Description of Subroutine Convect
Version 3.00
1 October 1996
Kerry A. Emanuel

A. General use

The subroutine is designed to be used in time-marching models of mesoscale to global-scale dimensions. It is meant to represent the effects of *all* moist convection, including shallow, non-precipitating cumulus. It also contains a dry adiabatic adjustment scheme.

Since the method of calculating the convective fluxes involves a relaxation toward quasi-equilibrium, *subroutine CONVECT must be run for at least several time steps to give meaningful results.* At the first time step, the tendencies and convective precipitation will be zero. If the initial sounding is unstable, these will rapidly increase over successive time steps, depending on the values of the constants ALPHA and DAMP. Thus the user interested in convective fluxes and precipitation associated with a single initial sounding (i.e., without large-scale forcing) should still march CONVECT forward enough time steps that the fluxes have returned back to zero; the net tendencies and precipitation integrated over this time interval are then the desired results. But it should be cautioned that these quantities will not necessarily be independent of other model parameters such as the time step. CONVECT is very much built on the philosophy that convection, to the extent it can be represented in terms of large-scale variables, is never very far away from statistical equilibrium with the large-scale flow.

To achieve a smooth evolution of the convective forcing, CONVECT should be called at least every 20 minutes during the time integration. CONVECT will work at longer time intervals, but the convective tendencies may become noisy.

The thermodynamics used in CONVECT account for the dependence of heat capacities and gas constants on water substance and the temperature dependence of the latent heat of vaporization. The linear increase of L_v with decreasing temperature is carried right through to all temperatures below freezing. While this is artificial, it partially makes up for the lack of latent heat of fusion. The user should alter the values of the thermodynamic constants at the beginning of CONVECT and also at the beginning of TLIFT to be consistent with those used by the calling program.

CONVECT checks all levels to determine whether convection can originate from that level, starting from the surface upwards. Convection is assumed to originate only from the *first* level at which the air is conditionally unstable to upward displacement. This means that CONVECT can represent convection originating from an elevated level, but not when and where boundary layer convection is occuring.

The main item one should pay attention to in implementing CONVECT (based on experience with several global models) is its interaction with the boundary layer formulation of the model. In particular, *it is essential that the subcloud layer in convecting regions have lapse rates that are NOT sub-adiabatic; that is, the potential temperature, as defined within CONVECT, must not be increasing with height in the boundary layer.* If potential temperature increases with height in the subcloud layer, CONVECT may not produce convection, regardless of the degree of conditional instability.

B. Preparation

IT IS OF CRITICAL IMPORTANCE THAT TWO WORK ARRAYS OF DIMENSION (I,J) BE DEFINED IN THE CALLING PROGRAM, WHERE I AND J ARE HORIZONTAL INDICES COVERING THE ENTIRE MODEL DOMAIN. (THE J INDEX IS UNNECESSARY IN A 2-D MODEL.) THESE ARRAYS SHOULD BE INITIALIZED TO ZERO BUT NOT SUBSEQUENTLY ALTERED BY THE CALLING PROGRAM. AT EACH CALL TO CONVECT, THE FLOATING POINT SCALAR CBMF AND THE INTEGER SCALAR NCOUNT THAT ARE USED IN THE CALL TO CONVECT MUST BE DEFINED SUCH THAT

CBMF= ARRAY1(I,J),

and

NCOUNT=ARRAY2(I,J).

(These are memory arrays and are discussed in Section D1 below.)

Next, the parameter NA at the beginning of the subroutine must be set to a value larger than (by at least 1) the value of ND, which is the dimension of the one-dimensional thermodynamic arrays in the calling program.

The thermodynamic constants, described in detail in Section 2 below, should be made consistent with those used by the calling program, both at the beginning of CONVECT and at the beginning of TLIFT.

Finally, the value of ALPHA may be altered to give smoother integrations, but care should be taken that it is not too small (evidenced by very unstable soundings) or too big (evidenced by highly intermittent convection). DAMP can also be altered but should lie in the range 0.01-0.2.

C. A note about energy conservation

The conservation of energy by the scheme is slightly complicated by the fact that mass is lost from the atmosphere when precipitation reaches the surface. Since most models do not allow a net loss or gain of atmospheric mass, we compensate for this mass loss by adding dry air at the surface at a rate equal to the mass loss owing to precipitation.

