# A Toy Cumulus Ensemble Model

David J. Raymond

March 21, 2017

## 1 Introduction

This documents the development of a toy cumulus ensemble model which interacts with the large scale via Sobel and Bretherton's (2000) weak temperature gradient approximation. First results of the model are published in Raymond and Zeng (2005).

## 2 Governing equations

We first present the fundamental dynamic equations. Then the thermodynamic and cloud physical assumptions are presented, followed by the implementation of the weak temperature gradient approximation.

### 2.1 Fundamental equations

We start with the mass and momentum equations

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \boldsymbol{v}) = 0 \tag{1}$$

and

$$\frac{\partial \rho \boldsymbol{v}}{\partial t} + \nabla \cdot (\rho \boldsymbol{v}\boldsymbol{v} - K\boldsymbol{D}) + \nabla p + g\rho \boldsymbol{k} = \rho(\boldsymbol{F}_s - \boldsymbol{F}) - \rho\mu(\boldsymbol{v} - \boldsymbol{v}_0), \tag{2}$$

where $\rho$ is the density, $\boldsymbol{v}$ is the velocity, $\boldsymbol{v}_0(z)$ is a target horizontal velocity profile, $\mu(z)$ is an externally specified damping profile turned on only in the stratosphere for the purpose of absorbing upward-propagating gravity waves, $p$ is pressure, $\boldsymbol{F}_s$ is the force due to surface stresses, $\boldsymbol{F}$ is an external momentum sink, and $\boldsymbol{D}$ is the deformation rate tensor

$$D_{ij} = \frac{1}{2}\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right). \tag{3}$$

For the thermodynamic equations we have

$$\frac{\partial \rho s}{\partial t} + \nabla \cdot (\rho \boldsymbol{v} s - K \nabla s) = \rho(S_{ss} + S_{sr} - S_{se}), \tag{4}$$

where s is specific moist entropy, $S_{ss}$ is the source of entropy from surface fluxes, $S_{sr}$ is the source from radiation, $S_{se}$ is the external entropy sink, and $K$ is the eddy mixing coefficient,

$$\frac{\partial \rho r_t}{\partial t} + \nabla \cdot (\rho \boldsymbol{v} r_t - K \nabla r_t) = \rho(S_{rs} + S_{rp} - S_{re}), \tag{5}$$

where $r_t = r + r_c$ is total cloud water mixing ratio, $r$ is the vapor mixing ratio, $r_c$ is the mixing ratio of advected condensate, $S_{rs}$ is the source of total cloud water mixing ratio from surface evaporation, $S_{re}$ is the external sink of total cloud water mixing ratio, $S_{rp}$ is minus the conversion rate of cloud water to precipitation, and

$$\frac{\partial \rho r_r}{\partial t} + \nabla \cdot [\rho(\boldsymbol{v} - w_t \boldsymbol{k}) r_r - K \nabla r_r - K_n \frac{\partial r_r}{\partial z}] = -\rho S_{rp}, \tag{6}$$

where $r_r$ is rainfall mixing ratio.

The radiative source of specific moist entropy $S_{sr}$ is calculated either from a fixed profile or from an interactive radiative transfer calculation. The fixed profile is either a constant equivalent potential temperature sink (tapered to zero close to the tropopause) or a profile provided in the profile file. The interactive radiation is computed by the toy radiation package of Raymond (2001) as updated. The most current documentation is that in the documentation file for the sigma model, starting at sigma-104 (r101). The radiative cooling is not called at every time step in order to save computation. Calling the radiation about every 10 time steps reduces the computational load of the radiation to 10-20% of the total time.

The eddy mixing coefficient is non-zero when the square of the deformation rate minus twice the Brunt frequency is positive:

$$K = \rho C \left[\sum_{i,j} D_{ij} D_{ij} - \frac{2g}{\theta}\Gamma_e\right]^{1/2} \Delta z^2 \tag{7}$$

where $C \approx 1$ and $\Delta z$ is the vertical grid size. We set $K = 0$ when the quantity inside the square brackets is negative. This corresponds to having the Richardson number greater than 1/4. The effective stability is

$$\Gamma_e = A \frac{\partial \theta_e}{\partial z} + (1 - A) \frac{\partial \theta}{\partial z}, \tag{8}$$

where $\theta$ is the potential temperature, $\theta_e$ is the equivalent potential temperature, $A = [(H - 1)/\delta H + 1]/2$, and where $H = r_t/r_s$, $r_s$ being the saturation mixing ratio. We limit $A$ to the range $0 \leq A \leq 1$.

The rain mixing ratio equation has additional numerical viscosity to suppress $2\Delta z$ wiggles in the vertical which only occur in this variable:

$$K_n = \lambda \rho (\Delta z^2 / \Delta t) \tag{9}$$

where $\lambda = 0.001$ seems to produce adequate smoothing.

The above-mentioned sinks of momentum, moisture, and equivalent potential temperature just balance the sources of these quantities due to convective and radiative processes in the steady state, and thus on the average are equivalent to the large scale *sources* for these quantities. They therefore approximate the source terms for the large-scale dynamical equations. The large-scale sink term is omitted from the rainwater equation since the time scale for this equation is in general much shorter than that for large-scale vertical advection. The forms of the sink terms are discussed later in the section on the weak temperature gradient approximation.

## 2.2 Thermodynamics and cloud physics

We use an approximate formula for the specific moist entropy $s$,

$$s = C_P \ln(T/T_R) - R_d \ln(p/p_R) + (L_c + L_f) r / T_R \tag{10}$$

where $T$ is the temperature, $T_R = 300$ K is a constant reference temperature, $p_R = 100000$ Pa is a constant reference pressure, $L_c = 2.50 \times 10^6$ J kg$^{-1}$ and $L_f = 3.33 \times 10^5$ J kg$^{-1}$ are the latent heats of condensation and freezing, $R_d = 287$ J kg$^{-1}$ K$^{-1}$ is the gas constant for dry air, and the specific heat of air at constant pressure is $C_p = 1005$ J kg$^{-1}$ K$^{-1}$. The equivalent potential temperature is related to the entropy by

$$\theta_e = T_R \exp(s/C_p). \tag{11}$$

The saturated moist entropy $s_s$ is obtained by replacing the mixing ratio with the saturated mixing ratio in the entropy formula. The saturation mixing ratio $r_s = \epsilon e_s(T)/p$, where $\epsilon = 0.622$, and $e_s(T)$ is the saturation vapor pressure in Pascals at temperature $T$ (in Kelvins), approximated by Teton's formula:

$$e_s(T) = 611.2 \exp[17.67(T - 273.15)/(T - 29.65)]. \tag{12}$$

The ideal gas law

$$T = \frac{p}{\rho R_d}, \tag{13}$$

is used along with the definition of potential temperature

$$\theta = T(p_R/p)^\kappa \tag{14}$$

to derive a diagnostic expression for the pressure as a function of density and potential temperature:

$$p = (\rho R_d \theta)^\gamma / p_R^{\gamma-1}, \tag{15}$$

where $\gamma = C_p/C_v = 1.4$ is the ratio of specific heats, and $\kappa = 1 - 1/\gamma$.

The particle terminal velocity $w_t$ is externally specified with different values above ($w_{ti}$) and below ($w_{tw}$) the freezing level. The total water sink term is given by

$$S_{rp} = -\lambda_r(r_t - r_s), \quad r_t > r_s \tag{16}$$

and

$$S_{rp} = -\lambda_e r_r(r_t - r_s), \quad r_t < r_s, \tag{17}$$

where $r_s$ is the saturation mixing ratio. The constants $\lambda_r$ and $\lambda_e$ are externally specified rates. The parameter $\lambda_e$ is constant and $\lambda_r$ has different values above ($\lambda_{ri}$) and below ($\lambda_{rw}$) the freezing level.

