Theta-Sigma Coordinate Model

David J. Raymond

July 9, 2013

1 Governing equations

1.1 Basic equations

The sigma isentropic coordinate system has as its vertical coordinate

$$\sigma = \frac{\theta - \theta_B}{\theta_T - \theta_B}, \tag{1}$$

where θ is the potential temperature and θ_B and θ_T are values on the bottom and top surfaces of the computational domain. We assume that θ_T is constant in time and uniform over the domain top. Inverting, we get

$$\theta = \sigma(\theta_T - \theta_B) + \theta_B. \tag{2}$$

The governing equations for our sigma isentropic coordinate model are as follows. Momentum:

$$\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v} + S_\sigma \frac{\partial \mathbf{v}}{\partial \sigma} + \nabla M - \Pi \nabla \theta + f\mathbf{k} \times \mathbf{v} - \frac{1}{\eta} \nabla \cdot (K_h \nabla \mathbf{v}) - \frac{1}{\eta} \frac{\partial}{\partial \sigma} \left(K_\sigma \frac{\partial \mathbf{v}}{\partial \sigma} \right) = \mathbf{S}_v \tag{3}$$

Mass:

$$\frac{\partial \eta}{\partial t} + \nabla \cdot (\mathbf{v}\eta) + \frac{\partial}{\partial \sigma} (S_\sigma \eta) = 0 \tag{4}$$

Total cloud water mixing ratio:

$$\frac{\partial r_t}{\partial t} + \mathbf{v} \cdot \nabla r_t + S_\sigma \frac{\partial r_t}{\partial \sigma} - \frac{1}{\eta} \nabla \cdot (K_h \nabla r_t) - \frac{1}{\eta} \frac{\partial}{\partial \sigma} \left(K_\sigma \frac{\partial r_t}{\partial \sigma} \right) = S_r \tag{5}$$

The Coriolis parameter is f, the horizontal velocity (in sigma isentropic space) is $\mathbf{v}$, the sigma isentropic coordinate vertical velocity is

$$S_\sigma = \frac{d\sigma}{dt} = \frac{S_\theta - (1 - \sigma)(S_B + \Delta \mathbf{v}_B \cdot \nabla \theta_B)}{\theta_T - \theta_B}, \tag{6}$$

where $S_\theta = d\theta/dt$, S_B is S_θ at the surface, and $\Delta \mathbf{v}_B = \mathbf{v} - \mathbf{v}_B$, with $\mathbf{v}$ being the horizontal (constant σ) air velocity and $\mathbf{v}_B$ being $\mathbf{v}$ at the surface. The Montgomery potential is M, the mass density in sigma isentropic coordinates is η, and the total cloud water mixing ratio is r_t. The quantities K_h and K_σ aare "horizontal" and "vertical" eddy mixing coefficients, discussed later. The horizontal coordinates are geometrical positions x and y and time is t. The external horizontal force is $\mathbf{F}$ and the total cloud water mixing ratio source is S_r. The Montgomery potential is defined

$$M = \theta\Pi + \Phi, \tag{7}$$

where Π is the Exner function (defined below) and $\Phi = gz$ is the geopotential.

Technically, the η equation should not have eddy mixing terms, as the velocity is defined in a density-weighted sense. However, the integration becomes quite unstable without the resulting smoothing of the η field.

The pressure p can be obtained by integrating the density:

$$\frac{\partial p}{\partial \sigma} = -g\eta \tag{8}$$

subject to the pressure being fixed at some level, such as the top of the domain. The Exner function is related to the pressure by

$$\Pi = C_p(p/p_R)^\kappa \tag{9}$$

where p_R is a constant reference pressure and $\kappa = (C_p - C_v)/C_p$, C_v being the specific heat at constant volume. Finally, the Montgomery potential is obtained by integrating the Exner function:

$$\frac{\partial M}{\partial \sigma} = (\theta_T - \theta_B)\Pi \tag{10}$$

subject to the condition at the surface

$$M_B = \theta_B \Pi_B + gh, \tag{11}$$

where g is the acceleration of gravity and $h(x, y)$ is the height of the terrain above the zero level. The geopotential is obtained by solving equation (10),

$$\Phi \equiv gz = M - \theta\Pi \tag{12}$$

where z is the height above the zero level.

The surface potential temperature is governed by:

$$\frac{\partial \theta_B}{\partial t} + \mathbf{v}_B \cdot \nabla\theta_B = S_B \tag{13}$$

where the subscripted B indicates a surface value.

Not shown explicitly in equation (3) is a relaxation of the mean zonal wind profile to the initial specified profile. This is used when we desire to keep this mean profile fixed.

1.2 Flux form of equations

Defining $\mathbf{u} = \eta\mathbf{v}$ and $\eta_t = \eta r_t$, we can write the momentum, equivalent potential temperature, and total cloud water equations in flux form:

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{v}\mathbf{u} - K_h \nabla \mathbf{v}) + \frac{\partial}{\partial \sigma}\left(S_\sigma \mathbf{u} - K_\sigma \frac{\partial \mathbf{v}}{\partial \sigma}\right) + \eta\nabla M - \eta\Pi\nabla\theta + f\mathbf{k} \times \mathbf{u} = \eta\mathbf{S}_v \tag{14}$$

$$\frac{\partial \eta_t}{\partial t} + \nabla \cdot (\mathbf{v}\eta_t - K_h \nabla r_t) + \frac{\partial}{\partial \sigma}\left(S_\sigma \eta_t - K_\sigma \frac{\partial r_t}{\partial \sigma}\right) = \eta S_r. \tag{15}$$

2 Source terms

We set $\mathbf{S}_v$, S_θ, S_r to values defined by the diabatic parameterization being used, or to fixed functions. The details of how the parameterization is incorporated are given below. The surface temperature distribution and the topography are externally specified. In all cases, since θ_T is assumed constant, S_θ must be zero at the top of the domain.

The zonal momentum source has an additional artificial term beyond that supplied by parameterizations which takes the form

$$S_{vxaux} = \mu_{vxrelax} \left[v_{xref}(y, \sigma) - v_{x0}(y, \sigma, t) \right] \tag{16}$$

where $v_{x0}(y, \sigma, t)$ is the zonal average of the zonal wind at the current time step and $v_{xref}(y, \sigma)$ is the value of $v_{x0}(y, \sigma, t)$ at the initial time. The effect of this term is to relax the zonally averaged zonal wind structure to that of the initial zonally averaged wind, with a more stringent relaxation occurring for larger values of $\mu_{vxrelax}$. In this way the initial zonally averaged wind structure can be maintained in the face of natural tendencies to change it. This is useful for creating zonally uniform balanced states.

2.1 Incorporation of diabat3 package

2.1.1 Cumulus and surface flux parameterizations

The cumulus parameterization is an adjustment scheme, with separate calculations made for shallow and deep convection. The forms of the convective source terms for each type of convection are the same; for equivalent potential temperature θ_e, total cloud water mixing ratio r_t, and the horizontal velocity $\mathbf{v}$, they are

$$S_{eci} = \left(\overline{\theta}_e + L(z) - \theta_e(z) + \frac{F_e}{M} \right) a(z) \tag{17}$$

$$S_{ri} = \left(\overline{r}_t - r_t(z) + \frac{F_r}{M} \right) a(z) - C_R(z) \tag{18}$$

$$\mathbf{S}_{vi} = \left(\overline{\mathbf{v}} - \mathbf{v}(z) + \frac{\mathbf{F}_v}{M} \right) a(z) \tag{19}$$

where F_e, F_r, and $\mathbf{F}_v$ are the surface fluxes of the corresponding variables, M is the mass per unit area in the column extending from the surface to the top of the convection.