Let h be the specific static energy of the air in the system:

$$h = [c_{pd}(1-Q) + c_\ell Q]T + L_v q + gz, \tag{1}$$

where c_{pd} is the heat capacity of dry air at constant pressure, c_ℓ is the heat capacity of liquid water, L_v is the (temperature-dependent) latent heat of vaporization, gz is the geopotential, q is the specific humidity, Q is the total specific water content, and T is temperature.

If it is assumed that all kinetic energy generated during an adjustment is convected back into nonkinetic energy, then following individual samples of air

$$\frac{dh}{dt} = (c_\ell - c_{pd})T\frac{dQ}{dt}. \tag{2}$$

The quantity on the right represents the heat loss owing to the loss of mass by precipitation, minus the heat gained when that mass is replaced by dry air. Since mass is lost by CONVECT by precipitation falling out at the surface, and regained when dry air is assumed to enter from the surface, this becomes

$$\frac{dh}{dt} = (c_\ell - c_{pd})T^1\frac{1}{p_0 g}\int_0^{p_0}\frac{\partial q}{\partial t}dp, \tag{3}$$

where T^1 is the temperature at the lowest level and p_0 is the surface pressure. The integral represents the net water loss (precipitation). Expanding the derivative of h,

$$\frac{dh}{dt} = \frac{\partial h}{\partial t} + \frac{\partial}{\partial x}(uh) + \frac{\partial}{\partial y}(uh) + \frac{\partial}{\partial p}(\omega h), \tag{4}$$

and noting that the scheme does not generate horizontal velocities and that $\omega = 0$ on $p = p_0$, $p = 0$, then if (4) is substituted into (3) and the result integrated over the mass of the atmosphere, we have

$$\int_0^{p_0}\frac{\partial h}{\partial t}dp = (c_\ell - c_{pd})T^1\int_0^{p_0}\frac{\partial q}{\partial t}dp. \tag{5}$$

Substituting (1) for h in (5) and accounting for the temperature dependence of L_v, the above becomes

$$\int_0^{p_0} [c_{pd}(1-q) + c_{pv}q] \frac{\partial T}{\partial t} dp + \int_0^{p_0} [L_v + (c_\ell - c_{pd})(T - T^1)] \frac{\partial q}{\partial t} dp = 0. \qquad (6)$$

The scheme almost precisely satisfies the finite difference version of (6). The user should note in particular that the heating is given by

$$[c_{pd}(1-q) + c_{pv}q] \frac{\partial T}{\partial t};$$

This should be consistent with, for example, the conversion of radiative heating to temperature change in the calling program. (If one does not want to consider the dependence of heat capacity on water substance, then set $c_{pv} = c_{pd}$ in CONVECT.) Moreover, if the surface energy balance of the model is important, then care should be taken to assure that the latent heat coefficient at the surface in the calling program has the same value as L_v (at the surface) in CONVECT. If the temperature dependence of L_v is neglected in the calling program, then setting $c_\ell = c_{pv}$ will eliminate it in CONVECT as well. For models with simple thermodynamics in which the heat capacity of air and the latent heat of vaporization are both constant, setting $c_\ell = c_{pv} = c_{pd}$ eliminates the variability of L_v and heat capacity in CONVECT, reducing (6) to the conventional form

$$\int_0^{p_0} \left[c_p \frac{\partial T}{\partial t} + L_v \frac{\partial q}{\partial t} \right] dp = 0. \qquad (7)$$

Energy conservation in the scheme is not exact, owing to the use of upstream differencing in the vertical advection terms, but is usually within 1% of the latent heating contribution.

D. Description of the code

1. *The call to convect*

The basic design of the scheme allows the user to input one-dimensional arrays of temperature, specific humidity, saturation specific humidity and pressure, as well as three scalars representing surface conditions, and receive as output the convective precipitation and time tendencies of temperature and specific humidity as well as some quantities necessary for having the convective downdrafts interact with the surface heat and moisture fluxes. The input arrays of temperature, specific humidity and saturation specific humidity are subject to dry adiabatic adjustment by the scheme, so the returned arrays may be altered accordingly. If the dry adiabatic adjustment itself results in condensation, the condensed water is assumed to precipitate and the latent heat is added to the layer.