## 2.3 Weak temperature gradient approximation

### 2.3.1 Conventional WTG

We define a quantity $w_{wtg}$, which is the domain-mean vertical velocity obtained from the weak temperature gradient approximation of Sobel and Bretherton, i. e., it is that imaginary vertical velocity consistent with keeping the horizontal-mean potential temperature profile fixed in time:

$$w_{wtg} = \left(\frac{\partial \overline{\theta}}{\partial z}\right)^{-1} S_\theta \tag{18}$$

The quantity $S_\theta$ is the large-scale potential temperature source due to convective heating.

In the boundary layer this relationship breaks down because the potential temperature gradient becomes weak. Because of this we replace $w_{wtg}$ in the boundary layer by a linear interpolation between the value of $w_{wtg}$ at the boundary layer top $z = b$ and the zero surface value.

We assume that the large-scale heating is related to a deviation from an assumed reference profile $\theta_0(z)$:

$$S_\theta = \lambda_{st} M(z)(\overline{\theta} - \theta_0). \tag{19}$$

The quantity $\lambda_{st}$ is the assumed potential temperature relaxation rate, and is modulated by a mask $M(z) = \sin(\pi z/h), z < h$, where $h$ is the height of the tropopause, which keeps this relaxation from acting at the surface and the tropopause. We assume that $M(z) = 0$ for $z > h$.

### 2.3.2 Spectral WTG

The classical weak temperature gradient (WTG) vertical velocity can be written in terms of the horizontally averaged potential temperature excess $\theta'$ over the reference profile as

$$w_{wtg}(z) = \frac{\overline{\theta}'(z)}{\tau(d\overline{\theta}/dz)}, \tag{20}$$

where $\tau$ is the relaxation time, $\overline{\theta}(z)$ is the mean potential temperature profile, and where the mask has been set to $M(z) = 1$ for $z < h$ here. The relaxation time $\tau = 1/\lambda_{st}$ is specified in terms of the horizontal scale $L$ of the potential temperature anomaly, and hence the scale of the heating, and the speed $c$ of the gravity waves that disperse the potential temperature anomaly:

$$\tau = L/c. \tag{21}$$

This has the problem that $\tau$ is only valid for potential temperature anomaly profiles with vertical scale equal to the vertical scale of the gravity wave. Real profiles may be spectrally rich and be dissipated by gravity waves of many different vertical scales.

We solve this problem by decomposing the scaled potential temperature anomaly into a superposition of approximate gravity wave vertical eigenmodes, taken here to be sine functions bounded above by the tropopause and below by the surface,

$$\frac{\overline{\theta}'(z)}{(d\bar{\theta}/dz)} = \sum_j \Theta_j \sin(m_j z), \qquad 0 \leq z \leq h, \tag{22}$$

where $h$ is the height of the tropopause and the vertical wavenumber is

$$m_j = \frac{j\pi}{h}, \qquad j = 1, 2, 3, .... \tag{23}$$

The number of vertical modes equals $h/\Delta z$ where $\Delta z$ is the vertical grid size. Inversion of this sine series yields the coefficients

$$\Theta_j = \frac{2}{h} \int_0^h \frac{\overline{\theta}'(z) \sin(m_j z)}{(d\bar{\theta}/dz)} dz. \tag{24}$$

The WTG vertical velocity is then written

$$w_{wtg} = \sum_j \frac{\Theta_j \sin(m_j z)}{\tau_j} \tag{25}$$

where $\tau_j$ are the relaxation times for the various vertical scales. If $\tau_j = \tau$ independent of $j$, then this expression reduces to the convectional WTG vertical velocity (20).

The relaxation times can be written in terms of $L$ and the gravity wave speeds for each vertical mode,

$$\tau_j = L/c_j, \tag{26}$$

where a simple model of hydrostatic gravity waves yields

$$c_j = N/m_j \tag{27}$$

with $N$ being the approximate Brunt-Väisälä frequency of the troposphere, assumed constant. Rewriting in terms of $j$, we have

$$w_{wtg} = \frac{Nh}{\pi L} \sum_j \frac{\Theta_j \sin(j\pi z/h)}{j}. \tag{28}$$

As with conventional WTG, lower bounds have to be placed on $d\bar{\theta}/dz$ in order to keep the analysis from blowing up.

### 2.3.3 Water vapor, entropy, and horizontal momentum

The external sink of total water mixing ratio is a combination of entrainment from the surrounding environment, large-scale vertical advection by the mean vertical velocity $w_{wtg}$, and an imposed relaxation to the reference profile:

$$S_{re} = \lambda_{xm}(\overline{r}_t - r_{t-ent})\frac{1}{\rho_0}\frac{\partial \rho_0 w_{wtg}}{\partial z} + w_{wtg}\frac{\partial \overline{r}_t}{\partial z} + \lambda_{sm}(\overline{r_t} - r_{t0}), \qquad (29)$$

We set $r_{t-ent}$ to $\overline{r}_t$ at detraining levels, i. e., where $\partial \rho_0 w_{wtg}/\partial z < 0$, and to the reference profile $r_{t0}$ where $\partial \rho_0 w_{wtg}/\partial z > 0$. A similar equation is written for the entropy sink:

$$S_{se} = \lambda_{xm}(\overline{s} - s_{ent})\frac{1}{\rho_0}\frac{\partial \rho_0 w_{wtg}}{\partial z} + w_{wtg}\frac{\partial \overline{s}}{\partial z} + \lambda_{sm}(\overline{s} - s_0). \qquad (30)$$

The term containing $\lambda_{sm}$ represents an Adam Sobel style additional relaxation of the moisture and moist entropy toward the reference profile whereas the term containing $\lambda_{xm}$ controls the lateral entrainment, being set either to zero or one. The Sobel relaxation rate $\lambda_{sm}$ is set to any desired value reflecting the ambient flow of moisture into the system, as via a rotational wind (Sobel et al, 2007).

We can make the thermodynamic reference profiles $s_0$ and $r_{t0}$ functions of time in order to explore the effect of time-varying background conditions.

The external sink of momentum in (2) is a relaxation toward a reference profile of horizontal velocity $\boldsymbol{v}_0$ at a rate $\lambda_{sd}$:

$$\boldsymbol{F} = \lambda_{sd}(\overline{\boldsymbol{v}} - \boldsymbol{v}_0). \qquad (31)$$

We ignore vertical advection of momentum by $w_{wtg}$ because in general we make the effect of relaxation strong enough to force the mean wind to the specified wind profile.

### 2.3.4 Reference profiles

The simplest reference profiles for potential temperature and mixing ratio are taken from radiative-convective equilibrium (RCE) calculations with the model. These profiles are introduced into the calculation by the initialization file. The profiles can be altered as desired by modifying the initialization file. In addition to potential temperature and mixing ratio, a specified profile of radiative moist entropy source must be included in the initialization file. This way, non-standard fixed radiative heating profiles can be introduced. Finally, a mean sea surface temperature must be included. All of these quantities can be made time dependent in the initialization file. The height must be the first index field in this case and the time the second. If the model time exceeds the maximum time in the index field, the model returns to the initial time in the initialization file, allowing the easy introduction of periodic forcing.

It is unwise to use actual data for initialization profiles, as the model thermodynamics differs from nature's thermodynamics. In this case, introducing variable profiles as perturbations on model RCE profiles is a viable strategy.

# 3 Numerical issues

Lax-Wendroff differencing is used, with cell-centered fields. Boundary conditions in $x$ and $y$ are periodic and there are rigid lids top and bottom in the $z$ direction. The lids pass through the centers of the top and bottom cells as illustrated in figure 1.

