-table scheme described by Zhu et al. (1992), but (2.1) is particularly easy to invert into $\theta = \theta(\sigma, \zeta)$ and we take advantage of this below. In this section, equations that are used in the EPIC model are marked with bold equation numbers. The traditional definition of σ is $\sigma = p/p_{\text{bot}}$, but for mathematical convenience KA97 use the definition $\sigma_{\text{KA}} = (p - p_{\text{bot}})/(p_{\text{top}} - p_{\text{bot}})$, which ranges from 0 at the surface to 1 at the top of the model. In place of these, we favor a definition that makes σ roughly proportional to geometrical distance, the point being that the mountains on Venus, Earth, and Mars are all similar in scale when measured in kilometers, whereas their surface pressures are on the order of 10^5 , 10^3 , and 10^1 mbar ($1 \text{ mbar} \equiv 1 \text{ hPa}$), respectively. Thus, with the mild restriction $p_{\text{top}} > 0$, we define σ to be proportional to the natural logarithm of pressure

$$\sigma \equiv \frac{\log(p/p_{\text{bot}})}{\log(p_{\text{top}}/p_{\text{bot}})}, \quad (2.2)$$

which also has the benefit of making the σ coordinate for shallow and deep gas-giant models more uniform.

Many of the details of our present hybrid model have evolved to be different than in KA97, and so we describe below each change and the circumstances that led us to the algorithms we currently use. Regarding notation, we write the functions \tilde{f}

and \tilde{g} with tildes to avoid any confusion with the Coriolis parameter and gravity, respectively, and ζ in (2.1) is not to be confused with relative vorticity. In the EPIC source code we refer to the hybrid coordinate as `sigmatheta`, and sometimes abbreviate it `sgth`.

2.1. The functions $\tilde{f}(\sigma)$ and $\tilde{g}(\sigma)$

The idea of (2.1) is that at the bottom of an atmospheric model, where θ typically ceases to be a viable vertical coordinate, ζ becomes a function of σ . Once away from the bottom, the coordinate transitions into θ itself, thereby gaining the motion-following advantages of isentropic coordinates (Hsu and Arakawa, 1990; Schaack et al., 2004). In addition, Konor and Arakawa point out that one can effectively tilt ζ by a small amount, for example 3 K over the vertical span of the model, to handle unstable regions (superadiabatic temperature lapse-rate regions, $\partial\theta/\partial\sigma < 0$) and thereby maintain ζ as a monotonically increasing vertical coordinate even when there exist neutrally stable or slightly unstable regions of the atmosphere. The boundary conditions on $\tilde{f}(\sigma)$ and $\tilde{g}(\sigma)$ corresponding to (2.1) are

$$\begin{aligned} \tilde{f}(1) &= 0, & \tilde{g}(1) &= 1, \\ \tilde{f}(0) &= \zeta_0, & \tilde{g}(0) &= 0, \end{aligned} \quad (2.3)$$

where ζ_0 is a free parameter corresponding to the value of the coordinate at the bottom of the model.

To ensure that ζ is monotonic, KA97 find $\tilde{f}(\sigma)$ by solving the ordinary differential equation (their 2.37)

$$\frac{d\tilde{f}}{d\sigma} + \frac{d\tilde{g}}{d\sigma}\theta_{\min} + \tilde{g}\left(\frac{d\theta}{d\sigma}\right)_{\min} = 0. \quad (2.4)$$

They give as an example $(d\theta/d\sigma)_{\min} = -3 \text{ K}$ and $\theta_{\min} = 250 \text{ K}$. If both these parameters are constants, then (2.4) integrates to yield

$$\tilde{f}(\sigma) = \theta_{\min} - \tilde{g}(\sigma)\theta_{\min} + \left(\frac{d\theta}{d\sigma}\right)_{\min} \int_{\sigma}^1 \tilde{g} d\sigma, \quad (2.5)$$

such that the hybrid coordinate becomes

$$\zeta(\theta, \sigma) = \theta_{\min} + \tilde{g}(\sigma)(\theta - \theta_{\min}) + \left(\frac{d\theta}{d\sigma}\right)_{\min} \int_{\sigma}^1 \tilde{g} d\sigma. \quad (2.6)$$

The term involving the integral of \tilde{g} provides the tilt that overcomes any $(d\theta/d\sigma)_{\min} \leq d\theta/d\sigma < 0$ regions. With this formulation, KA97 “have reduced the problem of obtaining a hybrid σ - θ vertical coordinate to a selection of $g(\sigma)$.”

We have discovered a problem with having $\tilde{f}(\sigma)$ be linked to $\tilde{g}(\sigma)$ in this manner, however. Consider the case when $\tilde{g}(\sigma)$ is exactly zero for some range of σ near the surface, say 0 to σ_{σ} . The expectation from (2.1) is that $\tilde{f}(\sigma)$ will take over and provide ζ with σ dependence, like in a pure σ -coordinate model. However, it turns out there is no “ σ ” left in $\tilde{f}(\sigma)$ when it is linked in this manner, rather

$$\tilde{g}(\sigma) = 0, \quad \sigma \leq \sigma_{\sigma};$$

$$\Rightarrow \tilde{f}(\sigma) = \theta_{\min} + \left(\frac{d\theta}{d\sigma} \right)_{\min} \int_{\sigma_{\sigma}}^1 \tilde{g} d\sigma = \text{constant},$$

$$\sigma \leq \sigma_{\sigma}. \quad (2.7)$$

In fact, much of the σ dependence in ζ near the surface comes from the $\tilde{g}(\sigma)\theta$ term rather than the $\tilde{f}(\sigma)$ term in the KA97 formulation. This is at most a minor weakness because there is a wide spectrum of choices of $\tilde{g}(\sigma)$ that work well with (2.4), (2.5). Nevertheless, we have changed our definition of $\tilde{f}(\sigma)$ to be

$$\tilde{f}(\sigma) = (1 - \tilde{g}(\sigma))(\zeta_0 + (\zeta_1 - \zeta_0)\sigma), \quad (2.8)$$

where the parameters ζ_0 and ζ_1 are two constants that are chosen appropriately such that ζ is monotonic and reasonable; they supplant the two parameters θ_{\min} and $(d\theta/d\sigma)_{\min}$. In the code they are `grid.zeta0` and `grid.zeta1` and are set in `epic_initial.c`.

For $\tilde{g}(\sigma)$, we use an exponential profile similar to KA97, except that we build in a pure-sigma region at the bottom and use the definition of σ given by (2.2). We denote the corresponding values of σ , ζ , and the pressure at the top of the pure-sigma region with the subscript “ σ ,” such that σ_{σ} , ζ_{σ} , and p_{σ} refer to this transition level [similar notation is employed by Schaack et al. (2004)]. Likewise, we let k_{σ} be the vertical index of the layer that has this transition level as its top interface, in other words, the first layer inside the pure- σ region counting downward. Our current function for $\tilde{g}(\sigma)$ is

$$\tilde{g}(\sigma) = \begin{cases} 0, & 0 \leq \sigma \leq \sigma_{\sigma}, \\ (1 - e^{-\alpha(\sigma - \sigma_{\sigma})}) / (1 - e^{-\alpha(1 - \sigma_{\sigma})}), & \sigma_{\sigma} < \sigma \leq 1, \end{cases} \quad (2.9)$$

with a default value of $\alpha = 50$, which gives our $g(\sigma)$ a similar shape to KA97’s $g_{KA}(\sigma_{KA})$ when plotted with respect to pressure, as shown in Fig. 1. This figure corresponds to the case of a surface pressure of 1000 mbar, whereas for mountainous regions the surface pressure and the corresponding pressure at the top of the pure-sigma region, p_{σ} , would be substantially less. Note that although p_{σ} varies with topography, σ_{σ} and ζ_{σ} are constants in the model. In any particular application, one should pick values of σ_{σ} and α that ensure that the pure-sigma portion of the model contains the region in which potential temperature can realistically be mixed by planetary boundary layer (PBL) processes. An illustration of how ζ departs from θ for typical models of Earth and Jupiter is shown in Fig. 2.

2.2. Calculation of potential temperature and pressure

In a pure-isentropic coordinate model one does not predict the potential temperature, θ , it is the coordinate itself and so is a given constant in each layer. This changes with the hybrid approach because θ can vary on ζ surfaces. We have come to appreciate that there is more to the handling of θ than simply adding in the appropriate prognostic equation; we find the most subtle aspect of the hybrid approach is that one is tempted to treat θ as both a prognostic variable and a diagnostic variable at the same time.

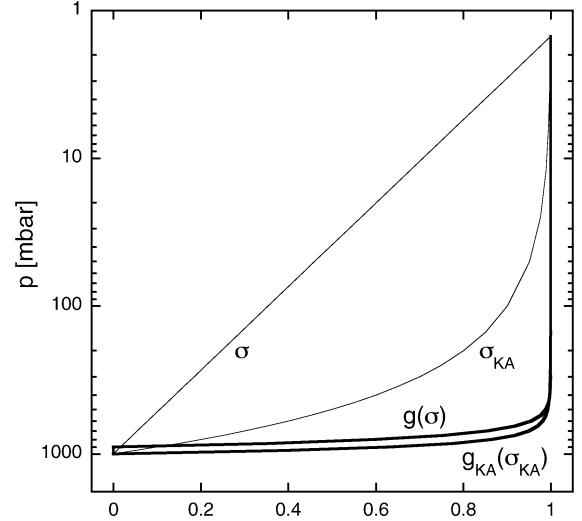


Fig. 1. The functions σ and $\tilde{g}(\sigma)$ vs pressure, p , used to define the EPIC model hybrid vertical coordinate, ζ . The case shown corresponds to Earth with $p_{\text{top}} = 1.5$ mbar, $p_{\sigma} = 900$ mbar, $p_{\text{bot}} = 1000$ mbar, and $\alpha = 50$; see (2.1)–(2.3) and (2.9). The functions used by Konor and Arakawa (1997, KA97) are included for comparison. In this terrain-following scheme, p_{σ} is not a constant but scales with p_{bot} in the case of topography (not shown), as discussed in the text.

