

The Explicit Planetary Isentropic-Coordinate (EPIC) Atmospheric Model

T. E. Dowling

Department of Mechanical Engineering, University of Louisville, Louisville, Kentucky 40292
E-mail: dowling@fiolab.spd.louisville.edu

A. S. Fischer

Department of Earth, Atmospheric, and Planetary Sciences, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139

P. J. Gierasch and J. Harrington

Department of Astronomy, Cornell University, Ithaca, New York 14853

R. P. LeBeau, Jr.

Department of Earth, Atmospheric, and Planetary Sciences, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139

and

C. M. Santori

Department of Applied Physics, Stanford University, Stanford, California 94305

Received June 23, 1997; revised October 17, 1997

We describe a new general circulation model (GCM) designed for planetary atmospheric studies called the EPIC model. This is a finite-difference model based on the isentropic-coordinate scheme of Hsu and Arakawa (1990. *Mon. Wea. Rev.* 118, 1933–1959). We report on previously undocumented modifications, additions, and key practical issues that experience running the model has revealed to be important. The model integrates the hydrostatic primitive equations, which are valid for large-scale atmospheric dynamics and include gravity waves (buoyancy waves), planetary waves (Rossby waves), and horizontally propagating sound waves (Lamb waves), but not vertically propagating sound waves because of the hydrostatic approximation. The vertical coordinate is entropy in the form of potential temperature, which coincides with material surfaces for adiabatic motion. This means that there is no vertical velocity except where there is heating, which improves accuracy and helps the model maintain conservation properties over long integrations. An isentropic vertical coordinate is natural for the atmospheres of Jupiter, Saturn, Uranus, and Neptune, which are believed to have essentially adiabatic interiors that match up with the bottom of the model and is also excellent for middle-atmosphere studies on any planet. Radiative processes are parameterized by Newtonian cooling, and the latent heat of ortho–para hydrogen conversion is included when appropriate, with a suitably defined mean potential temperature. The model is written with

general map factors that make it easy to configure in oblate spherical, cylindrical, or Cartesian coordinates. The code includes optional Message Passing Interface (MPI) library calls and hence runs on any Unix-based parallel computer or network cluster. An optional graphical user interface to commercial visualization software facilitates control of the model and analysis of output. Memory is allocated dynamically such that the user does not recompile to change horizontal or vertical resolution or range. Applications to date include comet impact forecasts and hindcasts for Jupiter, meridional circulation studies of Uranus and Neptune, and the accompanying paper on three-dimensional simulations of Neptune's Great Dark Spot (1998. *Icarus* 132, 239–265). © 1998 Academic Press

1. INTRODUCTION

This paper describes the general circulation model (GCM) used in the companion paper by LeBeau and Dowling (1998) to study the dynamics of Neptune's Great Dark Spot (GDS). An early version of this GCM was used to make predictions of the response of Jupiter's atmosphere to the Shoemaker–Levy 9 comet impacts and successfully predicted that there would be no long-lasting vortices spawned from that event (Harrington *et al.* 1994). Together with introductory material, three main technical compo-

nents are discussed here: the model's adiabatic, diabatic, and numerical stability components. We also include a section containing practical implementation details and two appendices describing technical aspects of the model related to hydrogen thermodynamics. We spell out only those details of the model, about 25% of the total, that represent previously undocumented modifications and additions to published algorithms, together with enough interstitial material between the details to provide a coherent description of the overall GCM. Technical accounts of computer models can make notoriously tedious reading, and this one is no exception, except when it comes time to design one's own model, in which case the more clearly written, anecdotal descriptions one can lay one's hands on the better, and it is in this spirit that we have composed this account.

1.1. Design Goals

A GCM allows one to model an atmosphere realistically without artificial boundary conditions, to pull together many different types of observations into a coherent global picture, and to make predictions. It also allows one to study idealized or localized processes to gain further insight into the underlying dynamics. It serves as a test bed for new ideas, a self-consistency check, and makes clear what observational input is most sensitive to a particular process. In addition, we seek to take the traditional roll of the GCM one step further by specifically designing this one to be, from its inception, a tool for comparative-planetology studies, with the long-term goal of accurately simulating all the dozen or so atmospheres in the Solar System with a single model.

The main goal of the present GCM is to provide a flexible, state-of-the-art model for studies of planetary atmospheric dynamics and thermodynamics. In particular, there is a need for a GCM of the gas-giant atmospheres. The excellent spacecraft observations of Voyager, Galileo, the Hubble Space Telescope, and the Galileo Probe data and upcoming Cassini mission, continue to give us clear views across multiple wavelengths of many fascinating phenomena on the gas-giant planets. Key phenomena include long-lived vortices, multiple jet streams, bursts of cloud activity, and organized planetary-scale waves. Although these all involve complicated processes, they occur in an environment that is fundamentally simpler than experienced by Earth's atmosphere. There are no mountain ranges or air-sea interfaces on the gas-giant planets, and the time-dependence of their eddies is much more predictable, in part because there is enough room on a gas-giant planet for eddies to act more like individuals, instead of like crowded inhabitants of a small domain (Earth's atmosphere is only large enough to hold 6–10 eddies around its circumference). The price one pays

for having a giant domain is the need for parallel computers to resolve eddies. The typical eddy scales in the atmospheres of Earth and Jupiter both happen to be about the same, about 1000 km, but Jupiter has over 100 times the surface area of Earth. Earth's ocean basins are likewise giant in size because the scale of oceanic eddies is only ~ 30 km. Parallel computers are now being successfully used to run eddy resolving models in oceanography, planetary meteorology, and terrestrial meteorology.

Our GCM is called the Explicit Planetary Isentropic-Coordinate (EPIC) atmospheric model. The adjective "Explicit" refers to the time step and to the finite-difference algorithm, and also to the goal of keeping the model's source code simple to understand and easy to run on parallel computers by avoiding the global operations (processor-to-processor communications) inherent in spectral schemes and implicit timesteps. "Planetary" refers both to the fact that the model is implemented for the four gas-giant planets and for the middle atmospheres of all planets, jovian or terrestrial (this is the only GCM we are aware of that first queries the user to "Choose a planet"), and to the fact that the model is written using general polar map factors and therefore can be run in oblate spherical, cylindrical, or Cartesian geometry. "Isentropic-Coordinate" refers to the vertical coordinate, potential temperature, which is the meteorologist's preferred form of entropy. This means that for adiabatic motions the model's vertical direction benefits from the motion-following aspects of a Lagrangian scheme, or, in other words, there is no vertical velocity except where there is heating.

1.2. Governing Equations

The governing equations in the EPIC model are the hydrostatic primitive equations with potential temperature, θ , as the vertical coordinate. For simple cases with nonreacting constituents, a potential temperature can straightforwardly be defined in terms of the specific entropy (here "specific" indicates "per unit mass"), S , by

$$dS = C_p d \log \theta, \quad (1)$$

where C_p is specific heat at constant pressure. Meteorologists prefer θ and astrophysicists prefer S , but either is acceptable as an isentropic coordinate. Adopting the former, the first law of thermodynamics takes the form (in the nonreacting case)

$$\dot{\theta} \equiv \frac{D\theta}{Dt} = \frac{\theta}{C_p T} \dot{Q}, \quad (2)$$

where \dot{Q} is the specific heating rate and T is temperature. The quantity $\dot{\theta}$ plays the role of vertical velocity in isen-

tropic coordinates, such that the material (Lagrangian) time derivative is given by

$$\frac{D}{Dt} = \frac{\partial}{\partial t} + (\mathbf{v} \cdot \nabla) + \dot{\theta} \frac{\partial}{\partial \theta}, \quad (3)$$

where t is time, \mathbf{v} is the horizontal velocity, and ∇ is the horizontal gradient operator, both defined along constant- θ surfaces.

For the case when $\kappa = R_{\text{gas}}/C_p$ is a constant, where R_{gas} is the gas constant (including the molecular weight factor), it is straightforward to show that the term $C_p T/\theta$ in (2), called the Exner function, equals $C_p (p/p_0)^\kappa$, where p_0 is a reference pressure (Holton 1992, p. 52). However, for the cold (50–100 K) H_2 atmospheres of Uranus and Neptune, C_p varies by 30% over the temperature range encountered in the tropospheres, and to further complicate matters, it is a different function for the ortho ($\uparrow\uparrow$) and para ($\uparrow\downarrow$) states of H_2 . In other words, ortho hydrogen has a different potential temperature than para hydrogen, and care must be taken to define a mean potential temperature properly for the two-component system. The H_2 component of an atmosphere is treated by adding a prognostic mass variable, the fraction of para hydrogen, f_p , which entails an additional continuity equation and modifications to both the pressure-gradient term in the horizontal momentum equations and to the equation of hydrostatic balance, as described in Section 3. The remainder of this section introduces the standard primitive isentropic-coordinate equations without these f_p extensions. A standard derivation is presented by Dutton (1976).

The horizontal equations of motion (omitting viscosity) may be written

$$\frac{\partial u}{\partial t} = +qvh - \dot{\theta} \frac{\partial u}{\partial \theta} - \frac{1}{r} \frac{\partial}{\partial \phi} (M + K), \quad (4a)$$

$$\frac{\partial v}{\partial t} = -quh - \dot{\theta} \frac{\partial v}{\partial \theta} - \frac{1}{R} \frac{\partial}{\partial \lambda} (M + K). \quad (4b)$$

Here u is zonal wind, v is meridional wind, ϕ is longitude, λ is latitude, ∇ is the horizontal gradient operator, \hat{k} is local vertical direction, Φ is geopotential, $M = C_p T + \Phi$ is Montgomery potential, which in isentropic vertical coordinates plays a role similar to pressure in geometric-height coordinates, and K is specific kinetic energy. The factors r and R are the zonal and meridional radii of curvature (map factors). The quantity q is potential vorticity, which for a shallow atmosphere (i.e., large horizontal-scale dynamics) is given by

$$q = \frac{\zeta + f}{h}, \quad (5)$$

where $\zeta = \hat{k} \cdot (\nabla \times \mathbf{v})$ is relative vorticity, $f = \hat{k} \cdot 2\mathbf{\Omega}$ is planetary vorticity, also called the Coriolis parameter, and $\mathbf{\Omega}$ is the planet's angular velocity. For a shallow, statically stable atmosphere, the variable h in (5) is given by

$$h = -\frac{1}{g} \frac{\partial p}{\partial \theta}, \quad (6)$$

and plays the role of density in (x, y, θ) space. It is sometimes called the layer thickness by analogy to the shallow-water equations, because for adiabatic motions the isentropic-coordinate model reduces to the shallow-water model if one sets $R_{\text{gas}} = 1$ and $C_p = 1$. The EPIC model grew out of a shallow-water model and retains this special case as an option.