The first line of parameters in the subroutine call contains one-dimensional arrays T, Q, QS, P, and PH, corresponding to temperature (Kelvin), specific humidity (gm gm^{-1}), saturation specific humidity (gm gm^{-1}), pressure (mb), and pressure at the half-levels (mb). *The arrays must be defined so that their first element is the value corresponding to the first grid point above the ground level.* The temperature, specific humidity, and saturation specific humidity must be defined at the same levels as pressure (P). The array PH contains pressures at the model half-levels; the first element of PH should correspond to a half level *below* (i.e., at a higher pressure than) the first grid level at which T, Q, QS, and P are defined. As mentioned above, the arrays T, Q and QS are returned unaltered, unless dry-adiabatic adjustment occurs. The arrays P and PH are always returned unaltered. All the arrays except PH are dimensioned ND, while PH is dimensioned ND+1.

The scalar quatitites TG, QG and PG are, respectively, the temperature (K), specific humidity (gm/gm) and pressure (mb) of the surface itself. If the surface is ocean, TG is the sea surface temperature and QG is the saturation specific humidity corresponding to the sea surface temperature. If the surface is land, then TG is the ground temperature while QG should be considered to be the specific humidity of air in immediate contact with the surface.

The next set of parameters in the call statement are single-integer values and one single floating point value, DELT. The first, ND, is the dimension of the arrays T, Q, QS,P,FT and FQ. Note that PH is dimensioned ND+1. The integer NL is the number of levels one wishes checked for convection. It must be less than or equal to ND -1. *NL must be at least one more than the maximum number of model levels, starting at the surface, that will be affected by moist convection during the integration.* For example, if the model levels are numbered upward from the surface and level 8 is the maximum level you expect convection ever to penetrate to, NL should be 9. In general, NL will be a model level located in the lower stratosphere.

It is important that the calling program predicts or specifies temperature, pressure, specific humidity, and saturation specific humidity up to at least one level higher than the highest level moist convection will penetrate to. Setting NL equal to NDP1-2 is safe, but some computational overhead will occur in checking levels that prove to be unaffected by convection.

The next parameter on the second line, DELT, is the time step in seconds between calls to convect. It is used to check that a CFL condition is met. If it is not, the value of IFLAG (discussed below) is altered, and the convective mass fluxes are reduced to insure that the CFL condition is satisfied.

IFLAG is an integer that contains information about the performance of the scheme. Standard values are 0 and 1. If the returned value is zero, no moist convection has occurred and if it is 1, moist convection has happened. A value of 0 will occur when air lifted from the lowest level is negatively buoyant at the first level above cloud base, when the potential temperature just below cloud base is greater than that at the surface, when $T(1) < 250$ K or when $Q(1) < 0$. A returned value of 2 indicates that the moist convection was bypassed because the lifted condensation pressure of boundary layer air was less than 200 mb or greater than 2000 mb (the latter being an indication that something is wrong with the input variables). Likewise, a returned value of 3 indicates that the level of cloud base is at or above the highest level convection is assumed to penetrate; that is, ICB $\geq$ NL-1. A value of IFLAG of 2 or 3 returns to the calling program without doing moist convection. Finally, when IFLAG = 4, the convective mass fluxes are large enough that a CFL condition is violated. This is not fatal, in the sense that the subroutine calculation continues. One may expect this to occur at initialization, if the initial soundings appear to be very unstable to CONVECT, or in cases where the mass flux becomes excessive due to the release of large, previously stored quantities of convective available potential energy (e.g., Oklahoma in May). If it happens often, try reducing the value of ALPHA. If this does not work, then either the vertical grid interval or the time step should be reduced.

The final set of variables in the call statement include the floating point one-dimensional output arrays FT and FQ, the output floating point scalars PRECIP, WD, TPRIME and QPRIME, and the storage scalars CBMF (floating point) and NCOUNT (integer).

The output array FT contains the moist-convectively produced tendency of absolute temperature in degrees Kelvin per second. Likewise, the array FQ contains the moist-convectively produced tendency of specific humidity (*not* mixing ratio), in gm gm^{-1} s^{-1}. The scalar PRECIP returns the convectively produced precipitation, in mm d^{-1}. If convection does not occur, all of these values are set to zero.