Three grids are used in the model, the $I$ grid, the $J$ grid, and the $L$ grid. The $I$ grid is the normal grid in which field data are stored – grid-cell-centered values, with no repetition in the $x$ and $y$ directions. The $J$ grid consists of grid corner values. Thus, there is one more point each in the $x$, $y$, and $z$ directions in the $J$ grid than there is in the $I$ grid. The $I$ grid is contained entirely inside the $J$ grid. Notice that the top and bottom grid points in the $J$ grid are located on the upper and lower rigid surfaces. The $J$ grid repeats once in the $x$ and $y$ directions. The lowest and highest levels of the $J$ grid are outside of the computational domain. Fluxes at these points are filled by mirror conditions at the horizontal rigid surfaces. Fields are extended by linear interpolation.

The $L$ grid is cell-centered like the $I$ grid except that it is extended one point each in the $+x$ and $+y$ directions using periodic boundary conditions in these directions, with similar extensions in the $-x$ and $-y$ directions.

The Lax-Wendroff differencing proceeds as follows: Starting with input data on an $I$ grid, fluxes are computed on an $L$ grid using periodic domain extension. Intermediate results are then computed using the $L$ grid data on a $J$ grid. The final results on an $I$ grid are then computed from the intermediate results on the $J$ grid.

Lax-Wendroff tends to produce some mild two-delta-$x$ wiggles in the hor-

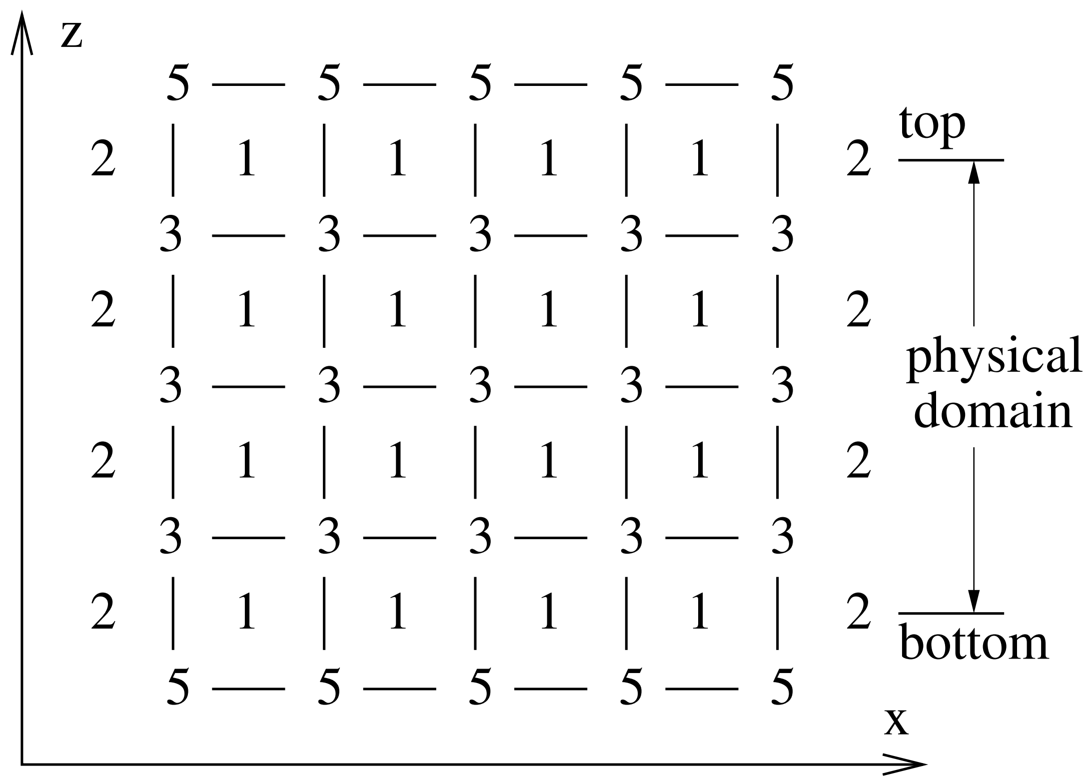


Figure 1: Computational grid. 1: $I$ grid. 2: Extension of $I$ grid to make $L$ grid. 3: $J$ grid as computed from $L$ grid. 5: vertical extension of $J$ grid. Structure in the $y$ direction is the same as in the $x$ direction.

izontal and vertical, which need to be filtered out. This is done horizontal wiggles by adding the term

$$\lambda_h F(\rho\chi)\Delta t/\rho \tag{32}$$

to the prognostic equation for dependent variable $\chi$, where

$$\begin{aligned} F_{i,j}(\rho\chi) &= 0.25(\rho_{i+1,j}\chi_{i+1,j} + \rho_{i-1,j}\chi_{i-1,j} \\ &+ \rho_{i,j+1}\chi_{i,j+1} + \rho_{i,j-1}\chi_{i,j-1}) \\ &- \rho_{i,j}\chi_{i,j}. \end{aligned} \tag{33}$$

The $i$ and $j$ indices indicate the $x$ and $y$ dimensions. For $\chi = 1$, the filtering of the density $\rho$ is accomplished. The constant $\lambda_h$ typically takes a value in the $0.001 - 0.01$ range.

No numerical filtering is done in the vertical outside of the extra smoothing on the precipitation field described above.

The sea surface temperature is allowed to vary sinusoidally in the $x$ direction, with

$$T_{ss} = T_{ss0} - \delta T_{ss} \cos(2\pi x/L) \tag{34}$$

where $T_{ss}$ is the mean sea surface temperature, $\delta T_{ss}$ is its deviation, and $L$ is the size of the domain in the $x$ direction.

The surface flux sources are concentrated in the lowest model layer and are derived from the bulk flux formula:

$$S_{es} = C_d U_e [s_{ss} - s(0)]/(\Delta z/2) \tag{35}$$

where $C_d = 0.001$ is the drag coefficient, $s_{ss}$ is the saturated sea surface entropy, $s(0)$ is the entropy of the air at the lowest model level, $\Delta z$ is the vertical grid size, and where

$$U_e = [v_x(0)^2 + v_y(0)^2 + W^2]^{1/2} \tag{36}$$

is the effective surface wind, $W \approx 3$ m s$^{-1}$ being an externally set parameter;

$$S_{rs} = C_d U_e [r_{ss} - r_t(0)]/(\Delta z/2) \tag{37}$$

where $r_{ss}$ is the saturation mixing ratio at the sea surface temperature and pressure;

$$\boldsymbol{F}_s = -C_d U_e \boldsymbol{v}(0)/(\Delta z/2). \tag{38}$$

The surface fluxes are dumped in a layer of thickness $\Delta z/2$ because this is the part of the lowest grid box which is above the surface.

The contributions of eddy mixing, the external sink terms, and the precipitation generation term are only included in the final step of the Lax-Wendroff scheme since a very short time step is dictated by the CFL criterion on sound waves. Since these are relatively slow processes, the resulting error should be small. The vertical moisture advection uses upstream differencing.

The sink terms in the governing equations and the associated horizontal averaging associated with WTG are not calculated at every time step, as these quantities should not change rapidly in time. They also break the efficient parallelization of the code when they are calculated, as the master process must transfer a significant amount of data from all slave processes.

# 4 Release notes

## 4.1 Sound

### 4.1.1 Sound-001

In this version gravity is turned off, potential temperature takes a constant value of 300 K, and there is no moisture. Moisture variables are not implemented. The $z$ boundary condition is also periodic. Initialization is via a centered density-pressure anomaly of size determined on the command line.

This version has no explicit damping. It works well in propagating weak sound waves, but misbehaves in large amplitude situations when shock waves develop.

### 4.1.2 Sound-002

In this version the periodic boundary conditions in $z$ are replaced by mirror boundary conditions at the top and bottom. This is equivalent to a rigid floor and ceiling with free-slip conditions there.

Sound waves reflect nicely off of these surfaces.

