Introduction to EULAG
(and cloud modeling in general)

Wojciech W. Grabowski

NCAR, Boulder, USA

SMALL-SCALE DYNAMICS:

In the spirit of the Boussinesq approximation, moisture and condensate affect gas dynamics equations only though the buoyancy term

$$\frac{d\mathbf{u}}{dt} = -\frac{1}{\rho}\nabla p - g\mathbf{k} + ...(\text{Coriolis, turbulence, etc})$$

$$\rho = \rho_o(z) + \rho'$$

$$p = p_o(z) + p'$$

$$(\rho_o + \rho')\frac{dw}{dt} = -\frac{\partial p_o}{\partial z} - \rho_o g - \frac{\partial p'}{\partial z} - \rho' g + ...$$

$$p = \rho R T$$

$$p'/p = \rho'/\rho + T'/T$$

$$p \sim 1000 \text{ hPa}$$

$$p' \sim \rho u^2 \sim 1 \text{ hPa}$$

$$\frac{d\mathbf{u}}{dt} = -\frac{1}{\rho_o}\nabla p' - g\mathbf{k}\frac{\rho'}{\rho_o} + ...$$

For small-Mach number flows ($|\mathbf{u}| \ll c_s$; c_s - speed of sound):

$$\frac{\rho'}{\rho_o} \approx -\frac{T'}{T_o}$$

$$\frac{d\mathbf{u}}{dt} = -\frac{1}{\rho_o}\nabla p' + g\mathbf{k}\frac{T'}{T_o} + ...$$

Density temperature T_d: the temperature dry air has to have to yield the same density as moist cloudy air

$$T_d = T\frac{1+q/\epsilon}{1+q+Q}$$

T - air temperature

q - water vapor mixing ratio ($\sim 10^{-3}$)

Q - condensate mixing ratio (cloud water, rain, ice, snow, etc.; $\sim 10^{-3}$)

$$\epsilon = \frac{R_d}{R_v} \approx 0.622$$

$$T_d \approx T\left[1+\left(\frac{1}{\epsilon}-1\right)q - Q\right]$$

$$T_d \approx T\left(1+0.61q-Q\right)$$

T, q and Q – thermodynamics (and much more!)

Water vapor is a minor constituent:

mass loading is typically smaller than 1%; thermodynamic properties (e.g., specific heats etc) only slightly modified;

Suspended small particles (cloud droplets, cloud ice):

mass loading is typically smaller than a few tenths of 1%, particles are much smaller than the smallest scale of the flow; multiphase approach is not required, but sometimes used (e.g., DNS with suspended droplets, Lagrangian Cloud Model)

Precipitation (raindrops, snowflakes, graupel, hail):

mass loading can reach a few %, particles are larger than the smallest scale the flow; multiphase approach needed only for very-small-scale modeling

***Continuous medium approach**: apply density as the main field variable (density of water vapor, density of cloud water, density of rainwater, etc…)*

$$\frac{\partial \rho_v}{\partial t} + \nabla(\rho_v \mathbf{u}) = S \quad \text{or} \quad \frac{d\rho_v}{dt} + \rho_v \nabla \mathbf{u} = S$$

In practice, mixing ratios are typically used. Mixing ratio is the ratio between the density (of water vapor, cloud water…) and the air density.

Mixing ratios versus specific humidities…

$$\frac{\partial \rho_a}{\partial t} + \nabla(\rho_a \mathbf{u}) = 0 \quad \text{or} \quad \frac{d\rho_a}{dt} + \rho_a \nabla \mathbf{u} = 0$$

$$\frac{\partial \rho_v}{\partial t} + \nabla(\rho_v \mathbf{u}) = S \quad \text{or} \quad \frac{d\rho_v}{dt} + \rho_v \nabla \mathbf{u} = S$$

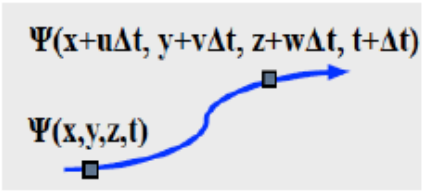
$$\text{mixing ratio :} \quad q = \frac{\rho_v}{\rho_a}$$

$$\frac{dq}{dt} = \frac{S}{\rho_a}$$

$$\text{specific humidity :} \quad Q = \frac{\rho_v}{\rho_v + \rho_a}$$

$$\frac{dQ}{dt} = \frac{\rho_a}{\rho_v + \rho_a} \frac{S}{\rho_v + \rho_a}$$

Lagrangian versus Eulerian formulation



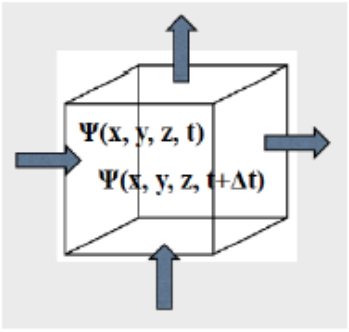
$$\frac{D\Psi}{Dt} = S$$

or

$$\frac{\partial \Psi}{\partial t} + \mathbf{u} \cdot \nabla \Psi = S$$

combined with dry air continuity equation:

$$\frac{\partial \rho_a}{\partial t} + \nabla(\rho_a \mathbf{u}) = 0$$



gives:

$$\frac{\partial \rho_a \Psi}{\partial t} + \nabla(\rho_a \mathbf{u} \Psi) = \rho_a S$$

For the anelastic system:

$$\frac{\partial \Psi}{\partial t} + \frac{1}{\rho_o} \nabla(\rho_o \mathbf{u} \Psi) = S$$

$$\rho_o = \rho_o(z)$$

And we also need equation for the temperature:

First Law of Thermodynamics:

$$dq = du + p\ dv \tag{1}$$

dq - heat (per unit mass) added to the system
du - increase of internal energy (per unit mass)
$p\ dv$ - work (per unit mass) performed by the system

$$du = c_v\ dT, \quad pv = RT, \quad v = 1/\rho, \quad c_v + R = c_p$$

$$dq = c_p\ dT - \frac{RT}{p} dp \tag{2}$$

Indtroducing *potential temperature* as:

$$\theta = T \left(\frac{p_{oo}}{p} \right)^{R/c_p} \tag{3}$$

where p_{oo}=const (typically 1000 mb), (1) can be written as:

$$d\theta = \frac{\theta}{c_p T} dq \tag{4}$$

And we also need equation for the temperature:

$$\frac{d\theta}{dt} = \frac{\theta}{c_p T} S$$

where $S = \frac{dq}{dt}$ in the heat source per unit mass [in J kg^{-1} s^{-1}]

$S = 0$ - adiabatic motions

$S \neq 0$ - motions with diabatic processes (heating due to radiative transfer, phases changes, chemical reactions, etc)

For phase changes of water substance:

$$S = L \frac{dq}{dt}$$

where L is the latent heat (of condensation, freezing, or sublimation), and $\frac{dq}{dt}$ is the change of corresponding water mixing ratio

Modeling of cloud microphysics: ***solving a system of PDEs (advection/diffusion type) coupled through the source terms:***

$$\frac{d\theta}{dt} = S_\theta$$

$$\frac{dq_v}{dt} = S_{q_v}$$

for $i = 1, N$:

$$\frac{dq_c^{(i)}}{dt} = S^{(i)}$$

θ - potential temperature
q_v - *water vapor* mixing ratio
$q_c^{(i)}$ - *condensed water* mixing ratios
$S^{(i)}$ - sources/sinks for condensed water (phase changes, transfer from one category to another, sedimentation, etc.)