Notice that (4) is written with an emphasis on the potential vorticity, q ; specifically, the terms qvh and quh are written as such, rather than in their simpler forms $(\zeta + f)v$ and $(\zeta + f)u$. This is to facilitate numerical conservation of total energy and total potential enstrophy, $\iint fhq^2 dx dy$, regardless of grid resolution, using finite-difference algorithms developed at UCLA by Arakawa and coworkers. These are implemented as discussed in Section 2. Potential vorticity is a particularly important diagnostic variable because it is conserved following the motion for inviscid and adiabatic flow (see the discussion by Holton 1992, pp. 110–112)

$$\dot{\theta} = \frac{D\theta}{Dt} = 0 \Rightarrow \frac{Dq}{Dt} = 0. \quad (7)$$

The map factors, r and R in (4), are kept as general functions of latitude in the EPIC model, so that the user can choose between spherical, cylindrical, and Cartesian coordinates. For the choice of oblate spherical coordinates, they are set to

$$dx = r(\lambda) d\phi; \quad r(\lambda) = \frac{R_e}{(1 + (R_p/R_e)^2 \tan^2 \lambda)^{1/2}}, \quad (8a)$$

$$dy = R(\lambda) d\lambda; \quad R(\lambda) = \frac{r(\lambda)/\cos \lambda}{\sin^2 \lambda + (R_e/R_p)^2 \cos^2 \lambda}, \quad (8b)$$

where R_e is the planet's equatorial radius, R_p is polar radius, and λ is planetographic latitude, which is the angle between the equatorial plane and the local vertical direction (as compared to planetocentric latitude, which is the angle measured through the center of the planet). The Coriolis parameter is given by $f = 2\Omega \sin \lambda$. Also available as a choice at initialization is a local f -plane model (constant- f model) using any number of layers in either Cartesian or cylindrical coordinates. For example, a 50-layer, cylindrically symmetric f -plane configuration has

proved useful for studies of comet-impact gravity waves on Jupiter (Dowling 1995a).

The vertical momentum equation assumes hydrostatic balance, which takes the operational form

$$\frac{\partial M}{\partial \theta} = \frac{C_p T}{\theta}. \quad (9)$$

Conservation of mass, or the continuity equation, is written

$$\frac{\partial h}{\partial t} = -\nabla \cdot (h\mathbf{v}) - \frac{\partial}{\partial \theta} (h\dot{\theta}), \quad (10)$$

which makes apparent the role of h as the density variable. For the case of thermodynamically active H_2 , additional terms appear in (4) and (9) and an additional continuity equation is necessary for f_p , as discussed in Section 3. The model also includes optional continuity equations for the advection of water and methane, in anticipation of moist convective parameterizations.

2. ADIABATIC TERMS

This section describes the discretization of the continuous equations of the previous section for the case of adiabatic motion, $\dot{\theta} = 0$. The EPIC model is based on the finite-difference scheme of Hsu and Arakawa (1990). Their paper includes a detailed discussion of the many benefits of isentropic coordinates, including the elimination of vertical advection when the motions are adiabatic, thereby increasing numerical accuracy, higher resolution of fronts, and better conservation of potential vorticity, total angular momentum, and total energy for vertically discrete systems.

The only major drawback to isentropic coordinates is the technical problem of handling the intersection of constant-entropy surfaces with the ground on terrestrial planets. This problem is of no concern when modeling the gas-giant planets and of little or no concern when modeling middle-atmosphere dynamics on any planet. Another drawback is that in order to use θ as the vertical coordinate, the atmosphere must be statically stable with θ increasing monotonically with altitude. As θ approaches a constant inside a gas giant, h approaches infinity, and in convectively unstable regions h is negative and hence θ can no longer be used as the vertical coordinate. Both the shallow-atmosphere approximations implicit in the assumption of hydrostatic balance and the positive stability restriction prohibit the EPIC model from directly simulating the convecting interior of a gas-giant planet. A future goal is to develop a hybrid-coordinate model that will allow us to form a complete coupled atmosphere–interior model for the gas-giant planets.

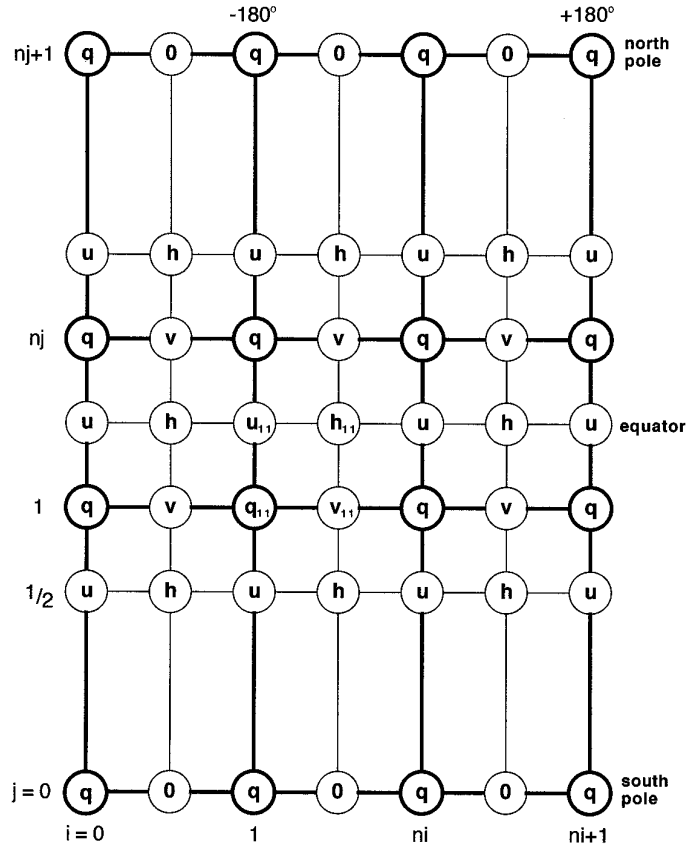


FIG. 1. Horizontal distribution of variables u , v , h , and q in the staggered C -grid. The indexing is for the spherical geometry case. The extra latitude spacing at the poles is discussed in the text.

2.1. Horizontal Discretization

The distribution of the prognostic variables, u , v , and h (or p) on a finite-difference grid within a given isentropic layer is based on the staggered C -grid scheme introduced by Arakawa and Lamb (1977) and illustrated in Fig. 1, which shows the placement and indexing of the variables for spherical geometry. One can see immediately that u , v , h , and q each has its own grid. In the vertical direction, these variables are all at the same level, whereas the pressure, p , which is a vertical integral of h , is staggered in the vertical by half a layer spacing (Fig. 2). The extra spacing in latitude indicated at the poles results from the triangular-shaped segments there (Fig. 3). Notice that q is positioned where the relative vorticity, $\zeta = v_x - u_y$ in the Cartesian case, would most naturally fall. Similarly, the horizontal divergence of velocity, $D = u_x + v_y$ in the Cartesian case, is most naturally defined on the h -grid. We take the convention that the q -grid is the “home” or “whole-number index” grid. Half-index notation is convenient for explanations, but is obviously not used in the code itself, instead, we use the index scheme illustrated by q_{11} , u_{11} , v_{11} , and

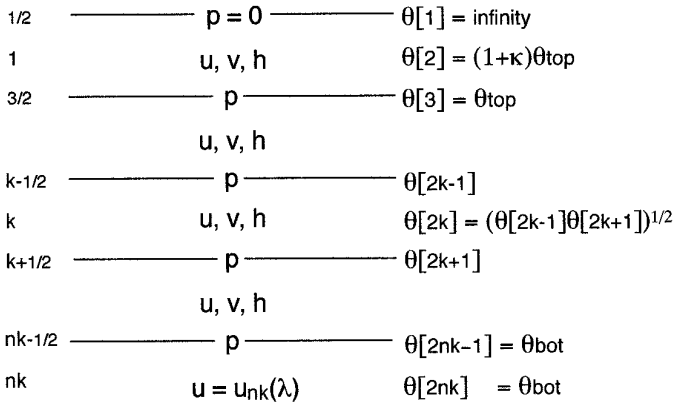


FIG. 2. Vertical distribution of variables in the EPIC model. Half-integer indices are handled by multiplying k by 2, as indicated on the right side of the figure.

h_{11} in the middle of Fig. 1. For example, $U(\mathcal{J}, \mathcal{I})$ refers to the zonal wind located at the same longitude and one-half spacing northward in latitude relative to the position of $Q(\mathcal{J}, \mathcal{I})$. For quantities that are needed on every grid point, such as the Coriolis parameter, $f(y)$, we multiply the corresponding index by 2, such that $f(2*\mathcal{J}+1)$ refers to the grid location $j+1/2$.

One consequence of staggering the variables is that there is an ambiguity as to how to evaluate a variable on a site other than its own, for example, consider (4a) with $\dot{\theta} = 0$. One sees that although the Bernoulli streamfunction, $B \equiv M + K$, is defined on the h -grid and is therefore naturally positioned for the longitudinal gradient in (4a), to calculate $\partial u/\partial t$ one needs to average q , v , and h onto the u -grid. The same holds for $\partial v/\partial t$. Arakawa-type schemes pay particular attention to how q is so averaged, because they seek to control whether total potential enstrophy is dissipated or conserved on the grid, in addition to conserving the total energy. The Hsu and Arakawa scheme is a second-order scheme and hence only two such quantities may be conserved on the grid. It is an art to choose which two because the continuous equations admit an infinite number of such conservation laws.¹ Referring to Fig. 1, one sees that there are four v points surrounding each u point. The quantity vh is defined on the v -grid by averaging the two closest values of h , so there are also four vh points surrounding each u point. Arakawa and Lamb (1981) introduced a

¹ Any h -weighted function of q , say $hF(q)$, integrated over the domain is conserved because $DF(q)/Dt = 0$. Like most inviscid systems in geophysical fluid dynamics, this is a Hamiltonian system, and these “extra” conservation laws, which are called Casimir invariants in the Hamiltonian formalism, render the system noncanonical. However, their existence allows for the construction of powerful theorems on shear stability and eddy saturation, as pioneered by Arnol’d in the 1960s and reviewed by Shepherd (1990).

notation whereby α , β , γ , and δ represent a particular average of nearby q values at each of these vh sites. They also included two uh sites, multiplied by averaged q values denoted by ε , to add flexibility; these terms must vanish as the grid spacing tends to zero for consistency. Analogous weightings are arrayed for (4b). The choice of q averaging determines the scheme’s conservation properties. Arakawa and Lamb (1981) derived the unique scheme that yields conservation of total energy and total potential enstrophy on the grid and showed that this scheme gives superior results for long integrations in many situations. Other alternatives using this q -averaging framework have been investigated as well. For example, Hsu and Arakawa (1990) derived a scheme that conserves total energy and total enstrophy for the case of nondivergent mass flux, which is not quite as general as the Arakawa and Lamb scheme, but is more robust in the sense that it has the added property that it is not corrupted by the presence of massless ($h \rightarrow 0$) regions in the model. Both these options are available in the EPIC model.

There is also an issue regarding how to discretize the continuity equation (10) to calculate $\partial h/\partial t$. A substantial body of work exists on numerical schemes for mass advection, the main problem being that naive schemes do not guarantee positive-definite mass at each point because of truncation errors. The EPIC model implements two versions of the continuity equation, a straightforward version that just calculates the divergence of hv using the map factors appropriate to the geometry at hand, but otherwise without fanfare, and the advanced predictor–corrector scheme developed by Hsu and Arakawa (1990). The straightforward scheme has the advantage that it is simple, but the disadvantage that it does not guarantee positive mass. Hsu and Arakawa’s predictor–corrector scheme has the advantage that it is positive-definite and handles steep gradients, and therefore works well with low-mass (low- h) situations, but the disadvantage is that it is complicated. A model that can handle low-mass situations is desirable for a number of reasons. For example, a relatively clean way to handle the intersection of isentropic surfaces with the ground is to set h to a vanishingly small value past the point of intersection, which is the method employed by Hsu and Arakawa. Low-mass situations can also crop up internally when fronts develop, which is important for terrestrial atmospheres and may be important for gas-giant atmospheres (little work has been done to study fronts in the gas-giant setting). The simulations reported by LeBeau and Dowling (1998) used the straightforward advection scheme, which proved to be adequate. The predictor–corrector scheme took us longer to implement and to test, but is now the model’s default mass-advection scheme.