## 4.2 Gravity

### 4.2.1 Gravity-001

In this version the potential temperature and equivalent potential temperature fields are activated and gravity is added. Hydrostatic initialization is done.

The hydrostatic initialization doesn't work very well because the time stepping has a different notion of hydrostatic equilibrium than the initialization does, due to numerics.

### 4.2.2 Gravity-002

A couple of fixes near the upper and lower boundaries alleviated but didn't completely fix the initialization problem.

### 4.2.3 Gravity-004

We changed the location of the rigid lids from the lower and upper boundaries of the lower and upper cells to the centers of these cells. This makes application of the boundary condition and the initialization easier. Also a bug was fixed in the final step integration. The integration wasn't completely consistent with the Lax-Wendroff algorithm at this stage, resulting in a nasty instability.

The model now makes nice clean gravity waves.

### 4.2.4 Gravity-007

Added a Richardson number based eddy mixing coefficient, which seems to smooth out gradients as expected.

### 4.2.5 Gravity-008

There are no changes of substance. However, the starting information is now read from a file rather than the command line. The name of the start file and the revision number of main.c are also put into the output Candis files.

## 4.3 Moist

### 4.3.1 Moist-001

Add moist thermodynamics. The VERSION parameter is now the snapshot value (moist-001 in this case). Use an effective moist Richardson number criterion for eddy viscosity calculation. There is still no rain production.

The eddy diffusivity doesn't work very effectively to smooth out fluctuations in this case, because it is zero in regimes where smoothing is needed.

### 4.3.2 Moist-002

The calculation of the eddy diffusivity is revised so that the square of the shear is replaced by the square of the deformation rate, and the factor of four is removed from the Brunt frequency term.

This version does a much better job of smoothing out wiggles where needed. Tests involving the release of a three-dimensional dry thermal in a stratified environment suggest that the optimal eddy mixing coefficient $C \approx 1$. Smaller values result in aliasing which can produce unphysical values of potential temperature.

### 4.3.3 Moist-003

Change the calculation of the effective stability for the computation of the eddy mixing coefficient. Make the effective stability equal to half moist, half dry just at saturation, with a linear transition to full moist or full dry for positive or negative relative humidity excursions of magnitude $\delta H$. (Relative humidity, defined here as the total cloud water mixing ratio divided by the saturation mixing ratio, can be greater than unity when there is condensed cloud water.)

Tests show that the development of three-dimensional convective plumes is very insensitive to the value of $\delta H$ over the range of values $0.05 - 0.20$. Let's choose $\delta H = 0.1$ as a default value.

## 4.4 Rain

### 4.4.1 Rain-001

A crude simulation of rain production, evaporation, and fallout is now included. The particle terminal velocity is taken to be constant and rain production and evaporation are accounted for by simple linear rate processes.

### 4.4.2 Rain-002

The horizontally averaged profiles of wind, equivalent potential temperature, and total cloud water mixing ratio are now relaxed to externally determined profiles. The external wind profiles are defined by their surface value, shear, and curvature. The external thermodynamic profiles are defined as before.

### 4.4.3 Rain-003

Surface fluxes are added, assuming that the underlying surface is ocean of specified temperature. The relaxation to externally determined profiles is eliminated for the lowest model level and the surface fluxes are dumped totally into this layer. A bug fix was made which brings the surface pressure much closer to the specified reference pressure.

### 4.4.4 Rain-004

No change in the model physics. Just add the saturated equivalent potential temperature and the rainfall rate to the output.

### 4.4.5 Rain-005

Change all floats to doubles, as it appears that we are having some precision problems. On the upper and lower boundary conditions, set vertical fluxes to zero on the topmost and bottommost cell boundaries even though the bounding surfaces are centered in the top and bottom cells. This is needed to make the model truly flux-conservative.

### 4.4.6 Rain-006

Add the option of a fixed radiative cooling rate up to the tropopause.

### 4.4.7 Rain-007

Add damping of the momentum equation in the stratosphere for the purpose of absorbing upward-propagating gravity waves. The Newtonian damping drives the velocity to the above-specified external wind profile at a rate which starts at zero at the tropopause and increases linearly with height to a maximum value at the top of the domain.

This keeps normal eddy mixing from occurring in the stratosphere. Eddy mixing is bad there, because it transports heat, which changes the stratospheric temperature profile.

#### 4.4.8 Rain-008

Add a spatially constant source term to the mass continuity equation which tends to keep the mean surface pressure fixed as the domain average temperature changes in the model. This works by gradually adjusting the total amount of mass in the model domain. We set the adjustment rate to $\nu = 0.001$ s$^{-1}$.

#### 4.4.9 Rain-009

Define separate thermodynamic and dynamic relaxation times, so that, for instance, we can force the wind profile to a specified value, but can leave the thermodynamic profiles unconstrained.

#### 4.4.10 Rain-010

Perfect the table-based technique for calculating $\theta$ from $\theta_e$. Also fix a bug in the calculation of $\theta$ in the first time step.

#### 4.4.11 Rain-011

Revise the computation of rainfall rate – compute it internally in timestep from the surface rain flux rather than trying to infer it from the rainwater content and terminal velocity at the surface. This is likely to be more accurate.

#### 4.4.12 Rain-013

Allow total water and rain water mixing ratios to potentially become negative, but protect against bad effects of this. Also made a slight technical revision of how rainflux was calculated.

#### 4.4.13 Rain-014

We are getting a negative spike of total cloud water at the level at which the eddy mixing coefficient goes to zero. Try doing 1-2-1 smoothing on eddy mixing coefficient to extend non-zero values one level higher.

#### 4.4.14 Rain-015

The above technique for eliminating negative total water values didn't work, as the model blew up after a few hundred iterations. Apparently it didn't like the smoothed eddy mixing coefficient.

Try alternate approach – conservatively borrow total water from the level below to eliminate negative total water. Work from the top down, so that new negative values produced by the borrowing are successively eliminated. As total water increases as one goes down, this iterative process should end well before reaching the surface.

#### 4.4.15 Rain-016

Oops! Introduced a major blunder in rain-014 – the eddy mixing calculation was moved out of the main space loop. However, this removed point-by-point information that was needed in the subsequent calculation and thus completely messed up the eddy mixing fluxes! Fix this problem by returning the eddy mixing calculation to the main loop. Also, it is now clear that the real reason the calculation blew up in rain-14 is not the smoothing of the eddy mixing coefficient, but the problem noticed and fixed here.

This seems to work satisfactorily, in that the calculation keeps the total water non-negative without blowing up or introducing other distortions.

#### 4.4.16 Rain-017

Introduce an alternate way to specify the initial profile – a one-dimensional Candis file containing horizontally averaged fields from a previous run.

#### 4.4.17 Rain-018

Rain-017 doesn't work right. Fix an initialization bug.

#### 4.4.18 Rain-019

The initial profiles of potential temperature and total water mixing ratio can now be modified by the equations

$$\theta_0 \rightarrow \theta_0 + \Delta\theta f(z) \tag{39}$$

and

$$r_t \rightarrow r_t + \Delta r_t f(z) \tag{40}$$

where

$$f(z) = (z/z_s)\exp(1 - z/z_s). \tag{41}$$

This way, the effects of perturbations to radiative-convective equilibrium or other specified state can be investigated.

#### 4.4.19 Rain-020

Add a random perturbation factor to the initial humidity profile of the form

$$r(z) = r_0(z)[1 + HX(z/z_s)\exp(1 - z/z_s)], \tag{42}$$

where $z_s = 3000$ m, $H$ is a constant specified in the input, and $X$ is a random variable uniformly distributed over $[-0.5, 0.5]$.

#### 4.4.20 Rain-021

We forgot to initialize the wind fields from the initial profile when a profile file is specified. Fix this problem.

#### 4.4.21 Rain-022

Give the water vapor mixing ratio and the equivalent potential temperature different relaxation times. In addition, some minor cleanup of the code is done.