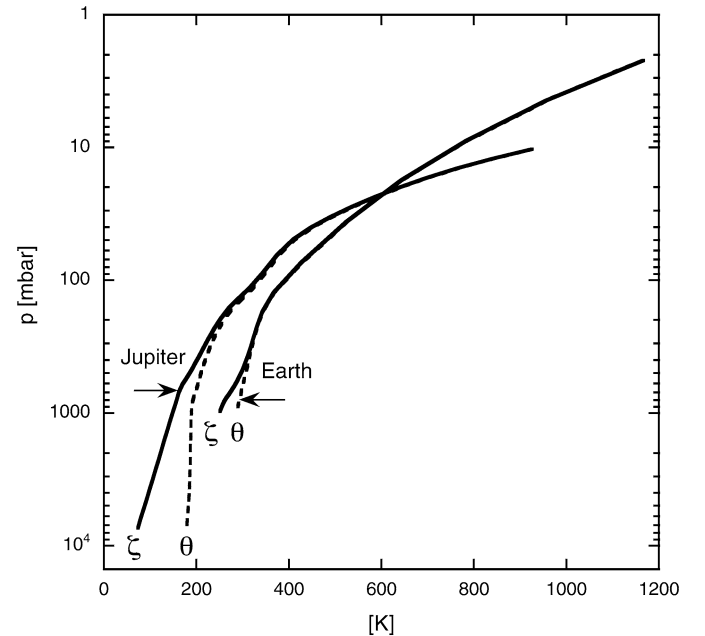


Fig. 2. Hybrid vertical coordinate, ζ , and potential temperature, θ , versus pressure, p , for typical EPIC models of Earth and Jupiter. The arrows indicate the initial pressure, p_{σ} , used to set the corresponding coordinate parameter, ζ_{σ} . For altitudes at and below this value ($\zeta \leq \zeta_{\sigma}$) the vertical coordinate is a function of σ only. Above, it is a function of σ and θ that asymptotes to θ at a rate controlled by the parameter α in (2.9), which here is set to 10 for both planets.

When θ is treated as a prognostic variable, its tendency may be connected directly to the heating rate, \dot{Q} , by

$$\frac{d\theta}{dt} \equiv \dot{\theta} = \frac{\dot{Q}}{\Pi}, \quad (2.10)$$

where $\Pi = C_p^r T/\theta$ is the Exner function.¹ To integrate (2.10), one supplies heating terms for the right-hand side and an advection scheme to handle the left-hand side; let us call the result θ_{prog} . Following KA97, we position both θ_{prog} and p on the layer interfaces, which is known as the Charney–Phillips grid, because this arrangement is less prone to spurious short-wave growth (Arakawa and Konor, 1996).

For a hybrid model, the story does not end with the addition of a prognostic equation for θ because the potential temperature can also be treated as a diagnostic variable given (2.1) and the pressure expressed in terms of σ

$$\theta_{\text{diag}} = \frac{\zeta - \tilde{f}(\sigma)}{\tilde{g}(\sigma)}, \quad \tilde{g}(\sigma) > 0. \quad (2.11)$$

The KA97 approach may be viewed as equivalent to finding the hybrid vertical velocity, $\dot{\zeta}$, that makes $\theta_{\text{prog}} = \theta_{\text{diag}}$ for each timestep. Since the KA97 scheme ensures self-consistency and has been successfully implemented, one might ask why we seek an alternative. One reason is that there are potential drawbacks of a practical nature to be considered. For example, the KA97 scheme requires iteration to get $\theta_{\text{prog}} \rightarrow \theta_{\text{diag}}$, but the number of iterations needed to achieve precision can be open-ended, and in addition one must delay the calculation of the vertical velocity to the last, which may hamper flexibility.

Our approach is different. We eliminate the iteration and reduce the restriction on the ordering of the calculation of $\dot{\zeta}$ by relaxing the requirement that $\theta_{\text{prog}} = \theta_{\text{diag}}$. To develop this idea, we have carried out experimentation of three different algorithms for how to calculate θ . We give only the briefest mention of the first two because we encountered numerical instabilities with each that we were ultimately not able to overcome. The first, which we call theta blending, uses the $\tilde{g}(\sigma)$ function to combine the prognostic and diagnostic values together

$$\theta = \tilde{g}\theta_{\text{diag}} + (1 - \tilde{g})\theta_{\text{prog}} = \zeta - \tilde{f} + (1 - \tilde{g})\theta_{\text{prog}}. \quad (2.12)$$

There is no division by $\tilde{g} = 0$ in (2.12) and it provides a smooth transition from the prognostic to diagnostic role played by θ . However, we were never able to get the numerics to remain stable for longer than about a month of model time using (2.12), and consequently we do not recommend theta blending. The second idea, which we call theta nudging, uses $\theta = \theta_{\text{prog}}$ everywhere, but adds to the prognostic equation a small, artificial, Newtonian-cooling-type term

$$\begin{aligned} \frac{\partial \theta_{\text{prog}}}{\partial t} &= \dots - \frac{\tilde{g}}{\tau_{\text{cool}}} (\theta_{\text{prog}} - \theta_{\text{diag}}) \\ &= \dots - \frac{1}{\tau_{\text{cool}}} (\zeta_{\text{prog}} - \zeta_{\text{diag}}), \end{aligned} \quad (2.13)$$

where $\zeta_{\text{prog}} = \tilde{f} + \tilde{g}\theta_{\text{prog}}$ and ζ_{diag} is the fixed value of the vertical coordinate for each level. Unfortunately, as with theta

blending, we were unable to find an implementation of theta nudging that avoided numerical instability, and so we were forced to move on. It would be interesting to study the numerical stability of theta blending and theta nudging analytically to determine whether or not they can be made stable.

Fortunately, our third idea proved to work extremely well and is what we have implemented. We simply use the diagnostic value of θ everywhere it is available

$$\theta = \theta_{\text{diag}}, \quad \tilde{g}(\sigma) > 0. \quad (2.14)$$

We use double precision for the terms in (2.11) to avoid roundoff problems and have not encountered any such issues. As shown in the next section, the diabatic heating, \dot{Q} , affects the model via the hybrid vertical velocity rather than through (2.10), just as it does in a pure-isentropic-coordinate model. As with any sigma model, for the sigma portion of our model where θ cannot be diagnosed, $0 \leq \sigma \leq \sigma_\sigma$, we treat it as a prognostic variable and use $\theta = \theta_{\text{prog}}$. For the layer values of θ , we use θ_{diag} in the hybrid portion and an arithmetic average of the interface values above and below (Konor and Arakawa, 1997) in the sigma portion.

Analogous to the prognostic vs diagnostic issue for potential temperature, there is a complementary issue for pressure. First, we recognize that a practical advantage of using a pressure-based coordinate in a hydrostatic model is that the model layers can never pinch together or cross, which they sometimes do when using θ as the vertical coordinate in low static-stability regions. We would like to retain as much of this advantage as possible. Second, in the pure-sigma region, $0 < \sigma \leq \sigma_\sigma$ or $\zeta_0 < \zeta \leq \zeta_\sigma$ (not including the surface), our definitions (2.2) and (2.8) yield the diagnostic value

$$p_{\text{diag}} = p_{\text{bot}} (p_{\text{top}}/p_{\text{bot}})^{\frac{\zeta - \zeta_0}{\zeta_1 - \zeta_0}}, \quad 0 < \sigma \leq \sigma_\sigma. \quad (2.15)$$

To apply p_{diag} everywhere it is available, we let the model predict the bottom pressure and the pressure everywhere outside of the pure-sigma region, but then use (2.15) to set $p = p_{\text{diag}}$ inside the pure-sigma region itself.

2.3. Hybrid vertical velocity

EPIC is still a hydrostatic model in this work, and consequently the hybrid vertical velocity, $\dot{\zeta}$, is a diagnostic variable. There is more than one way to calculate $\dot{\zeta}$ and the choice made is a distinguishing feature of a hybrid model. Konor and Arakawa (1997) proceed by taking the partial derivative of $\zeta = F(\theta, p, p_{\text{bot}})$ with respect to t , holding ζ constant, to get

$$\left. \frac{\partial F}{\partial \theta} \right|_{p, p_{\text{bot}}} \frac{\partial \theta}{\partial t} + \left. \frac{\partial F}{\partial p} \right|_{\theta, p_{\text{bot}}} \frac{\partial p}{\partial t} + \left. \frac{\partial F}{\partial p_{\text{bot}}} \right|_{\theta, p} \frac{\partial p_{\text{bot}}}{\partial t} = 0. \quad (2.16)$$

They then use the energy and mass conservation equations in Eulerian form to replace the prognostic tendencies $\partial \theta / \partial t$, $\partial p / \partial t$ and $\partial p_{\text{bot}} / \partial t$ in (2.16), and factor $\dot{\zeta}$ out from the vertical advection terms (their 2.25).

We tried Konor and Arakawa's approach early on, but ran into a few issues. First, for a given timestep, one must wait to calculate $\dot{\zeta}$ until after the contributions to the prognostic

¹ We do not write $\Pi = C_p^r (p/p_0)^\kappa$ in the EPIC model because $\theta \neq T(p_0/p)^\kappa$ for an active mixture of gases with different entropies, as is the case of the cold (50 K) *ortho-para* hydrogen mixtures in the upper tropospheres of Uranus and Neptune (Dowling et al., 1998). The method used for calculating the buoyancy frequency, N , when θ is generalized in this fashion is given in Appendix A.

tendencies from all other physical processes besides vertical advection have been included. Second, if the vertical-velocity calculation mixes variables from old and new timesteps, such as in KA97, then iteration is needed to improve accuracy; Konor and Arakawa iterate on their calculation “two or more times” to reduce such errors. As we reported at the 1st Hybrid-Isentropic Modeling Workshop held in Madison, WI, 15–16 August 2000, for this purpose we developed a globally convergent iteration loop based on a model-trust region algorithm for finding roots to nonlinear problems developed by Dennis and Schnabel (1996). We found that Dennis and Schnabel’s double-dogleg algorithm (their A6.4.4) yields reliable and economical convergence to self-consistency for this problem. But in the end, it has proven to be more flexible to pursue a scheme for calculating $\dot{\zeta}$ that does not require iteration or a rigid ordering of the calculation, and we have settled on the following noniterative scheme.

We proceed by taking the full derivative of $\zeta = F(\theta, p, p_{\text{bot}})$ with respect to t (rather than the partial derivative)

$$\dot{\zeta} = \frac{\partial F}{\partial \theta} \dot{\theta} + \frac{\partial F}{\partial p} \dot{p} + \frac{\partial F}{\partial p_{\text{bot}}} \dot{p}_{\text{bot}} \quad (2.17)$$

and then use the conservation equations in Lagrangian form (rather than Eulerian form) to replace $\dot{\theta}$, \dot{p} , and \dot{p}_{bot} , in (2.17). Application of (2.1) and (2.10) yields

$$\dot{\zeta} = \tilde{g}(\sigma) \frac{\dot{Q}}{\Pi} + F_p \dot{p} + F_{p_{\text{bot}}} \dot{p}_{\text{bot}}, \quad (2.18)$$

where the subscripts on F denote partial differentiation. We need the hybrid vertical velocity, ζ , on the layer interfaces, which means we need \dot{p} on the layer interfaces. Let $h = -(1/g)\partial p/\partial \zeta$ be the hybrid density, that is, the mass divided by the hybrid volume element, which is carried in the layers, and count layers down from the top such that the whole integer k is the index for values inside layer k , and $k + 1/2$ refers to the bottom of layer k (the interface between layers k and $k + 1$). Integrate h_k from the top of the model down in finite-difference form to determine $p_{k+1/2}$

$$p_{k+1/2} = p_{\text{top}} + \sum_{m=1}^k g h_m (\zeta_{m-1/2} - \zeta_{m+1/2}), \quad (2.19)$$

and then take the full time derivative

$$\dot{p}_{k+1/2} = \dot{p}_{\text{top}} + \sum_{m=1}^k \{ g \dot{h}_m (\zeta_{m-1/2} - \zeta_{m+1/2}) + g h_m (\dot{\zeta}_{m-1/2} - \dot{\zeta}_{m+1/2}) \}. \quad (2.20)$$