2.2. Modification at Poles

The main technical challenge for any spherical finite-difference scheme is how to handle the regions about the

geostrophic balance of the abyssal zonal-wind profile. The simplest example is the 1 1/2 layer shallow-water model, also called the “reduced gravity” or “equivalent barotropic” model, which has been applied to Jupiter’s atmosphere by a number of researchers (see Dowling 1995b). The EPIC model is an “ n 1/2” layer model that allows us to model the vertical direction realistically.

The layer value for pressure, p_k , has the following complicated definition in the Hsu and Arakawa scheme

$$p_k = \left(\frac{1}{1 + \kappa} \frac{p_{k+1/2}^{1+\kappa} - p_{k-1/2}^{1+\kappa}}{p_{k+1/2} - p_{k-1/2}} \right)^{1/\kappa}. \quad (14)$$

This choice allows one to “define the available potential energy for a discrete system in a way parallel to what Lorenz (1955) did for the continuous case” (Hsu and Arakawa 1990, p. 1941). We follow Hsu and Arakawa’s specification that layer values of θ be equal to the geometric mean of the interface values, $\theta_k = (\theta_{k-1/2} \theta_{k+1/2})^{1/2}$. The authors mention that one could also use the arithmetic mean, but they favor the geometric mean for reasons that are left unspecified (but note that the geometric mean yields a value of θ_k that is closer to the bottom of the layer, where there is more mass). The scheme requires that the top layer use the special value $\theta_1 = (1 + \kappa) \theta_{3/2}$. For our gas-giant application, the abyssal layer has constant potential temperature, such that $\theta_{nk} = \theta_{nk-1/2}$. There is no restriction that the jump in θ from layer to layer be constant, and we have variously employed layer spacings that are proportional to $\log p$, θ , and $\log \theta$, depending on the problem; the user chooses between these or customizes the layer spacing at initialization.

Both pressure and Montgomery potential are calculated by vertical integrals of h , and therefore require boundary conditions. One can either treat h as the prognostic mass variable and integrate to get p whenever it is needed, starting with the boundary condition $p = 0$ at the top of layer $k = 1$, or else treat p as the prognostic mass variable and take a derivative to get h whenever needed. The version of the EPIC model used by LeBeau and Dowling (1998) carries h as the prognostic variable, a holdover from its shallow-water heritage, but more recent versions of the model use p , because recovering h requires just a local derivative with respect to θ . To calculate M at each time step, we begin in the abyssal layer and work our way upward by applying hydrostatic balance (9). The zonal wind in the abyssal layer, u_{nk} , connects with the Bernoulli streamfunction, $B_{nk} = M_{nk} + K_{nk}$, where K_{nk} is the kinetic energy per unit mass, via the meridional momentum equation (4b) with $v_{nk} = 0$ and $\hat{\theta} = 0$ to yield

$$(\zeta_{nk} + f)u_{nk} = -\frac{1}{R} \frac{\partial}{\partial \lambda} (M_{nk} + K_{nk}). \quad (15)$$

In general, if one were to specify a meridional wind in the deep layer, $v_{nk} \neq 0$, then obtaining B_{nk} would require inverting a two-dimensional elliptic operator for each time step. We do not encounter this complication because we have $v_{nk} = 0$. Integrating (15) with respect to latitude yields

$$M_{nk}(\lambda) = M_{nk}(\lambda_0) + K_{nk}(\lambda_0) - \int_{\lambda_0}^{\lambda} (\zeta_{nk} + f)u_{nk} R d\lambda - K_{nk}(\lambda). \quad (16)$$

To center the integrand properly, the code uses (16) with $(\zeta + f)u$ averaged onto the v -grid in the same manner as quh for the v momentum equation (4b). The starting latitude is typically the south pole, $\lambda_0 = -\pi/2$, where $M \equiv C_p T + gz$ is assigned some convenient value, say $M_{nk}(\lambda_0) = (C_p T)_0$ or $M_{nk}(\lambda_0) = 0$; for gas-giant simulations it typically does not matter since the zero of the geopotential, gz , is usually arbitrary. Having calculated $M_{nk}(\lambda)$, the rest of the layers are filled in upward from the bottom by integrating (9) with respect to θ . In practice, $M_k(\phi, \lambda)$ is used everywhere it is needed for the calculations of layer k , and then it is integrated in place to yield $M_{k-1}(\phi, \lambda)$, and so on, such that we only allocate a two-dimensional (horizontal) array for M .

2.4. Time Discretization

The time discretization of the momentum equation in the EPIC model is not the one described in Hsu and Arakawa (1990)—for expediency, they used the Euler-backward scheme, which is easy to implement and has the desirable property that it damps high-frequency gravity waves (Haltiner and Williams 1980). However, it requires two function evaluations per time step and hence runs twice as slowly as schemes that require a single function evaluation. For the EPIC model, we use the third-order Adams–Bashforth scheme to integrate u and v . Durran (1991) compares this time step favorably against other popular single-evaluation schemes like the leapfrog time step. Unlike the leapfrog time step, it is appropriate for dissipative terms as well as for conservative terms, does not suffer from a time-splitting numerical instability, and is more accurate. It uses more memory than the leapfrog time step because it requires two previous time tendencies for each prognostic variable, however, memory is no longer a constraining resource, and so the third-order Adams–Bashforth scheme fits our needs perfectly. The value of a prognostic variable, u for example, at time $t + \Delta t$ is marched forward by

$$u[t + \Delta t] = u[t] + \frac{\Delta t}{12} \left(23 \frac{\partial u}{\partial t}[t] - 16 \frac{\partial u}{\partial t}[t - \Delta t] + 5 \frac{\partial u}{\partial t}[t - 2\Delta t] \right). \quad (17)$$

All the information needed to characterize one time step of an EPIC model run is stored in a single file, including the model size and all parameters. The model produces such a file on demand, and it may be used to restart a run. The generic file name `epic.dat` typically is used for an initial file, $t = 0$, and subsequent file names are labeled by the simulation time. The model begins by reading the top of an `epic.dat` file to determine the size of the grid, ni , nj , nk , after which it dynamically allocates all the necessary arrays. In this manner, we do not need to recompile the code to change the number of layers, the horizontal resolution, or the range of pressures, latitudes, or longitudes covered by the model. The model assigns values to utility arrays like f , θ , and the map factors r and R , and then reads in the remainder of `epic.dat` to get the initial values of all prognostic variables and their tendencies. An `epic.dat` file valid for time t stores $u[t]$, $\partial u/\partial t[t - \Delta t]$, and $\partial u/\partial t[t - 2\Delta t]$ for each grid point and analogous arrays for the other prognostic variables. The tendencies $\partial u/\partial t[t]$, $\partial v/\partial t[t]$, and $\partial h/\partial t[t]$ are not stored to disk but are generated from $u[t]$, $v[t]$, and $h[t]$.

As mentioned above, the time discretization of the continuity equation in the EPIC model follows one of two options. LeBeau and Dowling (1998) use the straightforward evaluation of the continuity equation and apply (17) to h as well as to u and v . The second option uses Hsu and Arakawa's predictor-corrector to get a new h , call it $h^{\text{pc}}[t + \Delta t]$, which is then converted into a third-order Adams-Bashforth tendency by reversing (17)

$$\begin{aligned} \frac{\partial h^{\text{pc}}}{\partial t}[t] &= \frac{1}{\Delta t} \frac{12}{23} (h^{\text{pc}}[t + \Delta t] - h[t]) \\ &+ \frac{16}{23} \frac{\partial h}{\partial t}[t - \Delta t] - \frac{5}{23} \frac{\partial h}{\partial t}[t - 2\Delta t]. \end{aligned} \quad (18)$$

We do this because we need the freedom to add to the mass tendency the $\partial(h\theta)/\partial\theta$ term, plus hyperviscosity terms to control numerical instability and source-sink terms for continuity equations of thermodynamic mass variables such as f_p . The total mass tendency is then used in (17) to march the mass variable forward in time.

3. HYDROGEN THERMODYNAMICS

Voyager IRIS observations indicate that the hydrogen ortho/para ratio is not in thermodynamical equilibrium on the gas-giant planets, so entropy is not a unique function of T and p . The problem is that cold H_2 is a two-component system in which ortho hydrogen and para hydrogen have different heat capacities and different potential temperatures. Appendix A describes how we average to obtain a mean potential temperature, θ . Henceforth, we adopt θ as defined in Appendix A to be the model's vertical coordi-

nate. The ensuing modifications to the standard isentropic-coordinate equations of Section 1 are described in this section.

3.1. Pressure Gradient

To express the horizontal equations of motion in isentropic coordinates, the pressure gradient is written in the form

$$\frac{1}{\rho} \nabla|_z p = \frac{1}{\rho} \left(\nabla|_{\theta} p - \frac{\partial p}{\partial z} \nabla|_{\theta} z \right), \quad (19)$$

where $z(x, y, \theta, t)$ is the height of a surface of constant θ . Pressure may be related to θ , T , and the para hydrogen fraction, f_p (Appendix A). On a constant- θ surface the ideal-gas law and hydrostatic balance combine with (19) to yield

$$\frac{1}{\rho} \nabla|_z p = C_p \nabla|_{\theta} T + C_p^{(R)} \frac{T}{\theta} \frac{\partial \theta}{\partial f_p} \nabla|_{\theta} f_p + \nabla|_{\theta} g z, \quad (20)$$

where $C_p^{(R)}$ is a constant reference heat capacity and g is gravity, which is assumed to be constant since we are working with the shallow-atmosphere approximation. The mean heat capacity, C_p , is a function of temperature. It appears outside of the gradient operator in the first term on the right-hand side of (20), but this can be simplified by introducing the enthalpies, H , for each constituent and for the mixture, which we reference to zero at zero temperature

$$H_i = \int_0^T C_{pi}(T') dT', \quad H = \sum_i X_i H_i, \quad (21)$$

where X_i is the concentration of constituent i by number. Express X_i in terms of f_p for ortho hydrogen and para hydrogen

$$X_o = (1 - f_p) X_{\text{H}_2}, \quad X_p = f_p X_{\text{H}_2}, \quad (22)$$

where X_{H_2} is the number concentration of H_2 in the atmosphere. The enthalpy gradient is

$$\nabla|_{\theta} H = C_p \nabla|_{\theta} T + X_{\text{H}_2} (H_p - H_o) \nabla|_{\theta} f_p. \quad (23)$$

Substituting (23) into (20) yields

$$\frac{1}{\rho} \nabla|_z p = \nabla|_{\theta} M + \left(C_p^{(R)} \frac{T}{\theta} \frac{\partial \theta}{\partial f_p} - X_{\text{H}_2} (H_p - H_o) \right) \nabla|_{\theta} f_p, \quad (24)$$

where $M \equiv H + g z$ is the Montgomery potential that appears in the standard formulation (4). The two-compo-

