Zentralanstalt für Meteorologie und Geodynamik

Documentation on:
Condensation sources and sedimentation

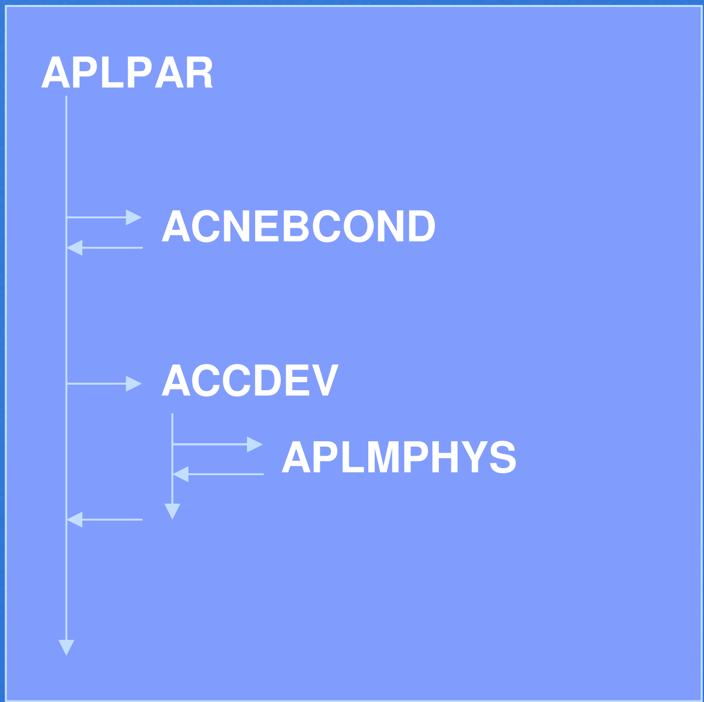
Christoph Wittmann

Overview

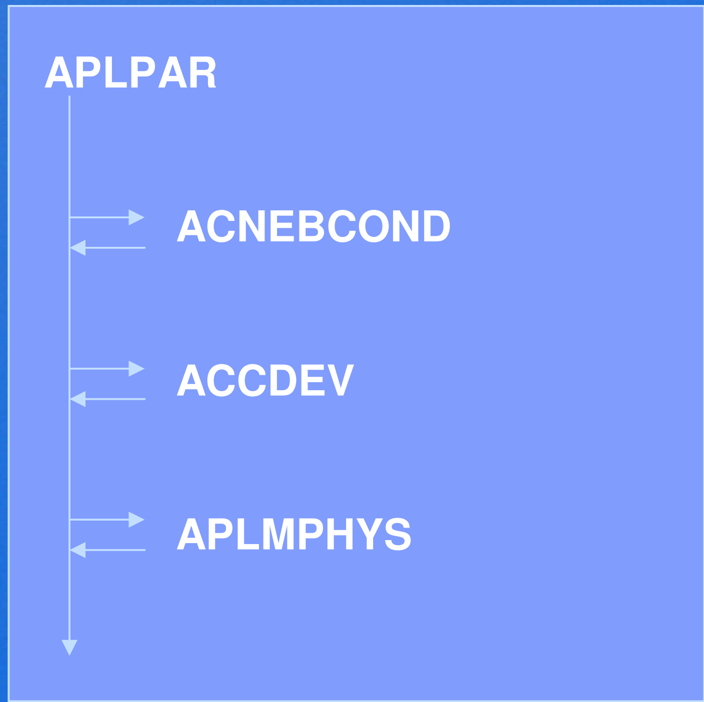
- ACNEBCOND: computation of cloudiness and critical relative humidity for the resolved condensation processes
- ACCDEV: computation of stratiform condensation/ evaporation fluxes
- APLMPHYS: computation of precipitation fluxes and linked pseudo-fluxes (autoconversion, evaporation/melting, condensation)

CALL SEQUENCE

LSTRAPRO=.TRUE.



L3MT=.TRUE.



ACNEBCOND I

computation of cloudiness and critical relative humidity for the resolved condensation processes

INCOMING ARRAYS

i.a.

PQ	q_v
PQW	q_w
PQI	q_i
PQL	q_l
PHUC	RH_{crit}

OUTGOING ARRAYS

PNEBCOND	n
PHCRICS	RH_{crit}
PQSAT	q_{sat}
PRMF	α_i

used modules: YOMPHY, YOMPHY0, YOMPHY2, YOMCST

I. Preliminary computation of ice fraction α_i (PRMF), FONICE

$$\alpha_i = \begin{cases} 1 - \exp\left\{ -\left(\frac{T - T_f}{2\Delta T} \right)^2 \right\} & \text{for} \quad T < T_f \\ 0 & \text{for} \quad T \geq T_f \end{cases}$$

II. Critical relative humidity RH_{crit} (PHCRICS)

LXRCDEV=.TRUE.