The scalars WD, TPRIME and QPRIME are for use in coupling this convection scheme to the calling program's formulation of surface fluxes. They are, respectively, a convective downdraft outflow velocity scale, and the perturbations from the grid-area mean values of temperature and specific humidity. An example of how they might be used is as follows:

Suppose the surface fluxes, F, in the callling program are given by simple aerodynamic flux formulae of the form

$$F = C|\mathbf{V}|(x_s - x), \tag{1.1}$$

where C is a dimensionless constant, $|\mathbf{V}|$ is the near-surface wind speed, x_s is the surface value of the quantity in question, and x is the value of that quantity in the atmosphere near the surface. Now it can be shown that the grid-area average value of F can be approximated by

$$F \simeq C\left(\overline{|\mathbf{V}|}(x_s - \overline{x}) - \overline{v'x'}\right), \tag{1.2}$$

where

$$\overline{|\mathbf{V}|} = \sqrt{u^2 + v^2 + w*^2 + w_d^2}, \tag{1.3}$$

and

$$v' = \left(\sqrt{u^2 + v^2 + w*^2 + w_d^2} - \sqrt{u^2 + v^2 + w*^2}\right). \tag{1.4}$$

Here, u and v are the grid area-averaged values of the scalar horizontal wind components (predicted by the calling program), $w*$ is a dry covective velocity scale, $\overline{x}$ is the grid area-averaged value of x, x' is the outflow deviation of x from its mean value, and w_d is the outflow velocity scale. In the output of CONVECT, w_d=WD and x'=TPRIME for temperature and QPRIME for specific humidity. The representation of the effect of convective outflows on surface fluxes was found to be important in single-column tests using the TOGA/COARE data set.

The scalar floating point variable CBMF is the convective cloud base upward mass flux, in $Kg\ m^{-2}\ s^{-1}$, and the integer scalar NCOUNT keeps track of the number of time steps with or without convection at each grid column. *It is essential that CBMF and NCOUNT be set to zero at the beginning of an integration and that they be unaltered by the calling program between successive calls to CONVECT at the same horizontal grid point.* That is, for each vertical column of grid points, the calling program must "remember" the last values of CBMF and NCOUNT at that vertical column between calls to CONVECT.

2. *Preliminaries*

After the call statement, various arrays are dimensioned. *Note here that it is necessary to set a parameter value for NA; this should usually equal ND.*

The next group contains all the parameters used by the scheme. Their values have been optimized to produce the best possible forecasts of relative humidity using a single-column model driven by TOGA/COARE data. Their definitions are as follows:

ELCRIT: The critical liquid water content (gm/gm) in the warm parts of cloud, above which all condensed water is assumed to be converted to precipitation. The actual autoconversion threshold value decreases linearly from the value given by ELCRIT to zero over the temperature range from 0 C to the value given by TLCRIT.

TLCRIT: The critical temperature (C) below which the autoconversion threshold is zero.

COEFFMIX: A dimensionless constant that determines the rate at which undilute air mixes with its enviroment, according to the equation

$$\delta M = \mathrm{COEFFMIX} * M \frac{|\delta B|}{B}, \tag{2.1}$$

where M is the undilute mass flux, δM is the rate of mixing in a given altitude interval, $|\delta B|$ is the absolute value of the rate of change of undilute buoyancy over the same altitude interval, and B is the buoyancy itself.

SIGD: The fractional area assumed to be covered by the unsaturated downdraft.

SIGS: The fraction of the precipitation shaft that is assumed to occur in the environment (as opposed to in cloud).

WTRAIN: The terminal velocity (m/s) of rain.

OMTSNOW: The terminal pressure velocity (Pascals/s) of snow.

COEFFR: A dimensionless coefficient governing the rate of evaporation of rain.

COEFFS: A dimensionless coefficient governing the rate of evaporation of snow.

DTCRIT: The critical difference in virtual temperature (K) between an adiabatically lifted parcel and its environment below which no convection is permitted to occur.

DTMAX: The critical magnitude of the "negative area" (here measured in degrees Kelvin) to which the cloud base mass fluxes relax the convecting atmosphere.

SLFRAC: A dimensionless coefficient determining the gradients of temperature and specific humidity in an assumed surface layer. Parcels ascending into cloud base are assumed to have a temperature and specific humidity a fraction SLFRAC of the range between their mixed layer values and their values at the surface itself.