BULK MODEL OF CONDENSATION:

$$\frac{d\theta}{dt} = \frac{L_v \theta}{c_p T} C_d$$

$$\frac{dq_v}{dt} = -C_d$$

$$\frac{dq_c}{dt} = \quad C_d$$

θ - potential temperature
q_v - *water vapor* mixing ratio
q_c - *cloud water* mixing ratio
L_v - latent heat of condensation/evaporation
C_d - condensation rate
Note: θ/T function of pressure only ($\approx \theta_o/T_o$)

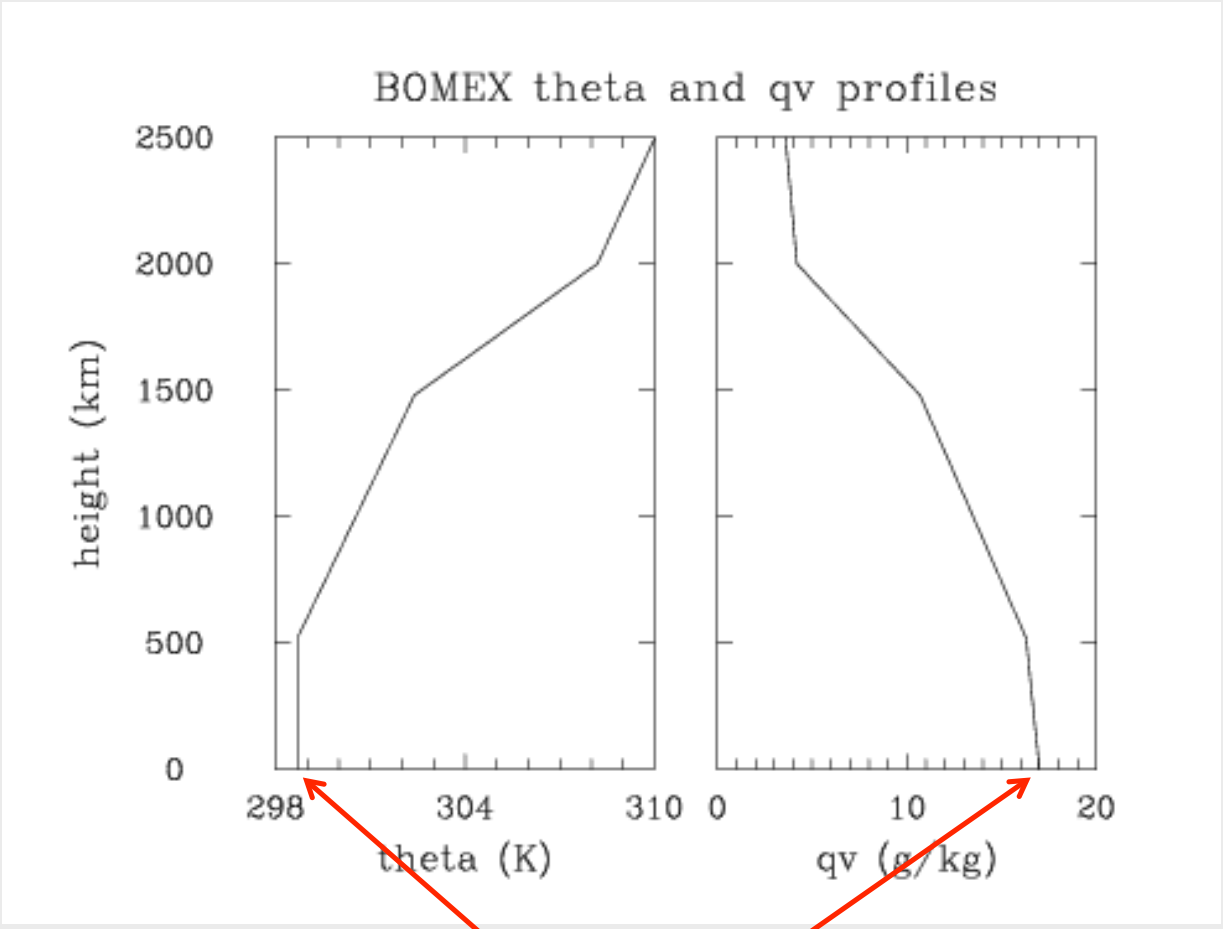
C_d is defined such that cloud is always at saturation, which is a very good approximation:

$$q_c = 0 \quad \text{if} \quad q_v < q_{vs}$$

$$q_c > 0 \quad \text{only if} \quad q_v = q_{vs}$$

where $q_{vs}(p,T) \approx 0.622 \dfrac{e_s(T)}{p}$ is the water vapor mixing ratio at saturation

A very simple (but useful) model: rising adiabatic parcel…



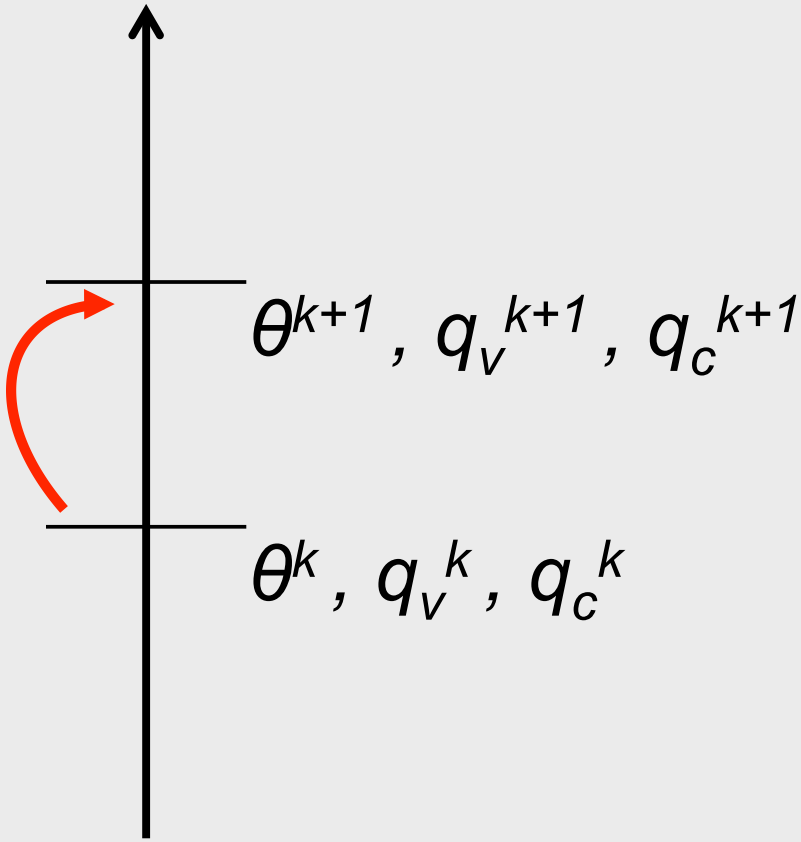
$$\frac{d\theta}{dt} = \frac{L_v \theta}{c_p T} C_d$$

$$\frac{dq_v}{dt} = -C_d$$

$$\frac{dq_c}{dt} = C_d$$

… by solving these equations.