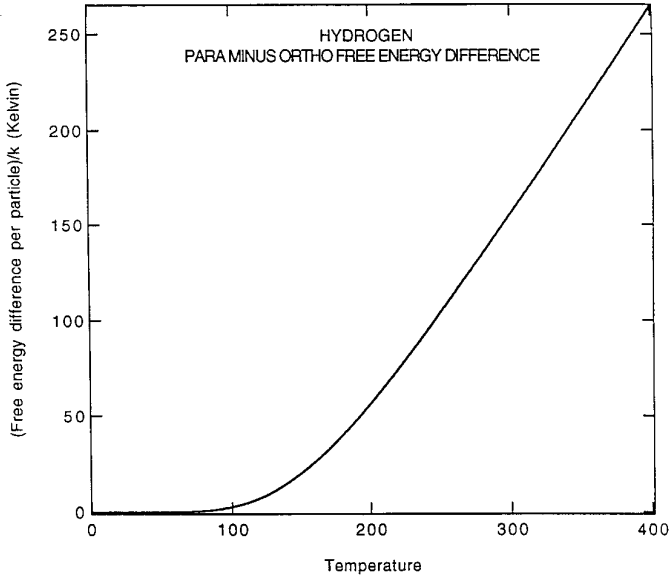


FIG. 4. The Gibbs free energy difference, $F_{H_2}(T)$, as defined in the text. Units are per particle, divided by k_B .

ment modification is the $\nabla|_{\theta} f_p$ term in (24). Its coefficient can be simplified (see B.5)

$$\begin{aligned} C_p^{(R)} \frac{T}{\theta} \frac{\partial \theta}{\partial f_p} &= X_{H_2} C_{pH_2}^{(R)} T (\log \theta_p - \log \theta_o), \\ \Rightarrow C_p^{(R)} \frac{T}{\theta} \frac{\partial \theta}{\partial f_p} - X_{H_2} (H_p - H_o) &= X_{H_2} \{ (TS_p - H_p) \\ &\quad - (TS_o - H_o) \} |_{\theta}^T, \\ &\equiv X_{H_2} F_{H_2}(T). \end{aligned} \quad (25)$$

The quantity $F_{H_2}(T)$ in (25) is the difference between the Gibbs free energies of para and ortho hydrogen, each normalized to zero at zero temperature. Pressure dependence cancels out, so that F_{H_2} is only a function of temperature (Fig. 4). The pressure gradient (20) becomes

$$\frac{1}{\rho} \nabla|_z p = \nabla|_{\theta} M + F(T) \nabla|_{\theta} f_p, \quad (26)$$

where $F(T) = X_{H_2} F_{H_2}(T)$ has units of energy per mass and depends on the mixture via the factor X_{H_2} as well as on the hydrogen thermodynamics. In the EPIC model, f_p and M are both carried on the h -grid, which means that the gradients in (26) are naturally centered for the momentum equation.

3.2. Hydrostatic Equation

In isentropic coordinates, hydrostatic balance is most conveniently written in terms of derivatives with respect

to θ . The objective is to obtain an expression for $\partial M / \partial \theta$. We start with

$$\frac{\partial p}{\partial \theta} = -\rho g \frac{\partial z}{\partial \theta}. \quad (27)$$

Writing $d\theta$ in terms of dT , dp , and df_p , the derivative of pressure can be rewritten

$$\begin{aligned} \frac{\partial p}{\partial \theta} \Big|_{x,y,t} &= -\frac{C_p^{(R)}}{R_{\text{gas}}} \frac{p}{\theta} + \frac{C_p}{R_{\text{gas}}} \frac{p}{T} \frac{\partial T}{\partial \theta} \Big|_{x,y,t} \\ &\quad + \frac{C_p^{(R)}}{R_{\text{gas}}} \frac{p}{\theta} \frac{\partial \theta}{\partial f_p} \Big|_{p,T} \frac{\partial f_p}{\partial \theta} \Big|_{x,y,t}. \end{aligned} \quad (28)$$

In parallel with the development leading to (26), the last two terms of (28) can be written in terms of the vertical enthalpy gradient. Using (27), (28), and $M = H + gz$, the modified hydrostatic equation is

$$\frac{\partial M}{\partial \theta} + F(T) \frac{\partial f_p}{\partial \theta} = C_p^{(R)} \frac{T}{\theta}. \quad (29)$$

4. DIABATIC TERMS

This section describes the heating processes in the model, which manifest themselves as vertical mass flux, $h\dot{\theta}$. How the different heating terms change with respect to latitude and time can be one of the more fascinating aspects of a model's evolution. At any time, the user can direct the model to record the separate heating contributions for a given time step into a file for later analysis.

4.1. Vertical Placement of Heating

We depart from the scheme of Hsu and Arakawa (1990) on the vertical placement of the heating rate, \dot{Q} , in (2). They evaluate $h\dot{Q}$ inside the layers and $h\dot{\theta}$ on the interfaces. The no-flux boundary condition, $h\dot{\theta} = 0$, is applied at the top of the top layer and at the bottom of the bottom layer, and they specify (their 5.48)

$$\begin{aligned} (\theta_k - \theta_{k+1/2}) \left[h\dot{\theta} \frac{C_p^{(R)} T}{\theta} \right]_{k+1/2} &+ (\theta_{k-1/2} - \theta_k) \left[h\dot{\theta} \frac{C_p^{(R)} T}{\theta} \right]_{k-1/2} \\ &= (\theta_{k-1/2} - \theta_{k+1/2}) [h\dot{Q}]_k, \end{aligned} \quad (30)$$

to find $h\dot{\theta}$ at each of the interior interfaces, given $h\dot{Q}$ in the layers. We have detected several problems with this specification. In the first place, there is no mention of how to handle the right-hand side of (30) for the top layer $k = 1$, the problem being $\theta_{1/2} = \infty$. One way to proceed is to set $[h\dot{\theta} C_p T / \theta]_{1+1/2} = [h\dot{Q}]_1$. In the second place, (30) is overspecified. If we start with $h\dot{\theta} = 0$ at the top and

work our way down using (30) to evaluate $h\dot{\theta}$ on the interfaces, then the bottom value is specified before we can apply the no-flux boundary condition there. There is no mention by Hsu and Arakawa of the need to restrict $h\dot{Q}$ to avoid this clash. One can let $h\dot{\theta}$ at the bottom of the model (which is the top of the abyssal layer, $k = nk - 1/2$) be whatever came from (30), since there is no reason to prevent mass flux at this interface in a gas-giant atmosphere. In the third place, we found that staggering $h\dot{Q}$ and $h\dot{\theta}$ tends to produce an undesirable sign flip of $h\dot{\theta}$ from layer to layer. This numerical instability can grow to destroy a simulation and was the cause of much consternation in the early development of the EPIC model. Consider the case of constant $[h\dot{Q}]_k$, for example.

For all of the above reasons, we have abandoned (30) and the staggering of $h\dot{Q}$ and $h\dot{\theta}$. Instead, we just specify $h\dot{Q}$ at the interfaces directly where $h\dot{\theta}$ is needed and find $h\dot{\theta}$ by

$$[h\dot{\theta}]_{k+1/2} = \left[h\dot{Q} \frac{\theta}{C_p^{(R)}T} \right]_{k+1/2}. \quad (31)$$

This clears up all the problems encountered with (30). Note that h is not cancelled out on both sides of (31) because the quantity $h\dot{\theta}$ is called for in the equations rather than $\dot{\theta}$. Equation (31) necessitates a definition of h at the interfaces, which we take to be

$$h_{k+1/2} = \begin{cases} -\frac{1}{g} \frac{p_k - p_{k+1}}{\theta_k - \theta_{k+1}} & 1/2 < k + 1/2 < nk - 1/2, \\ -\frac{1}{g} \frac{p_k - p_{k+1/2}}{\theta_k - \theta_{k+1/2}} & k + 1/2 = nk - 1/2, \end{cases} \quad (32)$$

where p_k is the layer value of pressure given by (14). (Note that the quantity $h_{1/2}$ is never needed.)

We are now ready to consider heat sources. As is typical with GCMs, complicated radiative and thermodynamic processes are represented by simple relaxation parameterizations in the EPIC model.

4.2. Ortho–Para Hydrogen Conversion

The difference in heat capacities between ortho and para hydrogen gives rise to a latent heat source when conversion occurs. The first law of thermodynamics gives (in per-particle form)

$$dq = du + pd \left(\frac{k_B T}{p} \right). \quad (33)$$

By use of (A.4), (A.8), (A.9), and (A.10) the right-hand side of (33) can be written in terms of differentials of θ

and f_p . Switching over to per-mass form again, $\theta = D\theta/Dt$ then satisfies

$$\frac{C_p^{(R)}T}{\theta} \dot{\theta} = \dot{Q} + X_{\text{H}_2} [(U_o - U_p) - C_{p\text{H}_2}^{(R)} T \log(\theta_o/\theta_p)] \frac{Df_p}{Dt}, \quad (34)$$

where U_i is internal energy per mass. In (34) it is assumed that the total hydrogen concentration, ortho plus para, is constant. The internal energy difference, $U_o - U_p$, is due to the rotational component only, and is positive because ortho ($\uparrow\uparrow$) rotational energy cannot drop to zero even at low temperature. Note that the absolute energies are to be used in (34), not the enthalpies introduced above that were referenced to zero at zero temperature.

The factor Df_p/Dt in (34) is the conversion rate from ortho to para hydrogen following the motion. Following standard practice, we parameterize this as a simple relaxation to equilibrium

$$\frac{Df_p}{Dt} = -\frac{1}{\tau_{fp}} (f_p - f_{peq}(T)). \quad (35)$$

The equilibrated fraction of para hydrogen is a function of temperature, $f_{peq}(T)$. The relaxation time, τ_{fp} , is not well constrained for the gas-giant planets, which is the reason for most of the uncertainty regarding this latent-heat source. A discussion may be found in Conrath and Gierasch (1984). The nominal EPIC model assumes

$$\frac{1}{\tau_{fp}} = \frac{1}{3 \times 10^8 \text{ s}} \left(\frac{p}{1 \text{ bar}} \right), \quad (36)$$

which is an estimate of the conversion rate from hydrogen–hydrogen collisions and is considered to be as slow as the process is likely to proceed, since any paramagnetic contaminants will speed up the conversion.

The EPIC model is currently being used to interpret the disequilibrium of para hydrogen observed on all four gas-giant planets, especially Uranus and Neptune, and to simulate this disequilibrium (Dowling *et al.* 1996). A complete description of our results will appear in a forthcoming paper.

4.3. Newtonian Cooling

As is common practice for GCMs, radiative cooling is parameterized by Newtonian cooling, which is the relaxation process

$$\dot{Q}_{\text{rad}} = -\frac{1}{\tau_{\text{rad}}} C_p^{(R)} (T - T_{\text{cq}}), \quad (37)$$

where T_{eq} and τ_{rad} are the radiative–equilibrium temperature profile and time constant, respectively. Hsu and Arakawa (1990) show that in isentropic coordinates, Newtonian cooling may be expressed simply as

$$h\dot{\theta} = -\frac{1}{\tau_{\text{rad}}}(p - p_{\text{eq}}). \quad (38)$$

We used (38) initially because of its simplicity (p is a prognostic variable), but eventually reverted back to the form (37) for the EPIC model, because radiation is most straightforwardly parameterized in terms of temperature. This means that we evaluate $T(p, \theta, f_p)$, $\tau_{\text{rad}}(p)$, and $T_{\text{eq}}(\lambda, p, t)$, at each heating point and for each time, t . Time variation enters through the equilibrium temperature, T_{eq} , which is diurnally averaged but seasonally varying. For the gas-giant planets, evaluation of τ_{rad} and T_{eq} are done using cubic-spline interpolations on data tables produced by Conrath *et al.* (1990).