#### 4.4.22 Rain-023

Change the potential temperature profile perturbation to an equivalent potential temperature profile perturbation. Also, make both the equivalent potential temperature and mixing ratio perturbations fractional in nature, i. e.,

$$\theta_{e0} \rightarrow \theta_{e0}(1 + \Delta_e f(z)) \tag{43}$$

and

$$r_{t0} \rightarrow r_{t0}(1 + \Delta_r f(z)) \tag{44}$$

where

$$f(z) = (z/z_s)\exp(1 - z/z_s). \tag{45}$$

## 4.5 MPI

### 4.5.1 MPI-002

This implements MPI for parallel processing without changing any of the physics. The domain is split into equal pieces along the x-axis. Specifying one process makes it revert to the non-MPI behavior. The number of processes must be odd, one control process plus an even number of worker processes.

### 4.5.2 MPI-003

The equivalent potential temperature sink term is set proportional to the difference between the actual mean and target *potential temperature* profiles rather than the equivalent potential temperature profiles. This should have the effect of driving the system toward the specified potential temperature profile rather than a specified equivalent potential temperature profile.

Previously, the surface values were not forced by the sink terms. They now are.

### 4.5.3 MPI-004

The moisture external source term is changed to emulate moisture advection while the equivalent potential temperature external source term is rewritten in terms of potential temperature and moisture sources. The former is taken as a relaxation toward a specified potential temperature profile.

### 4.5.4 MPI-005

Change the velocity initialization so that it is initialized by the specified parameters in the initialization file rather than by the profile file.

### 4.5.5 MPI-006

Fix a bug in the calculation of the mean profile of vy.

## 4.6 WTG

### 4.6.1 WTG-001

Add vertical advection to the moisture sink term in order to impliment Sobel and Bretherton's weak temperature gradient approximation for tropical dynamics. This didn't work well and is abandoned because use of the heating profile to get the vertical velocity breaks down at low levels where the vertical velocity is very important to the moisture budget. Thus, WTG breaks down where it is needed the most.

### 4.6.2 MPI-007

We return to a moisture profile forced to a specified profile like the other variables.

### 4.6.3 WTG-002

Fix the problem with wtg-001 by increasing the lower limit on the theta gradient to 3 K/km.

### 4.6.4 WTG-003

Define a weak temperature gradient vertical velocity in the free troposphere and interpolate this linearly to zero in the boundary layer. Add a new parameter which specifies the top of the boundary layer. It is called the "wtgbase".

### 4.6.5 WTG-004

Got the sign wrong on the total water sink term! Fix. Also fix up some notation in the documentation.

### 4.6.6 WTG-005

Show stopper bug in the eddy mixing coefficient calculation! An extra factor of $\Delta z$ was included in the calculation of the square of the Brunt frequency.

Fix.

#### 4.6.7 WTG-006

Put in multiplier in the Brunt frequency part of the criterion for eddy mixing to make the effective cutoff 0.25 in Richardson number for pure vertical shear of the horizontal wind.

#### 4.6.8 WTG-007

The minimum vertical potential temperature gradient for computing the WTG vertical velocity was set too large at 0.001 K $m^{-1}$. Set to 0.0001 K $m^{-1}$.

#### 4.6.9 WTG-008

The relaxation parameters are made to taper linearly to zero at the surface from their value at the boundary layer top.

#### 4.6.10 WTG-009

Change the equivalent potential temperature profile perturbation back into a potential temperature profile perturbation, i. e.,

$$\theta_0 \rightarrow \theta_0(1 + \Delta_\theta f(z)), \tag{46}$$

thus undoing part of what was done in Rain-023. The mixing ratio perturbation thus leaves the potential temperature perturbation untouched rather than the equivalent potential temperature perturbation.

#### 4.6.11 WTG-010

Remove the change of wtg-008, so that the relaxation parameters don't taper in the boundary layer. The wind forcing tapering was affecting the surface fluxes.

#### 4.6.12 WTG-011

Make the weak temperature gradient vertical velocity a result of both a tendency for relaxation to the reference potential temperature profile and a tendency for harmonic oscillation about it. This is consistent with the idea that a stable environment oscillates up and down when it is perturbed, with a damping due to the gradual radiation of energy to the surrounding environment. Major reorganization was done on the documentation.

#### 4.6.13 WTG-012

Make the minimum value of $d\theta/dz$ an external parameter called *minthgrad*. This was set by default before to 0.0001 K $m^{-1}$.

#### 4.6.14 WTG-013

Remove the harmonic oscillation term introduced in WTG-011. Remove moisture advection and replace with a moisture entrainment term. This will hopefully solve the problem of runaway upward motion at upper levels.

#### 4.6.15 WTG-014

Do 1-2-1 smoothing on the divergence of the WTG vertical velocity, as some two-delta-x oscillations show up.

#### 4.6.16 WTG-015

Add a sinusoidal height-dependent mask to the computation of the heating profile. This nullifies the effect of gravity wave adjustment at the surface and the tropopause.

#### 4.6.17 WTG-016

Add an option to make $S_r$ a simple relaxation to a reference profile rather than the response to the weak temperature gradient vertical velocity and entrainment.

#### 4.6.18 WTG-017

Make radbrk (the height at which radiative cooling starts to decrease) an input parameter rather than something set at compile time.

#### 4.6.19 WTG-018

Define an eddy viscosity separate from the eddy mixing coefficient which just includes the deformation rate term and not the stability term. This is needed because of the development of extensive two-delta-$x$ noise. This just weakens the convection without getting rid of the two-delta-$x$ noise. Also tried adding viscous smoothing to density equation, but this just makes the two-delta-$x$ noise problem worse.

#### 4.6.20 WTG-020

The Lax-Wendroff scheme as it stands doesn't allow convectional filtering, since the original fields are not extended in the $-x$ and $-y$ directions. This enlarges the $L$ grid to encompass that possibility. Otherwise the code should be have as in wtg-017, i. e., with both separate eddy viscosity and smoothing of the density equation removed.

#### 4.6.21 WTG-021

Do lots of internal rearrangement. All of the grid expansion is now done on the $I \rightarrow L$ grid transformation. The $J$ grid is computed completely from the $L$ grid, with no extensions needed. Also, the initial fluxes are computed directly on the $L$ grid, so that they don't have to be extended. This should save considerable interprocess communication time. The results still should be identical to those of wtg-017. They are, and the code works with bounds checking turned on and in single as well as multiple process mode.

#### 4.6.22 WTG-022

Add a horizontal filter to all fields at full timestep stage. This seems to take out the two-delta-$x$ wiggles in simple tests.

#### 4.6.23 WTG-023

Make the reference profiles for the WTG calculation unchanged by the perturbation variables, so that the test region can be started from a non-reference state.

## 4.7 Snow

### 4.7.1 Snow-001

The particle fall velocity and the cloudwater to precipitation conversion rate now take on different values above and below the freezing level. This allows a crude simulation of the effects of snow production, which falls more slowly than raindrops. However, the calculation is oversimplified still, because, rain and snow cannot coexist above the freezing level. Also, the conversion of cloud water to snow is not a function of temperature beyond the fact that it can only occur when the temperature is less than freezing.

### 4.7.2 Snow002

The moisture source term is changed so that it is a combination of a fractional tendency (rrelax) to represent horizontal moisture advection and the previously existing entrainment and vertical advection terms.

### 4.7.3 Snow003

Positive rrelax actually causes drying in snow-002, due to the peculiar way the source/sink terms are done. Change the sign of rrelax.

## 4.8 Meso

### 4.8.1 Meso001

Allow for mesoscale variations in sea surface temperature in the form of a sinusoid in the $x$ direction.