ACPLUIE_PROG formula:

Modification of PHUC according to model-mesh size using characteristic space scales for ice and liquid water

LSMGCDEV=.TRUE.

Formula after Lopez:

PHCRICS=f (RH_{Cmax}, RH_{Cmin}, PVETAF)

ACNEBCOND III

III. Cloud cover n (PNEBCOND) and saturation specific humidity q_{sat} (PQSATS)

LXRCDEV=.TRUE.

modified Xu-Randall scheme

LSMGCDEV=.TRUE.

Smith-Gerard scheme

LXRCDEV=.TRUE.

Xu-Randall typed relation

$$n = \left(\frac{q_v}{q_w}\right)^r \frac{\alpha q_c}{\alpha q_c + (q_w - q_v)^\delta},$$

,fullfilling

$$q_v = q_w(RH_c(1-n)+n) \quad \text{and} \quad q_c = q_t - q_v,$$

function x(s):

$$x_{ref} = x(s) = \underbrace{d_c}_{\text{ZDC}} \overbrace{(1-\frac{1}{s})}^{\text{ZNI}} \left(1 + \frac{A}{\underbrace{d_c s^\delta}_{\text{ZFAC}}} \overbrace{\frac{1}{\underbrace{(1-\frac{d_c}{s})^r}_{\text{ZNPI}} - (1-\frac{1}{s})}}^{\text{ZDNII}}\right)$$

for

$$x_{ref} = \frac{q_t}{q_w} - RH_c.$$

The root of x(s) is found in a Newton-loop with 3 iterations (ITER) with control variable s (ZSITER):

$$\overbrace{s^{n+1}}^{\text{ZSITER}} = \overbrace{s^n}^{\text{ZSITER}} + \frac{1}{2}\left(\frac{\overbrace{x_{ref}}^{\text{ZX0}} - \overbrace{x(s^n)}^{\text{ZXN}}}{\underbrace{\frac{\partial x(s^n)}{\partial s}}_{\text{ZDXN}}}\right)^2 \frac{1}{s^n}$$

→

$$n = 1 - \frac{1}{\text{ZSITER}} \quad \text{and} \quad q_{sat} = q_w$$

LSMGCDEV=.TRUE.

- PQSATS (using thermodynamical functions FOEW, FOQS)
- NSMTBOT=0: recompute PRMF for lowest level (JLEV=KLEV) by using average of T and surface temperature Ts as input for FONICE
- LSMTPS=.TRUE. : smoothing of PQSATS for layers containing liquid condensate and in the case of dry adiabatic unstable profile; NSMTPB and NSMTPA defining numbers of levels below/above actual level
- PNEBCOND:

$$n = 1 - \frac{1}{2}\left(1 + \frac{\frac{q_t}{q_{sat}} - 1}{1 - RH_c}\right)^2 \quad \text{for} \quad q_t \geq q_{sat}$$

$$n = \frac{1}{2}\left(1 - \frac{\frac{q_t}{q_{sat}} - 1}{1 - RH_c}\right)^2 \quad \text{for} \quad q_t < q_{sat}$$

- Smoothing of cloud-cover profile:
  - NSMDNEB=1: averaging of cloud cover n via (central weighted) running mean
  - NSMDNEB=2: allow maximum gradient of cloud cover between two levels

ACCDEV I

Computation of resolved condensation flux (and/or precipitation fluxes + pseudo-fluxes)

INCOMING ARRAYS

i.a.

PQ	q_v
PQW	q_w
PQI	q_i
PQL	q_l
PHCRICS	RH_{crit}
PNEBCOND	n
PQSATS	q_{sat}
PMRF	α_i
PQN	q_s
PQL	q_r

OUTGOING ARRAYS

PFCSQL	P_{lc}
PFCSQN	P_{sc}
PFPLSL	P_l
PFPLSN	P_s
PFASL	P_{la}
PFASN	P_{sa}
PFESL	P_{le}
PFESN	P_{se}

used modules: YOMPHY, YOMPHY0, YOMPHY2, YOMCST

I. Initialzing condensation fluxes (minus evaporation) in highest level

II. Computation of latent heat with FOLH: L_v, L_s

III. Computation of condensation fluxes (PFCSQL,PFCSQN)

LXRCDEV=.TRUE.

ACPLUIE_PROG scheme

LSMGCDEV=.TRUE.