4.4. Additional Heat Sources

Another heating source in the model is heat conduction. For most tropospheres and stratospheres this is negligible and is not included. However, when radiative cooling is weak, such as in Triton’s tenuous atmosphere, then heat conduction is included.

An artificial “hasten” factor is available that multiplies all relaxation rates in the model and is nominally set to unity so that it has no effect. Setting this factor to a value larger than unity allows one to narrow artificially the separation between dynamical time scales, which are measured in hours, days, and weeks, and diabatic time scales, which are measured in decades and centuries for the gas-giant planets. This is useful during the exploratory phase of a new project.

It is often the case that we do not want the overall heating to cause the average mass in each layer to change, or in other words, we do not want the layer interfaces to drift upward or downward. Mathematically, this means $\iint h\hat{\theta}R \, d\phi \, d\lambda = 0$, and is easy to ensure by making an adjustment to the heating rate after all the normal contributions to the right-hand side of (34) have been summed together. Conceptually, we view this adjustment as a refinement of the radiative–equilibrium temperature profile and so introduce it (when requested) by modifying T_{eq} in (37) to be

$$T_{\text{eq}}^{(\text{adj})} = T_{\text{eq}} - \frac{\iint \frac{1}{T} \dot{Q}_{\text{tot}} hR \, d\phi \, d\lambda}{\iint \frac{1}{T} \frac{C_p^{(R)}}{\tau_{\text{rad}}} hR \, d\phi \, d\lambda}, \quad (39)$$

where \dot{Q}_{tot} is the sum of all preadjustment contributions to the right-hand side of (34), including the Df_p/Dt latent-heat term. The size of the adjustment is monitored and the user is alerted if it exceeds a predefined limit.

The latent-heat contributions from phase changes of water (Jupiter and Saturn) and methane (Uranus and Neptune), and moist-convective parameterizations, are currently under development for the model. We plan to study how these processes influence the zonal winds in global models of gas-giant atmospheres and to compare them with terrestrial models.

5. NUMERICAL STABILITY TERMS

In this section, we discuss the various terms in the model that are introduced to prevent numerical instability. All of these represent standard practice in GCM modeling, and so we only report the choices we have settled on and why. From an engineering point of view, our efforts to simulate Neptune’s atmospheric dynamics have proved to be an excellent shakedown of the EPIC model. We were left with the impression that a GCM that can successfully simulate Neptune without succumbing to numerical instability can handle anything. For clarity of presentation, we express the equations in this section assuming adiabatic motion, $\hat{\theta} = 0$.

5.1. Hyperviscosity

In addition to the inviscid terms in the horizontal momentum equation (4), we follow the standard procedure of adding hyperviscosity to parameterize subgridscale processes and to dampen high-frequency computational modes that can accumulate to cause numerical failure. The hyperviscosity terms take the form $(-1)^{n-1} \nu_{2n} \nabla^{2n} \mathbf{v}$, where n is a positive integer, such that the adiabatic momentum equation in vector form is

$$\frac{\partial \mathbf{v}}{\partial t} = -q\hat{k} \times (h\mathbf{v}) - \nabla(M + K) + \sum_{n=1}^{n_{\text{max}}} (-1)^{n-1} \nu_{2n} \nabla^{2n} \mathbf{v}. \quad (40)$$

Spectral models allow very high values of n to be applied with little additional computational burden, because the ∇^{2n} operator reduces to a simple multiplication in transformed space. This is not the case for finite-difference schemes, where we actually must calculate the Laplacians. We have nominally limited the range of n to $n_{\text{max}} = 3$ in the EPIC model; the user can elect to apply any combination of ν_2 , ν_4 , and ν_6 to a run. Molecular viscosity in gas-giant atmospheres is negligible on week-to-month time scales for large-scale dynamics, and all the runs in LeBeau and Dowling (1998) set $\nu_2 = 0$. The time scale for viscous decay is proportional to the length scale of the disturbance raised to the $2n$ th power. Higher-order viscosities are preferable

to ν_2 for damping computational modes because they act much more strongly on grid-scale features than on large-scale features such as vortices and jets. For Neptune simulations we generally use ν_6 hyperviscosity, but occasionally turn on ν_4 hyperviscosity during initial vortex adjustments to “burn off” transient grid-scale waves.

The Laplacian operator in (40) acts on the horizontal velocity vector, $\mathbf{v} = (u, v)$. Hyperviscosity is also applied to the continuity equation(s), in which case it involves the usual Laplacian of a scalar variable. For the case of (40), however, the components are not what one might expect:

$$\nabla^2 \mathbf{v} \neq (\nabla^2 u, \nabla^2 v). \quad (41)$$

One way to see the complication is to write $\nabla^2 \mathbf{v} = \nabla \cdot (\nabla \mathbf{v})$ and note that $\nabla \mathbf{v}$ is a tensor rather than a vector. We record the details here because they are not easy to find elsewhere (but see Gill 1982, pp. 91–93, for a related discussion). As is usually the case, one seeks an identity that replaces the troublesome term with known quantities. It is easiest to think in three dimensions, to which we now jump without changing notation. The appropriate identity is

$$\nabla^2 \mathbf{v} \equiv \nabla(\nabla \cdot \mathbf{v}) - \nabla \times (\nabla \times \mathbf{v}). \quad (42)$$

The right-hand side of (42) is composed of only vector and scalar quantities and serves as the definition of the Laplacian of a three-dimensional vector. We have not included vertical hyperviscosity in the EPIC model, which is to say, our vertical hyperviscosity (and viscosity) coefficients are zero. Since the EPIC model is a shallow-atmosphere model, for this discussion the vertical coordinate may be approximated as Cartesian with a map factor of unity. It is therefore sufficient to consider the barotropic case of (42), where $\mathbf{v} = (u, v, 0)$ and u and v do not depend on height. The vertical component of relative vorticity and the horizontal divergence are given in general by

$$\zeta = \frac{1}{rR} \left(\frac{\partial}{\partial \phi} (Rv) - \frac{\partial}{\partial \lambda} (ru) \right), \quad (43a)$$

$$D = \frac{1}{rR} \left(\frac{\partial}{\partial \phi} (Ru) + \frac{\partial}{\partial \lambda} (rv) \right). \quad (43b)$$

These expressions are easy to remember with the help of a sketch of an area segment if one recalls that vorticity is circulation divided by area and divergence is outward flux divided by area. The general definition for the curl of a vector, $\mathbf{v} = (u, v, w)$, with coordinates (x_1, x_2, x_3) , map factors (h_1, h_2, h_3) , and unit vectors $(\hat{e}_1, \hat{e}_2, \hat{e}_3)$, is given by the determinant (GradshTEYN and Ryzhik 1980, p. 1085)

$$\nabla \times \mathbf{v} \equiv \frac{1}{h_1 h_2 h_3} \begin{vmatrix} h_1 \hat{e}_1 & h_2 \hat{e}_2 & h_3 \hat{e}_3 \\ \partial/\partial x_1 & \partial/\partial x_2 & \partial/\partial x_3 \\ h_1 u & h_2 v & h_3 w \end{vmatrix}. \quad (44)$$

Our geometry has $\mathbf{v} = (u, v, 0)$, $(x_1, x_2, x_3) = (\phi, \lambda, z)$, and $(h_1, h_2, h_3) = (r, R, 1)$. Applying (44) twice to $\mathbf{v} = (u, v, 0)$ yields

$$\nabla \times (\nabla \times \mathbf{v}) = \left(\frac{1}{R} \frac{\partial}{\partial \lambda} \zeta, -\frac{1}{r} \frac{\partial}{\partial \phi} \zeta, 0 \right). \quad (45)$$

Therefore, using (42), the components of the horizontal Laplacian of velocity are

$$\nabla^2 \mathbf{v} = \left(-\frac{1}{R} \frac{\partial}{\partial \lambda} \zeta + \frac{1}{r} \frac{\partial}{\partial \phi} D, \frac{1}{r} \frac{\partial}{\partial \phi} \zeta + \frac{1}{R} \frac{\partial}{\partial \lambda} D \right). \quad (46)$$

Details of our implementation of (46), including code fragments for the staggered C -grid, are given in LeBeau’s (1997) Ph.D. thesis. With regard to parallel computing, we gain a simplification in the bookkeeping by calculating the Laplacian in two steps: first we compute ζ and D and then update the processor edges, and second we compute the components in (46) and update the processor edges a second time. This allows us to work with single-thickness edge padding (the boundary data that overlaps with adjacent processors). If we carried out the Laplacian operation as a single step we would need double-thick pads for each processor.

5.2. Stability Criteria for Hyperviscosity

Along with the Courant–Friedrichs–Lewy (CFL) limit on the time step associated with the fastest information-carrying waves in the model (Haltiner and Williams 1980), there is also an analogous limit on the time step associated with the hyperviscosity terms. This means that for a given time step and grid spacing, we can choose the hyperviscosity coefficients to be only so strong. We therefore need to calculate the maximum allowable values in order to prevent accidentally exceeding them, which would ironically cause the hyperviscosity terms to become numerically unstable. The maximum coefficients may be estimated by applying a Von Neumann stability analysis (Fletcher 1991). To start, consider the finite-difference equation

$$u_{i,j}^{(n+1)} = u_{i,j}^{(n)} + \Delta t f(u), \quad (47)$$

where $u_{i,j}^{(n)}$ is the finite-difference velocity at time step n and longitude–latitude location (i, j) , and $f(u)$ is the linear, finite-difference representation of the viscous term (not to be confused with the Coriolis parameter). Define \hat{u} as the

exact solution and $\xi = u - \hat{u}$ as the error. Then the linearity of f gives

$$\xi_{i,j}^{(n+1)} = \xi_{i,j}^{(n)} + \Delta t f(\xi). \quad (48)$$

The analysis proceeds by writing the error in the form

$$\xi_{i,j}^{(n)} = G^n e^{I(i\Delta x)} e^{J(j\Delta y)}, \quad (49)$$

where $I = \sqrt{-1}$, hence $G = \xi^{(n+1)}/\xi^{(n)}$. For the error not to increase requires that $|G| \leq 1$, which sets a criterion for numerical stability.

The EPIC model uses the third-order Adams–Bashford time step (17), which in this notation gives

$$u^{(n+1)} = u^{(n)} + \frac{\Delta t}{12} (23f^{(n)}(u) - 16f^{(n-1)}(u) + 5f^{(n-2)}(u)). \quad (50)$$

In (50), set $f^n(u)$ equal to the Cartesian, centered finite-difference form of $\nu_2 \nabla^2 u$ and apply (49). Then divide through by $G^n e^{I(i\Delta x)} e^{J(j\Delta y)}$ to obtain

$$G^3 - (1 - 23s)G^2 - 16sG + 5s = 0, \quad (51)$$

where here s (not to be confused with specific entropy) is given by

$$s = \frac{1}{3} \frac{\nu_2 \Delta t}{(\Delta x)^2} [\sin^2(\Delta x/2) + \delta^2 \sin^2(\Delta y/2)]; \quad \delta = \Delta x/\Delta y. \quad (52)$$

Equation (51) is a cubic equation in G that can be solved to get equations for the roots $G(s)$, which reveal that $|G| \leq 1$ is satisfied so long as $s \leq 1/22$. The constraint is therefore

$$0 \leq \nu_2 \leq \frac{3/22}{1 + \delta^2} \frac{(\Delta x)^2}{\Delta t}. \quad (53)$$

For higher powers of hyperviscosity, the cubic equation (51) is the same but the definition of s in (52) changes. The general constraint is

$$0 < \nu_{2n} \leq \frac{3/22}{(3n-2)(1+\delta^2)^n} \frac{(\Delta x)^{2n}}{\Delta t}, \quad n \geq 1. \quad (54)$$

The EPIC model uses

$$\nu_{2n}^{\max} \frac{\Delta t}{(\Delta x)^{2n}} = \left\{ \frac{1}{3}, \frac{1}{30}, \frac{1}{240}, \frac{1}{800} \right\}, \quad n = \{0, 1, 2, 3\}, \quad (55)$$

which for $n \geq 1$ is (54) with $\delta \sim 1$ and an engineering safety multiplier of order 2 and ν_0^{\max} is the appropriate

coefficient for Rayleigh drag, which is not captured by the algebra leading to (54). If the hyperviscosity coefficients are nondimensionalized by ν_{2n}^{\max} , then the typical values used in LeBeau and Dowling (1998) are $\nu_{0,2,4} = 0$ and $\nu_6/\nu_6^{\max} = 0.7$. The user is prompted to choose these at the beginning of a run. The grid spacings are evaluated at 45° latitude for a spherical model, which marks the start of high-latitude filtering.