As in KA97, we henceforth assume $\dot{p}_{\text{top}} = 0$. Now, discretize the continuity equation in Lagrangian form as

$$\dot{h}_k = -h_k \left\{ D_k + \frac{\dot{\zeta}_{k-1/2} - \dot{\zeta}_{k+1/2}}{\zeta_{k-1/2} - \zeta_{k+1/2}} \right\}, \quad (2.21)$$

where $D_k \equiv (\vec{\nabla} \cdot \vec{v})|_{\zeta}$ is the horizontal divergence on ζ surfaces. Combining (2.18), (2.20), and (2.21) yields the direct formula

we use to calculate the hybrid vertical velocity:

$$\dot{\zeta}_{k+1/2} = \tilde{g}(\sigma_{k+1/2}) \frac{\dot{Q}_{k+1/2}}{\Pi_{k+1/2}} - F_p \sum_{m=1}^k g h_m D_m \Delta \zeta_m - F_{p_{\text{bot}}} \sum_{m=1}^{nk} g h_m D_m \Delta \zeta_m, \quad (2.22)$$

where $\Delta \zeta_k \equiv \zeta_{k-1/2} - \zeta_{k+1/2}$ and $k = nk$ is the bottom layer. The partial derivatives on F are evaluated at the $k + 1/2$ position by finite differencing (using machine-precision accuracy) rather than analytically, to keep from having to rework the expressions if and when F is modified. In the EPIC source code, $\Delta \zeta_k$ is `grid.dsgth[2*K]`, where the doubled index is defined such that $2*K$ and $2*K+1$ correspond to k and $k + 1/2$, respectively. This indexing convention is used often, for example `grid.lat[2*J+1]` refers to the latitude in degrees associated with u and h in the model’s staggered C-grid [illustrated in Dowling et al. (1998)].

3. Examples

Here, we illustrate the flexibility of the hybrid vertical coordinate with a standard Earth test case and a published Neptune vortex-drift case. The first initiates the model into the realm of comparative GCM studies and the second compares the model to its former self. In addition to describing the details of these two cases, we also use this section to explain some key technical aspects in context.

3.1. Earth: Held–Suarez test case

The Held and Suarez (1994) benchmark uses simple Newtonian cooling and Rayleigh drag that mimics the forcing of Earth’s atmosphere well enough to produce realistic midlatitude jets. We do not yet have similar benchmarks for Jupiter, Venus, and the other planets, but they would be valuable to develop.

For vertical resolution, the Held–Suarez paper uses 20 layers in traditional sigma coordinates, with $\sigma = 0$ ($p = 0$) formally at the top. The EPIC model gives the user the choice of specifying that the undisturbed layers be spaced even in log pressure or pressure, or that they be read from a file. If we use pressure spacing with 20 layers and a top pressure of 1 mbar, then our top layer spans approximately 1 to 50 mbar, which presents us with a problem because this factor of 50 change in pressure and the corresponding large change in the potential temperature yields top-layer values of temperature that are not sufficiently accurate. So, we use the “from file” option to add an additional 4 layers to the top, bringing the count to $nk = 24$ layers; the 25 undisturbed interfaces are placed at 1.5, 3, 6, 12, 25, 50, 100, 150, 200, 250, 300, 350, 400, 450, 500, 550, 600, 650, 700, 750, 800, 850, 900, 950, and 1000 mbar. For horizontal resolution, the highest-resolution case in the original paper is 2° latitude by 2.5° longitude, so we set $nj = 88$ and $ni = 128$ to get $d\lambda = 2.0^\circ$ and $d\phi = 2.8^\circ$ [see Dowling et al. (1998) for a discussion of the formula $d\lambda = 180^\circ/(nj + 1 + \sqrt{2})$ for a globe; ni is taken to be a power of two to facilitate the model’s high-latitude fast

Fourier transform or FFT filter]. For the hybrid-coordinate parameters we use $\zeta_0 = 260$ K and $\zeta_1 = 700$ K in (2.8) and $\alpha = 50$ and $\sigma_\sigma = 0.016$ in (2.9), such that the top of the pure-sigma portion of the model corresponds to the top of layer 23, i.e. $k_\sigma = 23$ and initially $p_\sigma = 900$ mbar. The interior meridional grid spacing is $\Delta y = 221.4$ km and the timestep is $\Delta t = 150$ s. We use the 3rd-order Adams–Bashforth timestep for the momentum equations. The eighth-order hyperviscosity parameter is set to $\nu_8 = (0.5)[(1/2400)\Delta y^8/\Delta t]$ and the divergence damping parameter is set to $\nu_{\text{div}} = (0.5)[(1/30)\Delta y^2/\Delta t]$, where the factors in square brackets are treated by the code as the dimensional units for the case of the 3rd-order Adams–Bashforth timestep (unity corresponds to the maximum numerically stable value, divided by an engineering safety factor of approximately two).

The U.S. Standard Atmosphere is used for the initial $T(p)$ profile (with the temperature set to 200 K for altitudes above 120 mbar to be consistent with the forcing). When we initialize the model with zero winds and no perturbations, it runs without breaking translational symmetry in the zonal direction

or mirror-image symmetry in the meridional direction, which demonstrates that the model code is symmetric; our longest such run is 400 days. Held and Suarez break the symmetry by adding small perturbations (unspecified) to the initial conditions. In the run shown below, we break the symmetry by adding two weak anticyclones to the otherwise at-rest initial conditions, with amplitudes of 1 m s^{-1} , major, minor, and vertical semi-axes (a, b, c) of 6° , 3° , and 2 scale heights, initially positioned at (lon. [deg], lat. [deg], p [mbar]) = (30, -33 , 500) and (-30 , 33, 500). The sponge at the top of the model is turned off. The turbulence scheme is turned on and provides convective adjustment via its Richardson-number-dependent vertical diffusion [see (5.8)]. As is prescribed, we first run the model for 200 days and then keep track of the statistics over the span from 200 to 1200 days. We extract the variables every 48 h to produce 501 samples, from which we calculate the same mean and eddy statistics as in the original paper.

The statistical measures used in the original paper (Held and Suarez, 1994, HS94) are applied to the EPIC output and

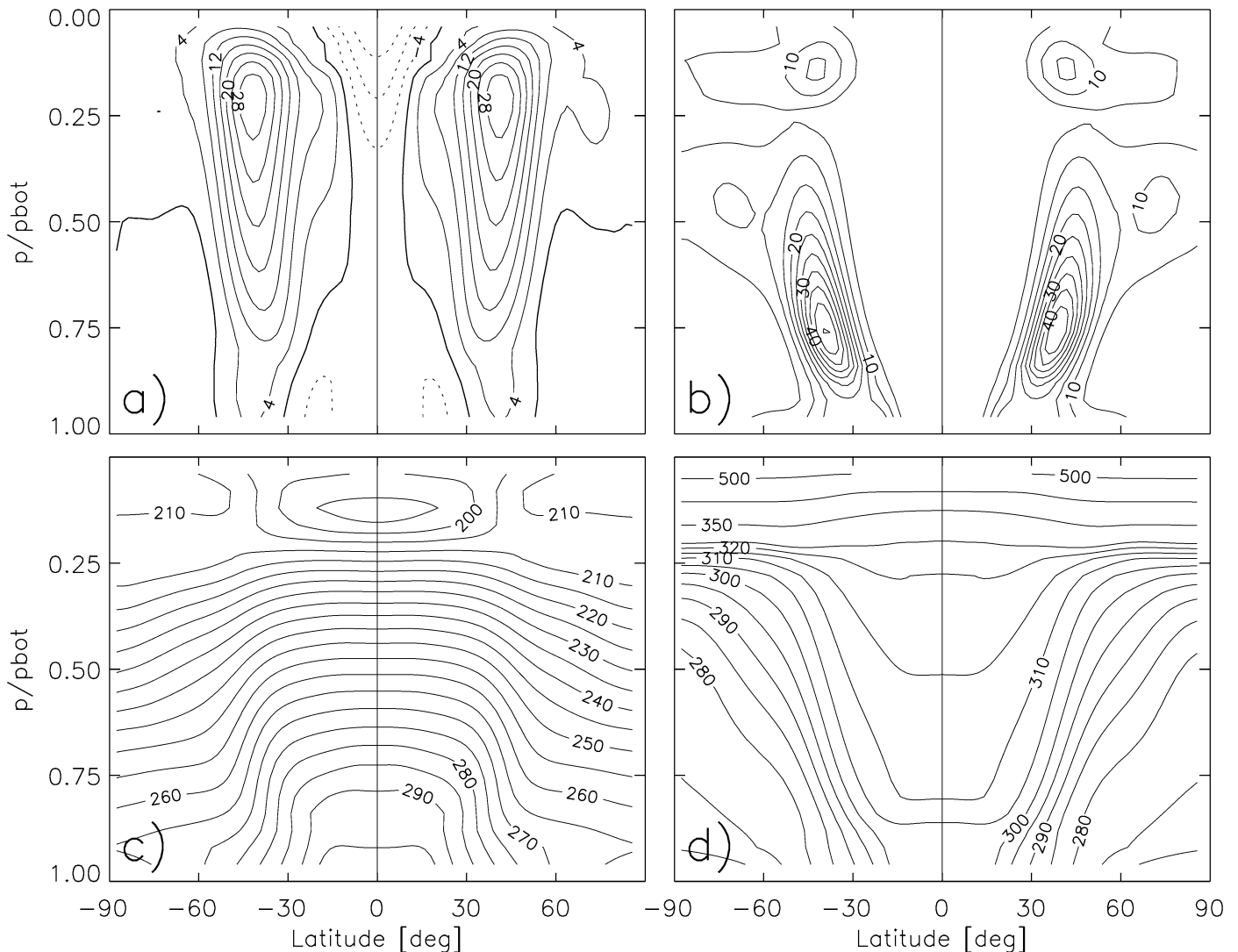


Fig. 3. EPIC model results for the Earth-atmosphere benchmark of Held and Suarez (1994, HS94). The statistics are generated by extracting variables from the run every two days from 200 to 1200 days, for a total of 501 samples. (a) Mean zonal wind [m s^{-1}] averaged over time and longitude; (b) mean of the square of the temperature eddies [K^2]; (c) mean temperature [K]; (d) mean potential temperature [K], with the contours in 5 K intervals except 325, 350, 400, and 500 K.