The quantity $a(z)$ is the rate function, which equals

$$a(z) = \lambda_c(1 + \alpha z/D)\Lambda(D - z) \tag{20}$$

where α controls the shape of the $a(z)$ profile, D is the depth of the convective layer, λ_c is a constant defining the maximum mixing rate, and where the Heaviside function $\Lambda(x) = 1$ for $x > 0$ and $\Lambda(x) = 0$ for $x < 0$. For deep convection we set $\alpha = \alpha_d$, whereas $\alpha = \alpha_s$ for shallow convection. The same value of λ_c is used for shallow and deep convection.

The function

$$L(z) = \lambda_l(z - z_l) \tag{21}$$

where z_x is adjusted so that

$$\int_0^D L(z)a(z)dz = 0. \tag{22}$$

The quantity λ_l is an adjustable parameter to control the magnitude of $L(z)$. The purpose of $L(z)$ is to cause the target profile of θ_e to increase with height, as seems to occur in the natural atmosphere.

The overbar applied to any variable $\chi(z)$ is

$$\overline{\chi} = \frac{1}{M}\int_0^D \rho(z)a(z)\chi(z)dz \tag{23}$$

where $\rho(z)$ is the atmospheric density profile and where

$$M = \int_0^D \rho(z)a(z)dz. \tag{24}$$

Equations (17)-(19) are designed so that, for instance,

$$\int_0^D \rho(z)(\overline{\theta}_e - \theta_e)a(z)dz = 0, \tag{25}$$

so that the first two terms represent a conservative redistribution or adjustment which attempts to homogenize vertically the variable in question. The third term on the right of these equations represents the vertical distribution of surface fluxes. This distribution is weighted by the function $a(z)$.

The vertical profile $C_R(z)$ represents the conversion of total cloud water (vapor plus cloud droplets) to precipitation, and it takes a rather complex form

$$C_R = \lambda_s(r_t - \mathcal{H}_c r_s)\Lambda(r_t - \mathcal{H}_c r_s)\mathcal{T}(\Gamma - \Gamma_t, \Gamma_s)\mathcal{T}(z - z_p, z_p) - \lambda_e(r_s - r_t)\Lambda(r_s - r_t)r_p \tag{26}$$

involving two terms. The second term represents the evaporation of precipitation. The quantities r_s and r_p are the saturation and precipitation mixing ratios and λ_e is a constant controlling the evaporation rate.

The first term represents the formation of precipitation in regions that are saturated on the large scale, and λ_s is the rate constant controlling this process. The function $\mathcal{T}(x, y)$ is a throttle function; i. e.,

$$\mathcal{T}(x, y) = \begin{cases} 0, & x < -y/2 \\ x/y + 1/2, & -y/2 < x < y/2 \\ 1, & x > y/2 \end{cases} . \tag{27}$$

The throttle function is like the Heaviside function $\Lambda(x)$ with a gradual transition from "off" to "on". The quantity $\Gamma = d\theta_e/dz$ and Γ_t is a constant threshold value of Γ, while Γ_s defines the range over which the throttle function turns on. The purpose of the first throttle function in the second term of (26) is to suppress the conversion of cloud water to precipitation in saturated regions where $\Gamma < 0$. Since grid-scale saturation with negative Γ is unstable, it cannot exist for long. I interpret such regions as being mainly unsaturated, but pierced by saturated columns of convection of sufficient density to render the average cloud water content saturated. The instability in this case resides solely in the convective columns, and it is inappropriate to turn on "stratiform" processes in this case. The second throttle function causes precipitation to turn on gradually with height, starting at $z = z_p/2$ and completing at $z = 3z_p/2$. Setting z_p to twice the boundary layer depth causes precipitation to begin turning on at the top of the boundary layer.

The vertical profile of precipitation rate $P = \rho w_t r_p$ is assumed to adjust instantly to conditions in the grid box, and thus obeys the time-independent vertical advection equation

$$\frac{1}{\rho}\frac{\partial P}{\partial z} = -C_R, \tag{28}$$

subject to $P = 0$ at the top of the domain. Thus, integrating (28) down from the top yields the precipitation rate at the surface.

As noted above, separate calculations are made for shallow and deep convection. Shallow, planetary boundary layer (PBL) convection is assumed to have the prescribed depth b. The depth D of deep convection is assumed to be the level of neutral buoyancy for parcels with average characteristics of the PBL, i. e., in the height range $0 < z < b$. In calculating this, the highest level of neutral buoyancy is taken, thus ignoring possible inversions and stable layers at intermediate altitudes.

Observations over warm tropical oceans generally show positive buoyancy for PBL parcels everywhere in the troposphere except for a stable layer just above the PBL. The partitioning between shallow and deep convection in each grid box is decided on the basis of the convective inhibition in this layer. The parameter ϵ is taken as a measure of this inhibition:

$$\epsilon = \mathcal{T}(\theta_{eb} - \theta_{et}, \Delta\theta_e), \tag{29}$$

where $\Delta\theta_e$ is a constant, θ_{eb} is the roughly the mean equivalent potential temperature in the height range $0 < z < b$ and θ_{et} is the mean saturated equivalent potential temperature in the range $b < z < 2b$. The formal definitions of θ_{eb} and θ_{et} are

$$\theta_{eb} = \sum_i \theta_{ei} \exp(-z_i^2/b^2) \bigg/ \sum_i \exp(-z_i^2/b^2) \tag{30}$$

and

$$\theta_{et} = \sum_i \theta_{esi} \exp\left[-(z_i/b - 1)^2\right] \bigg/ \sum_i \exp\left[-(z_i/b - 1)^2\right] \tag{31}$$

where the index i indicates the model level and θ_{ei} and θ_{esi} are the equivalent potential temperature and the saturated equivalent potential temperature at level i. The summation is over all model levels. Technically for a true average, the levels should be weighted by the mass per unit area at each level, but the inclusion of this weighting can contribute to various forms of numerical instability.

Actual profiles of convective source terms are computed as weighted averages of the shallow and deep convective profiles, indicated respectively by subscripted s and d:

$$S_{ec} = \epsilon S_{ecd} + (1 - \epsilon) S_{ecs} \tag{32}$$

etc.

Parameter	Value	Meaning
λ_c	0.06 ks^{-1}	convective mixing rate
α_d	0	structure parameter for deep mixing
α_s	-1	structure parameter for shallow mixing
λ_l	2 K km^{-1}	linear bias in θ_e target profile
λ_s	0.03 ks^{-1}	precipitation rate parameter
Γ_t, Γ_s	2 K km^{-1}(both)	lower bound of $d\theta_e/dz$ for strat precip
λ_e	250 ks^{-1}	rate parameter for evaporation of rain
z_p	4 km	minimum height of rain formation
w_t	$5, 1$ m s^{-1}	precip terminal velocity (rain,snow)
b	2 km	depth of PBL
$\Delta\theta_e$	4 K	slop in convective throttle
ΔT	0.5 K	slop in surface flux throttle
C_D	0.001	surface drag coefficient
W	3 m s^{-1}	minimum wind for surface fluxes

Table 1: Typical values of parameters used in the cumulus parameterization.