ALPHA and DAMP: Coefficients governing the rate of relaxation of the cloud base mass flux towards its equilibrium value, as governed by the equation

$$\Delta \text{CBMF} = 0.1 * \text{ALPHA} * (\Delta T_v + \text{DTMAX}) - \text{DAMP} * \text{CBMF}, \tag{2.2}$$

where ΔCBMF is the change of cloud base mass flux over a time step, CBMF is the cloud base mass flux itself, and ΔT_v is the minimum value of the difference between the virtual temperature of a parcel lifted from the surface layer and its environment below its level of free convection.

The parameter PPMIN is the minimum pressure (mb) below which CONVECT will not check for unstable parcel origin. With PPMIN set to the standard value, convection is assumed to originate somewhere between the surface and 600 mb. Setting PPMIN to a higher pressure value will reduce computational overhead a little.

After this comes the specification of thermodynamic constants and gravity; these specifications are also made at the beginning of subroutine TLIFT. *These should be consistent with the constants used in the calling program.* They are the heat capacity at constant pressure of dry air, the heat capacity of water vapor, the heat capacity of liquid water, the gas constant for dry air, the gas constant of water vapor, and the latent heat of vaporization at 0° C. Note that the actual latent heat coefficient used in CONVECT is temperature dependent, according to

$$L_v = L_{v0} + (c_{pv} - c_\ell)(T - 273.15). \tag{2.2}$$

This is used even when $T < 0°$ C, making the latent heat coefficient too large at temperatures below freezing. On the other hand, we do not include the latent heat of fusion in the calculation.

In the loop DO 5 ... the arrays FT and FQ and the scalars PRECIP, WD, TPRIME and QPRIME are initialized to zero. IFLAG is also set equal to 0. The array of potential temperature, TH, is calculated.

3. The counter

The integer constant NCOUNT is used as a counter. It is initially incremented by 1. If moist convection occurs, its value is reset to 2 at the end of the program. If its value is equal to 1 (i.e., at the first time step) or greater than 12 (i.e., if ten time steps have passed without convection) then CBMF is set equal to 0.

4. Dry adiabatic adjustment (loop DO 30 ...)

This adjustment is performed internally even if the calling program does some kind of adjustment, to insure the numerical stability of the moist convective calculations.

Working down from the top level, the highest level jn for which $\theta^j < \overline{\theta^{ij}}$ is determined, if any (at present, the check is on θ, not virtual potential temperature). Here, $\overline{\theta^{ij}}$ is the pressure-weighted mean potential temperature in the layer between levels i and j. The adjustment is performed over the levels i to jn, inclusive. The mass-weighted mean values of enthalpy (AHM) and specific humidity (QM) are determined, and Q is set equal to QM in the layer. The rest of the loop calculates θ^j and T^j according to

$$\theta^j = \frac{\sum_{j=i}^{jn}\left[\left(c_{pd}(1-q^j)+c_{pv}q^j\right)T^j\Delta p^j\right]}{\sum_{j=i}^{jn}\left[\left(c_{pd}(1-q_m)+c_{pv}q_m\right)\left(\frac{p^j}{p_0}\right)^{R/c_p}\Delta p^j\right]}, \tag{4.1}$$

$$T^j = \theta^j\left(\frac{p^j}{p_0}\right)^{R/c_p}. \tag{4.2}$$

This preserves the mass-weighted average enthalpy; i.e., the kinetic energy of the adjustment is assumed to be dissipated as heat.

After the adjustment, the adjusted layer is checked for supersaturation. If this occurs, the saturation is removed and the latent heat is added to the saturated layer. The condensed water is assumed to precipitate without evaporation and is added to the variable PRECIP.

If the adjustment results in instability at the top of the adjusted layer, the adjustment is repeated.

This is repeated for each value of i, working downward.

After the dry convective adjustment has been performed, the temperature and specific humidity of the lowest model level are adjusted upward slightly to reflect the presence of a surface layer. Typically, it is air from this lowest model level that participates in moist convection. The upward adjustment of T and Q depends on the values of SLFRAC, TG, QG and PG. The mixed layer values of T and Q are then adjusted downward so that the total enthalpy of the boundary layer is conserved.

5. *Calculation of geopotential and liquid water static energy (Between statements 35 and 40)*

Specific volume is integrated over pressure to give geopotential, GZ, and the environmental liquid water static energy, H, is calculated. The array CPN is the weighted heat capacity at constant pressure.

This marks the beginning of a loop through parcel origin levels. Note that convection is assumed to originate only at the first level with conditionally unstable air.