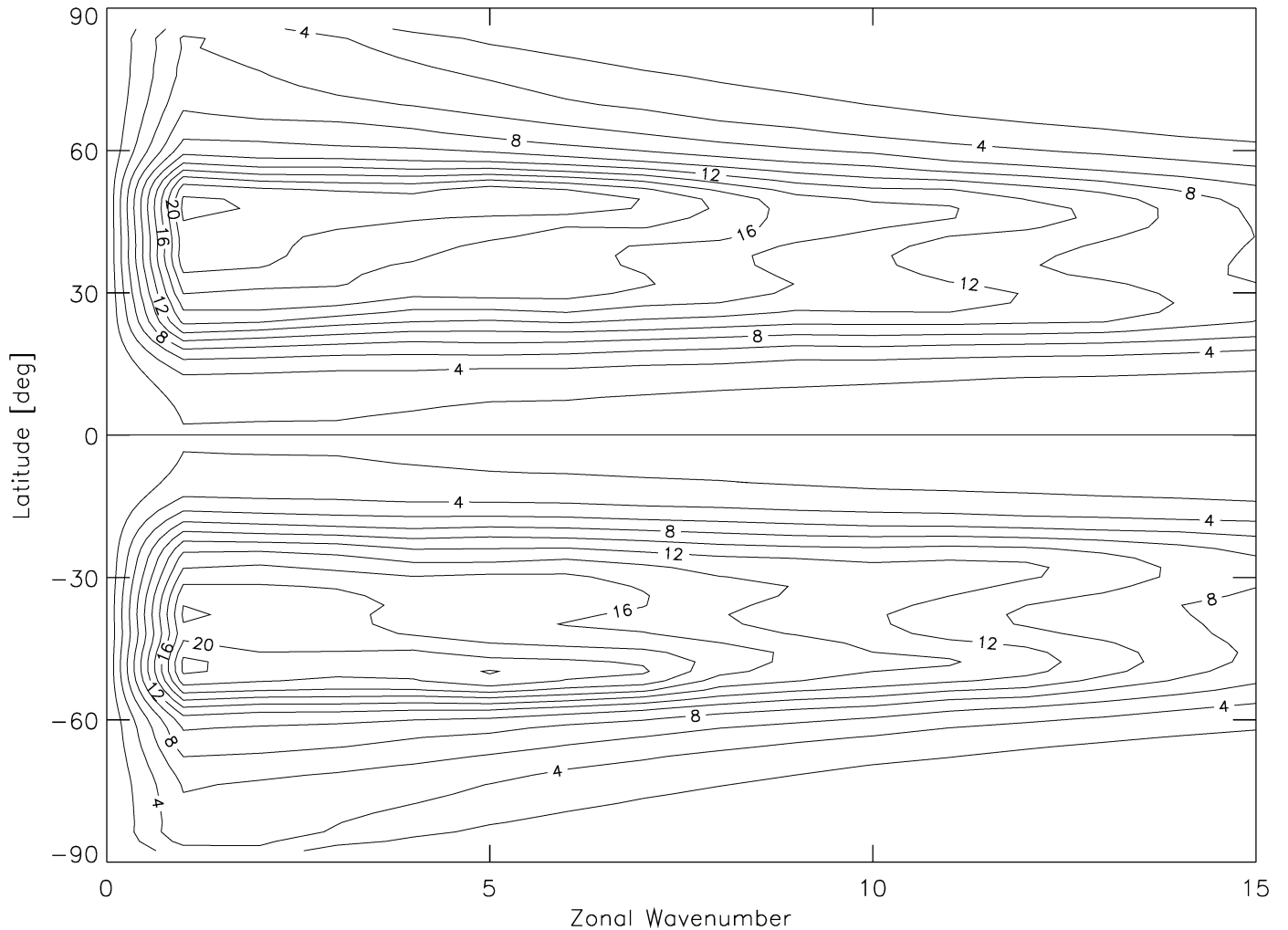


Fig. 4. As in Fig. 3, but for the zonal-wavenumber spectrum of the square of the zonal-wind eddies [$\text{m}^2 \text{s}^{-2}$], averaged over time and height.

shown in Figs. 3 and 4. To make the plots, we first interpolate our results onto traditional sigma surfaces (using IDL’s `interpol` function with `\quadratic`), which we label “p/pbot,” and then calculate the mean and eddy fields. Fig. 3a shows the zonal wind averaged over time and longitude. Midlatitude jets form as expected and the zonal-wind structure reproduce the original paper’s results in detail, a notable result given the differences in the vertical coordinates. This simulation also adds a page of history to the EPIC model because it is the model’s maiden zonal-wind spinup experiment. Significant deviations can be made to arise at the top of the model if we turn on the sponge, at the poles if we change back to the high-latitude filter used in the 1998 paper, and at the bottom if we revert to the crude convective-adjustment scheme we had been employing, all of which underscore the indispensable utility of standard benchmarks like Held–Suarez for vetting new GCMs. Fig. 3b shows the squared eddy temperature averaged over time and longitude. It matches the original as well as the zonal-wind plot. Intriguingly, Held and Suarez remark that “an unrealistic feature of the results is the penetration of the temperature variance near the surface well into the Tropics,” whereas our results do not appear to exhibit such a penetration. Figs. 3c and 3d show

the mean temperature and potential temperature, which match the original results point for point, except for a slightly smaller vertical gradient in the surface tropics. Fig. 4 shows the zonal wavenumber spectrum of the squared eddy zonal wind, averaged over height and time. The power in our model is spread out to higher wavenumbers than in the original paper, and the peaks at wavenumber 5 in the original paper are only faintly seen in our plot, but we have similar structure at wavenumbers 0, 1, and 2 and similar midlatitude storm tracks. It may be of interest to pinpoint why our eddy zonal winds have a flatter spectrum than that obtained by Held and Suarez, and in particular whether this is related to the differences in the vertical coordinate, but this is outside the scope of the present study.

3.2. Neptune: Great Dark Spot test case

For a gas-giant example we return to the Neptune Great Dark Spot (GDS) study of LeBeau and Dowling (1998) that was published in conjunction with Dowling et al. (1998). Two key results from that work are that the model GDS anticyclones drift equatorward at a rate that is proportional to the environmental potential-vorticity gradient, and the gradient needed to match the observed 1.2° per month equatorward drift of the GDS is

much less than the value associated with the standard polynomial fit to the zonal-wind data. Here, we redo LeBeau and Dowling's highest-resolution simulation, Run 90, to see what effect the hybrid coordinate might have on these experiments. This is a sensitive test because reproducing the equatorward track of the GDS with two different GCMs is in the same equivalence class as reproducing the poleward track of a hurricane with two different GCMs. To minimize the changes we use the same model configuration as in 1998 wherever possible. For completeness and to facilitate any future comparisons we include a full description of the parameters here.

The domain for the LeBeau and Dowling simulations is a quarter sphere in which latitude, λ , ranges from -90° to 0° with $n_j = 128$ and $\Delta\lambda = 0.7^\circ$, and longitude, ϕ , ranges from -90° to 90° periodically with $n_i = 256$ and $\Delta\phi = 0.7^\circ$. The original run used $\Delta t = 50$ s, but here we take advantage of improvements to the model's numerical stability as described in Section 5 below and use $\Delta t = 90$ s. To match the original we use a sixth-order hyperviscosity with $\nu_6 = 1.05 \times 10^{27} \text{ m}^6 \text{ s}^{-1}$ and here the turbulence model is turned off. Vertically, the model consists of 11 layers (" $10\frac{1}{2}$ " layers) and there have been two changes besides the hybrid coordinate that need to be mentioned. The first one is trivial, the labeling has been changed such that the gas-giant abyssal layer is now counted as $nk + 1$ instead of nk , consequently in the new model one sets $nk = 10$ to match the simulations in LeBeau and Dowling (1998); the idea is to emphasize the boundary-condition nature of the abyssal layer (note that the bottom layer in terrestrial-class models such as the first test case above, which the old model did not include, is counted as nk). The second is that the old model formally had zero pressure and infinite potential temperature at its top boundary whereas in the new model the top corresponds to a finite pressure and a finite potential temperature. We have come to prefer the finite approach because it yields a top layer that behaves more like the other layers. Matching LeBeau and Dowling's vertical configuration in the new EPIC model yields the following undisturbed pressure levels for the layer interfaces: 2.3, 8.2, 28.0, 96.9, 180.2, 334.9, 662.6, 1157.4, 2151.6, 3999.6, and 7435.0 mbar. For the hybrid-coordinate parameters we use $\zeta_0 = 48.9$ K and $\zeta_1 = 130.3$ K in (2.8) and $\alpha = 50$ and $\sigma_\sigma = 0.077$ in (2.9), such that $k_\sigma = 10$.

LeBeau and Dowling idealized the environmental conditions in their study in order to better control the effects of the mean state on the vortex and to make it easier to connect with existing theory. Planetary parameters for Neptune are gravity, $g = 11.0 \text{ m s}^{-2}$, angular velocity, $\Omega = 1.083 \times 10^{-4} \text{ s}^{-1}$, equatorial radius, $r_e = 24764$ km, polar radius, $r_p = 24343$ km, gas constant $R_{\text{gas}} = 3480 \text{ J kg}^{-1} \text{ K}^{-1}$, specific heat $c_p = 9280 \text{ J kg}^{-1} \text{ K}^{-1}$, and $\kappa = R_{\text{gas}}/c_p = 0.375$. For Run 90, the temperature sounding profile, $T(p)$, is given for $p \geq 300$ mbar by their (13) for the case $N_0^2 = 0.64 \times 10^{-4} \text{ s}^{-2}$, and for $p < 300$ mbar by a standard Voyager sounding as shown in their Fig. 3; this profile is included in the EPIC model distribution (in the subdirectory `epic/data/neptune`).

The initial wind profile is a function of latitude only (barotropic) such that the potential vorticity is proportional to the absolute vorticity. For the Run 90 case the zonal wind, u ,

is idealized to have a constant absolute-vorticity gradient of $1.87 \times 10^{-12} \text{ m}^{-1} \text{ s}^{-1}$ from the equator down to -53° latitude, labeled the ' $Q_y = 1/3$ ' case in LeBeau and Dowling (1998). Here we use this and the other zonal-wind profiles from the 1998 study corresponding to $Q_y = 0, 2/3$, and 1 (polynomial fit), which are also now included in the model distribution (when integrating LeBeau and Dowling's (8), note that $(\zeta + f)_0$ is $-6.5 \times 10^{-5} \text{ s}^{-1}$ rather than the printed $-6.5 \times 10^{-4} \text{ s}^{-1}$).

In the old model, to complete the initialization of the prognostic variables for the zonally symmetric configuration (prior to introduction of the vortex) we need to find the pressure field that is in steady-state (gradient) balance with the given zonal-wind profile, which in general can be a function of latitude and the vertical coordinate. This is done in a three-step process. First, a single sounding profile, $T(p)$ is applied to one chosen latitude, λ_{tp} , and the hydrostatic-balance equation in isentropic-coordinate form is integrated vertically to determine the spacing of the Montgomery potential, M . Second, the gradient-balance equation is integrated horizontally out from these positions to yield the Bernoulli streamfunction $B = M + K$ everywhere, after which the (known) kinetic energy per mass, K , is subtracted to yield M everywhere. Third, the hydrostatic-balance equation is applied to M vertically everywhere to determine the pressure. To do the horizontal integration in the second step we take pains to use precisely the same numerics on the grid as in the model's meridional momentum equation. The result is a steady, meridional-plane balance between the pressure and wind fields (mass and momentum fields) that is exact to roundoff error on the model's grid, and manifests itself as a steady solution from onset when the model is subsequently run. Obtaining such a solution to the nonlinear, steady-state equations in the manner just described is possible because we have three key elements: (i) a steady-state solution in two dimensions, (ii) a hydrostatic model, and (iii) a potential function, M .