A simplified definition of equivalent potential temperature is used in this parameterization:

$$\theta_e = \theta \exp(\gamma r_t), \tag{33}$$

where $\gamma = L/(C_p T_R)$, $L = L_c + L_f$ being the sum of the latent heats of condensation and freezing, C_p the specific heat of air at constant temperature, and $T_R = 300$ K a constant reference temperature. The total cloud water mixing ratio r_t is used instead of the vapor mixing ratio, because the difference between the two is likely to be small when averaged over grid boxes typical of large-scale models.

The potential temperature source term is computed from the equivalent potential temperature and total water source terms using

$$S_\theta = \theta[(S_{ec} + S_{er})/\theta_e - \gamma S_r] \tag{34}$$

where S_{ec} and S_{er} are the convective and radiative contributions to the equivalent potential temperature source, S_r is the convective source of total water mixing ratio, and γ is defined above. The total precipitation rate is computed similarly:

$$P = \epsilon P_d + (1 - \epsilon) P_s. \tag{35}$$

The source terms are smoothed both in the vertical and horizontal before use in the governing equations.

Surface fluxes for each variable χ are calculated using a simple bulk formula:

$$F_\chi = \mathcal{T}(T_{ss} - T_b, \Delta T)\rho_b C_D(|\mathbf{v}_b|^2 + W^2)^{1/2}(\chi_{ss} - \chi_b) \tag{36}$$

where T_{ss} is the temperature of the surface, T_b is the temperature of the air in the boundary layer adjacent to the surface, ΔT and W are a constants, C_D is the (constant) drag coefficient,

Symbol	Reference Value	Variable Name
p_R	1000hPa	pressure
T_R	300K	temperature
ρ_R	$1.2\mathrm{kgm}^{-3}$	density
r_R	$0.02\mathrm{gg}^{-1}$	vapor mixing ratio

Table 2: Reference values of thermodynamic variables.

a subscripted b indicates a boundary layer value, and a subscripted ss indicates a surface value. The throttle function turns the fluxes off when $T_b > T_{ss} + \Delta T/2$.

Over ocean, it is set to the saturation mixing ratio at the temperature and pressure of the sea surface. Over dry land, the surface value of the mixing ratio is set to the saturation mixing ratio at the temperature and pressure of the sea surface times a "surface saturation fraction" $\mathcal{S}_s$, which is discussed later in the surface model. The surface value of the equivalent potential temperature is computed from the temperature and pressure of the surface and the surface mixing ratio computed above.

All source terms, S_θ, S_e, S_r, and $\mathbf{S}_v$, are subjected to 1-2-1 smoothing in the vertical before they are ingested into the large-scale model. In order to prevent numerical instabilities, we also currently limit the allowed range of the vertical potential temperature gradient: $1 < \partial\theta/\partial z < 20$ K km^{-1}.

Table 1 lists values of parameters used in the cumulus parameterization.

2.1.2 Radiation parameterization

The majority of this section is drawn from the description of the radiation parameterization in Raymond and Torres (1998).

A toy empirical scheme for radiation is used here in which the great range of absorptivities occurring across the infrared spectrum is approximated by a small number of fixed absorptivities, each associated with some fraction of the spectrum. Water vapor bands in particular are assumed to be lumped into 11 categories spread over 5 orders of magnitude in optical depth. Carbon dioxide is categorized by a single absorptivity, while one other category accounts for atmospheric infrared windows and possible continuum absorption by water vapor. Thus 13 categories are defined overall. (The original version of the model had 6 categories for water vapor, but in higher vertical resolution models, this resulted in noisy vertical profiles.) The water vapor band categories receive solar infrared radiation from above as well as terrestrial radiation from below. Clouds are assumed to increase the absorptivity at their level in proportion to their condensed water content. However, no account is taken of the tendency of clouds to scatter rather than absorb solar infrared radiation. Ozone is not included since its main effects are in the stratosphere and a proper treatment would require an interactive model of ozone concentration, which is beyond the scope of this work.

This model can be criticized on many theoretical grounds. However, after tuning a small number of coefficients, it yields reasonable agreement with radiative heating rates predicted by the National Center for Atmospheric Research's CCM2 radiation model (Briegleb, 1992a,b; Hack, Boville, Briegleb, Kiehl, Rasch, and Williamson, 1993) over a fairly wide range of atmospheric conditions, and therefore appears to catch at least some of the essential features of atmospheric radiative transfer. It also has the advantage of being computationally inexpensive.

Upward and downward streams, I_{i+} and I_{i-}, are defined for each category of infrared radiation. These streams are assumed to satisfy

$$\frac{dI_{i+}}{d\tau_i} = f_i \sigma_{SB} T^4 - I_{i+} \qquad \frac{dI_{i-}}{d\tau_i} = -f_i \sigma_{SB} T^4 + I_{i-}, \tag{37}$$

where τ_i is the optical depth associated with category i, $T(\tau_i)$ is the absolute temperature of the air, σ_{SB} is the Stefan-Boltzmann constant, and f_i is the fraction of the thermal infrared spectrum associated with this radiation category.

The equation for I_{i+} is integrated upward from the surface, while that for I_{i-} is integrated downward from the top of the domain. Boundary conditions on I_{i+} and I_{i-} are that $I_{i+}(surface) = f_i \sigma_{SB} T_s^4$, where T_s is the surface temperature, and $I_{i-}(top) = S_i \cos(\phi_Z)$, where S_i is the assumed diurnal-mean downward solar radiative flux at the model top in this category, assuming that the sun is straight overhead, and ϕ_Z is the solar zenith angle. (The solar contribution is set to zero if the sun is below the horizon.)

The vertical profile of optical depth for each category is obtained by integrating the equation

$$d\tau_i = \rho(C_i \kappa_i + r_l \kappa_c) dz \tag{38}$$

upward from the surface, where $\tau_i = 0$. In this equation ρ is the air density, C_i is a factor which depends on the particular radiation category, κ_i is the absorptivity for the gas represented by this category, r_l is the condensed water mixing ratio, and κ_c is the cloud absorptivity.

Raymond and Zeng (2000) showed that the cloud absorptivity for water droplets of uniform radius R is approximately

$$\kappa_c = 3/(4R\rho_w)$$

where ρ_w is the density of liquid water. For a droplet radius of 10 μm, $\kappa_c = 7.5 \times 10^4$ m^3 kg^{-1} km^{-1}. In a model with spatially coarse grid cells, not all of the area of a grid cell may be covered with cloud. To cover this possibility in a crude way, we define a cloud fraction input parameter and make two radiative computations in each column, one with cloud water, the other without. The actual radiative heating profile is the weighted average of the cloudy and clear calculations.

For carbon dioxide we assume $C_i = (\rho/\rho_R)(T/T_R)^{1/2}$, where ρ_R and T_R are constant reference values for density and temperature. This equation introduces the effect of pressure broadening. For the water vapor bands, $C_i = (r/r_R)(\rho/\rho_R)(T/T_R)^{1/2}$, which makes the radiative absorptivity scale with the density of water vapor with an additional factor for pressure broadening. For the window-continuum band we assume $C_i = (r/r_R)^2$. Reference values of parameters (indicated by subscripted Rs) are listed in table 2, while values of f_i, κ_i, and S_i are given in table 3. The cloud absorptivity, κ_c, is set to zero in the present work.