6. *Lifted condensation level*

The formula used is very similar to that of Bolton (*Mon. Wea. Rev.,* 1980) but computationally faster. PLCL is the lifted condensation pressure of surface air.

7. *Level of cloud base, ICB*

Next, the first level *above* cloud base, ICB, is found. The maximum allowed value of the cloud base mass flux (CBMF) is determined here as well.

8. *Lifted parcel properties, subroutine TLIFT*

The call to TLIFT has as input the arrays P, T, Q, QS, and GZ, and the integer ICB. All of these are unaltered by TLIFT. It returns the absolute temperature (TP) of a parcel lifted adiabatically from the level NK, the total amount of adiabatic condensed water (CLW) in gm gm^{-1}, and a *partial* calculation of the virtual temperature of the lifted parcel which does not yet include water loading effects. The routine is also passed the values of ND and NL. The subroutine TLIFT is described at the end of this document. Note that TLIFT is called twice: Once to determine parcel properties up through cloud base and once again to determine parcel properties above cloud base.

9. *Test for convection*

The next block of code tests whether the column will experience moist convection. Moist convection will occur if the undilute lifted parcel at ICB has a virtual temperature that exceeds that of its environment by more than DTCRIT *and* if the atmosphere is not subadiabatic between the parcel origin level and ICB−1.

If convection does not occur, the next level is checked in loop DO 800 ….

10. *Calculation of parcel precipitation efficiencies and virtual temperatures*

    (loops DO 60 … and DO 64 …)

The parcel precipitation efficiencies, EP, are determined here. The method of doing this is discussed above in the description of the parameter values.
The virtual temperature of the environment, TV, is calculated here and the calculation of the lifted parcel virtual temperature, TVP, is finished. This quantity includes the effect of cloud water (but not precipitation) loading.

11. *Initialization of working arrays*

Working arrays are initialized to zero. Also note that the temperature-dependent latent heat coefficient, LV, is calculated as well as its ratio, LVCP, to the weighted mean heat capacity.

12. *Cloud top*

The cloud top, INB, is here initially defined as one level greater than the highest level at which the lifted parcel is positively buoyant. This may not be greater than NL−1. A check is then performed to insure that there are no intervening stable layers; if there are, then INB is redefined to be at the base of the first such layer.

13. *Reinitialize CBMF*

At the beginning of the integration and if there has been no moist convection at this grid column for 10 time steps, reset CBMF to zero.

14. *Calculation of lifted parcel liquid water static energy (loop DO 95 ...)*

The parcel's liquid water static energy is increased by removal of precipitation:

$$h = [c_{pd}(1 - q) + c_{pv}q]T - L_v\ell + gz \tag{14.1}$$
$$\Delta h = [L_v + (c_{pd} - c_{pv})T]\Delta\ell, \tag{14.2}$$

where $\Delta\ell$ is the amount of condensed water removed from the sample. HP is the lifted parcel's liquid water static energy after removal of precipitation.

15. *Calculation of cloud base mass flux (CBMF) and undilute updraft mass fluxes, (M).*

The cloud base mass flux and the undilute mass fluxes to each level are calculated according to equations (2.1) and (2.2) listed above under the description of parameter values. CBMF is then limited to a magnitude less than that which would violate a CFL condition.

16. *Calculation of mixing fractions and mixed air properties (loop DO 170 ...)*

First, QTI is the cloud water in the undilute updraft after precipitation has been removed. In loop DO 160 ... we calculate the mixing fraction of environmental air, SIJ, that produces no temperature tendency when air mixed at level i detrains at level j. The formula for this is

$$\sigma^{ij}h^i + (1 - \sigma^{ij})h_p^i = h^j + (c_{pv} - c_{pd})T^j(Q^{ij} - q^j), \tag{16.1}$$

where

$$Q^{ij} = \sigma^{ij}q^i + (1 - \sigma^{ij})q_t^i. \tag{16.2}$$

Here, Q^{ij} is the total amount of water substance in the mixture (called QENT in the program). Note that the initial calculation assumes that no precipitation has formed in the mixed air draft. Also note that in all calculations in this subroutine, it is assumed that all water substance in cloud has a heat content proportional to the environmental temperature, rather than the cloud temperature, at that level. This is a very minor approximation but makes the mixing calculations much more straightforward.