Take a parcel from the surface and move it up…



$$\theta^{k+1} = \theta^k + \frac{L_v \theta_e}{c_p T_e} \Delta q$$

$$q_v^{k+1} = q_v^k - \Delta q$$

$$q_c^{k+1} = q_c^k + \Delta q$$

$$\Delta q = ?$$

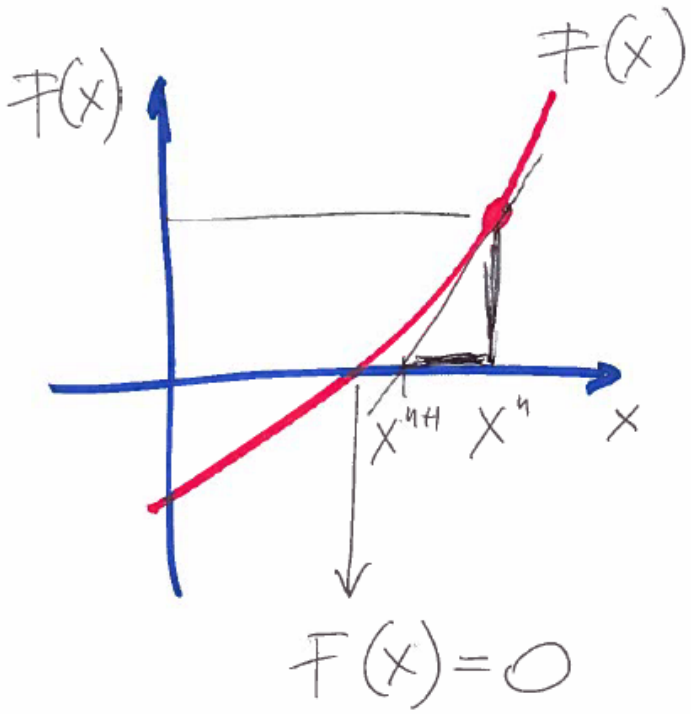
$$q_v^{k+1} = q_{vs}(\theta^{k+1})$$

$$q_v^k - \Delta q = q_{vs}(\theta^k + \frac{L_v \theta_e}{c_p T_e} \Delta q)$$

The nonlinear equation for Δq can be solved using the Newton-Raphson method…

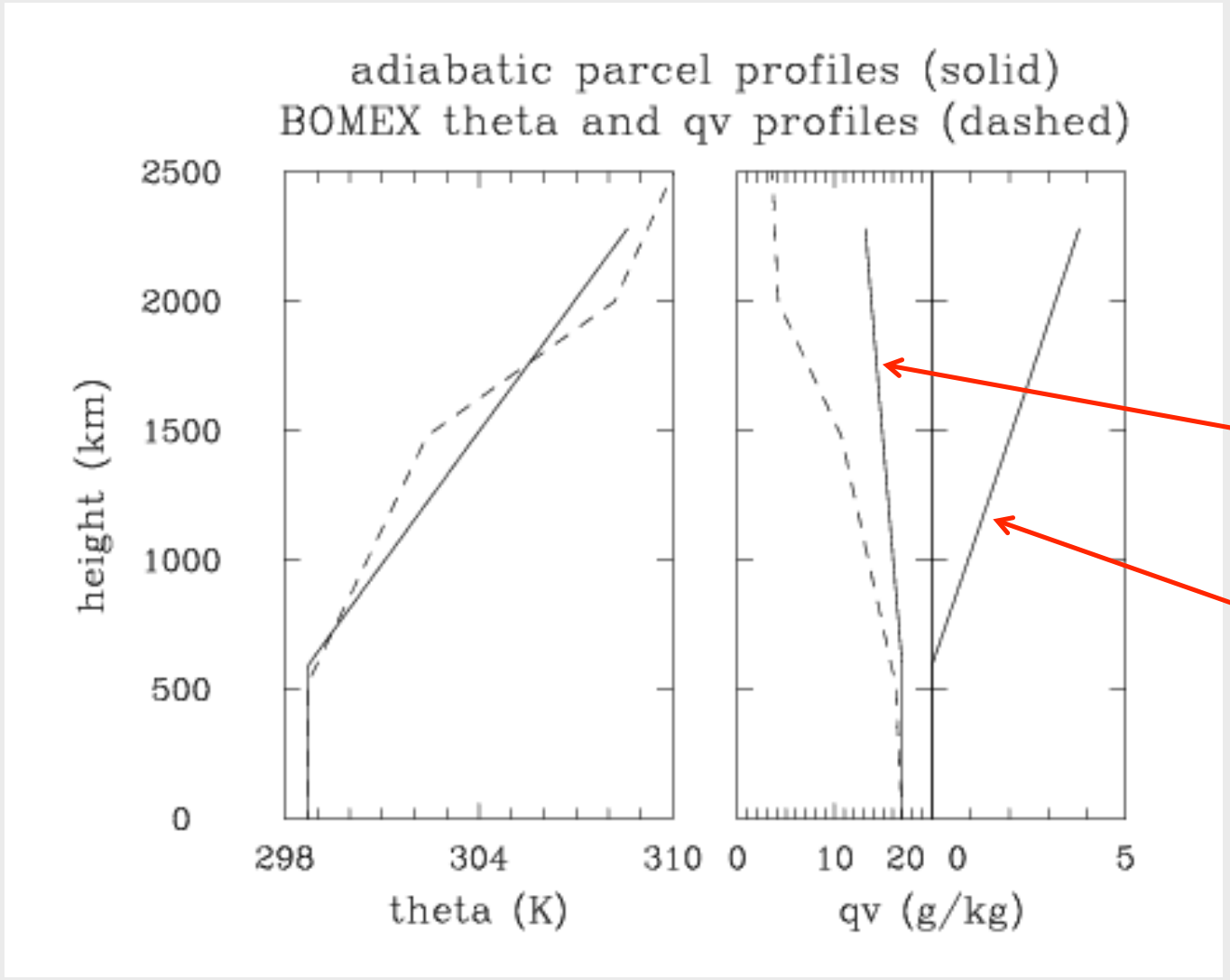
F(x) = 0

x = ?



$$F'(x^n) = \frac{F(x^n)}{x^n - x^{n+1}}$$

$$x^{n+1} = x^n - \frac{F(x^n)}{F'(x^n)}$$



Look not only on the patterns (i.e., processes), but also on specific numbers (e.g., temperature change, mixing ratios, etc).