The upward energy flux due to radiation is

$$F_{rad} = \frac{\partial}{\partial z} \sum_i [I_{i+}(z) - I_{i-}(z)] - \cos(\phi_Z) \left[AF_{sun} - \sum_i S_i \right] \tag{39}$$

Category	f_i	κ_i	S_i	purpose
1	0.11	$2\mathrm{m}^3\mathrm{kg}^{-1}\mathrm{km}^{-1}$	$0\mathrm{Wm}^{-2}$	carbon dioxide
2	0.20	0.5	160	windows and continuum
3	0.13	1	35	water vapor bands
4	0.11	3.16	26	water vapor bands
5	0.09	10	19	water vapor bands
6	0.08	31.6	13	water vapor bands
7	0.07	100	10	water vapor bands
8	0.06	316	6	water vapor bands
9	0.05	1000	3	water vapor bands
10	0.04	3160	0	water vapor bands
11	0.03	10000	0	water vapor bands
12	0.02	31600	0	water vapor bands
13	0.01	100000	0	water vapor bands

Table 3: Values of constants for each radiation category.

where ϕ_Z is the solar zenith angle as before, A is the mean albedo of the earth, and F_{sun} is the solar constant at the radius of the earth's orbit. The second term on the right side of (39) represents the downward flux of visible solar radiation, assumed to pass through the atmosphere unhindered except by clouds, the radiative effects of which are handled approximately by the mean albedo. In cases in which averaging is made over the diurnal cycle, we set

$$\cos(\phi_Z) = 1/\pi, \tag{40}$$

which is approximately valid for the tropics.

The resulting radiative source term for equivalent potential temperature is

$$S_{er} = -\frac{\theta_e}{\rho \Pi \theta} \frac{\partial F_{rad}}{\partial z}. \tag{41}$$

Note that the visible light part of F_{rad}, being constant, drops out of (41).

2.1.3 Land surface model

The land surface temperature T_{ss} evolves according the the governing equation

$$\frac{\partial T_{ss}}{\partial t} = -\lambda_T T_{ss} \left[\frac{F_{rad}(0) + \Pi(0) F_e}{\sigma T_{ss}^4} \right] \tag{42}$$

where λ_T is an assumed rate constant for evolution of the surface temperature, $F_{rad}(0)$ and $\Pi(0)$ are surface values of the upward radiative flux and the Exner function, F_e is the surface flux of equivalent potential temperature, and σ is the Stefan-Boltzmann constant. The equivalent potential temperature flux term represents the loss of energy from the surface to both latent and sensible heat fluxes into the atmosphere. The scaling factor of σT_{ss}^4 in the denominator is introduced since the surface radiation scales with the black body flux from the surface.

If the surface is represented by a single layer of thickness d_s, density ρ_s, and specific heat C_s, then the rate constant would be

$$\lambda_T = \frac{\sigma T_{ss}^3}{\rho_s d_s C_s}. \tag{43}$$

Most land surface models consist of several layers. These models do not admit the simple time dependence expressed by (42). However, the present treatment should be useful as a toy model for the surface temperature.

Over the sea a mixed layer ocean model is implemented by replacing λ_T by a much longer λ_S in equation (42). A model for λ_S is given by equation (43) with ocean values inserted.

The surface "water column" W_s is assumed to obey the equation

$$\frac{\partial W_s}{\partial t} = P - F_r - \lambda_W W_s \tag{44}$$

where P is the precipitation rate, F_r is the surface evaporation rate, and λ_W is the rate parameter for the contribution of runoff and infiltration of water, both of which remove water from the surface-accessible column. The surface saturation fraction is assumed to be

$$\mathcal{S}_s = 1 - \exp(-W_s/W_C) \tag{45}$$

where W_C is a critical value of the surface water column. The surface water column rate parameter is probably of order 1 day^{-1} and the critical water column is likely to be of order 10 mm.

2.2 Ideal1 parameterization

This constitutes an alternative treatment of the parameterization of diabatic effects. The present version is linear and incorporates thermodynamic sources and surface friction. Base values of source terms are assumed to cancel out and only perturbation values are represented. A linear surface friction is added.

The potential temperature perturbation source is assumed to be

$$S_\theta = B \sin(m_0 z) \exp[\nu(z - h/2)] \bigg/ \int_0^{\sigma_t} \eta(\sigma) \sin(m_0 z) \exp[\nu(z - h/2)] d\sigma \tag{46}$$

where B is determined below, $m_0 = \pi/h$ where $z = h$ is the height of the tropopause, σ_t is the height of the tropopause in sigma coordinates, and ν is a skew factor in the vertical profile of S_θ. We assume that

$$B = L(P - R)/C_p \tag{47}$$

where P is the perturbation precipitation rate, given by the equation

$$P = P_1 + P_2 = \alpha(H - H_m) - \mu_{cin} \left(\frac{\rho_R C_p}{L}\right) [\theta'_{es}(d) - \theta'_e(0)] \tag{48}$$

where

$$H = \int_0^1 \eta r_t d\sigma \tag{49}$$

is the precipitable water, H_m is the mean value of the precipitable water, and α is a rate constant. The first term produces precipitation in proportion to the excess of the precipitable water over its mean value. In the second term $\theta'_{es}(d)$ is the perturbation saturated equivalent potential temperature at height d, $\theta'_e(0)$ is the perturbation equivalent potential temperature at the surface, and μ_{cin} is a rate constant. The perturbation is relative to the spatial average. In this term precipitation is controlled by convective inhibition. The term R is the perturbation radiative cooling rate expressed in equivalent precipitation rate terms as

$$R = -\epsilon P_1. \tag{50}$$

The idea is that radiative cooling decreases as type 1 precipitation increases, due to the associated increase in moisture and cloudiness.

We further split P_2 into P_{2s} and P_{2t}, associated respectively with the $\theta'_{es}(d)$ term and the $\theta'_e(0)$ term in (48).

The perturbation evaporation rate is given by

$$E = \rho_R C_D \Delta q \frac{U_R(U - U_R)}{(U_R^2 + W^2)^{1/2}} \tag{51}$$

where Δq is the difference between the saturation mixing ratio at the sea surface temperature and the boundary layer mixing ratio. For the purposes of this model Δq is assumed to be constant. The parameter U is the boundary layer zonal wind and U_R is a constant reference value of U, while W is a gustiness contribution to the effective surface wind.

The vertical profile of water vapor mixing ratio is initially assumed to have constant relative humidity as a function of height equal to S. The source term for water vapor is

$$S_r = (E - P)\exp(-z/z_q) \Big/ \int_0^1 \eta \exp(-z/z_q) d\sigma \tag{52}$$

where z_q is the scale height of water vapor in the atmosphere. This source term increases or decreases the relative humidity as required by the surface evaporation and precipitation. The source term for equivalent potential temperature is

$$S_e = S_\theta + LS_r/C_p. \tag{53}$$

The values of r_t and θ calculated at the surface using (5) and (13) are not used to compute $\theta'_e(0)$ for the purpose of convective inhibition control of convection, due the uncertainty in the actual profile of mixing ratio. Instead, we use the simplified equation

$$\theta'_e(0) = (\lambda_s/\rho_R)F'_{es} \tag{54}$$

where we approximate the surface θ_e flux by $F'_{es} = (L/C_p)E'$.