5.3. High-Latitude Filtering

As previously mentioned, on a longitude–latitude grid the zonal spacing, $\Delta x = r d\phi$, shrinks to zero at the poles and takes with it the computationally stable (CFL) time step. The EPIC model handles this problem by the standard technique of low-pass zonal filtering of variables near the poles. Low-pass filtering effectively reduces the zonal grid spacing and allows the model to run with a reasonable time step. There have been many suggestions made for what type of filtering to use; the introduction to Kar *et al.* (1994) contains a good summary. The goal of many filtering schemes is to selectively filter only those zonal terms that give rise to gravity waves. Some filtering schemes introduce spurious sources of vorticity and divergence at the poles because they do not exactly preserve the irrotational properties of the pressure gradient term. Kar *et al.* propose a filtering scheme that does not suffer this problem, which they describe in their Appendix B. Their method is to filter the pressure before calculating the pressure gradient to overcome the effect of the latitudinal dependence of the filter function.

Over the course of development and testing of the EPIC model, we have spent considerable time trying these selective filtering schemes, including the one described by Kar *et al.* Unfortunately, we have found that they are not reliable, at least not in the gas-giant planet context. Numerical instability inevitably leaks through the defenses at the poles and kills our runs. Neptune’s zonal winds are the strongest found in the Solar System, and this is likely to be a major source of the problem. Therefore, we use a filter that works in practice, even though it may suffer minor problems like spurious vorticity creation at the poles.

It is not a good idea to filter the prognostic variables themselves, because the smoothing would then be cumulative. Instead, we filter the tendencies. We find that if we leave one or another tendency variable out of the filtering (like mass), problems inevitably arise, so we filter all of them. The filtering is applied only poleward of 45° . Each tendency variable is first transformed by a Fast-Fourier Transform (FFT), then multiplied by a zonal filtering function, $\Gamma_j(s)$, where here s is the spatial frequency index corresponding to longitude, and then transformed back. The expression for $\Gamma_j(s)$ involves $\Delta x = r(\lambda)\Delta\phi$ and $\Delta y = R(\lambda)\Delta\lambda$, which are expressed in the model code in terms

of their reciprocals, $m = 1/\Delta x$, and $n = 1/\Delta y$ (to exploit the fact that multiplication is significantly faster than division). The expression for $\Gamma_j(s)$ we have settled on is

$$\Gamma_j(s) = \min \left\{ 1, \frac{[(n/n_0)/(m/m_0)]^{\alpha_1}}{\sin^{\alpha_2}[\Delta\phi(s-1)/2]} \left[\frac{mn}{m \times n} \right]_j \right\}; \quad (56)$$

$$s = \{2, ni/2 + 1\},$$

where $\alpha_1 = 1$, $\alpha_2 = 1/2$, $m_0 = 1/\Delta x(45^\circ)$, and $n_0 = 1/\Delta y(45^\circ)$. The factor $mn/(m \times n)$ in (56) requires explanation because it appears to equal just unity. In fact, it does equal unity in the interior of the model, but not so at the poles, where the reciprocal area element, denoted mnemonically by mn , is not equal to $m \times n$ because of the triangular shape of the elements (Fig. 3), a subtlety noted in Appendix B of Kar *et al.* (1994). Arakawa and Lamb (1977, p. 250) used a filter function similar to (56), but with $\alpha_1 = \alpha_2 = 1$, and they only applied it to the zonal pressure gradient and zonal mass flux terms, a scheme which they derived from a linear stability analysis of the system equations. Kar *et al.* (1994) used $\alpha_1 = \alpha_2 = 2$, which they applied only to the pressure fields used in both the zonal and meridional pressure gradient terms. Ours is a pragmatic approach. By taking the square root, $\alpha_2 = 1/2$, we flatten the filter function at the high-frequency end, thereby strengthening the effect of the filter, which we have found in practice to be necessary for Neptune.

With regard to parallel computing, because an FFT is a global process that requires a large amount of communication between grid points, it is relatively inefficient to divide the zonal direction among processors when using high-latitude zonal filtering. For different reasons, the vertical direction is notoriously difficult to subdivide, because of the vertical integrals related to calculating the pressure and Montgomery potential. Therefore, the current version of the EPIC model spreads the work across parallel processors by assigning one or more complete longitude-height planes to each processor. Although one generally wishes to minimize the surface area to volume ratio of processor subdomains in order to minimize the edge communication between processors, our long subgrids in practice are not as inefficient as they may sound, because there is significant communications overhead on most parallel systems that does not scale with the length of the edges. To date, the number of processors available to us has not significantly exceeded our typical number of latitude gridpoints, n_j , such that our practice of cutting only in the meridional direction has not prevented us from taking full advantage of available computer resources.

5.4. Damping in the Top Layers

A fundamental problem in numerical atmospheric modeling arises from the fact that atmospheres do not have

top boundaries, but atmospheric models do. Ideally, one wants a model that does not reflect waves impinging upon its top boundary. For linear-wave problems one can apply a precise radiation boundary condition that eliminates reflection for each different wavelength. An analogy from electronics is the impedance-matching terminator placed at the end of a coaxial cable. For fully nonlinear problems, this is a much more difficult proposition. A good discussion can be found in the introduction of Klemp and Lilly (1978). For the EPIC model, we follow the standard practice of adding a dissipation term to the upper layers to dampen upward propagating waves and reduce their reflection. Effective use can be made of either a viscosity term, $\nu_2 \nabla^2 \mathbf{v}$, or a simple Rayleigh drag term, $-\nu_0(\mathbf{v} - \mathbf{v}_0)$, where \mathbf{v}_0 may be set to the initial velocity profile or to zero. The EPIC model uses Rayleigh drag with \mathbf{v}_0 nominally set to the initial velocity profile.

It is important to introduce the dissipation gradually with altitude to minimize reflections. We refer to the top layers containing this added dissipation as the ‘‘sponge.’’ Counting the layers down from $k = 1$ at the top, the EPIC model is configured to include layers $k = 1$ through $k_{\text{sp}} = [nk/5]$ (nearest integer) in the sponge, in other words, the top 20% of the layers nominally participate in the sponge. For models with $nk \leq 4$ we forgo the sponge. The Rayleigh damping coefficient, $\nu_0^{\text{sp}}(k)$, is introduced gradually as

$$\nu_0^{\text{sp}}(k) = \frac{1}{5\Delta t} \frac{1}{2} \left(1 - \cos \left[\pi \frac{k_{\text{sp}} + 1 - k}{k_{\text{sp}}} \right] \right), \quad (57)$$

$$k = \{1, k_{\text{sp}}\},$$

which is the functional form proposed by Klemp and Lilly (1978). The strongest value, $\nu_0^{\text{sp}}(1) = (5\Delta t)^{-1}$, is chosen to avoid numerical instabilities while making the sponge’s dissipation strong. It should be noted that unless k_{sp} is rather large, the sponge is only minimally effective. For the runs reported by LeBeau and Dowling (1998) we have only $k_{\text{sp}} = 2$, which means we still suffer some reflections from the top of the model. However, because our focus for that project is on coherent-vortex dynamics seated in Neptune’s troposphere, this is a minor issue.

6. IMPLEMENTATION

Operating system. The EPIC model consists of individual source-code files organized into various subdirectories and is compiled via a hierarchy of Unix makefiles that are controlled by a top makefile (makefiles keep track of which components have been modified and recompile only those that need to be). The bulk of the code is written in C. Subroutines written in either C or Fortran may be added, for example, the model’s thermodynamical functions are unmodified Fortran subroutines developed over many years by P.J.G. and colleagues.

Fortran in a C environment. The ability to link to “legacy” Fortran subroutines without rewriting them into C yields several practical advantages. To do this, the makefiles for the EPIC model first compile the Fortran subroutines with an `f77` or equivalent compiler to produce object files (`.o` files). Second, they use a C compiler, `cc` or equivalent, to link these with the object files generated from C source code. There are three simple (but obscure) tricks to linking Fortran object files to C object files using a C compiler. The first trick is necessary because Fortran math functions inevitably call a small handful of primitive functions that the C compiler does not know about and that are not in the standard math library (`libm.a`). The fix is to hunt down the requisite object libraries and link to them. Unfortunately, the names for these libraries vary from one computer platform to the next, but eventually one tracks them down. The second trick is to determine what internal name the particular computer at hand gives to Fortran subroutines. For example, if the Fortran source code has a subroutine named `myfunc(x)`, the possibilities for its internal name include `myfunc(x)`, `MYFUNC(X)`, `myfunc_(x)`, and `myfunc__(x)` (we have encountered all four). The third trick is to remember to pass all arguments by reference (by pointer). For convenience, we usually write a short “wrapper code” C function for each Fortran function that uses the proper internal name and passes the arguments to the Fortran subroutine by reference. With these issues taken care of, Fortran subroutines become part of the model without further modification.

Parallel computing. The most recent version of the EPIC model uses the Message Passing Interface (MPI) standard (Gropp *et al.* 1994, Snir *et al.* 1996). This means that the model runs equally well on massively parallel computers, symmetric multiprocessors (SMP’s), or clusters of Unix computers linked by a network. The simulations in LeBeau and Dowling (1998) were all run on the MIT Earth Resources Laboratory’s nCUBE 2 parallel computer. This computer is configured with a total of 512 processors; we typically used 64 of these processors for a run. For historical reasons, we used the nCUBE propriety message passing library rather than MPI library calls, but the two are virtually identical, and switching over to the new standard proved to be fast and easy.

Handling large output volume. The EPIC model is a high-resolution GCM, and as such, generates large quantities of output. A typical simulation in LeBeau and Dowling (1998) is for 50 days, and a typical time step is $\Delta t = 90$ s, which means that the runs are 48,000 time steps long. Obviously, we do not store the complete model state for every time step. The model’s small Δt is a consequence of the fact that we are integrating the primitive equations and therefore have fast-moving gravity waves that must be

resolved. The third-order Adams–Bashforth time step, discussed above, requires saving the current state of the primitive variables and two previous time tendencies. Consequently, one time frame is 7.3 Mb for a double-precision, $nk = 11$ model with 128×64 gridpoints per layer.