Invariant variables:

total water,

liquid water potential temperature,

equivalent potential temperature.

If $\overline{\theta}/\overline{T} = \text{const}$ (shallow convection approximation)

$$\frac{d\theta}{dt} = \frac{L_v\overline{\theta}}{c_p\overline{T}}C_d$$

$$\frac{dq_v}{dt} = -C_d$$

$$\frac{dq_c}{dt} = C_d$$

can be converted into:

$$\frac{d\theta_I}{dt} = 0$$

$$\frac{dQ}{dt} = 0$$

θ_I is one of the two:

$\theta_e = \theta + \frac{L_v\overline{\theta}}{c_p\overline{T}}q_v$ - equivalent potential temperature

$\theta_l = \theta - \frac{L_v\overline{\theta}}{c_p\overline{T}}q_c$ - liquid water potential temperature

$Q = q_v + q_c$ - total water mixing ratio

Adding rain or drizzle:

THE DISTRIBUTION OF RAINDROPS WITH SIZE

By J. S. Marshall and W. McK. Palmer[1]
McGill University, Montreal
(Manuscript received 26 January 1948)

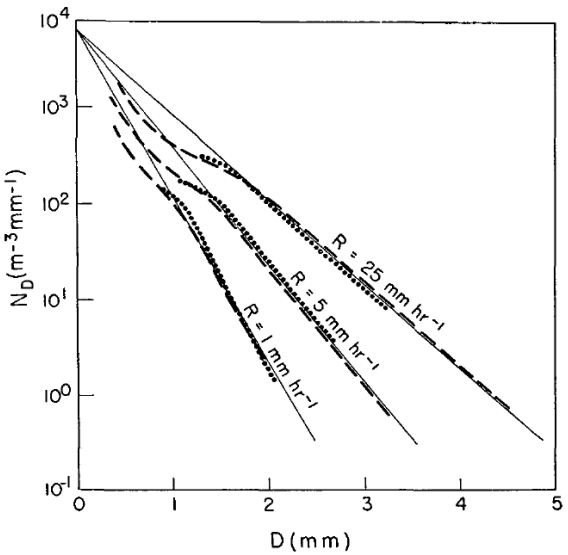


FIG. 2. Distribution function (solid straight lines) compared with results of Laws and Parsons (broken lines) and Ottawa observations (dotted lines).

WARM RAIN BULK MODEL (Kessler 1969):

$$\frac{d\theta}{dt} = \frac{L_v \theta}{c_p T}(C_d - EVAP)$$

$$\frac{dq_v}{dt} = -C_d + EVAP$$

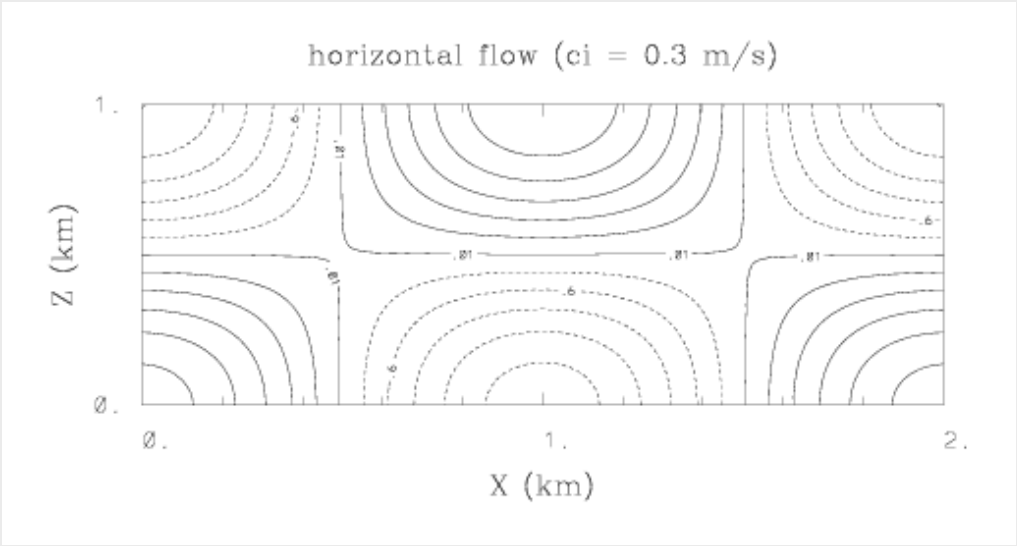
$$\frac{dq_c}{dt} = C_d - AUT - ACC$$

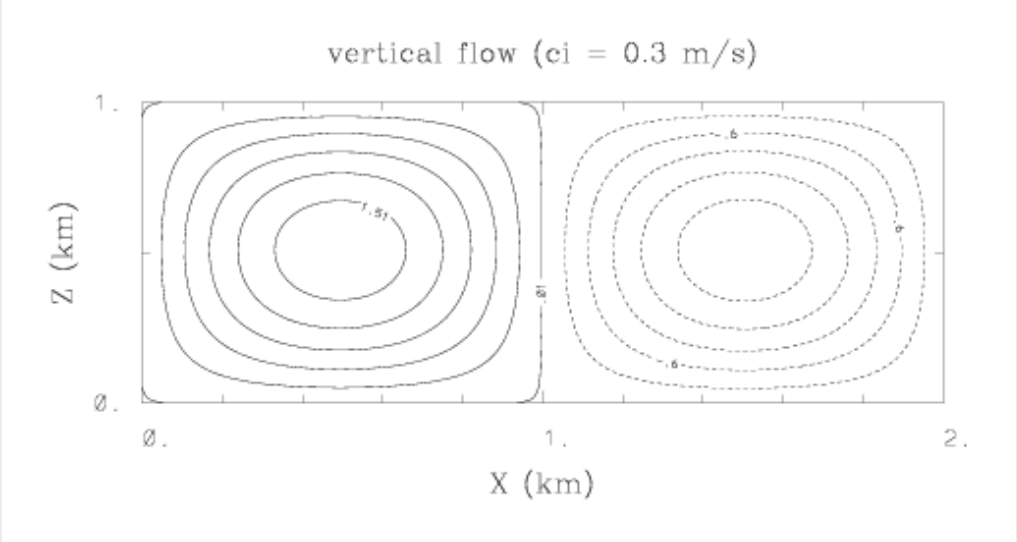
$$\frac{dq_r}{dt} = \frac{1}{\rho}\frac{\partial}{\partial z}(\rho q_r v_t) + AUT + ACC - EVAP$$