A problem that can, and often does, arise with the pure-isentropic-coordinate model occurs when the specified u and T combine to yield crossing isentropes. This is a signal that one has requested too much vertical shear in a region of low static stability. When this happens, we reduce the vertical shear in the problematical region until the isentropes cease to cross.

Setting up a steady-state, zonally symmetric model with the hybrid vertical coordinate would proceed in the same manner as above, except that the pressure-gradient term in the horizontal-momentum equations does not come in the form of a single scalar function completely underneath a gradient operator, in other words we are missing number (iii), the potential function. Lack of this one element prevents us from accomplishing the horizontal integration in step two above. It is important to understand that this does not cause a problem with running the model because the model never assumes a steady-state balance as it time-marches the prognostic equations forward. But, it does cause a problem with initializing the model, both in two dimensions because it takes away the straightforward means described above for setting up a perfectly balanced meridional plane, and in three dimensions because it takes away even the lesser approximation of geostrophic balance expressed in terms of a streamfunction, as discussed below. The latter turns out to

be the more serious problem, but for the former we still must decide how to proceed. We have elected to fall back to pressure coordinates, which have a potential function, the geopotential, and also have the distinct advantage over isentropic coordinates that isobaric surfaces do not cross for a hydrostatic model (because regular density, unlike mass divided by $dx dy d\theta$, is always nonnegative). We carry out the analogous three-step procedure outlined above for a fine grid in pressure coordinates, and then interpolate the results onto the model's hybrid-coordinate surfaces. As is well known, such an interpolation invariably degrades the gradient balance on the target grid. Fortunately, we have found that even for demanding configurations such as Jupiter's alternating jets, a steady-state solution may be obtained by running the model past this initial condition and allowing the natural geostrophic-adjustment process to settle the meridional plane into balance. For the Neptune case in hand, the initial wind field is simple enough that no such equilibration phase of the zonally symmetric model is found to be necessary.

Having specified the grid configuration and the zonally symmetric environmental conditions, we turn next to the problem of initializing the vortex itself, lately called "bogussing" in the terrestrial literature. Vortex studies using the EPIC model have typically initialized full-strength anticyclones or cyclones that are geostrophically balanced by specifying a perturbation to the Montgomery streamfunction, ΔM . In more recent papers the shape of this perturbation has been an ellipsoidal Gaussian (e.g., Stratman et al., 2001; Morales-Juberias and Dowling, 2005), but LeBeau and Dowling employed a somewhat different shape for which we have included here the parameters retrieved from the archived Run 90 source code. The horizontal cross-section is controlled by an elliptical radial variable $r_{el} \equiv (x^2 + AR^2 y^2)^{1/2}$, where $AR = 1.5$ is the aspect ratio and x and y measure horizontal distance in meters from the initial vortex center, which is placed at latitude $\lambda_0 = -29.7^\circ$ (the factor multiplying y^2 is correctly written AR^2 in the original source code, rather than AR as printed in the paper). The perturbation is applied to layers $k = 5, 6, 7, 8$, and 9 and includes a vertical amplitude factor $\alpha(k)$ that is set to 0.2, 0.6, 1.0, 1.0, and 0.5, respectively. The perturbation is applied inside of $r_{el} = L_{out} = 15171$ km and consists of an inner and outer part separated by $L_{in} = 4248$ km

$$\Delta M = \alpha(k) M_0 \frac{f}{f_0} \times \begin{cases} A_0 + A_2 (r_{el}/L_{in})^2, & 0 \leq r_{el} < L_{in}, \\ \text{sech}^2(r_{el}/a_0), & L_{in} \leq r_{el} < L_{out}, \end{cases} \quad (3.1)$$

where $M_0 = 35,000 \text{ m}^2 \text{ s}^{-2}$, $f = 2\Omega \sin(\lambda)$ is the Coriolis parameter, $f_0 = f(\lambda_0) = -1.074 \times 10^{-4} \text{ s}^{-1}$, $a_0 = 5057$ km, $A_2 = -(L_{in}/a_0) \text{sech}^2(L_{in}/a_0) \tanh(L_{in}/a_0) = -0.305$, and $A_0 = \text{sech}^2(L_{in}/a_0) - A_2 = 0.835$. These parameters were chosen to yield a southern-hemisphere anticyclone that resembles the Great Dark Spot as seen by Voyager 2, keeping in mind that we do not know the strength of the circulation inside the GDS due to the lack of cloud tracers. Section 5.4 below explains how the new model's initial prognostic variables are modified given the eddy streamfunction (3.1). As will become clear, one of our

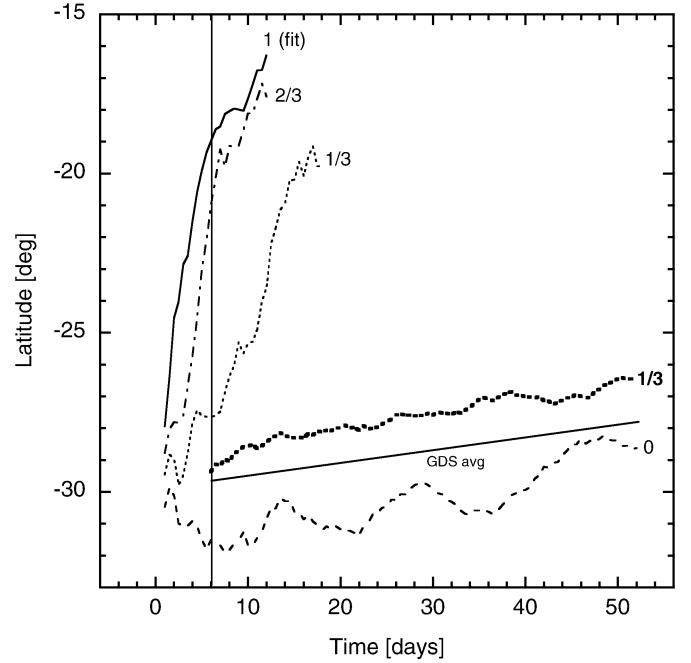


Fig. 5. Meridional drift of models of Neptune's Great Dark Spot (GDS). The straight line labeled "GDS avg" shows the 1.2° per month average drift rate of the GDS (Sromovsky et al., 1993). The bold dotted line "1/3" is the track of Run 90 from LeBeau and Dowling (1998), which was their most accurate simulation; the "1/3" refers to the value of the background potential-vorticity gradient, which was approximately one-third the value corresponding to the Sromovsky et al. (1993) polynomial fit to the zonal wind. The rest of the curves show runs made with the new model in which the configuration is identical to Run 90 except for the change to the hybrid vertical coordinate and the indicated variation in the potential-vorticity gradient.

highest priorities for future work is to develop a better means of introducing balanced eddies into the hybrid-coordinate model.

The general appearance of the GDS in the new model is the same as illustrated in the 1998 paper and so we turn immediately to the question of meridional drift. Fig. 5 shows the results of four runs made with the new model labeled by their nondimensional potential-vorticity gradients. This figure may be compared to Fig. 4 of LeBeau and Dowling (1998). Run 90 from the 1998 study is included as the bold dotted line labeled "1/3." We track the vortex center as in the 1998 study by finding the average position inside a suitable closed contour of potential vorticity multiplied by the average layer height, specifically the contour $-5.48 \times 10^{-5} \text{ s}^{-1}$ in layer $k = 8$. It is straightforward to distinguish this closed contour from the open one with the same value that exists near the equator by scanning up from the south pole.

In agreement with the 1998 study, the equatorward drift increases with increasing potential-vorticity gradient and the polynomial-fit value yields a drift that is much faster than the 1.2° -per-month average drift of the GDS. As noted by LeBeau and Dowling, the occasional poleward drift of the model vortices is realistic since the GDS itself occasionally meandered poleward. Quantitatively, however, the new model's drift rates are larger than in the old model. In all cases, during its first turnaround the vortex entrains environmental potential vorticity, and the degree to which this occurs appears to be the key to

the variation in behavior from the old model to the new. Even though we have used the streamfunction approach to initialize the Great Dark Spot in both studies, this technique yields poorer balance in the hybrid-coordinate case, especially in the intermediate altitude region where the coordinate is making the transition from $\tilde{f}(\sigma)$ to θ and the streamfunction approach is not accurate. The initial transient period for the vortex is still about 6 days as in the 1998 study, as marked by the vertical line in Fig. 5, but the amplitude of the adjustment is now larger. Notice in particular the larger equatorward movement in the first six days in the new model compared to the old, especially for the cases “1 (fit)” and “2/3.”

The biggest qualitative change between the old and new results is that in the new model there is a small meridional drift in the uniform potential vorticity case, and moreover it matches the observed drift rate, whereas $Q_y = 0$ implied no drift in the old model. This raises the intriguing possibility that in the region of the Great Dark Spot the potential vorticity is mixed into a uniform state, which would make Neptune’s environmental conditions qualitatively different than Jupiter’s and connect Neptune to a large body of theoretical papers on potential-vorticity mixing. One point that needs careful attention is the likelihood that the Great Dark Spot on Neptune did not form in a manner anything like the way the vortices have been introduced into either the old or new EPIC model, which means that such a study should pay particular attention to how the process leading to the genesis of the GDS ultimately affects its drift rate.

4. Phased-out components

In this section we point out those components of the EPIC model described in Dowling et al. (1998) that have become obsolete. We start by fixing a potential problem involving the continuity equation. In Dowling et al. (1998) we converted the results of mass advection from the predictor–corrector algorithm into a 3rd-order Adams–Bashforth tendency, with the idea that optional contributions to the mass tendencies could then be added, and the mass and momentum fields could both be advanced with the Adams–Bashforth timestep. We have since discovered that this is an ill-conceived idea that can lead to an inconsistent scheme. The inconsistency does not arise if advection is the only process affecting the mass variables. However, if any source/sink or other right-hand side terms are added to the mass tendency, their contribution to the integration will be too large by a factor of $23/12 \approx 2$. All processes affecting mass variables are now done sequentially in a consistent, time-splitting manner such that this problem cannot occur.

4.1. Shallow-water model

Dowling et al. (1998) mention that “the EPIC model grew out of a shallow-water model and retains this special case as an option.” During the Voyager spacecraft exploration of the gas giants, the lack of vertical resolution in one- and two-layer shallow-water models matched the paucity of vertical-structure data for Jupiter, Saturn, Uranus, and Neptune, while doing justice to the newly acquired, high-resolution cloud-motion data,

and thus the primitive shallow-water model saw some utility for modeling the data. But, as the vertical structure of gas-giant atmospheres has become better sampled, this utility has diminished; for example, the Galileo Probe and Orbiter have revealed substantial vertical-structure information for Jupiter’s atmosphere, and Cassini is currently doing the same for Saturn. Now, in order to do justice to these newly acquired three-dimensional observations involving wind, temperature, humidity, moist convection, *ortho*–*para*-hydrogen disequilibrium, and hydrocarbon distributions, there is a need for gas-giant models with realistic vertical structure. In the new distribution of EPIC, we drop the shallow water option, which helps unclutter the source code.