The surface friction takes the form

$$\boldsymbol{S}_v = -\lambda \boldsymbol{v}_s \exp(-\mu z) \tag{55}$$

where λ and μ are constant parameters.

3 Numerical implementation

3.1 Initialization

We initialize with separate constant values of isentropic density in the stratosphere and troposphere, and constant values of surface and tropopause potential temperature. A horizontally uniform x wind profile is specified.

Either an f plane or a β plane condition is imposed.

3.2 Numerical methods

We use a 3-D Lax-Wendroff method for the 3-D fields. These fields are all cell-centered. Before a time step is calculated, all fields are extended by two cells in positive and negative directions horizontally, and one cell vertically. Extension conditions are described below in the section on lateral boundary conditions. Upstream advective differencing is used for the surface potential temperature.

The horizontal eddy mixing coefficient K_h has two parts, a part proportional to the horizontal strain rate of the fluid, $\eta C_{KH}|D|\Delta x\Delta y$, where C_{KH} is a dimensionless constant and where

$$|D|^2 = D_{xx}^2 + D_{yy}^2 + 2D_{xy}^2, \tag{56}$$

with

$$D_{xx} = \frac{\partial v_x}{\partial x} \quad D_{yy} = \frac{\partial v_y}{\partial y} \quad D_{xy} = \frac{1}{2}\left(\frac{\partial v_x}{\partial y} + \frac{\partial v_y}{\partial x}\right), \tag{57}$$

and a sponge layer part which exists within a certain distance of the upper boundary, and within a specified distance of the N/S walls in the channel case. The lateral sponge layer may optionally be turned off. The maximum value at the top of the domain (and the N/S sides in the channel case) is $K_{max} = 0.25\eta\Delta x\Delta y/\Delta t$. The quantities Δx and Δy are the horizontal grid dimensions and Δt is the time step in the numerical method. Below the sponge level the mixing is set to a value which eliminates two-delta-x instability. Inside the sponge layer motion becomes increasingly damped with proximity to the wall, in order to absorb waves, thus preventing reflection of these waves off of the top and sides of the domain.

The vertical eddy mixing coefficient is defined

$$K_\sigma = \eta C_{KV}\left|\frac{\partial S_\sigma}{\partial \sigma}\right|\Delta\sigma^2 \tag{58}$$

where C_{KV} is a dimensionless constant similar to C_{KH}. Extensive tests suggest that $C_{KH} = C_{KV} = 1$ results in enough smoothing but not too much.

In addition, ∇^4 smoothing is applied to u_x, u_y, η, and η_t in both the vertical and the horizontal. The horizontal smoothing takes the form

$$\begin{aligned}\hat{\psi}_{i,j} = {} & (1-\kappa)\psi_{i,j} - \\ & \kappa[(\psi_{i+2,j} + \psi_{i-2,j} + \psi_{i,j+2} + \psi_{i,j-2})/20 + \\ & (\psi_{i+1,j+1} + \psi_{i+1,j-1} + \psi_{i-1,j+1} + \psi_{i-1,j-1})/10 - \\ & 2(\psi_{i+1,j} + \psi_{i-1,j} + \psi_{i,j+1} + \psi_{i,j-1})/5]\end{aligned} \tag{59}$$

while the vertical smoothing is given by

$$\hat{\psi}_k = (1 - \lambda)\psi_k - \lambda[(\psi_{k+2} + \psi_{k-2})/6 - 2(\psi_{k+1} + \psi_{k-1})/3], \tag{60}$$

where ψ is each of the above variables, and where the subscripts i and j represent x and y labels of grid cells and k represents σ labels. Currently we set $\kappa = 0$, i. e., we turn off this feature in the horizontal. Currently we set $\lambda = 0.01$.

Hard limits are put on the range of possible η values:

$$\eta_{min} \leq \eta \leq \eta_{max} \tag{61}$$

where

$$\eta_{min} = \frac{\rho(\theta_T - \theta_B)}{(d\theta/dz)_{max}} \qquad \eta_{max} = \frac{\rho(\theta_T - \theta_B)}{(d\theta/dz)_{min}}, \tag{62}$$

with the values of ρ and $\theta_T - \theta_B$ being ambient initial values:

$$\rho = \frac{p_R C_p (\Pi/C_P)^{1/\kappa}}{R\theta\Pi}. \tag{63}$$

If η in a grid cell goes outside of these limits, the excess or deficit in mass, moisture, and momentum is added to the grid cell below, and the whole process is propagated downward. If an excess or deficit remains at the surface, this is a fatal error. The limiting potential temperature gradients are $(d\theta/dz)_{min} = 1$ K km^{-1} and $(d\theta/dz)_{max} = 20$ K km^{-1}.

3.3 Lateral boundary conditions

Two choices exist for lateral boundary conditions. The first option is periodic conditions in both the x and y directions. The second option is that of a channel model, with periodic conditions in x and rigid, free-slip walls bounding the model in the y direction. In the channel model the extended values across the channel walls are mirror image values.

In the output files, low boundary values are repeated for display purposes on the high side in both x and y for the doubly periodic case. For the channel case, this is just done for the x dimension.

3.4 Model design

The model is written in the C language. Lessons from experience with both functional programming (Sisal) and modular programming (Modula, Oberon) have been applied. Functions applying to particular groupings of data are contained in modules dedicated to those data. Data are shared in these modules with static declarations at the beginning before any functions are defined. Communication of data between modules is solely via function arguments and return values. Macros are used to build up complex differencing operations in order to minimize the possibility of error. Multi-dimensional fields are represented by 1-D C-language arrays, with indexing done via macros. Function prototypes are defined in header files which are shared between communicating modules.

The modules defined at this stage are

1. Isentrope: This is the main calling program.
2. Params: This reads model control parameters from an input file, provides them to other modules, and participates in the construction of output files.
3. Step: This does model time steps, initializes the model at the beginning, and writes output files associated with each step. The dynamical core of the model is contained in this module.
4. Vert: This does various diagnostic vertical integrations and similar calculations.
5. Source: The source terms are calculated here or passed on to other modules.
6. Init: This is responsible for initializing fields.
7. Diabat3: Contains the diabatic parameterizations from the diabat3 package.
8. Ideal1: Contains a highly idealized linear parameterization.
9. Thermo: Contains thermodynamic constants and routines.
10. Index: Contains the macros and functions needed to implement indexing.
11. Grid: Allocates and frees storage.

4 History

4.1 Xforce

Only externally specified forcing is available in this series.

1. This has only externally imposed heating. The model seems to conserve mass very well, as indicated by volume integrating η. The initial mean state is one of constant η.
2. Fixed a bug in the half step computation (extended top and bottom levels not integrated), added a constant horizontal eddy mixing, and allow output file name to be selected from the parameter file.
3. Extend the eddy mixing to include the η equation. Zero vertical fluxes near upper and lower boundaries to aid in conservation. Make vertical index field reference cell centers rather than edges.
4. Minor cleanups: Put numerical parameters in parameter section of Candis output file, others in the comment part. Send time to source calculator. Better indexing for y-only variables such as Coriolis parameter.
5. Add options to control forcing. Parameters added to control radius in each direction, to create a Gaussian mountain, and to turn off the forcing after a specified time.

6. Separate out field initialization calculation into its own module. Also fix serious bug – half-step time step stopped being initialized when the calling sequence for step_do was changed!