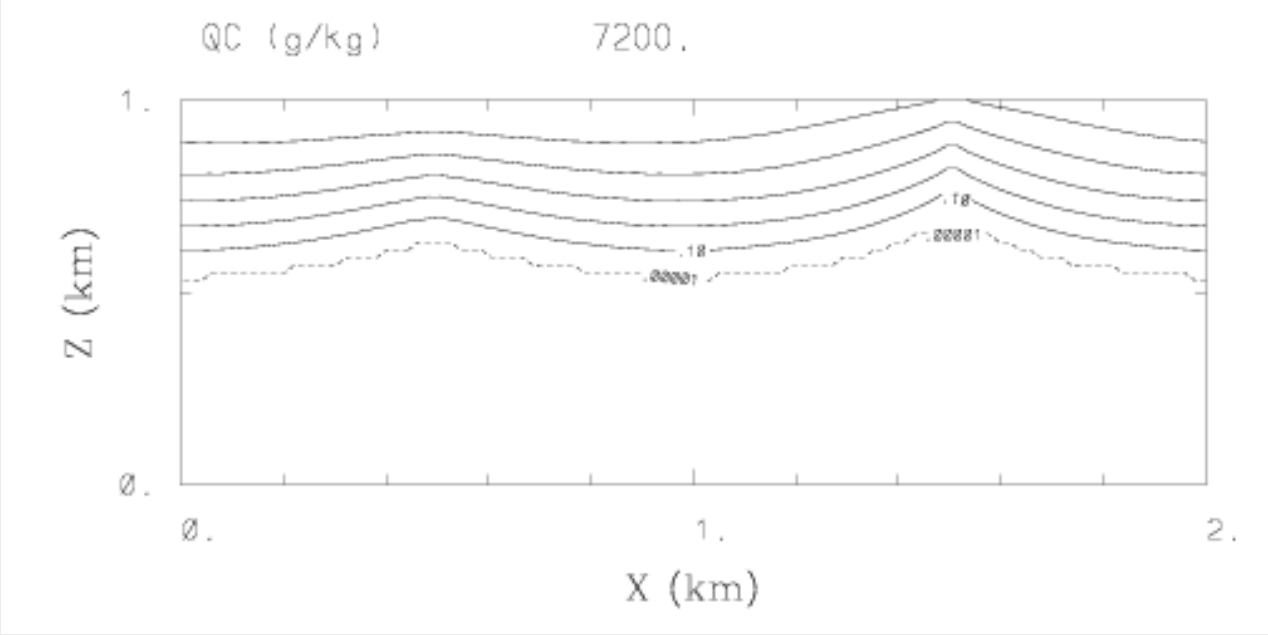
θ - potential temperature
q_v - *water vapor* mixing ratio
q_c - *cloud water* mixing ratio
q_r - *rain water* mixing ratio
C_d - condensation rate
$EVAP$ - rain evaporation rate
AUT - "autoconversion" rate: $q_c \rightarrow q_r$
ACC - accretion rate: $q_c, q_r \rightarrow q_r$

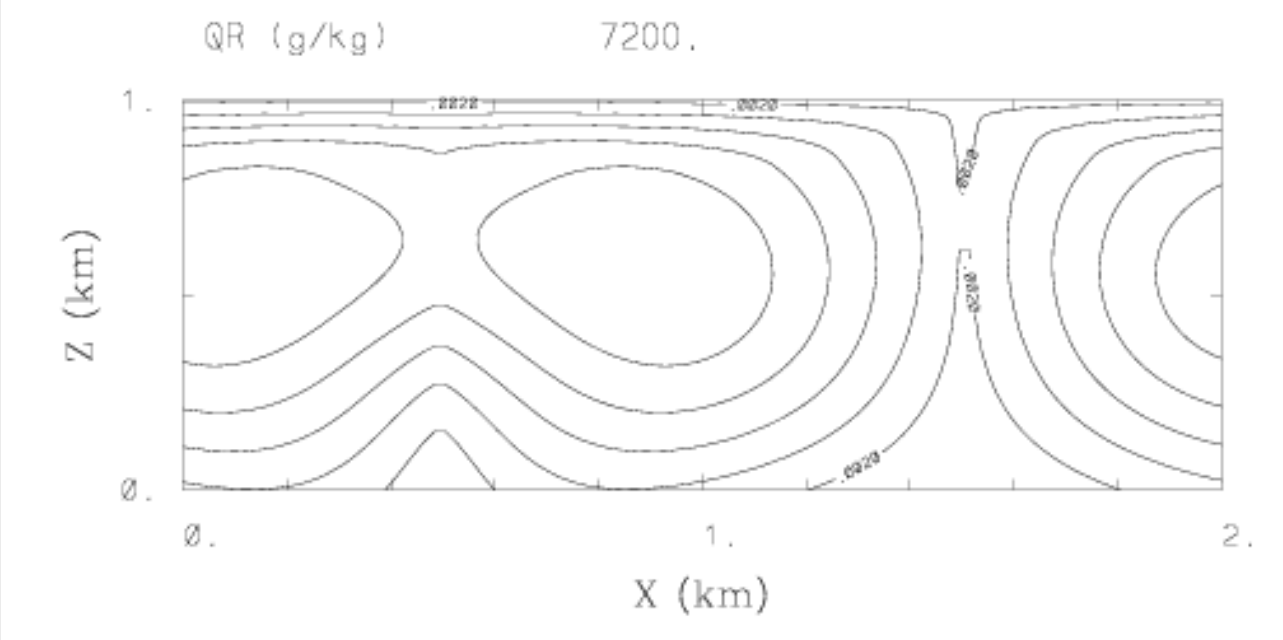
$v_t(q_r)$ - rain terminal velocity (typically derived by assuming a drop size distribution; e.g., the Marshall-Palmer distribution $N(D) = N_o exp(-\Lambda D)$, $N_o = 10^7$ m^{-4}).

We need something more complicated than a rising parcel as rain has to fall out. One possibility is to use the kinematic (prescribed flow) framework…









Cloud water and rain (drizzle) fields after 2 hrs (almost quasi-equilibrium…)

BabyEULAG documentation and code

Equations to be solved

The equations solved by the babyEULAG in the warm-rain version are (e.g., Grabowski and Smolarkiewicz, MWR 1996; Grabowski 1998):

$$\frac{d\mathbf{v}}{dt} = -\nabla\pi + \mathbf{k}B + \mathbf{D}_{\mathbf{v}}, \qquad (1a)$$

$$\nabla(\bar{\rho}\mathbf{v}) = 0, \qquad (1b)$$

Dynamics (fluid flow)

$$\frac{d\theta}{dt} = \frac{L\theta_e}{c_p T_e}(C_d - E_r) + D_\theta, \qquad (1c)$$

$$\frac{dq_v}{dt} = -C_d + E_r + D_{q_v}, \qquad (1d)$$

$$\frac{dq_c}{dt} = C_d - A_r - C_r + D_{q_c}, \qquad (1e)$$

$$\frac{dq_r}{dt} = \frac{1}{\bar{\rho}}\frac{\partial}{\partial z}(\bar{\rho} v_t q_r) + A_r + C_r - E_r + D_{q_r}. \qquad (1f)$$

Thermodynamics (temperature and moisture variables)

Buoyancy: $$B \equiv g\left[\frac{\theta - \theta_e}{\bar{\theta}} + \epsilon(q_v - q_{v_e}) - q_c - q_r\right], \qquad (2)$$

$$d/dt \equiv \partial/\partial t + \mathbf{v}\cdot\nabla$$

The subgrid-scale diffusion terms (last terms on rhs; D above) are all zero in babyEULAG. This is in the spirit of the implicit large-eddy simulation (ILES) for small-scale dynamics.