## 4.9 Rad

### 4.9.1 Rad001

Add the toy radiation package of Raymond (2001), which is used to calculate a mean radiative cooling profile for the whole domain, applied at the next time step. Thus, cloud-radiation interactions exist, but the effect of cloud-induced variations on the radiation field is averaged out. The radiation package is not called every time step.

#### 4.9.2 Rad002

Make fixed radiative cooling as done before rad001 an option. Also change from MPICH MPI implimentation to LAM.

#### 4.9.3 Rad003

Eliminate the option of altering the initial profiles from the reference profiles. Instead, provide the option of making the reference profiles functions of time. Change the form of the profile alteration to $z^2 \exp[2(1-z)]$.

#### 4.9.4 Rad003-2

Add pure 2-D (x-z) version of model from Sharon Sessions. These are incorporated as separate files main2.c, timestep2.c, indexing2.c, and sample2.in.

#### 4.9.5 Rad004, Rad004-2

Same code as in rad003, simply add a cumulative rainfall variable. Modify both 2-D and 3-D versions.

#### 4.9.6 Rad005, Rad005-2

Same code as in rad003, add a cumulative surface evaporation rate in both 2-D and 3-D versions.

#### 4.9.7 Rad006, Rad006-2

Same code as in rad003, forgot to multiply by dt to get cumulative evaporation rate – fixed.

#### 4.9.8 Rad007

Add simple 1-2-1 smoothing to the weak temperature gradient vertical velocity when it is calculated. This variable is otherwise subject to the production of 2*dz oscillations. Abandon the 2-D versions of the code for now.

## 4.10 Dyn

### 4.10.1 Dyn001

Add the possibility of a time-dependent reference profile as the main forcing mechanism. Abandon moisture advection (rrelax – this was undocumented anyway!) as well as time-dependent parameters for changing the reference profile. Eliminate the difference between reference and initial profiles and end all reference profile variables with “0” rather than “ref”.

### 4.10.2 Dyn002

Fix a bug in the moisture source term introduced when the rrelax parameter was eliminated.

### 4.10.3 Dyn003

The density and equivalent potential temperature are not needed in the initialization file, since they are recomputed in hydrostatic2 immediately after getprofile is called. Therefore, removed these from the required init file contents.

### 4.10.4 Dyn004

Add the reference saturated equivalent potential temperature as an output field. Otherwise, no change.

Reorganize release notes section of documentation to make subsections and subsubsections.

### 4.10.5 Dyn005

Calculate $S_e$ in a manner identical to $S_r$ rather than computing it indirectly using $S_\theta$ and $S_r$. Also, fix the entrainment part of this calculation so that the uppermost and lowermost levels are included.

### 4.10.6 Dyn006

Calculate and output cumulative environmental sinks ($S_\theta$, $S_r$, $S_e$, $\boldsymbol{F}$) and all surface fluxes. These constitute time integrals of the quantities of interest.

These cumulative values are needed because the quantities are highly time-variable (like rainfall) and reliable averages cannot be determined from time sampling.

#### 4.10.7 Dyn007

Fix some minor bugs introduced by the last change.

#### 4.10.8 Dyn008

Forgot to compute the cumulative radiation heating. Fix.

#### 4.10.9 Dyn009

Make initial temperature profile calculation second order accurate. Also, set the surface temperature to the initial sea surface temperature instead of to 300 K.

#### 4.10.10 Dyn010

The initial potential temperature profile calculation had some errors which reduced it to first order accuracy. Fix.

#### 4.10.11 Dyn011

Add some declarations needed to make the code compile with gcc-4.xxx. No substantive changes.

#### 4.10.12 Dyn012

Add new input variables to initialize both potential temperature and mixing ratio fields locally and randomly.

#### 4.10.13 Dyn013

Added checkpointing capability in which a checkpoint file is written every time a normal output file is written. If the model terminates abnormally, it can be restarted from the point that the last checkpoint file was written. The checkpoint file is named "checkfile.basename" where "basname" is the basename given in the input file.

#### 4.10.14 Dyn014

Fix bug introduced by the checkpointing. Also, lag the checkread() call by process rank to avoid hitting the master server all at once.

#### 4.10.15 Dyn015

Conservative fourth-order smoothing is added in the vertical. This is needed to suppress $2\Delta x$ oscillations in rainwater which tend to develop in quiescent conditions. Values of $\lambda_v > 0.001$ have a bad effect on mean profiles but still don't solve the problem. This smoothing will probably be taken out in the next revision.

#### 4.10.16 Dyn016

The fourth order smoothing doesn't solve the problem. Try an additional conservative viscous smoothing in the vertical on the rainwater only, which is the main variable with a $2\Delta z$ oscillation problem. Also, backup the checkpoint file before writing a new one in case the program is interrupted during a checkpoint file write.

#### 4.10.17 Dyn017

Add a factor "rhswitch" which multiplies the initial 3-D relative humidity by a constant factor with this value, which will generally be either zero or one. This is so experiments can be made to see if multiple equilibria exist with different relative humidities. This switch is brought out as an initial parameter.

#### 4.10.18 Dyn018

Make model dump core on floating point exceptions. Actually, for some reason core doesn't actually dump with lam-mpi. However, at least execution terminates.

#### 4.10.19 Dyn019

Fixed a problem in the calculation of theta from theta-e and pi in extreme cases. This showed up when floating point exceptions were turned on.

#### 4.10.20 Dyn020

Fixed a problem of floating point exceptions occurring in masked space during intialization. This only occurs when multiple processes are active, but doesn't affect previous results since it occurs in regions where calculations aren't made.

#### 4.10.21 Dyn021

Make the model run when an even number of processors is assigned – was one processor if necessary.

#### 4.10.22 Dyn022

The above fix didn't work. Had to do some additional MPI magic to be able to use an even number of processors.

#### 4.10.23 Dyn023

No change to the code – fix some documentation errors.

#### 4.10.24 Dyn024

Add the option of a horizontally uniform imposed heating with a sinusoidal vertical structure over a specified vertical range for a specified time interval. The amount of heating is specified as a maximum fractional value of the accumulated temperature change over the heating interval. The vertical structure of the heating is fundamental mode, first harmonic, second harmonic, etc., depending on the integer value of the mode parameter. The heating enters as an additional contribution to the radiative heating term beyond the effects of radiation.

#### 4.10.25 Dyn025

Fix a bug in the imposed heating code.

#### 4.10.26 Dyn026

Bring the surface exchange and drag coefficient out as an input variable rather than being set internally.

#### 4.10.27 Dyn027

Fix a bug in the checkfile generation discovered by Mike Herman – a needed variable was omitted.

#### 4.10.28 Dyn028

Fix another similar bug discovered by Mike Herman.

#### 4.10.29 Dyn029

Add a specified additive moisture relaxation in addition to the temperature and dynamic relaxations, following Adam Sobel. This actually relaxes both the mean moisture and the equivalent potential temperature of the computational domain to the reference profile. Yes, the new variable mrelax was included in the checkfile generation!

#### 4.10.30 Dyn030

Add an input parameter which is a seed for the random number generator. This way ensembles of identical runs (aside from the initial randomization) can be made.

#### 4.10.31 Dyn031

Add an optional command line argument which gives a new value of the number of time steps on restart from a checkfile. This allows runs to be extended without redoing the entire run.

### 4.11 Dyn3d

Enhance the parallelization to split up the domain into a checkerboard (2d in x and y) in stead of just slices in x.

#### 4.11.1 Dyn3d001

Initial working version.

#### 4.11.2 Dyn3d002

Fixed a problem in the direction of communication between adjacent checkerboards. (This didn't exist in the earlier non-checkerboard version.)

#### 4.11.3 Dyn3d003

Fix a small documentation error in equation (2). (Density left out of $\mu$ term.)

#### 4.11.4 Dyn3d004

Add the option of using the full 3-D radiative equivalent potential source directly rather than averaging horizontally.