The quantity ALTEM is the condensed water in the mixture at level j. It is calculated by expanding the Clausius-Clapeyron equation about the environmental temperature at level j:

$$q_m^{*j} \simeq q^{*j} + \left(\frac{\partial q^*}{\partial T}\right)_p^j \left[T_m^j - T^j\right], \tag{16.3}$$

where q_m^* is the saturation specific humidity of the mixture, q^* is that of the environment, and T_m is the temperature of the mixture. The condensed water in the mixture is just

$$\ell^{ij} = Q^{ij} - q_m^{*j}. \tag{16.4}$$

To solve the preceding two equations for ℓ^{ij} we need to know T_m^j. But the condition that detrainment of the mixture at j produces no temperature change is equivalent to:

$$c_{pd}(T_m^j - T^j) - L_v \ell^{ij} = 0. \tag{16.5}$$

These three relations are now solved for ℓ^{ij}, whose value is temporarily assigned to ALTEM.

The variable CWAT is the amount of condensed water remaining in the undilute updraft after precipitation has been removed. If ALTEM is greater than CWAT, or if the preceding calculation gives nonsensical values of σ^{ij}, a new calculation is performed assuming that precipitation forms. All condensed water in excess of CWAT is converted to precipitation. This means that to the left-hand side of (16.1) we must add the quantity

$$L_v \left[\ell^{ij} - \mathrm{CWAT} \right],$$

with ℓ^{ij} calculated according to (16.3), (16.4), and (16.5). The next block of code after IF((STEMP ... solves (16.1) with this addition, for SIJ ($= \sigma^{ij}$).

Then, the quantities QENT ($= Q^{ij}$) and ELIJ ($= \ell^{ij}$) are calculated. Both pertain to the mixed air *before* removal of precipitation.

The quantity MENT is a *provisional* calculation of the total mass flux of air mixed at level i and detraining at level j. NENT is a counter giving the total number of mixtures formed at level i that can detrain at any other level j.

17. *Calculation of detrained air properties when $j = i$*

It may happen that (16.1) cannot be met for any level $j \neq i$, meaning that mixing with the environment at i will not result in a buoyancy change sufficient to move the parcel to some other level within the cloud layer. In these cases, all of the undilute updraft ascending to i detrains at i. The block of code ending at statement 170 specifies the properties of this detrained air.

18. *Renormalization of entrained air mass fluxes, MENT*

The long block of code constituting the loop DO 200 ... performs a renormalization of MENT so as to result from an equal probability of mixing and to meet one other condition.

First, a critical value of σ is calculated that separates positively buoyant from negatively buoyant mixed air. (Such a value will not always exist; for example, when there is no condensed water in the cloud.) The rest of the code, through statement 180, weights MENT such that it results from an equal probability of mixing and so that ascending air always rises to the highest level for which (16.1) is satisfied.

Finally, an additional check is made to be sure that the renormalization operations have not eliminated all the entrained air fluxes. If it has, then the undilute updraft M is assumed to detrain directly at level i.

19. *Precipitating downdraft calculation*

First, the value of ϵ is checked at level INB. If it is zero, no precipitation has been formed (i.e., the convection is shallow) and this part of the code is skipped.

If precipitation is formed, then the downdraft loop DO 400 ... is performed, working down from INB.

In the next block of code, the total amount of precipitation formed at each level i is calculated; this called WDTRAIN.

Next, a temporary estimate of the specific humidity of the unsaturated downdraft, QP, is made by taking the average of the value of this quantity at the grid point above (which has been calculated accurately) and the value of Q in the environment. Although crude, it is an improvement over assuming that the evaporation is proportional to $q^* - q$ in the environment. The rest of the block solves equations (9), (12a), and (12b) of Emanuel (1991) to find the precipitation content of the downdraft, WATER, and the rate of evaporation of precipitation, EVAP. This involves solving a quadratic equation. This is done using forward differencing to avoid negative values of WATER, at the expense of some accuracy.

After the precipitation and evaporation have been calculated, an estimate of the downdraft mass flux (MP, defined to be positive) is made using the hydrostatic approximation (equation (15) of Emanuel, 1991). A small amount of inertia is added to the hydrostatic downdraft to prevent sudden changes in the downdraft mass flux over short distances. Near the end of the code, MP is tested for violation of the CFL condition; if it does violate CFL then its magnitude is appropriately limited. Also, to promote numerical stability, the downdraft mass flux is forced to decrease linearly near the surface.