DYNAMICS:

$$\frac{d\mathbf{v}}{dt} = -\nabla\pi + \mathbf{k}B \qquad \text{Dynamics (fluid flow)}$$

$$\nabla(\bar{\rho}\mathbf{v}) = 0,$$

$\mathbf{v} = (u,v,w)$ – fluid flow;
π – perturbation pressure; ensures that the flow is anelastic;
B – buoyancy (density temperature perturbations driving the flow)
$\bar{\rho} = \rho(z)$ – anelastic density profile (the only effect of compressibility)

THERMODYNAMICS:

$$\frac{d\theta}{dt} = \frac{L\theta_e}{c_p T_e}(C_d - E_r) \qquad \text{potential temperature}$$

$$\frac{dq_v}{dt} = -C_d + E_r \qquad \text{water vapor mixing ratio (m. r.)}$$

$$\frac{dq_c}{dt} = C_d - A_r - C_r \qquad \text{cloud water (condensate) m. r.}$$

$$\frac{dq_r}{dt} = \frac{1}{\bar{\rho}}\frac{\partial}{\partial z}(\bar{\rho}v_t q_r) + A_r + C_r - E_r \qquad \text{rain (precipitation) m. r.}$$

C_d - condensation rate; q_v --> q_c
E_r - precipitation evaporation rate; q_r --> q_v
A_r - autoconversion rate; q_c --> q_r
C_r - collection rate; $q_c + q_r$ --> q_r

condensate – follows the flow (no sedimentation)
precipitation – falls out (1st term on rhs represents sedimentation)

NOTE: Significantly more complex microphysics parameterizations have been used with 3D babyEULAG (see Grabowski JAS 2015, Grabowski and Jarecka JAS 2015, Grabowski and Morrison JAS 2016).

BULK MODEL OF CONDENSATION:

$$\frac{D\theta}{Dt} = \frac{L_v \theta}{c_p T} C_d$$

$$\frac{Dq_v}{Dt} = -C_d$$

$$\frac{Dq_c}{Dt} = C_d$$

θ - potential temperature
q_v - *water vapor* mixing ratio
q_c - *cloud water* mixing ratio
L_v - latent heat of condensation/evaporation
C_d - condensation rate
Note: θ/T function of pressure only ($\approx \theta_e/T_e$, i.e., environmental hydrostatic pressure)

C_d is defined such that cloud is always at saturation, which is a very good approximation:

$$q_c = 0 \quad \text{if} \quad q_v < q_{vs}$$

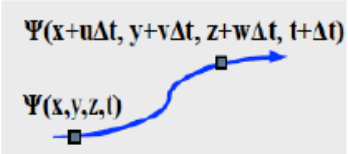
$$q_c > 0 \quad \text{only if} \quad q_v = q_{vs}$$

where $q_{vs}(p,T) \approx 0.622 \frac{e_s(T)}{p}$ is the water vapor mixing ratio at saturation

How these equations are solved?

First, the equations are converted from advective form to the flux (conservation) form:

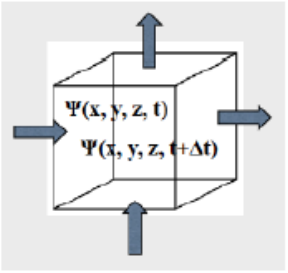
Lagrangian versus Eulerian formulation



$$\frac{D\Psi}{Dt} = S \quad \text{or} \quad \frac{\partial \Psi}{\partial t} + \mathbf{u} \cdot \nabla \Psi = S$$

combined with dry air continuity equation:

$$\frac{\partial \rho_a}{\partial t} + \nabla(\rho_a \mathbf{u}) = 0$$



gives:

$$\frac{\partial \rho_a \Psi}{\partial t} + \nabla(\rho_a \mathbf{u} \Psi) = \rho_a S$$

For the anelastic system:

$$\frac{\partial \Psi}{\partial t} + \frac{1}{\rho_o} \nabla(\rho_o \mathbf{u} \Psi) = S$$

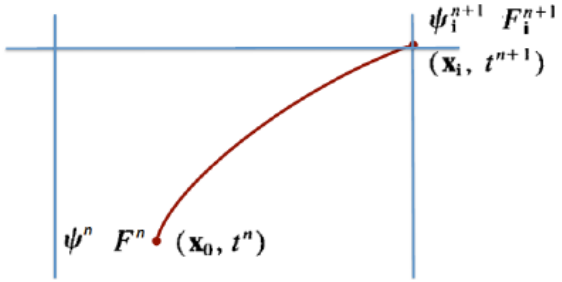
$$\rho_o = \rho_o(z)$$

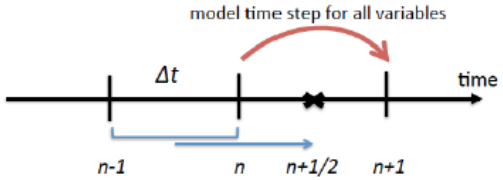
Note that advective and flux-form formulations are equivalent. Thus, while the model solves flux-form equations, one can apply the trajectory-wise thinking to design the numerical algorithm.

Second, the integration scheme follows the non-oscillatory forward-in-time (NFT) applying the trapezoidal (centered-in-time) integration. This can be schematically written as (Grabowski and Smolarkiewicz MWR 1996, Eqs. 6 and 7):

$$\frac{d\psi}{dt} = F \longrightarrow \psi_i^{n+1} = \left(\psi + \frac{1}{2}\Delta t F\right)_0 + \frac{1}{2}\Delta t F_i^{n+1}$$

n – time level
i – position on the grid (shorthand for i,j,k)
subscript $_0$ – value at the departure point $(\mathbf{x}_0, t^n)$ arriving at the grid point $(\mathbf{x}_i, t^{n+1})$.