7. Add an upper level sponge layer to absorb ascending gravity waves.

8. Add the option of channel boundary conditions with rigid, free-slip walls on the north and south boundaries.

9. Add parameter to produce initial surface potential temperature anomaly. Also, subtract the mean value off of the imposed heating at each level.

10. Add a strain rate dependent part of the eddy mixing coefficient.

11. Bug fixes: (1) Mirror rather than extrapolate in vertical extension. (2) Fix bug in vertical interpolation in vert.c.

12. Implement below surface values equal to lowest above surface values.

13. Add a variable height tropopause to the initialization.

14. Fix bugs in vertical interpolation to surface and in vertical integration to get the Montgomery potential.

15. Remove initial variable height (in θ space) surface and tropopause since these cause problems with the initial sea level pressure distribution. Add the option of positioning the heat source and terrain anywhere in the domain.

4.2 Diabat

Diabatic parameterization "diabat2" incorporated. The surface is ocean and of fixed temperature distribution.

1. Initial version with diabat2.

4.3 Sigma

Change from a pure isentropic coordinate system, which has difficult problems at the surface, to a sigma isentropic coordinate system, as used by Raymond (2001).

1. Initial version with no cumulus parameterization.

2. Fix minor bugs in fixed source term generation and exit with an error message when $\eta < 0.01$.

3. Make lower and upper boundary conditions symmetric reflection conditions.

4. First try at incorporating the diabatic parameterizations represented by the diabat3 package, version 1.16. Added full strength 1-2-1 smoothing on S_σ and conservative 1-2-1 smoothing at strength λ on the prognostic variables. This version is not satisfactory, as the smoothing on the dependent variables causes too much vertical transport.

5. Make the vertical mixing fourth order. Bring the individual contributions of convection and radiation to S_e to the output. Implement the parameters needed to switch between specified fixed radiation and interactive radiation.

6. Add vertical smoothing of η upon input into diabat3. The radiative-convective equilibrium behavior of the model is not bad at this point, with the vertical velocity going nearly to zero. There is still some two-delta-x noise in the vertical distribution of source terms, but the large-scale fields are relatively smooth.

7. Add the capability of importing starting profiles of η and r_t.

8. Forgot to extend some 2-D fields in x and y directions – fix.

9. Some more similar fixes. Add imposed surface temperature pattern. Go back to 2nd order smoothing in the vertical, as 4th order smoothing doesn't do the job.

10. Change vertical mixing from an after effect to incorporated into the main time stepping. Also add deformation-based vertical mixing.

11. Incorporate hard limiting of the values of η to a specified range related to limits in the vertical potential temperature gradient. Propagate excess or deficit mass downward to next grid cell and repeat. Hopefully one ends up with zero deficit at the surface.

12. Change to version toy-016 of diabat3. This only produces convective precipitation above 2 km.

13. Fix a cosmetic bug in the output of sthe_rad and sthe_cu fields.

14. Made vertical fluxes at surface and top zero. This should have been done before. Also fix one minor problem with channel boundary condition.

15. Rearrange location at which diabatic parameterizations are called in the model and smooth sth, srt, fx, fy both horizontally and vertically. Don't smooth ssig directly.

16. Do vertical smoothing of source terms via two passes with a 1-2-1 filter.

17. Correct a programming error which zeroed out the strain rate related part of the vertical mixing coefficient.

18. Propagate out of range densities upward rather than downward, so as to avoid the common problem of a low stability boundary layer stopping the simulation.

19. If leftover mass reaches the top in the upward propagation of out-of-range values of η, propagate it downward. Only if the fault reaches the surface, do we stop the integration.

20. Fix an indexing error which affected only y dimension, and add r_t to the list of variables that is expanded in the y direction.

21. Make surface temperature, topography, and imposed source terms subject to offsets from the domain origin.

22. Add a new input parameter, frict, which multiplies the diabat3 friction and surface flux terms. This is a way to turn off momentum transfer in simulations.

23. Rate limit the production of convective precipitation in diabat3 so that r_t at each level is not reduced by more than 30% at each level for each time step. This cuts out unsteadiness which develops even in radiative-convective equilibrium calculations.

24. Oops! The eddy flux was set proportional to the gradient of intensive quantities times η rather than the gradient of the intensive quantities by themselves. This is clearly wrong, so change. Also, remove the eddy terms from the mass continuity equation, which strictly shouldn't be there. Before they were needed for stability, but apparently not now.

25. Put constant mixing back into η equation to control numerical instability. This is the same as the constant mixing part of the eddy mixing in the other equations.

26. Add lateral sponge layers on the N/S walls when the channel mode is turned on. Revise the values of the eddy coefficient parameters.

27. Add an input parameter α which controls the shape of the mixing strength profile in diabat3. A value of $\alpha = -1$ is equivalent to the old style profile of diabat3, whereas $\alpha = 0$ is equivalent to the profile used in diabat2. (Diabat3 needs to be documented in full!)

28. More initialization options have been added. A zonally symmetric surface temperature perturbation option is now included in addition to the more general localized anomaly. In addition, an option for imposing random fluctuations on the humidity field is now included.

29. Fix the imported single column initialization so that the surface potential temperature is also initialized. Add an imported zonally symmetric initialization of surface potential temperature and 3-D fields η, r_t, v_x, v_y, and S_σ. Simplify the adjustment when η is out of range – don't adjust the other prognostic fields correspondingly. Also, simply set η_t to zero when it takes on a negative value. Remove the recalculation of intensive quantities from the output routine, as this is done more consistently earlier in the code.

30. Make the planetary boundary layer top an adjustable parameter set from the initialization file.

31. Added saturation mixing ratio as an output field.

32. Update version of diabat3 to toy-020. This adds two new parameters, pstiff and pht, which have to do with the dependence of convective precipitation production on height and humidity.

33. Fix error in the specification of surface temperature distribution.

34. Add options for starting the production of output files at some delayed time and for creating output at every timestep for some time interval (microscope option).

35. Fix the numbering of output files to be sequential for delayed starts and microscope option.

36. Make the intermediate step y extension in the channel case consistently one of mirroring. Also, add an input parameter which controls the strength of the lateral sponge damping.

37. Make setting $v_y = 0$ on the boundaries in channel flow explicit in the full time step. Note this is an approximation, as v_y next to the wall is defined half a grid length away from the boundary.

38. No substantive changes – just add diagostics if there is an η fault.

39. Fix a bug in the upstream differencing advection of surface temperature at channel walls. Since the y velocity is set to zero at the grid point adjacent to this boundary, the upstream differencing algorithm got fooled into making the wrong choice. This was rectified for both x and y advection.

40. Add the capability of initializing the topography and the surface temperature with an initialization file rather than with the limited set of options previously available.

41. Fix a bug in the surface temperature initialization which kept this feature from being turned off.

42. Add the first steps toward replacing the constant component of smoothing in the vertical and horizontal by hyper-smoothing, i. e., ∇^4 smoothing. Vertical hyper-smoothing is implimented and horizontal hyper-smoothing in y in the channel case is implimented.

43. Add full hyper-smoothing in both horizontal and vertical directions. Fix an error in the vertical implementation. Change the extension of the centered grid in the horizontal from one to two cells so that the horizontal hyper-smoothing may be done without explicit special cases at the boundaries. This change is extended to all code, so that the only reference to boundary values occurs in the extension subroutine. This should make parallelization easier in the future.