#### 4.11.5 Dyn3d005

Fix the treatment of cloud water in radiation, following fix in the sigma model.

#### 4.11.6 Dyn3d006

Update the version of the radiation code to that in sigma-104 (r101). Add local install options to the Makefile.

#### 4.11.7 Dyn3d007

The column/average radiation flag ptrad was implemented in the wrong sense, i.e., non-zero value resulted in area mean radiation. Fixed. Also renamed this variable 'columnrad', as 'ptrad' is rather non-intuitive.

#### 4.11.8 Dyn3d008

Fixed radiative heating at the break level (typically near 12 km) would go to zero under certain conditions. Fixed the problem.

#### 4.11.9 Dyn3d009

Added compile-time switch SYS to allow easy compilation on both Gryphon and UCAR machines. On Gryphon and other standard linux systems define SYS=0. On UCAR, define SYS=1. This comments out the core-dump enable routine, which is not recognized by the compiler on UCAR machines. Also, RAND MAX is redefined, since the one defined on UCAR systems is not the actual maximum random number returned by RANDOM().

#### 4.11.10 Dyn3d010

Add an option, madvect, to turn off vertical moisture advection and entrainment in the WTG treatment. madvect = 1. means it is turned on as was the previous default, madvect = 0. turns it off. It is expected that mrelax would be used in the place of advection.

#### 4.11.11 Dyn3d011

Create a separate sthwtg for the WTG vertical velocity calculation. The field sth now comes from from sthe and srt. (This calculation is approximate.) Switch 1-2-1 smoothing from vzwtg to sthwtg. Either place smooths out 2*dx oscillations in more or less the same way, but putting it in sthwtg is cleaner. Sthwtg and sth are in close agreement everywhere except the boundary layer, which is expected, as vzwtg doesn't come directly from sthwtg in this layer. Do some minor rearrangement of code to make it more rational.

#### 4.11.12 Dyn3d012

Do some code cleanup, isolating most MPI stuff in main routine in subroutines. Also, eliminate forced heating option.

### 4.12 Spec

#### 4.12.1 Spec001

Implement spectral WTG. Eliminate trelax, the original WTG relaxation time constant; thprime and thetagrad, previously scalars, become vectors. All WTG forcing is turned off by setting mrelax = madvect = 0. A new external parameter, spectral, selects the WTG scheme to use. Setting spectral $> 0$ activates the spectral WTG scheme. The original WTG scheme can be recovered by setting spectral = 0 and choosing a horizontal scale corresponding to the desired value of trelax, using $L = N_0 h/(\pi\lambda_{st})$, where $L$ and $h$ are the external parameters hscale and tpause, respectively, $N_0 = 1.0e-2$ s$^{-1}$is a constant Brunt Vaisala frequency, and $\lambda_{st}$ is the (now defunct) WTG relaxation parameter, trelax. An external function, simpson(), is added to main.c to compute Fourier coefficients for spectral WTG scheme. The vertically averaged buoyancy frequency for spectral WTG varies with each timestep and is computed using

$$N(t) = \frac{g}{nz} \sum_{i}^{nz} \left(d\overline{\theta}/dz\right)_i / \overline{\theta}_i,$$

where $nz$ is the number of grid cells below the tropopause.

The spectral WTG scheme is given by

$$w_{wtg-spectral} = \frac{Nh}{L\pi} \sum_{j}^{nj} \Theta_j \sin(m_j z)/j, \tag{47}$$

where $m_j = j\pi/h$. The Fourier sine series coefficients are given by

$$\Theta_j = \frac{2}{h} \int_0^h M(z) \frac{\left(\overline{\theta} - \theta_0\right)}{d\overline{\theta}/dz} \sin(m_j z) dz, \tag{48}$$

where $M(z) = 1$.

The original WTG scheme is approximately the Fourier sine series expansion of (18), expanded to $nj$ terms, with $\lambda_{st} = N_0 h/(\pi L)$. The original WTG scheme is therefore recovered from (47) and (48) when

$$M(z) = \begin{cases} \sin(\pi z/h) & z < h \\ 0 & z \geq h \end{cases},$$

the $j$ in the denominator of (47) is set to 1, 1-2-1 smoothing is applied to $M(z)\left(\overline{\theta} - \theta_0\right)$ in (48), and $w_{wtg}$ is defined by a linear interpolation to zero between wtgbase and the surface. Note that a slight difference remains between the WTG scheme used in Dyn3d012 and Spec001 due to the slight inaccuracy of the Gibb's phenomenon: since $\overline{\theta} - \theta_0$ can be nonzero at the surface and at the tropopause, the Fourier series must recreate a step function, which requires wavenumbers beyond those possible in the bandlimited series used here.

A bug in the calculation of finetime was fixed. The old finetime was erroneously defined as up to 1 output in the future.

### 4.12.2 Spec002

The spectral WTG gave slight downwelling in the boundary layer for a strong SST anomaly. This seems nonphysical. Now, both the original and spectral WTG vertical velocities have a linear decay to zero in the boundary layer.

### 4.12.3 Spec003

Revert to no linear reduction in PBL, in order to make spectral WTG as simple as possible. The external parameter 'spectral' now sets the number of Fourier modes to use in the expansion. A few bugs are fixed related to the checkfile. Extensive testing suggests that $nj = 8$ is the optimal number of modes. Using $nj > 17$ gives downwelling in the boundary layer above a strong SST warm anomaly. Slight documentation changes.

### 4.12.4 Spec004

On reconsideration of the behavior of the spectral WTG in the boundary layer, we've decided to use all of the modes ($nj = 60$). This gives the best accuracy since we use all possible modes up to the Nyquist wavenumber. Also, since we don't really know what the velocity should do in the boundary layer, there is no reason to truncate the modes at small $j$ in order to impose smoothing. The minimum vertical gradient of potential temperature, minthgrad, is now set to 0.0003 in order to make the velocity structure in the upper troposphere more realistic. A new variable vzwtg2 is added, which calculates the original WTG velocity w/o the Fourier expansion. This is to check the accuracy of the expansion, which can be affected by the Gibb's phenomenon whenever the vertical gradient of the heating profile is large. Lastly, a bug introduced in Dyn3d012 was fixed: the smoothing of sthwtg had caused nonzero WTG vertical velocity above the tropopause.

### 4.12.5 Spec005

Added newrain.sh, a script that handily outputs the rain production in the convective model. Also, refixed the bug in finetime so that finetime = time at the last iteration of the fine time-stepping loop in main.c (see notes for Spec001).

### 4.12.6 Spec006

Fix a bug having to do with time-dependent reference profiles. There was an off-by-one error in the computation of delt in getprofile.c that caused time shifts in the reference profile. This is in addition to the finetime bug mentioned above. Add archive folder with instructions on recovering earlier model versions from RCS.

### 4.12.7 Spec007

Fix a bug introduced in spec004, where the adjustment mask was imposed twice. Edit documentation. Set spectral = 15 in sample.in. This is the number of modes that gives seemingly useful values of vzwtg in the boundary layer. The WTG forcing in the free troposphere is ostensibly a weak function of the spectral parameter from spectral = 8 -> spectral = 60. For spectral > 18, a 4 K SST anomaly gives downwelling in the PBL. Setting spectral = 15 restricts the wavelength of resolved vertical features to approximatey the depth of PBL itself.

### 4.12.8 Spec008

Pull the repeated evaluation of a sine function in the spectral WTG model out of $iz$-$j$ loops and into a lookup table.

## 4.13 Entropy

This is a branch of spec008 to convert the model from use of the equivalent potential temperature to specific moist entropy. Legacy fields of equivalent potential temperature and the radiative source of equivalent potential temperature are written out at this point.

### 4.13.1 Entropy001

Initial version.

### 4.13.2 Entropy002

Initial (hopefully) useful version.