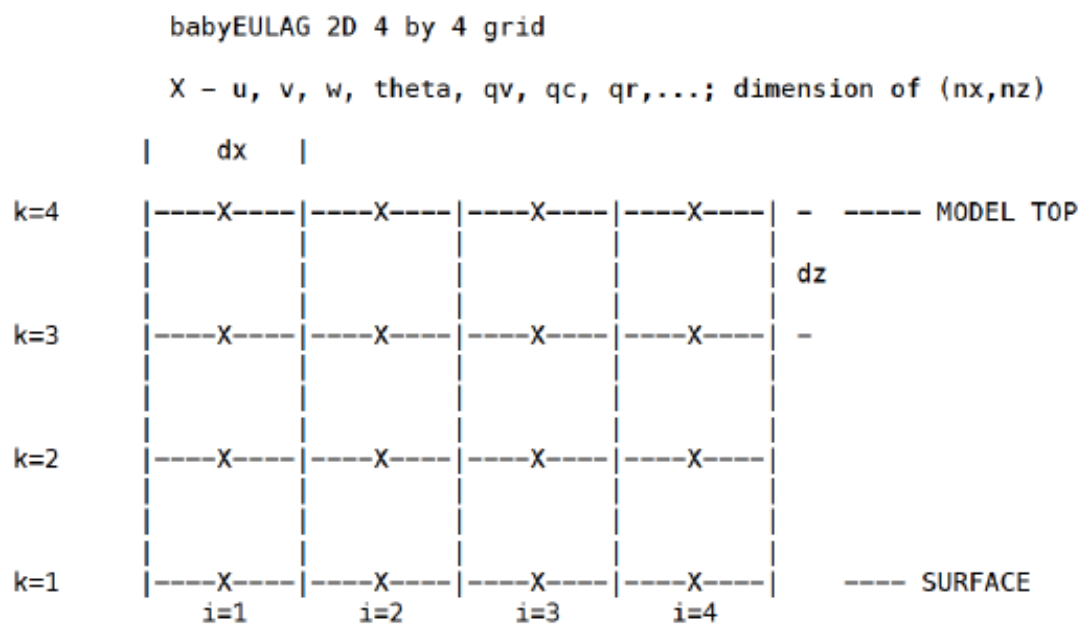
$$\psi_i^{n+1} = \left(\psi + \frac{1}{2}\Delta t F\right)_0 + \frac{1}{2}\Delta t F_i^{n+1}$$

To move model fields from n to $n+1$ time level, advecting velocities are needed at $n+1/2$ time level. This can be accomplished by various methods. A simple one is to linearly extrapolate fluid velocities from $n-1$ and n time levels. This is what babyEULAG does.

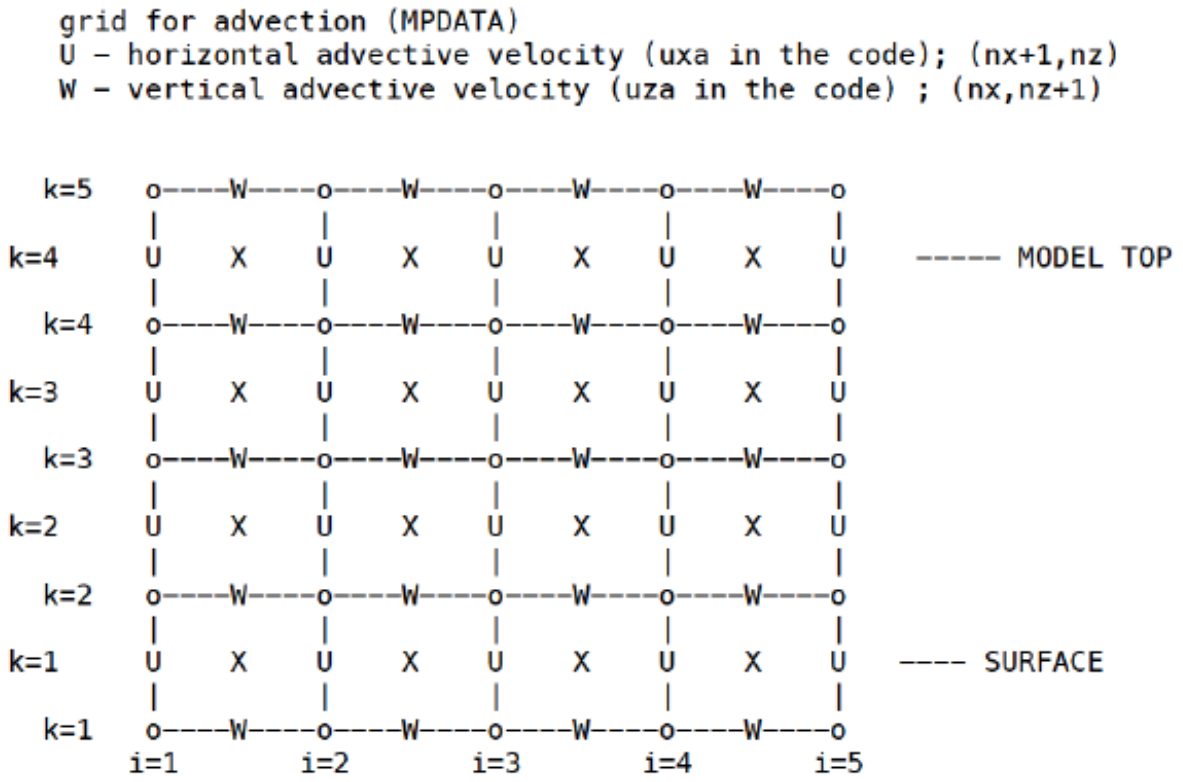
NOTE: See below more detailed discussion on how this implemented for momentum and thermodynamic variables.

Model grid

All model variables are collocated as illustrated below. Note that there are model levels at the surface and at the model top that helps impose boundary conditions.



However, for advection, one needs to prescribe advective fluxes at the cell edges as this is how MPDATA (Multidimensional Positive-Definite Advection Transport Algorithm) works. The C-grid type is then used and flow velocities are derived at cell edges by appropriate averaging.



Boundary conditions

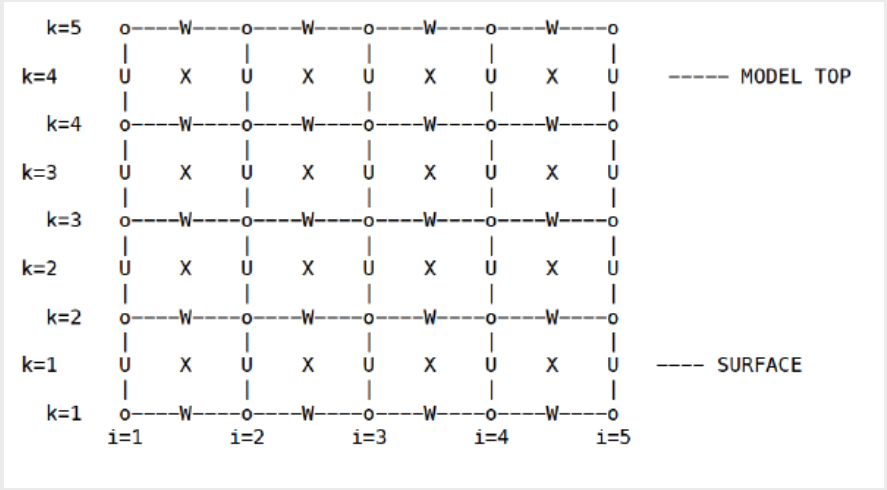
The code applies simple boundary conditions. In the horizontal, the domain is periodic, that is, $\Psi(i=1) = \Psi(i=nx)$ for all fields. Note that the periodicity also implies that:

$$U(i=1) = U(i=nx) \quad \text{and} \quad U(i=nx+1) = U(i=2).$$

In the vertical, rigid lid conditions are used. These imply that the advective fluxes across the surface and across the model top have to vanish. It follows that:

$$W(k=1) = -W(k=2) \quad \text{and} \quad W(k=nz+1) = -W(k=nz).$$

Note that these mean free-slip conditions for the horizontal momentum. If there are surface fluxes (of temperature, moisture, or momentum), these are best added separately (e.g., in a separate subroutine) and – if the vertical and horizontal resolution is insufficient to support boundary layer eddies like in LES – distribute the flux in the vertical mimicking the action of the eddies. For instance, the temperature flux varies approximately linearly with height in the convective boundary and this observation can be used to distribute the surface flux in convective situations.