44. Fix a flaw in the upstream advection macros which is rarely exercised, but catastrophic for the surface potential temperature when it happens.

45. Fix a bug at boundaries due to extension of fields not being done at right time. Also, remove all smoothing on source terms, as the hyper-smoothing seems to have fixed problems caused by noisy source terms.

46. Looks like the smoothing of source terms is really needed. Restore (correctly).

47. Make $\alpha = -1$ in diabat3 for the shallow convection case irrespective of what the input file says.

48. Remove all smoothing on the source terms. The noise this introduces used to crash the simulation, but apparently the introduction of hypersmoothing on the prognostic variables fixes this. Also, add partial documentation for diabat3.

49. Fix an important bug: In the call to diabat3, only the z profile in the SW corner of the domain was used over the whole domain. Also, add the first version of Zeljka Fuchs's idealized cumulus parameterization, called ideal1.

50. Change the initial heating function so that it can be made to move in the zonal direction while it is turned on.

51. Fixed a bug: the profile of S_θ was not being set to zero above the tropopause in the ideal1 parameterization; fixed. A relaxation term was added to S_r in the ideal1 parameterization in order to keep the vertical profile of relative humidity constant.

52. Add skew factor ν to the S_θ profile in the ideal1 parameterization.

53. Change the calculation of S_r by removing the relaxation term added in 51 and dumping all of the net surface flux in the range $0 < z < d$.

54. Add some standard declarations to make work with gcc-4.xx.

55. Alter the assumed form for S_r in the ideal1 package. See (52).

56. Decouple the computation of surface equivalent potential temperature from the profile aloft. Make $\theta_e^{'}(0)$ proportional to the surface equivalent potential temperature flux perturbation. This brings the computation into line with that in Raymond and Fuchs (2007). The units on mrelax (α) on input have changed to inverse kiloseconds from inverse seconds.

57. Split P_2 into P_{2s} and P_{2t} in the output diagnostics.

58. Major cleanup: the ideal1 model wasn't completely linear in a number of respects – fix. The saturation fraction rather than the precipitable water was used for forcing as in Raymond and Fuchs (2007) – fix. The documentation was in error in some respects for ideal1 – fix. Some reorganization is done on the documentation.

59. The value of the convective throttle slop $\Delta\theta_e$, which was previously set to 4 K, is now brought out as an input variable named theslop.

60. Add the option of a relaxation of the mean zonal wind to the initial mean zonal wind profile. Also, fix some inconsistencies in the notation for source terms.

61. New version of diabat3 – TOY-027. This version augments the vertical mixing rate with saturation fraction and has a smoother vertical profile of convective precipitation formation.

62. New version of diabat3 – TOY-028. The only change is to rate limit convective precipitation more when the stiffness parameter s is larger.

63. There was an error in the calculation of the horizontal eddy coefficient – the cross term in the strain rate (57) was calculated incorrectly. Fix.

64. Due to a bug in diabat3 (found by Michael Herman). S_θ was not calculated at every time step, but only when radiation routine was called – by default, every fifth time step. Fixed.

65. Add single column model from diabat3 to package.

66. Fix so that cumulus mixing takes on the maximum value in shallow convection and scales with saturation fraction for deep convection. Also, update and correct some of the documentation.

67. Add the radiation input parameter ϵ to the ideal1 parameterization.

68. Now under subversion version control. Renamed main routine sigma.c.

69. Change the zonal wind relaxation so that the zonal mean zonal wind relaxes to a meridional-height pattern rather than to a height profile. Note that this relaxation is currently undocumented.

70. Add a random initial density perturbation at low levels to match the initial random humidity perturbation. This creates a new parameter to indicate the typical fractional perturbation. Also, improve the documentation on the zonal wind relaxation. In the column model, recomputed hydrostatic balance after each call to the parameterization. This makes the solutions internally consistent.

71. Turn on hard adjustment for dry instability in diabat3. This can't happen in sigma coordinates, but it can happen in the single column model which uses geometrical height as an independent variable.

72. Increase the bounds on vertical potential temperature gradient from $[1, 20]$ K km^{-1} to $[1, 200]$ K km^{-1}. Split the coefficient *ck* for deformation-based eddy coefficients into horizontal and vertical coefficients *ckh* and *ckv*. Update some values in the input file.

73. Make the vertical profile of convective rate for deep convection adjustable in shape. Make the calculation of planetary boundary layer average quantities and the equivalent potential temperature threshold for convection smooth quantities. Turn off hard dry adjustment again, as it can cause stability problems. This should be handled elsewhere if needed.

74. Revert to the old way (pre-73) of calculating PBL average quantities and theta-e threshold. Also, revert to a maximum potential temperature gradient of 20 K km^{-1} for now. We are now back to the behavior of version 70 except that ck is still separated into horizontal and vertical parts and α is available as an external parameter.

75. Increase the bounds on vertical potential temperature gradient from $[1, 20]$ K km^{-1} to $[1, 200]$ K km^{-1} as in sigma-072.

76. Remove the smoothing of the density η before passing it to diabat3. This actually increases stability and fidelity.

77. Add vertical 1-2-1 smoothing of diabat3 source terms before applying them in the large-scale model. Go back to a limit of 20 K km on the vertical potential temperature gradient. Both of these changes appear to increase the stability of the model and reduce the probability of the development of small-scale inversions of numerical origin which crash the model.

78. Make boundary layer values and threshold values averages over grid levels surrounding the target value with a Gaussian weighting function centered on the target value. The levels are not weighted by the mass at each layer to avoid possible numerical instability.

79. Make the convective precipitation conversion term proportional to an additional power of $r_s(z)$ so that the convective precipitation is weighted more to lower levels.

80. Change the form of the convective mixing rate profiles to exponential functions of height. Remove the factor of 5 in front of the shallow mode convective mixing rate. This results in much smoother vertical profiles at low levels.

81. Add parameters to control the rate at which α varies with saturation fraction for deep convection. The parameter α controls the vertical profile of convective mixing.

82. Change the way the initial moisture profile is computed when the ideal1 parameterization is turned on. The mixing ratio takes the form

$$r_t(z) = H r_s(0) \exp(-z/z_q)$$

where H is the specified relative humidity, $r_s(0)$ is the saturation mixing ratio at the surface, and z_q is the specified mixing ratio scale height. This allows the ambient vertical profile of moist entropy to be varied. The value of $r_t(z)$ is not allowed to exceed $r_s(z)$.

83. For column model introduce a new parameter rtmul, which is a factor which increases the initial humidity without changing the potential temperature. Sigma is unchanged.

84. Introduce the estep parameter which makes the target θ_e profile increase or decrease linearly with height.

85. Change estep so that it is a dimensionless multiplier to the RMS variance of the θ_e in the convective layer. Thus, the difference between the target θ_e at the top and the bottom of the convective layer is this RMS variance times estep.

86. Undo the addition of an additional power of saturation mixing ratio to the convective precipitation rate formula in diabat3.c. This extra term allows the upper troposphere to stay saturated.

87. We go back to the Raymond and Fuchs (2009) treatment of $a(z)$ in which a is constant for deep convection and decreases linearly through the PBL for shallow convection. The code is also rearranged a bit in this department to make it clearer. Remove the estep parameter.