4.2. Fortran

The EPIC model has always been predominately written in the C programming language, but Dowling et al. (1998) mention that “Subroutines written in either C or Fortran may be added, for example, the model’s thermodynamical functions are unmodified Fortran subroutines,” and paper has a subsection that describes how to include Fortran in a predominately C source tree, including “tricks to linking Fortran object files to C object files using a C compiler.” We have since learned that the best trick is to use the Fortran compiler to do the linking, even if there is only one line of Fortran in a sea of C. Now, as we look at how the EPIC model has evolved, we realize that most of our portability issues have been related to Fortran, whereas new additions to the model have invariably been written in C. We therefore have converted the remaining Fortran subroutines into C, and have carefully verified that the old and new versions yield precisely the same output; in the process, we have improved the resolution of the table lookup components of the thermodynamics subroutines. We continue to welcome open-source additions in any language, but we plan to take the time to rewrite them into C, which helps with portability but also serves as a practical means of learning the details of contributed algorithms.

4.3. Data management

In the subsection on parallel computing, Dowling et al. (1998) point out that the output of a GCM like EPIC is voluminous, and so for the most part, “Instead of storing data to disks attached to the nCUBE, we send selected output from the nCUBE processors directly to our graphics workstation via Internet socket connections.” As in that paper, we use the Message Passing Interface (MPI), but the nCUBE is a distant memory; we built our current computer, COMPLINE (Comparative Planetology Linux Engine), a 40-node, Flat Neighborhood Network (FNN) cluster that runs at 35 GFLOP and is dedicated to running the EPIC model, from commodity components for \$25,000 in April 2002—now old news as well. Meanwhile, redundant arrays of inexpensive disks, or RAID systems, provide affordable terabyte storage, so we are back to storing our model output to disk and then post-processing it for all analysis work. We now have two types of output files, both

using the self-describing, portable network common data format (netCDF), for which “.nc” is the file suffix. A complete snapshot of the model is written at user-defined intervals to time-stamped `epic.nc` files, and this is the file format used to start or restart runs. Additionally, the user can decide which subset of variables, prognostic and/or diagnostic, to append at specified intervals to an `extract.nc` file, which is more disk-space efficient than the `epic.nc` format, but cannot be used to start a new run. We typically keep only the last timestep of a run in the `epic.nc` format, and do our post processing on `extract.nc` files. Regarding commercial third-party software, we no longer use AVS to control and visualize the model and have removed the corresponding runtime hooks. There are several analysis tools that input netCDF files and can be used to analyze EPIC-model output. Finally, we have ported EPIC to the Darwin Unix platform (Mac OS X). The only issue that arose in that port was related to network sockets, but since we no longer connect the cluster to an animation system in real time, we no longer need to establish our own network socket connections, and so those hooks have also been removed. The net result of these changes is a more portable and easier-to-manage open-source code.

5. New components

The EPIC model is updated steadily to improve functionality, runtime efficiency, portability, and readability. In this section, we briefly describe the main new components added to the model besides the hybrid vertical coordinate. An example of a simple upgrade that affects the entire model is the floating-point precision; the user now chooses this before compiling by setting the environment variable `EPIC_PRECISION` to 4 for single precision or 8 for double precision. Single precision has proven to be sufficient for short and medium-length simulations, including the sample runs in this paper, but double precision does a better job of conserving total mass and hence may be preferable for long simulations.

5.1. Dissipation terms

We follow the standard practice of controlling numerical instability with artificial, high-order dissipation terms, or hyperviscosity. Dowling et al. (1998) described the inclusion of hyperviscosity optionally up to sixth order that is applied to the horizontal winds as well as to the scalar prognostic variables. In the present version, we split the high-order dissipation by treating each dimension separately. This allows us to apply zonal hyperviscosity in an implicit manner, which eliminates numerical-stability problems near the poles. We do so efficiently by adding this as a second task for the zonal FFT, which is now applied to all latitudes, its first task being the high-latitude filter. The meridional direction is handled explicitly as before, except now in a one-dimensional fashion. These operators are available up to eighth order to match the order that is used in the Held and Suarez (1994) benchmark. The hyperviscosity coefficients are factored and

then folded in, order by order, to avoid floating-point overflows and underflows. We have added divergence damping as an optional means of controlling numerical instabilities associated with gravity waves (Skamarock and Klemp, 1992). The user sets the strength of all these high-order dissipation terms by specifying nondimensional values for the corresponding viscosity coefficients. Finally, the new optional turbulence model, described below, is in the standard form of a second-order dissipation with the two horizontal dimensions treated together explicitly and the vertical dimension treated implicitly to avoid numerical instabilities in thin-layer regions.

5.2. High-latitude zonal filter

We have changed the strategy for high-latitude filtering that is used to prevent numerical instabilities associated with the small zonal grid spacing on our longitude–latitude grid near the poles. For the momentum equations, it is preferable to apply the filter under the gradient operator of the pressure-gradient terms because this avoids spurious generation of vorticity. We were aware of this issue when designing the original model and tried the method of Kar et al. (1994) but had trouble with it, especially for Neptune’s strong winds at high latitudes, so we ended up filtering the wind tendencies themselves, which means the old model has the spurious vorticity generation problem. Fortunately, the trouble we experienced before has since been cleared up by the move to an implicit operator for high-order dissipation in the zonal direction, as described above. Interestingly, when we tested the Held–Suarez benchmark on the new model with our 1998 high-latitude filtering strategy, it yielded spurious jets in the polar regions in addition to the expected midlatitude jets, which prompted us to revisit the filtering question. The strategy we now employ is to filter the kinetic energy term under the gradient operator, plus the hybrid density, h , and the potential temperature, θ , which are the prognostic variables that generate the Montgomery potential and geopotential that appear under gradient operators in the pressure-gradient terms, but we no longer filter the winds themselves. In (56) of Dowling et al. (1998) we now use $\alpha_1 = \alpha_2 = 2$, meaning we now follow Kar et al. (1994).

5.3. Time-marching schemes

For advection of dry hybrid density, h_{dry} , and optional density-weighted mass mixing ratios, $h_{\text{dry}}q_i$, we use the KA97 update to the predictor–corrector scheme of Hsu and Arakawa (1990), which over the years has proved to be reliable. For numerical reasons this scheme enforces a small, positive lower limit to the density, which KA97 set to “1 mbar.” We need a more flexible minimum because, for example, the surface pressure on Mars is only about 6 mbar, so we set the minimum to 2% of the average initial density in each layer. We currently advect θ with a simple upwind scheme that is not h -weighted, because it has proven to be adequate and avoids repeated multiplication and division of θ by the interface value of h . The horizontal velocities are averaged onto the layer interfaces, where θ

resides in the Charney–Phillips grid, using the same weighting as in KA97.

Two schemes are now available for time-marching the horizontal momentum equations, the model's original 3rd-order Adams–Bashforth timestep and the standard Asselin-filtered leapfrog timestep. The pros and cons of each are discussed in detail by Durran (1991). Having both adds some flexibility that comes in handy when comparing to other models and when trying to isolate numerical instabilities. Durran cites Williamson (1983) for the value $\gamma = 0.06$ for the Asselin filter parameter, which prevents the odd–even instability of the leapfrog step, and this is the default value we use (GAMMA_ASSELIN in the code). Dissipative terms like Rayleigh drag and turbulent viscosity cannot be treated by the leapfrog step, and so we lag them, as is customary. For either timestep, we use the “poor-man's implicit” approach (C. Konor, private communication) in which the pressure-gradient terms are updated before being added to the wind tendencies, which allows for a somewhat larger timestep, Δt . To handle these multiple options, in the source code the prognostic momentum variables, U and V , and their tendency fields, DUDT and DVDT, now each have a time-plane index (two or three planes as required) in addition to their three spatial indices.

5.4. Initializing balanced eddies

In the terrestrial literature, methods for introducing full-strength, balanced eddies (bogussing) have been worked out based on reducing the problem to an elliptic equation (Kurihara et al., 1993; Wang, 1995). We do not yet have a similar technique implemented for our hybrid-coordinate model, but one would be desirable. We currently start by specifying a perturbation streamfunction. For $\zeta > \zeta_\sigma$, we interpret this streamfunction as the Montgomery potential and use the hydrostatic equation in the form $\partial M/\partial\theta = C_p T/\theta$ to get the perturbation temperature. In this region, as in the 1998 model, p is modified but not θ . For $\zeta \leq \zeta_\sigma$ (corresponding to $\tilde{g}(\sigma) = 0$), we switch and interpret the streamfunction as the geopotential, Φ , and use the hydrostatic equation in the form $\partial\Phi/\partial p = -RT/p$ to get the perturbation temperature, and now θ is modified but not p . Following the introduction of such vortices, we allow for a transient geostrophic adjustment period, just as with the previous model.

There has been recent progress reported in the literature regarding deriving balanced winds from streamfunctions. McIntyre and Roulstone (2002) show how balanced-flow theory can be optimized to yield an $O(Ro^2)$ corrected velocity, where $Ro = U/fL$ is the Rossby number,

$$\mathbf{u}^C = \mathbf{u}^G + \hat{\mathbf{z}} \times f^{-1} \mathbf{u}^G \cdot \nabla \mathbf{u}^G, \quad (5.1)$$

where \mathbf{u}^G is the $O(Ro)$ horizontal geostrophic wind vector. Given spherical map factors $dx = r d\phi$ and $dy = R d\lambda$ for longitude and latitude, respectively, the geostrophic velocity is

$$\mathbf{u}^G = (u^G, v^G) = \frac{1}{f} \left(-\frac{1}{R} \frac{\partial M}{\partial \lambda}, \frac{1}{r} \frac{\partial M}{\partial \phi} \right), \quad (5.2)$$

where $f = 2\Omega \sin \lambda$ is the Coriolis parameter, and here M stands for the appropriate streamfunction (Montgomery potential or geopotential). If we define ζ^G and K^G to be the relative vorticity and kinetic energy per mass that correspond to the geostrophic wind, then the components of (5.1) may be expressed most conveniently for the EPIC model as

$$\begin{aligned} \mathbf{u}^C = & \left(u^G - \frac{1}{f} \left\{ \zeta^G u^G + \frac{1}{R} \frac{\partial K^G}{\partial \lambda} \right\}, \right. \\ & \left. v^G - \frac{1}{f} \left\{ \zeta^G v^G - \frac{1}{r} \frac{\partial K^G}{\partial \phi} \right\} \right). \end{aligned} \quad (5.3)$$

This provides a straightforward means of improving the introduction of balanced eddies via a streamfunction and we now use (5.3) instead of (5.2) for this purpose.