We have developed an effective scheme for handling and archiving this large amount of output. 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. We have adopted a three-part analyze-as-you-go approach. (The specific numbers below are for the accompanying Neptune experiments.) First, a complete backup time frame is sent to our workstation and stored to its disk at intervals of 2 model days (unless otherwise noted, a reference to time indicates model time, not wall clock time, and 1 day = 24 h = 86400 s); in practice, two files were kept and subsequently overwritten, one for the new backup and one for the previous backup, to ensure always having a good one should a write to disk fail. Also every 2 days, a 0.2-Mb file, containing a zonal average of the primitive variables over a longitude range not including the local region of interest is stored to disk to monitor the environment. Second, at 6-h intervals, a 0.7-Mb “zoom save” is made containing all the data in a restricted cube that encompassed the region of interest. Third, at 1- or 2-h intervals, the primitive variables are sent to the graphics workstation, where they are rendered in three dimensions using commercial software (Advanced Visualization Systems, or AVS), and then recorded to videotape to provide an animated archive of the run. With this three-part approach we were able to avoid filling up our disks with a series of complete time steps and yet perform all the analysis work reported in the accompanying paper. For the rare occasions when we need data that we have not stored, we rerun the simulation from the last point saved.

Although the EPIC model can be compiled and run on a distributed computer environment without any graphical user interface (GUI), it comes with AVS modules that we have written to control, visualize, and analyze the numerical simulations. For definitions of computer-graphics terms and sample algorithms, the reference by Foley *et al.* (1990) has proved to be helpful; for example, we use their RGB to HSV color-space conversion algorithm (pp. 592–593). With these modules, the user sees a three-dimensional rendering of his model as a globe or as isentropic layers, with height modulated by a chosen variable, usually $-\log p$, and color modulated by any of a number of variables, including potential vorticity and temperature, plus two-dimensional contour plots and line plots that can sample any part of the model. We have found that making a videotape animation of our simulations as they proceed is an effective and efficient method for archiving the behavior of a run. Widgets representing various parameters of the

model are simple to implement via a GUI and allow for easy control of the model.

In closing, the design and implementation decisions we have described represent several years of experience running the EPIC model on workstations and parallel computers. It is impossible to predict what significant modifications will be necessary for future projects, but our hope is that they will represent incremental changes to this description. The companion paper by LeBeau and Dowling (1998) presents the results of a study of Neptune's Great Dark Spot using the model. Interested parties may obtain the source code from NASA's Planetary Data System (PDS) Atmospheres node.²

APPENDIX A

Mean Potential Temperature

Thermodynamic functions such as pressure, internal energy, specific heat, and specific entropy can be calculated from the partition function, Z , by treating the atmosphere as a mixture of constituents, following standard practice in chemical thermodynamics (e.g., Nash 1970). The development is more intuitive on a per-particle basis than on a per-mass basis, so here the adjective "specific" indicates "per particle."

The partition function for each ideal-gas constituent composed of N_i particles occupying volume V is given by

$$\frac{\log Z_i}{N_i} = \log \frac{V}{N_i} + \frac{3}{2} \log \left(\frac{m_H k_B T}{2\pi \hbar^2} \right) + \log \zeta_i^{\text{rot}} + \frac{3}{2} \log \mu_i + 1, \quad (\text{A.1})$$

where the constituents are indexed by i , $m_H = 1.66057 \times 10^{-27}$ kg is the mass of unit atomic weight, $\mu_i = m_i/m_H$ is molecular weight, $k_B = 1.38066 \times 10^{-23}$ JK⁻¹ is Boltzmann's constant, and $\hbar = 1.05459 \times 10^{-34}$ Js is Planck's constant divided by 2π . The rotational partition function for a particle, ζ^{rot} (not to be confused with relative vorticity), is given by

$$\zeta^{\text{rot}} = \sum_j g_j \exp \left(-\frac{\varepsilon_j}{k_B T} \right), \quad (\text{A.2})$$

where $\varepsilon_j = J(J+1)k_B \theta^{\text{rot}}$ is rotational energy, $\theta^{\text{rot}} = 87.567$ K for hydrogen, and $g_j = (2J+1)g_0$ are statistical weights. Ortho hydrogen corresponds to odd values of J and $g_0 = 3$, and para hydrogen corresponds to even values of J and $g_0 = 1$. For constituents other than hydrogen in gas-giant atmospheres, rotational levels are either not populated (helium) such that $\zeta^{\text{rot}} = 0$, or high rotational levels are occupied (water, methane) such that the effect of rotation is simply to add new degrees of freedom that can be accounted for by altering the coefficient of the translational term, which is the second term on the right-hand side of (A.1). If there are other phase changes to be included in addition to ortho–para hydrogen, then the liquid (or solid) states need to be included, the entropy needs to be modified, and the saturation vapor pressures need to be introduced, following well-documented procedures for handling water vapor in Earth models.

Once Z is known, other thermodynamical variables follow via partial derivatives. The internal energy, u (not to be confused with zonal wind), is

$$u_i = k_B T^2 \left. \frac{\partial}{\partial T} \frac{\log Z_i}{N_i} \right|_V = \frac{3}{2} k_B T + k_B T^2 \left. \frac{\partial \log \zeta_i^{\text{rot}}}{\partial T} \right|_V, \quad (\text{A.3})$$

$$u = \sum_i X_i u_i, \quad (\text{A.4})$$

and the specific heat at constant volume is (lower case to denote per particle)

$$c_{v_i} = \left. \frac{\partial u_i}{\partial T} \right|_V = \frac{3}{2} k_B + k_B \left. \frac{\partial}{\partial T} \left(T^2 \frac{\partial \log \zeta_i^{\text{rot}}}{\partial T} \right) \right|_V, \quad (\text{A.5})$$

$$c_V = \sum_i X_i c_{v_i}. \quad (\text{A.6})$$

The specific entropy for each constituent is

$$\frac{s_i}{k_B} = \frac{\log Z_i}{N_i} + \frac{u_i}{k_B T}. \quad (\text{A.7})$$

For each component, standard thermodynamic relations can be written as needed. For example, we will make use of

$$T ds_i = du_i + p d \left(\frac{k_B T}{p} \right). \quad (\text{A.8})$$

To define potential temperatures in a manner analogous to (1) in Section 1.2 for the single constituent case, we can write (per particle)

$$s_i - s_{i0} = c_{p_i}^{(R)} \log(\theta_i/T_0), \quad (\text{A.9})$$

where $s_{i0} = s_i(p_0, T_0)$, p_0 and T_0 are reference values of pressure and temperature, and $c_{p_i}^{(R)} = c_{p_i}(T_0)$. Equation (A.9) defines the individual potential temperature for each constituent, θ_i . We take the *mean potential temperature*, θ , to be defined by

$$c_p^{(R)} \log \theta = \sum_i X_i c_{p_i}^{(R)} \log \theta_i, \quad (\text{A.10})$$

where

$$c_p^{(R)} = \sum_i X_i c_{p_i}^{(R)}. \quad (\text{A.11})$$

We set T_0 to a low temperature such that $c_{p_i}^{(R)} = 2.5k_B$ for helium and both components of hydrogen. Other trace components like methane and water have only a small influence on the heat capacity and can either be neglected or accounted for with sufficient accuracy by using constant values of c_{p_i} . We have adopted $p_0 = 1$ bar and $T_0 = 1$ K (purely a mathematical device; the mixture would not remain an ideal gas at that temperature), such that the combination of constants in the Sackur–Tetrode equation of physical chemistry, which defines the entropy of an ideal gas (Nash 1970), is

$$\Phi_0 \equiv \log \left(\frac{k_B T_0}{p_0} \left(\frac{m_H k_B T_0}{2\pi \hbar^2} \right)^{\frac{3}{2}} \right) = -3.665. \quad (\text{A.12})$$

The mean potential temperature defined by (A.9)–(A.11) has the following desirable properties: (i) $\log \theta$ is additive in the components $\log \theta_i$, weighted by their concentrations by number; (ii) the individual $\log \theta_i$ each depend only on T and p ; (iii) $\log \theta$ is conserved if no heat is added and the composition is unchanged (which includes no conversion of ortho to para); (iv) For fixed p and f_p , θ is a monotonic function of T , so that a dynamical computation that carries the variables p , f_p , and θ uniquely carries the thermodynamic state; (v) Below about 40 K, where hydrogen

² The EPIC atmospheric model (© 1998) is by Timothy E. Dowling and is distributed under the provisions of the Free Software Foundation's General Public License.

rotational energy is negligible, the heat capacity is $c_p^{(R)}$ and the potential temperature reduces to standard form

$$\theta = T(p_0/p)^\kappa; \quad \kappa = k_B/c_p^{(R)}; \quad (\text{A.13})$$

(vi) above about 250 K, where hydrogen ortho and para energies are approximately the same, the heat capacity goes to another constant value, $c_p^{(\text{HIGH})}$, and the potential temperature reduces to

$$\theta = CT^\nu(p_0/p)^\kappa, \quad (\text{A.14})$$

where C is a constant that depends on the composition (as do the heat capacities). For pure hydrogen, $\nu = 3.5/2.5$, and for pure ortho and para, $C = 0.1218$ and 0.1888 , respectively.

In the EPIC model, $T = T(p, \theta, f_p)$ is evaluated by the algorithm

$$T_1 = \theta(p/p_0)^\kappa, \quad p_0 = 1 \text{ bar},$$

$$1/\kappa = c_p^{(R)}/k_B = 2.5(X_{\text{H}_2} + X_{\text{Hc}}) + 3.5(1 - X_{\text{H}_2} - X_{\text{Hc}}),$$

$$T = \begin{cases} T_1, & T_1 \leq 20 \text{ K}, \\ \text{table interpolation}, & 20 \text{ K} < T_1 < 600 \text{ K}, \\ \exp \left[\frac{\log(T_1)/\kappa + 2.5X_{\text{H}_2}}{2.5X_{\text{Hc}} + 3.5(1 - X_{\text{Hc}})} \right], & 600 \text{ K} \leq T_1. \end{cases} \quad (\text{A.15})$$

Use is made of (A.13) on the low-temperature end, $T_1 \leq 20 \text{ K}$, and of (A.14) on the high-temperature end, $600 \text{ K} \leq T_1$, the latter with the values $\log(0.1218) = -2.1054$ and $\log(0.1888) = -1.6671$ mentioned above. The evaluation in the middle range, $20 \text{ K} < T_1 < 600 \text{ K}$, is a bilinear interpolation on a data table of T values with dimensions $(n_{f_p} \times n_{T_1}) = (11 \times 111)$ and grid spacing $\Delta f_p = 0.1$ and $\Delta T_1 = 5.3 \text{ K}$. Pressure is not part of the table lookup because the rotational energy is independent of p . At initialization, a table in the form $T_1(T, f_p) = \theta(p, T, f_p)(p/p_0)^\kappa$ is calculated, and then interpolation is used (since the expression cannot be inverted analytically) to transpose the table into the form $T(T_1, f_p)$, also at initialization. For the evaluation of T at a layer interface, p and θ are naturally defined, and f_p on the interface is taken to be the h -weighted average of the layer values above and below.