More details about time stepping.

For the momentum, the time stepping is exactly as explained above. Note that thermodynamics is completed before momentum equations are finalized. Thus, thermodynamic variables are available to calculate buoyancy at *n+1* time level. The pressure gradient term is calculated last to ensure the "nondivergent" final flow. If u^* represents u at *n+1* time level that includes all sources at *n+1* level except for the pressure gradient term (i.e., buoyancy alone when only buoyancy and pressure gradient are considered as in 1a) and u stands for the final solution, then:

$$u - u^* = -\Delta t/2 \ \nabla\pi$$

Applying the anelastic continuity equation to u gives:

$$\nabla(\rho_o u) = \nabla[\rho_o (u^* - \Delta t/2 \ \nabla\pi)] = 0$$

This leads to the Poisson equation for the pressure field at the n+1 time level.

Note that if other terms are included in the momentum equation (e.g, the Coriolis force, surface fluxes) these need to be included through an un-centered in time approach, that is, as precipitation processes in the thermodynamics (see below).

For the thermodynamics, finding the forces due to precipitation processes at n+1 time level in the trapezoidal rule is cumbersome. One can use the predictor-corrector technique to do this, but because precipitation processes typically involve significantly longer time scale (say, tens of minutes), then an un-centered in time (Euler forward) time stepping is a sensible approach. This is what babyEULAG does. The thermodynamic equations are then written as (Grabowski and Smolarkiewicz MWR 1996):

$$\frac{d\psi}{dt} = F^F + F^S,$$

where superscript F refers to "fast forces" (like the condensation rate, infinitely fast in the bulk approach) and S refers to "slow forces" (i.e., those involved in precipitation formation and growth). Treating fast and slow forces through centered and un-centered in time gives:

$$\psi_i^{n+1} = \left(\psi + \Delta t F^S + \frac{1}{2} \Delta t F^F \right)_0 + \frac{1}{2} \Delta t F_i^F$$

Note that in the code, the fast and slow tendencies are group together. Slow tendencies need to be multiplied by 2 to account for 1/2 coefficient when applied before advection.

Final comment concerns precipitation sedimentation, the first term on rhs in Eq. 1f. It is included by a modification of the advecting vertical velocity, that is, by adding the precipitation fall velocity to the advective flow velocity. Since the fall velocity is calculated applying precipitation mixing ratio derived using an un-centered scheme, the precipitation sedimentation is calculated in the spirit of the un-centered in time scheme as well.

Explanation of the initialization in babyEULAG

Before entering the time-stepping loop ("MARCH FORWARD IN TIME"), the program sets the vertical grid ("call zstrtch"), initializes moisture parameters ("call moist_init") and initializes initial profiles (dry base-state, th0, rhe0; and moist environmental profiles, th_e, qv_e, tm_e as well as initial velocity profiles, ux_e and uy_e). Then the model sets initial 3D fields that may include random perturbations or an initial localized perturbation, like a bubble (the loop after "initial fields").

The time stepping starts with a call to velprd_1. This is the velocity predictor. Since MPDATA requires "n+1/2" time level velocities, these are extrapolated in time from "n-1" and "n" time level data (this is why the two level velocities are kept in the code). Note also that these velocities need to be interpolated in space to get them at cell edges (as required by the MPDATA advection routine) and multiplied by the air density. Finally, they are converted into "Courant numbers". These are uxa, uya and uza variables used in calls to mpdata. See velprd_1 for details.

Before calling mpdata, "n" time level velocities are into "n-1" velocities (loop below "save previous velocities"). Then some of the physics routines are called (e.g., surface fluxes, large-scale subsidence, etc). Note that tendencies in those routines are multiplied by 2 so they can be multiplied by dt/2 (rather then by dt) when they are included in the new time level calculations (the loop below "add half of the force").

Advection is performed by calls to mpdata. Variables that move with the air (momenta, theta, qv, and qc) are advected by uxa, uya, uza. Variables that move relative to the air (e.g., qr), are also advected by uxa and uya, but the vertical component of the advection velocity is modified to include fall velocity. This is done in a call to rain_fall that modifies uza. Additional comment is needed for the "liner" variable set just prior to calls to mpdata. This variable forces mpdata to complete just upwind advection, and it is typically done every a few model timesteps (say, every 6 timesteps) to provide additional smoothing of the advected fields. The model should run fine even without this option (e.g., liner=0 all the time).

After advection, the advecting velocities are no longer needed and they are used to temporary store velocity components (with partial forcings) after advection that are needed to calculate velocity forces once the new pressure is derived. See the loop below "save velocities after advection".

Having new thermodynamic fields (except for fast tendencies at "n+1" time level, just condensation in our case) allows calling thermodynamics routine, thermo. It does two things: 1) it completes the saturation adjustment at "n+1" time level ("condensation/evaporation" loop at the top of thermo). After that, all thermodynamic fields are already updated to the "n+1" time level because precipitation processes are calculated to the first order in time. So the new tendencies for slow (precipitation) processes can then be calculated (the loop "compute moist forces update"; loop 300 in thermo). And they are multiplied by 2 so they can be included in the same way as other forces (i.e., …+dt/2*force) before advection in the next timestep.

With new thermodynamic variables updated in thermo, new buoyancy is calculated in a loop below "add buoyancy". Note that previous time level buoyancy (as well as the pressure gradient force) have been already included before advection, so the buoyancy tendency is calculated by dt/2 for the centered in time integration (the loop below "apply").

At this stage, the only thing left is to calculate the new pressure to ensure that final velocities are divergence-free [i.e., div(rho_0 u)=0 to be exact]. This is done in calls to gcrk_2 and pforc_2. The fx, fy, and fz variables on exit from pforc_2 contain adjusted velocities. Having updated velocities at "n+1" time level and saved advected velocities that include forces from the "n" time levels allows calculation of the new velocity forces in the loop below "calculate velocity forces".

At this stage, all model variables are at "n+1" time level, calculation clock is advanced by dt, and the time step is completed.

Saving the data and the graphics

At the moment, the data from the simulation (3D snapshots) are written to an unformatted tape (fort.17). In some versions/simulations, surface precipitation is also written into another tape – it is averaged in time using every time step data and written out as the average over past number of time steps.

There is also a simple graphics output prepared using NCAR Graphics ("call plot_1" and "call plot_2"). If you prefer to use some other way of visualizing the results, you may simply save the data in a different way (say, using the netCDF format) and apply different graphics software (say, ncview for a quick look at the data). The 2D version of the code has netcdf output added thanks to the help of ICTP's Fabien Solmon (fsolmon@ictp.it)

.

Time to look at and run the code…