Finally, eqns. (14) of Emanuel (1991) are solved for the specific humidity of the downdraft, QP, and the precipitation reaching the surface, PRECIP, is calculated in units of mm d^{-1}. If the downdraft mass flux decreases downward, QP is calculated by making use of the conservation of moist static energy.

20. *Tendencies at lowest model level (statements 405 to 415)*

Upstream differencing is used in the subsidence terms. First, the total undilute updraft mass flux (originating at the lowest model level) is calculated and called AM. The tendency of temperature at the first level is due to:

1. Compensating subsidence warming,
2. Cooling by evaporation of precipitation,
3. Cooling owing to heat transport by precipitation,
4. Cooling owing to heat transport by water vapor in the downdraft,
5. Detrainment of air from unsaturated downdraft,

Likewise, the tendency of specific humidity at the lowest level is:

1. Detrainment of water vapor from unsaturated downdraft and evaporation of precipitation,
2. Compensating subsidence drying,
3. Possible detrainment from unsaturated downdrafts detraining at level 1.

21. *Tendencies above lowest model level*

First the total upward flux of mass in undilute and mixed updrafts through level $i+1$ is calculated; this is called AMP1. Next, the total downward flux of mass in penetrative downdrafts through level i is calculated; this is called AD and is defined to be positive.
Then the temperature tendencies are calculated. In order, the contributions are:

1. Compensating subsidence warming by updrafts, compensating adiabatic cooling due to penetrative downdrafts, and cooling by evaporation of precipitation.
2. Detrainment of air from undilute updrafts for which no entrainment fluxes occur.
3. Cooling by downward transport of heat by precipitation.
4. Cooling by downward transport of heat by water vapor in downdraft.

Likewise, the specific humidity tendency is contributed to by:

1. Drying by compensating subsidence and moistening by compensating ascent associated with penetrative downdrafts.
2. Moistening by detrainment from all drafts.
3. Moistening by evaporation of precipitation and convergence of the unsaturated downdraft flux of water vapor.

22. *Re-interpolation of cloud top (INB)*

Here the tendencies of temperature and specific humidity at levels INB and INB-1 are re-weighted in such a way that the cloud top detrainment effectively occurs at the real level of neutral buoyancy rather than at the model level INB.

23. *Homogenization of tendencies below cloud base*

This is once again done in the interest of numerical stability and because the physics of the scheme are too crude to justify any vertical structure of the moist convective tendencies in the subcloud layer, except at the top of the subcloud layer where subsidence warming and drying are physically meaningful and important.

If there has been no convection, the value of CBMF is relaxed towards zero.

This is the end of the main subroutine.

23. *Subroutine TLIFT*

This subroutine calculates a close approximation to the lifted temperature and adiabatic condensed water content of air from the lowest level. It accepts the input arrays P, T, Q, QS, and GZ, and the integer ICB. It leaves all of these unaltered. It returns the lifted parcel temperature and adiabatic condensed water content as well as a partial calculation of the parcel's virtual temperature. (Later in the main subroutine, when ϵ^i is calculated, the water loading term is added.)
First, the same thermodynamic constants are specified as in the main subroutine. Then the moist static energy at the lowest level, AH0, is calculated, and the temperature of air lifted dry adiabatically to one level below cloud base is found.
The principal calculations are performed in the loop DO 300 .... The calculation is based on conservation of moist static energy, again approximating the temperature of the water substance by the environmental temperature:

$$c_{pd}{T_p}^i + (c_\ell - c_{pd})q^1 T^i + gz^i + L_v^i {q_p}^i = \mathrm{AH0}, \tag{23.1}$$

where ${T_p}^i$ is the lifted parcel temperature and ${q_p}^i$ is the lifted parcel saturation specific humidity. The rate of change, S, of moist static energy with respect to temperature at constant pressure is calculated approximately, and the lifted parcel temperature (and saturation specific humidity) are calculated using Newton's method with two iterations. At the end of the iteration, the temperature is recalculated to insure exact conservation of h.
Next, the adiabatic condensed water is calculated. This allows the partial calculation of the lifted virtual temperature.

24. Problems

Kindly report any problems to the author:

K. Emanuel
Room 54–1620
Massachusetts Institute of Technology
Cambridge, MA 02139

(Internet) emanuel@texmex.mit.edu

25. Registration

Please send an email message to emanuel@texmex.mit.edu stating that you are using the code. This will also serve to place you on an update list.