5.5. Turbulence model

The version of EPIC described by Dowling et al. (1998) did not have a subgrid-scale closure scheme beyond hyperviscosity. One can do better by using a PDE turbulence model that has been verified against observations, in the best case using both atmospheric data for buoyancy and stratification effects and engineering data for near-wall effects. In theory, a large-eddy simulation (LES) model can account for both turbulent mixing in the interior of an atmosphere and the modified mixing and transport that occurs near the surface in a planetary boundary layer [PBL; see the discussion by Hong and Pan (1996)]. However, in practice the vertical grid resolution must often be extremely fine near the surface to match experimental data using LES. As a result, a variety of closure schemes that use more efficient parameterizations have been developed. Of these, we have settled on the detached-eddy simulation (DES) approach of Spalart et al. (1997) because of its flexibility and economy.

The framework for DES is the standard Reynolds-Averaged Navier–Stokes (RANS) viscosity model for turbulence, in which the effects of turbulence on the mean-flow are represented by the addition of Reynolds-stress terms to the momentum and energy equations

$$\frac{\partial u_i}{\partial t} = \dots - \frac{\partial(\overline{u'_i u'_j})}{\partial x_j}, \quad (5.4a)$$

$$\frac{\partial \theta}{\partial t} = \dots - \frac{\partial(\overline{u'_j \theta'})}{\partial x_j}, \quad (5.4b)$$

where u'_i and θ' are the perturbation velocities and perturbation potential temperature, respectively (the vertical momentum equation is supplanted by the hydrostatic approximation). To close these equations, the following standard approximations are adopted

$$\overline{u'_i u'_j} = -K_m \frac{\partial u_i}{\partial x_j}, \quad (5.5a)$$

$$\overline{u'_j \theta'} = -K_h \frac{\partial \theta}{\partial x_j}, \quad (5.5b)$$

where $K_m = \nu + \nu_t$ is the sum of the molecular and turbulent viscosities (“ m ” for “momentum”), and K_h is the eddy diffusivity of potential temperature (“ h ” for “heat”). To find ν_t , we start with the Spalart–Allmaras DES model (Spalart and Allmaras, 1992; Spalart et al., 1997), which introduces the variable $\tilde{\nu}$ and its nondimensional counterpart $\chi = \tilde{\nu}/\nu$, and relates these to ν_t by

$$\nu_t = \frac{\chi^3}{\chi^3 + \chi_0^3} \tilde{\nu}, \quad (5.6)$$

where $\chi_0 = 7.1$. The variable $\tilde{\nu}$ represents a new prognostic variable that is added when the DES turbulence model is turned on, and is governed by the equation

$$\begin{aligned} \frac{D\tilde{\nu}}{Dt} = c_{b1} \tilde{S} \tilde{\nu} + \frac{1}{\sigma} \left[\frac{\partial}{\partial x_j} \left((\nu + \tilde{\nu}) \frac{\partial \tilde{\nu}}{\partial x_j} \right) + c_{b2} \frac{\partial \tilde{\nu}}{\partial x_j} \frac{\partial \tilde{\nu}}{\partial x_j} \right] \\ - c_{w1} f_w (\tilde{\nu}/d)^2. \end{aligned} \quad (5.7)$$

The EPIC model uses the following empirically determined optimal values for the parameters in (5.7): $c_{b1} = 0.1355$, $c_{b2} = 0.6220$, $\sigma = 0.6667$, and $c_{w1} = 3.2391$; see Spalart et al. (1997) for the formulas for \tilde{S} in the production term and f_w in the destruction term. The variable d in the last term is critical to the DES approach and is discussed below.

The original DES model arose in an engineering context and does not take into account the suppression of turbulence in the vertical direction due to stable stratification or the enhancement due to convective instability, and so we have followed current meteorological practice by including a stability function, $F_c(Ri)$, that depends on the local Richardson number, $Ri = N^2/(\partial u/\partial z)^2$, and is applied as a multiplicative factor to the vertical-diffusion coefficient. This provides convective adjustment to the model via a relaxation process (rather than treating it as an instantaneous process). Following Collins et al. (2004), we assume an effective turbulent Prandtl number of unity and set

$$F_c(Ri) = \begin{cases} \{1 - 18Ri\}^{1/2}, & Ri \leq 0 \text{ (neutral or unstable),} \\ \{1 + 10Ri(1 + 8Ri)\}^{-1}, & Ri > 0 \text{ (stable),} \end{cases} \quad (5.8)$$

with $|Ri|$ in (5.8) limited to be less than or equal to 100. A non-local component to the vertical diffusion (e.g., Hong and Pan, 1996) has not yet been implemented in the EPIC model, and in general we expect the treatment of these subgrid-scale and convective adjustment processes to continue to improve as new algorithms are introduced.

The DES approach addresses the challenges of wall-bounded flows where LES-like unsteady turbulence is required but where the computational cost of near-wall LES is not practical. This is accomplished through the value of d in (5.7), which is set to

$$d = \min(d_w, C_{DES} \Delta), \quad (5.9)$$

where d_w is the distance to the wall (surface), $\Delta = \max(\Delta x, \Delta y, \Delta z)$ measures the local Cartesian grid spacing, and the scaling constant C_{DES} is set to 0.65, which reproduces the proper energy-decay spectrum in isotropic turbulence away

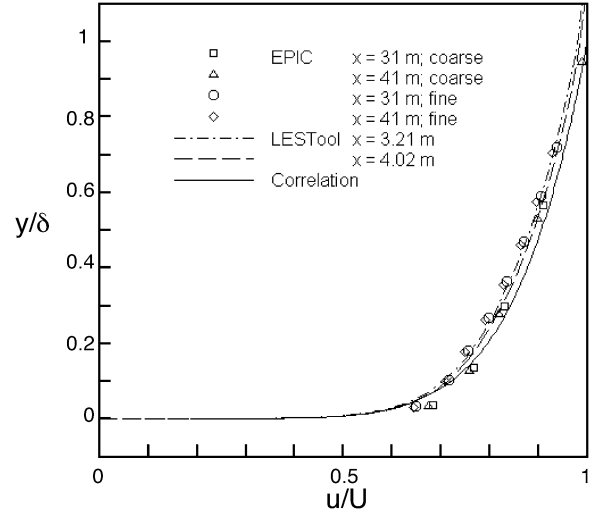


Fig. 6. Turbulent boundary layer velocity profiles using the Spalart–Allmaras DES algorithm. EPIC results are shown at two different stations along a flat plate; “coarse” and “fine” correspond to grids with 4 and 8 layers within the boundary layer, respectively. Also shown are calculations using the CFD code LESTool for the same turbulence algorithm on a finer grid (40 points within the boundary layer, first gridpoint within the viscous sublayer), and the traditional correlation $u/U = (y/\delta)^{1/7}$.

from walls (Strelets, 2001; Constantinescu et al., 2002; Forsythe et al., 2002). Near the surface, d equals d_w and the model is a traditional one-equation Spalart–Allmaras RANS turbulence model; away from the surface, d equals $C_{DES} \Delta$ and the model acts like a gridscale-dependent, LES model. Fig. 6 shows turbulent boundary layer velocity profiles for EPIC using the Spalart–Allmaras DES algorithm, compared to the CFD code LESTool.

6. Summary

The hybrid-coordinate idea appears to be the leading solution to the problem of designing an isentropic-coordinate model that behaves well at the bottom of the atmosphere. We have shown here that both the topography of terrestrial planets and the near-adiabatic lapse rate in the deep atmospheres of gas-giant planets can be handled with this approach, thereby killing two birds with one stone. It is the “one stone” aspect that may be the most significant, because it opens the door to the practical realization of accurately simulating all known atmospheres with a single model.

We started with the hybrid-coordinate formulation of Konor and Arakawa, but have reworked many of the details. In particular, we treat the potential temperature as a diagnostic variable rather than a prognostic variable in the hybrid portion of the model. In hindsight, perhaps it makes sense to first try to craft a hybrid isentropic coordinate by modifying or “downgrading” θ from being the coordinate to being a diagnostic variable, rather than by going all the way to making it a prognostic variable. That this idea and our other modifications work well is demonstrated in a preliminary fashion by our positive results for the Held–Suarez benchmark. We stress that this is just the beginning and there are more stringent tests that await the model in

the terrestrial setting, for example how accurately it simulates frontogenesis. We expect to make steady improvements to the model based on the results of such tests.

At the top of the list of new functionality needed for the hybrid EPIC model² is a better means of introducing balanced eddies, considering that the streamfunction approach we have employed in the past is now compromised. The technical details for this have largely been worked out in the context of sigma-coordinate models and simply need to be applied to the particular case of our theta–sigma hybrid. The inherent flexibility of design of the hybrid coordinate opens up the possibility that one might be able to single out a member of a family of acceptable profiles that optimizes the behavior of the model in some fashion (C. Leovy, personal communication), which is worth investigating.

Acknowledgments

This research was supported by NASA’s Planetary Atmospheres, Outer Planets Research, and EPSCoR Programs and NSF’s Planetary Astronomy Program. The authors thank Donald R. Johnson and an anonymous referee for their thoughtful and constructive reviews, Hank Dietz at the KAOS laboratory at the University of Kentucky (<http://aggregate.org>) for his contribution to the design of the Comparative Planetology Laboratory’s COMPLINE computer cluster and to the optimization of the EPIC source code as part of our mutual NASA EPSCoR collaboration, and Celal Konor for several insightful comments during the course of this research.