APPENDIX B

Criterion for Convection

Property (v) of θ in Appendix A turns out to be important to ensure that θ is monotonic, as the following consideration of static stability reveals. If a test bubble is moved upward or downward a small amount such that θ and f_p are constant in the bubble, the difference between the bubble's density and that of its new environment can be described sufficiently accurately by the first-order terms of a Taylor series expanded about the initial location

$$\delta\rho = \left(\frac{\partial\rho}{\partial p} \Big|_{\theta, f_p} - \frac{d\rho}{dp} \right) \Delta p = -\frac{\rho}{T} \left(\frac{\partial T}{\partial p} \Big|_{\theta, f_p} - \frac{dT}{dp} \right) \Delta p, \quad (\text{B.1})$$

where Δp is the pressure change from the move and $d\rho/dp$ and dT/dp are environmental gradients. The environmental temperature gradient may be expanded in terms of partial derivatives

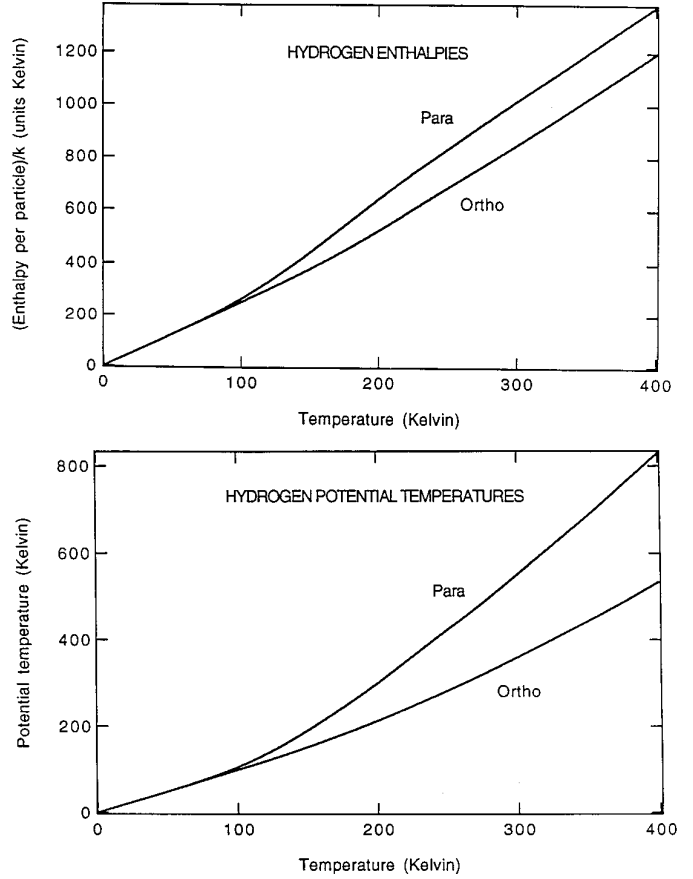


FIG. 5. Ortho and para hydrogen properties. The enthalpies are per particle, divided by the Boltzman constant, k_B , and in this paper are referenced to zero at zero temperature. The potential temperatures are evaluated at the reference pressure, p_0 , and are equal to the temperature for small values of the temperature.

$$\frac{dT}{dp} = \frac{\partial T}{\partial p} \Big|_{\theta, f_p} + \frac{\partial T}{\partial \theta} \Big|_{p, f_p} \frac{d\theta}{dp} + \frac{\partial T}{\partial f_p} \Big|_{p, \theta} \frac{df_p}{dp}, \quad (\text{B.2})$$

and substituted into (B.1) to yield

$$\delta\rho = \frac{\rho}{T} \left(\frac{\partial T}{\partial \theta} \Big|_{p, f_p} \frac{d\theta}{dp} + \frac{\partial T}{\partial f_p} \Big|_{p, \theta} \frac{df_p}{dp} \right) \Delta p. \quad (\text{B.3})$$

In agreement with the formulation for uniform composition, for the case of no gradient in f_p , (B.3) shows that the sign of $\delta\rho$ is the same as that of $\Delta\theta = \Delta p \cdot d\theta/dp$, since $\partial T/\partial\theta > 0$. To examine the case when there is a gradient in f_p , rewrite (B.3) as

$$\delta\rho = \frac{\rho}{T} \frac{\partial T}{\partial \theta} \Big|_{p, f_p} \left(\frac{d\theta}{dp} + \frac{\partial T/\partial f_p}{\partial T/\partial \theta} \Big|_{p, \theta} \frac{df_p}{dp} \right) \Delta p, \quad (\text{B.4})$$

$$= \frac{\rho}{T} \frac{\partial T}{\partial \theta} \Big|_{p, f_p} \left(\frac{d\theta}{dp} - \frac{\partial \theta}{\partial f_p} \Big|_{p, T} \frac{df_p}{dp} \right) \Delta p.$$

Wherever $d\theta/dp$ dominates, θ will be monotonic and will make a viable vertical coordinate for a stable atmosphere. However, there is potential

trouble in the troposphere because $d\theta/dp$ can become small. The derivative $\partial\theta/\partial f_p$ in (B.4) may be evaluated by writing (A.10) explicitly in terms of f_p using $X_o = (1 - f_p)X_{H_2}$ and $X_p = f_p X_{H_2}$ (the subscripts o and p denote “ortho” and “para”), and recalling that we have defined $c_{po}^{(R)} = c_{pp}^{(R)} \equiv c_{pH_2}^{(R)}$

$$c_p^{(R)} \log \theta = X_{H_2} c_{pH_2}^{(R)} ((1 - f_p) \log \theta_o + f_p \log \theta_p) + \dots \quad (\text{B.5})$$

Taking the partial derivative of (B.5) with respect to f_p yields

$$\left. \frac{1}{\theta} \frac{\partial \theta}{\partial f_p} \right|_{p,T} = X_{H_2} \frac{c_{pH_2}^{(R)}}{c_p^{(R)}} \log \frac{\theta_p}{\theta_o}. \quad (\text{B.6})$$

Because we have arranged to have $\theta_p = \theta_o$ at low temperatures (Fig. 5), and because the heat content of para hydrogen, relative to that at zero temperature, is greater, (B.6) shows that $\partial\theta/\partial f_p$ is positive. Therefore, (B.4) indicates that a vertical structure that is neutral against convection ($\delta\rho = 0$) will have θ increasing with height whenever f_p increases with height. This is the usual configuration in tropospheric regions of the gas-giant planets, because the equilibrium concentration of f_p increases as the temperature drops. If the potential temperatures of the separate constituents had been chosen to be equal in the high-temperature limit instead of the low-temperature limit, a nonmonotonic potential temperature in the troposphere would have been likely (this does not imply a change in the physics, only a mathematical limitation on using θ for the height variable).

ACKNOWLEDGMENTS

The authors thank Ted Charrette and Joe Matarese of the MIT Earth Resources Laboratory (ERL) for providing invaluable assistance getting the EPIC model to run on ERL’s nCUBE 2 parallel computer and ported to the Message Passing Interface (MPI) standard, Jim Friedson for advice on minimizing wave reflections off of the top of the model, Kari Backes for digitizing the temperature–pressure and zonal-wind data that are used to initialize models of Jupiter, Saturn, Uranus, and Neptune, Steve Summit for writing the device driver for the scan converter we use to make videotape archives, and Akio Arakawa and F. Michael Flasar for their insightful reviews. This research was supported by NASA’s Planetary Atmospheres Program.

REFERENCES

- Arakawa, A., and V. Lamb 1977. Computational design of the basic dynamical processes of the UCLA General Circulation Model. *Methods Comput. Phys.* **17**, 173–265.
- Arakawa, A., and V. Lamb 1981. A potential enstrophy and energy conserving scheme for the shallow water equations. *Mon. Weather Rev.* **109**, 18–36.
- Conrath, B. J., and P. J. Gierasch 1984. Global variation of the para hydrogen fraction in Jupiter’s atmosphere and implications for dynamics on the outer planets. *Icarus* **57**, 184–204.
- Conrath, B. J., P. J. Gierasch, and S. S. Leroy 1990. Temperature and circulation in the stratosphere of the outer planets. *Icarus* **83**, 255–281.
- Dowling, T. E. 1995a. Estimate of Jupiter’s deep zonal-wind profile from Shoemaker–Levy 9 data and Arnold’s second stability criterion. *Icarus* **117**, 439–442.
- Dowling, T. E. 1995b. Dynamics of jovian atmospheres. *Annu. Rev. Fluid Mech.* **27**, 293–334.
- Dowling, T., B. Conrath, P. J. Gierasch, and E. Ustinov 1996. Simulations of the diabatic circulations of Neptune and Uranus. *Bull. Am. Astron. Soc.* **28**, 1078.
- Durrant, D. R. 1991. The third-order Adams–Bashforth method: An attractive alternative to leapfrog time differencing. *Mon. Weather Rev.* **119**, 702–720.
- Dutton, J. A. 1976. *The Ceaseless Wind*. McGraw–Hill, New York.
- Fletcher, C. A. J. 1991. *Computational Techniques for Fluid Dynamics*, 2nd ed. Vol. I, pp. 85–88. Springer-Verlag, Berlin/New York.
- Foley, J. D., A. van Dam, S. K. Feiner, and J. F. Hughes 1990. *Computer Graphics: Principles and Practice*, 2nd ed. Addison–Wesley, Reading, MA.
- Gill, A. E. 1982. *Atmosphere–Ocean Dynamics*. Academic Press, San Diego.
- Gropp, W., E. Lusk, and A. Skjellum 1994. *Using MPI: Portable Parallel Programming with the Message-Passing Interface*. MIT Press, Cambridge, MA.
- Gradshteyn, I. S., and I. M. Ryzhik 1980. *Table of Integrals, Series, and Products*. Academic Press, San Diego.
- Haltiner, G. J., and R. T. Williams 1980. *Numerical Prediction and Dynamic Meteorology*, 2nd ed. Wiley, New York.
- Harrington, J., R. P. Lebeau, K. A. Backes, and T. E. Dowling 1994. Dynamic response of Jupiter’s atmosphere to the impact of Comet Shoemaker–Levy 9. *Nature* **368**, 525–527.
- Holton, J. R. 1992. *An Introduction to Dynamic Meteorology*, 3rd ed. Academic Press, San Diego.
- Hsu, Y.-J. G., and A. Arakawa 1990. Numerical modeling of the atmosphere with an isentropic vertical coordinate. *Mon. Weather Rev.* **118**, 1933–1959.
- Kar, S. K., R. P. Turco, C. R. Mechoso, and A. Arakawa 1994. A locally one-dimensional semi-implicit scheme for global gridpoint shallow-water models. *Mon. Weather Rev.* **122**, 205–222.
- Klemp, J. B., and D. K. Lilly 1978. Numerical simulation of hydrostatic mountain waves. *J. Atmos. Sci.* **35**, 78–107.
- LeBeau, R. P. 1997. *Simulations of Time-Dependent Three-Dimensional Vortices with Application to Neptune’s Great Dark Spot*. Ph.D. thesis, MIT.
- LeBeau, R. P., and T. E. Dowling 1998. EPIC simulations of time-dependent, three-dimensional vortices with application to Neptune’s Great Dark Spot. *Icarus* **132**, 239–265.
- Lorenz, E. N. 1955. Available potential energy and the maintenance of the general circulation. *Tellus* **7**, 157–167.
- Nash, L. K. 1970. *Elements of classical and statistical thermodynamics*, Book One and Book Two. Addison–Wesley, Reading, MA.
- Shepherd, T. G. 1990. Symmetries, conservation laws, and Hamiltonian structure in geophysical fluid dynamics. *Adv. Geophys.* **32**, 287–339.
- Snir, M., S. W. Otto, S. Huss-Lederman, D. W. Walker, and J. Dongarra 1996. *MPI: The Complete Reference*. MIT Press, Cambridge, MA.