#### 4.13.3 Entropy003

Rationalize the flags controlling WTG. The flag “wtg” turns of WTG. The flag “sobel” turns on vertical advection and horizontal relaxation of moisture and entropy a la Adam Sobel. Make “spectral” an integer variable and put an upper limit on its value. Otherwise, vertical advection and entrainment/detrainment a la Raymond et al. occurs. Put all the WTG stuff in a subroutine. Random code cleanup.

#### 4.13.4 Entropy004

Define TREF = 300 K (was FREEZING) for consistency with trunk. (This is likely to change later.) Make toyrad.c thermodynamic constants independent of thermo.h so that future changes won’t affect toyrad.

#### 4.13.5 Entropy005

Fix bug where sine0 wasn’t written to checkfile. Minor code cleanup.

#### 4.13.6 Entropy006

Other bugs were showing up when a restart with a checkfile was done. Made a thorough revamp of the creation and reading of a checkfile, putting all variables and parameters in the same order as in the main code. Several parameters and fields were missing and/or out of order.

#### 4.13.7 Entropy007

One more sneaky bug remained in the checkfile routines.

#### 4.13.8 Entropy008

Radically simplify the treatment of radiation. There is either a fixed radiative heating profile imposed by the initial input file or the profile file, or fully interactive radiation (no averaging) is done. Some additional code simplification was done on the WTG code.

#### 4.13.9 Entropy009

Readcheck bugfix. Put all space allocation for main() into external function.

#### 4.13.10 Entropy010

The Sobel moisture relaxation was incorrect in that there was no relaxation rate. A relaxation rate “sobel” was added to the input file and the moisture lateral entrainment was given independent control via the flag “entrain” (either 0 or 1; off or on). The quantity “sobel” should scale like the inverse of the expected time for the ambient flow to cross the convective region (s^-1).

#### 4.13.11 Entropy011

Add failsafe exit() to thetasat() in case of non-convergence.

#### 4.13.12 Entropy012

Add failsafe exit() to thetasat() in case of non-convergence.

#### 4.13.13 Entropy013

Remove option to use fewer Fourier modes. An incomplete set of modes leads to (maxnj/nj)*dz wiggles. Fixed radiation is now consistent with the theta-e model, such that it employs a fixed equivalent potential temperature sink, rather than a fixed moist entropy sink. Thus, stherad is now the assumed initialization profile for radiation.

#### 4.13.14 Entropy014

The model goes into serious oscillations if mumax is set to zero. This comes from non-zero vertical flux divergences of density, entropy, and possibly rt at the very top of the domain. There should be no flux divergences at this elevation. Therefore, force them to be zero there.

#### 4.13.15 Entropy015

To reduce wiggles in vz near the onset of damping at the tropopause, the damping function is smoothed. Also, the layer of nonzero damping is held adjacent to the model top but the layer thickness is held fixed (SPGDEPTH=5 km), so that a higher model top gives a higher damping onset.

#### 4.13.16 Entropy016

Add optional command line argument [new_mt] to change the output time resolution on restart from checkfile.

#### 4.13.17 Entropy017

Compute vertical gradient of theta using centered difference method and remove smoothing of sthwtg. Upstream thetagrad computation was causing 2dz wiggles in vzwtg. In multiple equilibrium experiments, thetasat() does not converge within 20 iterations when humidity is set to zero. Change iteration limit to 200 on calls to thetasat().

#### 4.13.18 Entropy018

Solve the problem of stratospheric drift by (1) mirroring the vertical fluxes at the top and bottom of the domain such that the interpolated fluxes at the actual surfaces are zero, and (2) removing the density adjustment term in the density governing equation – this term caused significant mass non-conservation. The potential temperature drift at the domain top is down to less than 1 K over 50 days with fixed radiation.

#### 4.13.19 Entropy019

The changes in flux boundary conditions in entropy018 necessitate a change in how surface fluxes are deposited in the lowest layer. They must now be deposited in a layer of thickness dz/2 rather than dz, since only half of the lowest grid cell is above the surface. Also, revert the stratospheric damping to the original, linear form starting at the specified tropopause, as the cosine form is causing dissipation to concentrate too much in a thin layer.

#### 4.13.20 Entropy020

The interactive radiation grid was shifted dz/2 down from the convective model grid. Fixed. This necessitated extrapolation of radiative heating to the surface and the domain top.

#### 4.13.21 Entropy021

Introduce a new diagnostic variable, srtrain, which is the net source term of the total cloud water associated with conversion of water substance to and from rain. Made tests that indicate the entropy and water budgets are reasonably well (though not perfectly) conserved. I'm not sure where the water and entropy leaks are – they may be due to truncation error.

#### 4.13.22 Entropy022

Rename variables sent –> sentwtg, srt –> srtwtg, and eliminate the variable sth as there is already an sthwtg. Rename the cumulative variables consistently as the variable name with "cum" appended. Add two new cumulative variables for sentrad and srtrain.

#### 4.13.23 Entropy023

Zero the initial radiation forcing, as this only acts for the first time step. (There was a bug in this code anyway.) Include the mean sea surface temperature as a variable returned by getprofile. Update the documentation in several areas. Include some important shell scripts for analyzing the output.

#### 4.13.24 Entropy024

Incorporate a new version of the radiation code from fermi020 that is expressed in terms of entropy and which returns results on grid edges rather than grid centers (actually, centered on the I grid). The fixed radiative cooling profile is specified in terms of a temperature tendency rather than an entropy or theta-e tendency.

#### 4.13.25 Entropy025

Change the code so that the radiation and time step routines are run lt times between calls to horizontal averaging and WTG calculations. The parameter lt has been added to the input file. The old parameter mt is equivalent to mt*lt in this version of the code; i.e., mt*lt is now the total number of loops of the radiation and time step routines per display interval in the new code. The quantity dt is still the time step on the finest time scale.

#### 4.13.26 Entropy026

Change the meaning of vzwtg2, so that it now represents $\overline{\theta}'(z)/(d\overline{\theta}/dz)$, scaled to the units of vzwtg. No taper is applied to this variable at all. In certain WTG simulations, vzwtg has 2 delta-z noise near the surface. It is generally associated with low static stability there. Since the WTG vertical velocity is constrained to be zero at the surface, the discontinuity takes the form of a localized peak. One can compare the spectral WTG vertical velocity with the raw scaled temperature anomaly in vzwtg2 to see if numerical oddities are produced by the Fourier expansion.

#### 4.13.27 Entropy027

Rewrote and simplified the Fourier transforms for spectral WTG. In particular, wrote as a finite Fourier transform, so that the forward transform is formally a sum rather than an integral. The results differ little from those of entropy024/025/026. The differences are sufficiently small that they are likely to be insignificant, and model results of these generations probably can be intermixed.

#### 4.13.28 Entropy028

Forgot to multiply the time step in the cumulative wtg source terms by lt. Fixed.

#### 4.13.29 Entropy029

Add new diagnostic output variables consisting of the vertical convective fluxes of momentum, entropy, total cloud water, and rain water. This does not change the calculation in any way.

#### 4.13.30 Entropy030

Change "radcool" in sample.in to an absolute temperature sink. Minor changes and additions to scripts. Stop multiplying water-related quantities by the number of seconds in a day. No changes to actual model calculations.

# 5 References

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

**Raymond,** D. J., and X. Zeng, 2005: Modelling tropical atmospheric convection in the context of the weak temperature gradient approximation. *Quart. J. Roy. Meteor. Soc.*, **131**, 1301-1320.

**Sobel,** A. H., G. Bellon, and J. Bacmeister, 2007: Multiple equilibria in a single-column model of the tropical atmosphere, Geophys. Res. Lett., 34, L22804, doi:10.1029/2007GL031320.

**Sobel,** A. H., and C. S. Bretherton, 2000: Modeling tropical precipitation in a single column. *J. Climate*, **13**, 4378-4392.