Appendix A. Calculation of buoyancy frequency

One consequence of the generalized definition of potential temperature, θ , that we use to handle mixtures, in particular the *ortholpara* mixture of molecular hydrogen (Dowling et al., 1998), is that the buoyancy (Brunt–Väisälä) frequency, N , does not satisfy the meteorology textbook relation $N^2 = (g/\theta) d\theta/dz$. We are thus motivated to derive a formula for N^2 for an arbitrary equation of state along the lines of Gill (1982). Consider a vertically oscillating air parcel that maintains pressure equilibrium with its environment and undergoes no drag forces, chemical reactions, radiation, or other diabatic processes. The environment is assumed to be in a hydrostatically balanced state. Assuming small-amplitude motion, the equation governing the displacement Δz of the parcel from its initial position is

$$\begin{aligned} \frac{d^2}{dt^2} \Delta z &= -g - \frac{1}{\rho_{\text{par}}} \frac{\partial p}{\partial z} = -\frac{g}{\rho_{\text{par}}} (\rho_{\text{par}} - \rho_{\text{env}}) \\ &= -N^2 \Delta z, \end{aligned} \quad (\text{A.1})$$

² The EPIC atmospheric model (©1998–2006 by Timothy E. Dowling) is distributed as open source under the provisions of the Free Software Foundation’s General Public License, and is available from the NASA Planetary Data System (PDS) Atmospheres node.

where ρ_{par} and ρ_{env} are the densities of the parcel and environment, respectively, and

$$\begin{aligned} N^2 &= \frac{g}{\rho_{\text{par}}} \left[\left(\frac{\partial \rho}{\partial z} \right)_{\text{par}} - \left(\frac{\partial \rho}{\partial z} \right)_{\text{env}} \right] \\ &= \frac{g}{\rho_{\text{par}}} \left[\left(\frac{\partial \rho}{\partial z} \right)_{\text{adi}} - \left(\frac{\partial \rho}{\partial z} \right)_{\text{env}} \right]. \end{aligned} \quad (\text{A.2})$$

The subscript “adi” refers to the adiabatic density gradient, which is the density gradient followed by the air parcel. The environmental density gradient is $(\partial \rho / \partial z)_{\text{env}} = -\rho_{\text{env}} g d\rho_{\text{env}}/dp$, since the environmental profile is a function of pressure only and $p = p(z)$. The adiabatic gradient may be written

$$\begin{aligned} \left(\frac{\partial \rho}{\partial z} \right)_{\text{adi}} &= \left(\frac{\partial \rho}{\partial p} \right)_T \frac{\partial p}{\partial z} + \left(\frac{\partial \rho}{\partial T} \right)_p \left(\frac{\partial T}{\partial z} \right)_{\text{adi}} \\ &= -\rho_{\text{env}} g \left[\left(\frac{\partial \rho}{\partial p} \right)_T + \left(\frac{\partial \rho}{\partial T} \right)_p \left(\frac{\partial T}{\partial p} \right)_s \right], \end{aligned} \quad (\text{A.3})$$

where $(\partial T / \partial p)_s$ is the temperature gradient at constant entropy, which is the temperature profile along the adiabat assuming the motion is thermodynamically reversible. We can obtain an expression for this gradient as follows. The entropy per mass, s , satisfies

$$ds = \left(\frac{\partial s}{\partial T} \right)_p dT + \left(\frac{\partial s}{\partial p} \right)_T dp. \quad (\text{A.4})$$

Using the facts that $c_p = T(\partial s / \partial T)_p$ and $(\partial s / \partial p)_T = -(\partial v / \partial T)_p$, where c_p is specific heat at constant pressure and $v \equiv 1/\rho$ is specific volume, and setting $ds = 0$ for an adiabatic, reversible process, we obtain

$$\left(\frac{\partial T}{\partial p} \right)_s = \frac{T}{c_p} \left(\frac{\partial v}{\partial T} \right)_p = \frac{\alpha T}{\rho c_p}, \quad (\text{A.5})$$

where $\alpha \equiv -\rho^{-1}(\partial \rho / \partial T)_p = \rho(\partial v / \partial T)_p$ is the isobaric volume thermal expansivity. The expression for $(\partial \rho / \partial z)_{\text{adi}}$ is therefore determined, and we can write the buoyancy frequency as

$$N^2 = g^2 \left[-\rho\beta + \frac{\alpha^2 T}{c_p} + \frac{d\rho_{\text{env}}}{dp} \right], \quad (\text{A.6})$$

where $\beta \equiv \rho^{-1}(\partial \rho / \partial p)_T$ is the isothermal compressibility, and we have approximated the factor $(\rho_{\text{env}} / \rho_{\text{par}})$ as unity for small-amplitude motion. The above result is equivalent to that derived by Gill (1982, p. 51) in his (3.7.1). In his expression, the environmental vertical density gradient is expressed using the environmental temperature and composition gradient, whereas in our expression, gradients of environmental temperature and composition are contained in the term $d\rho_{\text{env}}/dp$. Our formulation has the advantage that, unlike Gill, we make no assumptions about the equation of state or the nature of the compositional gradients in the environment (although for specific applications, such assumptions may be introduced in order to calculate $d\rho_{\text{env}}/dp$). We use (A.6) to calculate N^2 in the EPIC model (the function `get_brunt2`).

Example 1. For an ideal gas, $\beta = 1/p$ and $\alpha = 1/T$. Assuming the environmental composition is constant with height, we

can express the environmental density gradient in terms of the environmental temperature gradient by writing

$$\frac{d\rho_{\text{env}}}{dp} = \left(\frac{\partial\rho}{\partial p}\right)_T + \left(\frac{\partial\rho}{\partial T}\right)_p \frac{dT_{\text{env}}}{dp}. \quad (\text{A.7})$$

Inserting these expressions into (A.6) leads to the standard expression for an ideal gas,

$$N^2 = \frac{g}{T} \left(\frac{g}{c_p} + \frac{dT_{\text{env}}}{dz} \right). \quad (\text{A.8})$$

Example 2. Consider a liquid with an equation of state

$$\rho = \rho_0 [1 - \alpha(T - T_0) + \beta(p - p_0)], \quad (\text{A.9})$$

where α and β are evaluated at the reference state. Assuming an environment of constant composition, then

$$\begin{aligned} N^2 &= \alpha g^2 \rho_0 \left(\frac{\alpha T}{\rho_0 c_p} - \frac{dT_{\text{env}}}{dp} \right) \\ &= \alpha g \left(\alpha T \frac{g}{c_p} + \frac{dT_{\text{env}}}{dz} \right), \end{aligned} \quad (\text{A.10})$$

which matches (3.7.1) of Gill (1982) in the case of no salinity gradient.

Example 3. The incompressible case with $\alpha = 0$ and $\beta = 0$ yields the familiar (Turner, 1973)

$$N^2 = g^2 \frac{d\rho_{\text{env}}}{dp} = - \frac{g}{\rho_{\text{env}}} \frac{d\rho_{\text{env}}}{dz}. \quad (\text{A.11})$$

References

- Arakawa, A., Konor, C.S., 1996. Vertical differencing of the primitive equations based on the Charney–Phillips grid in hybrid σ – p vertical coordinates. *Mon. Weather Rev.* 124, 511–528.
- Collins, W.D., Rasch, P.J., Boville, B.A., Hack, J.J., McCaa, J.R., Williamson, D.L., Kiehl, J.T., Briegleb, B., Bitz, C., Lin, S.-J., Zhang, M., Dai, Y., 2004. Description of the NCAR Community Atmosphere Model (CAM 3.0). NCAR Technical Note NCAR/TN-464+STR.
- Constantinescu, G.S., Pacheco, R., Squires, K.D., 2002. Detached-eddy simulation of flow over a sphere. *AIAA Paper* 2002-0425.
- Dennis, J.E., Schnabel, R.B., 1996. Numerical Methods for Unconstrained Optimization and Nonlinear Equations. In: *Classics Appl. Math.*, vol. 16. SIAM, pp. 336–338.
- Dowling, T.E., Fischer, A.S., Gierasch, P.J., Harrington, J., LeBeau, R.P., Santori, C.M., 1998. The explicit planetary isentropic-coordinate (EPIC) atmospheric model. *Icarus* 132, 221–238.
- Durran, D.R., 1991. The third-order Adams–Bashforth method: An attractive alternative to leapfrog time differencing. *Mon. Weather Rev.* 119, 702–720.
- Forsythe, J.R., Squires, K.D., Wurtzler, K.E., Spalart, P.R., 2002. Detached-eddy simulation of fighter aircraft at high alpha. *AIAA Paper* 2002-0591.
- Gill, A.E., 1982. *Atmosphere–Ocean Dynamics*. Academic Press, pp. 50–51.
- Held, I.M., Suarez, M.J., 1994. A proposal for the intercomparison of the dynamical cores of atmospheric general circulation models. *Bull. Am. Meteorol. Soc.* 73, 1825–1830.
- Hong, S.-Y., Pan, H.-L., 1996. Nonlocal boundary layer vertical diffusion in a medium-range forecast model. *Mon. Weather Rev.* 124, 2322–2339.
- Hsu, Y.-J.G., Arakawa, A., 1990. Numerical modeling of the atmosphere with an isentropic vertical coordinate. *Mon. Weather Rev.* 118, 1933–1959.
- Kar, S.K., Turco, R.P., Mechoso, C.R., Arakawa, A., 1994. A locally one-dimensional semi-implicit scheme for global gridpoint shallow-water models. *Mon. Weather Rev.* 122, 205–222.
- Konor, C.S., Arakawa, A., 1997. Design of an atmospheric model based on a generalized vertical coordinate. *Mon. Weather Rev.* 125, 1649–1673.
- Kurihara, Y., Bender, M.A., Ross, R.J., 1993. An initialization scheme of hurricane models by vortex specification. *Mon. Weather Rev.* 121, 2030–2045.
- LeBeau, R.P., Dowling, T.E., 1998. EPIC simulations of time-dependent, three-dimensional vortices with application to Neptunes Great Dark Spot. *Icarus* 132, 239–265.
- McIntyre, M.E., Roulstone, I., 2002. Are there higher-accuracy analogues of semi-geostrophic theory? In: *Large-Scale Atmosphere–Ocean Dynamics. II. Geometric Methods and Models*. Cambridge Univ. Press, pp. 300–363.
- Morales-Juberias, R., Dowling, T.E., 2005. Simulations of high-latitude spots on Jupiter: Constraints on vortex strength and the deep wind. *Planet. Space Sci.* 53, 1221–1233.
- Schaack, T.K., Zapotocny, T.H., Lenzen, A.J., Johnson, D.R., 2004. Global climate simulation with the University of Wisconsin global hybrid isentropic coordinate model. *J. Climate* 17, 2998–3016.
- Skamarock, W.C., Klemp, J.B., 1992. The stability of time-split numerical methods for the hydrostatic and the nonhydrostatic elastic equations. *Mon. Weather Rev.* 120, 2109–2127.
- Spalart, P.R., Allmaras, S.R., 1992. A one-equation turbulence model for aerodynamic flows. *AIAA Paper* 1992-0439.
- Spalart, P.R., Jou, W.-H., Strelets, M., Allmaras, S.R., 1997. Comments on the feasibility of LES for wings, and on a hybrid RANS/LES approach. In: Liu, C., Liu, Z. (Eds.), *Advances in DNS/LES; 1st AFOSR International Conf. on DNS/LES*, Ruston, LA. Greyden Press, Columbus, OH, pp. 1–22.
- Sromovsky, L.A., Limaye, S.S., Frye, P.M., 1993. Dynamics of Neptune’s major cloud features. *Icarus* 105, 110–141.
- Stratman, P.W., Showman, A.P., Dowling, T.E., Sromovsky, L.A., 2001. EPIC simulations of bright companions to Neptune’s Great Dark Spots. *Icarus* 151, 275–285.
- Strelets, M., 2001. Detached eddy simulation of massively separated flows. *AIAA Paper* 2001-0879.
- Turner, J.S., 1973. *Buoyancy Effects in Fluids*. Cambridge Univ. Press, pp. 11–12.
- Wang, Y., 1995. An inverse balance equation in sigma coordinates for model initialization. *Mon. Weather Rev.* 123, 482–488.
- Williamson, D.L., 1983. Description of the NCAR Community Climate Model (CCMOB). NCAR Technical Note NCAR/TN-210+STR.
- Zhu, Z., Thuburn, J., Hoskins, B.J., Haynes, P.H., 1992. A vertical finite-difference scheme based on a hybrid σ – θ – p coordinate. *Mon. Weather Rev.* 120, 851–862.