88. Do some more code simplification and reintroduce non-constant profile of deep convective relaxation rate, $a(z)$.

89. No change to the calculation. Extract stratiform rain, penetrative convection rain, and rain evaporation rate as separate output fields.

90. Reintroduce the estep parameter introduced in version 85 and removed in version 87.

91. Introduce checks to keep the intermediate values of eta (η) and ert (η_t) in range in function step.

92. Change all memory allocation-free cycles to allocation on first call and no free.

93. Add a wall clock package which allows wall clock timing between arbitrary points in the code. This was invented because it was found that the code slowed down drastically after a few hundred time steps when run in the AMD64 architecture. The slowdown occurred in the radiation code in diabat3 and was traced to the Glib math exponential function, which would randomly take much longer to execute than normal. The reason for this is uncertain. However, as the exponential function is used extensively in the radiation code, a home-brewed exponential function called safeexp was created, which uses the regular exponential function to generate a lookup table. The slowdown of the code was eliminated with this modification. (Safeexp is used only in the radiation code, as its accuracy is limited to about 4 significant digits.) There were also some minor changes in the radiative code; a limitation on the max value of tau was lifted, as due to other checks, there was no reason for this anymore. The code appears to run as fast or faster than the original i386 version of the code (which doesn't exhibit the AMD64 slowdowns), but gives the same results.

94. Add an input parameter to change the seed of the random number generator so that different instances of an ensemble of runs can be generated which differ only in their random initialization.

95. Modify the suppression of stratiform rain by the equivalent potential temperature lapse rate. Change the throttle turn on range from 1 K km^{-1} – 3 K km^{-1} to 0 K km^{-1} – 2 K km^{-1}. This solves the problem of accumulation of water at upper levels in stratiform regions. No obvious bull's eye instabilities in an MJO simulation with 250 km resolution.

96. Implement heating and moistening mechanisms in order to perform Z. Kuang's linear response function analysis described in Kuang (2010). Tendencies are added to srt and sthe according to a new set of parameters listed in the input file: thm1, the time (ks) when the tendency is to begin; thm2, the time to end the tendency; dtempdt, the prescribed heating rate (K/day); dqdt, the prescribed moistening rate (g/kg/day);

gmode, a parameter decribing the location of the perturbed layer. An inner-loop time variable called finetime has been added to facilitate this mechanism.

97. Do minor code cleanup, add the saturation mixing ratio rsat to the output, and introduce a refinement of the interactive radiation code in diabat3. Documentation yet to come. Also, in column.c, put final hydrostatic adjustment call inside inner time loop.

98. Bring the terminal velocities of rain and snow out to the input file rather than setting them internally.

99. Add a land fraction parameter whose only function is to reduce the surface moisture flux to zero when this is unity. This is appropriate for a bone-dry land situation. Currently this parameter is specified for the imposed heating option. For the specified SST pattern option, this parameter is set to one minus the value of the land mask field in the SST file.

100. Actually, the land fraction calculation was done wrong. 1 - landfrac = landrh should multiply the saturated sea surface mixing ratio in both the surface evaporation rate and the surface flux of moist entropy.

101. The land case is still not done right! Fix.

102. Introduce linear surface friction of the form (F_x, F_y) = -lam*(vx, vy)*exp(-mu*z) into the ideal 1 parameterization. Also, a horizontally sinusoidal initial zonal wind was added in order to explore the nature of Ekman pumping.

103. Fix the cloud absorption part of the radiation code. There were extraneous multiplicative parameters in the calculation of optical depth. Changed to the form delta tau = cld*rl*rho*dz where cld is the input parameter for cloud absorption, rl is the cloud water mixing ratio in g/g, rho is the air density in kg/m3̌, and dz is the layer thickness in km. cld thus has the dimensions m^3/kg/km.

104. Make the code more efficient by calling only clear or cloudy radiation code when the cloud fraction is 0 or 1.

105. Column model changes.

106. Change diabat3 so that the surface model is only active over land. Thus, the SST stays fixed. Changed the sigma model to work with the new diabat3 with land surface. The surface moisture column is initialized to zero. The surface temperature and the land surface distribution are initialized either through a surface input file or an idealized initialization.

107. Put a separate temperature rate parameter for the sea surface layer into diabat3 and modify the sigma and column models to handle this parameter.

108. The individual rain components were not being computed correctly. Fix.

109. Document the estep parameter (λ_x) which controls the shape of the target θ_e profile without changing the mean value of this profile. Also change the form of $a(z)$ for shallow convection to an exponential. The control parameter α for deep convection is zero, while that for shallow convection is -4. This works in conjunction with a deeper boundary layer than used previously. The net result is a smoother transition between the zones of influence of shallow and deep convection.

110. Simplify the entire linear bias treatment for the θ_e target profile. The controlling parameter (formerly λ_x) is now called λ_l (estep parameter). Revert back to the linear treatment of $a(z)$ with $\alpha_d = 0$ and $\alpha_s = -1$. This seems to be more robust.

111. Unrevert back to the exponential treatment of $a(z)$ with $\alpha_d = 0$ and $\alpha_s = -3$. Do not allow the shallow mode to precipitate at all and shut off stratiform precipitation for elevations less than pscale.

112. Major simplification of diabat3: Penetrative precipitation is removed and stratiform precipitation commences at a relative humidity less than unity. It turns on gradually with height. We go back to a linear treatment of $a(z)$ with $\alpha_d = 0$ and $\alpha_s = -1$. The convective mixing rate is set to be the same for shallow and deep convection. With the specified values, mixing to completion occurs for slow processes, so the exact value doesn't seem to matter too much.

5 References

Briegleb, B. P., 1992a: Delta-Eddington approximation for solar radiation in the NCAR Community Climate Model. *J. Geophys. Res.*, **97**, 7603-7612.

Briegleb, B. P., 1992b: Longwave band model for thermal radiation in climate studies. *J. Geophys. Res.*, **97**, 11475-11485.

Hack, J. J., B. A. Boville, B. P. Briegleb, J. T. Kiehl, P. J. Rasch, D. L. Williamson, 1993: Description of the NCAR community climate model (CCM2). NCAR/TN-382+STR, 108 pp.

Kuang, Z., 2010: Linear response functions of a cumulus ensemble to temperature and moisture perturbations and implications for the dynamics of convectively coupled waves. *J. Atmos. Sci.*, **67**, 941-962.

Raymond, D. J., and D. J. Torres, 1998: Fundamental moist modes of the equatorial troposphere. *J. Atmos. Sci.*, **55**, 1771-1790.

Raymond, D. J., and X. Zeng, 2000: Instability and large-scale circulations in a two-column model of the tropical troposphere. *Quart. J. Roy. Meteor. Soc.*, **126**, 3117-3135.

Raymond, D. J., 2001: A new model of the Madden-Julian oscillation. *J. Atmos. Sci.*, **58**, 2807-2819.

Raymond, D. J., and Ž. Fuchs, 2007: Gravity and moisture modes in a simple atmospheric model. *Tellus*, **59A**, 627-640.

Raymond, D. J., and Ž. Fuchs, 2009: Moisture modes and the Madden-Julian oscillation. *J. Climate*, **22**, 3031-3046.

Raymond, D. J., and D. J. Torres, 1998: Fundamental moist modes of the equatorial troposphere. *J. Atmos. Sci.*, **55**, 1771-1790.