Smith-Gerard scheme

LXRCDEV=.TRUE.

$$P_{lc[jlev]} = P_{lc[jlev-1]} + \frac{\Delta p}{g\Delta t}\text{ZCONL}$$

$$P_{sc[jlev]} = P_{sc[jlev-1]} + \frac{\Delta p}{g\Delta t}\text{ZCONI}$$

ZCONL liquid condensation amount

ZCONI solid condensation amount

$q_t > q_v^*$: super-saturation

$$\begin{aligned}\text{ZCONL} &= (1-\alpha_i)\text{ZDQVN} \\ \text{ZCONI} &= \alpha_i\text{ZDQVN}\end{aligned}$$

$$\text{ZDQVN} = q_v - q_v^*$$

$$q_v^* = q_w\left(RH_c(1-n)+n\right)$$

$q_t <= q_v^*$: under-saturation

$$\begin{aligned}\text{ZCONL} &= -q_l \\ \text{ZCONI} &= -q_i\end{aligned}$$

LSMGCDEV=.TRUE.

$$P_{lc[jlev]} = P_{lc[jlev-1]} + \frac{\Delta p}{g\Delta t}\text{ZDQL}$$

$$P_{sc[jlev]} = P_{sc[jlev-1]} + \frac{\Delta p}{g\Delta t}\text{ZDQI}$$

ZDQL liquid condensation amount

ZDQI solid condensation amount

$$\text{ZDQL} = q_l^* - q_l \quad \text{and} \quad \text{ZDQI} = q_i^* - q_i$$

$$q_l^* = (1-\alpha_i)q_c \quad \text{and} \quad q_i^* = q_c - q_l^*$$

$$q_c = q_c\sigma_s$$

σ_s standard deviation of distribution describing q_c

with q_c= f (RH,RHc)

LSTRAPRO=.TRUE.

- call of APLMPHYS inside ACCDEV (LSTRAPRO=.TRUE.)

  need for updated q_l, q_s and q_v according to "actual" condensation/evaporation rates to pass them to APLMPHYS
- Precipitation fluxes are returned from APLMPHYS for LSTRAPRO=.TRUE.
- L3MT=.TRUE. : precipitation fluxes remain unaffected

APLMPHYS I

Computation of precipitation fluxes and connected pseudo-fluxes

INCOMING ARRAYS i.a.	
PQMP	q_v
PQIMP	q_i
PQLMP	q_l
PQRMP	q_r
PQNMP	q_s
PHCRICS	RH_{crit}
PNEBM	n
PQSATS	q_{sat}
PDQ	Δq
PLHV	L_v
PLHS	L_s

OUTGOING ARRAYS	
PFPLSL	P_l
PFPLSN	P_s
PFCSQL	P_{lc}
PFCSQN	P_{sc}
PFASL	P_{la}
PFASN	P_{sa}
PFESL	P_{le}
PFESN	P_{se}
PFALLR	ω_r
PFALLS	ω_s

used modules: YOMPHY, YOMPHY0, YOMPHY2, YOMCST

I. Initialization of fluxes and pseudo fluxes at the top

vertical loop: JLEV=KTDIA,KLEV

II. Local copies for actual layer, temperature dependencies

III. Preparation for sedimentation (fall speed, PDFs,...)

IV. Autoconversion (ACACON)

V. Collection (ACCOLL), called 2 times

VI. Evaporation and melting (ACEVMEL), called 2 times

VII. Final computation of fluxes for actual layer

VIII. Prepare fluxes at the top of the next layer (LRNUMX)

II. Local copies for actual layer, temperature dependencies

Temperature dependency of

- autoconversion process (time scale, critical thresholds)
- fall speed of snow (or fall speed ratio snow/rain)
- collection efficiencies

based on exponential function

$$\mathrm{ZEXPN} = \min\left(1, \exp(c_t(T - T_f)\right)$$

With c_t being geometrical average of Lopez values (differing c_t's for different processes)

III. Preparation for sedimentation (fall speed, PDFs,...)

Fall speed of liquid and solid species:

$$\omega_l = \Omega^r \left(\frac{P'_l}{\rho^4}\right)^{\frac{1}{6}} \qquad \omega_s = \Omega^r \left(\frac{P'_s}{\rho^4}\right)^{\frac{1}{6}} \frac{1}{\text{ZEVGSL}}$$

ZEVGSL: fall speed ratio snow/rain

Probabilities P (differing for liquid and solid species) to reach the bottom of actual layer during actual time step for:

- precipitating species already present in the layer : P1(Z)
- precipitating species falling from the layer above : P2(Z)
- Precipitating species produced locally (autoconv., ...) : P3(Z)

P1, P2, P3 refer on basic probability P0(Z) to reach the bottom of actual layer, with Z

$$Z_l = \max\left(\epsilon, \frac{\Delta p}{\rho g \Delta t}\frac{1}{\omega_l}\right)$$

IV. Autoconversion (ACACON)

CALL ACACON $\longrightarrow$ conversion rates ZACORL, ZACONI, ZACONL

Update of precipitation fluxes (including graupel) using probabilities:

$$P_l = P_{l[jlev-1]} P_2^l + \frac{\Delta p}{g \Delta t} (P_1^l q_r + P_3^l \mathsf{ZACORL}'),$$

$$P_n = P_{n[jlev-1]} P_2^s + \frac{\Delta p}{g \Delta t} (P_1^s q_s + P_3^s \mathsf{ZACONI}'),$$

$$P_g = P_{g[jlev-1]} P_2^l + \frac{\Delta p}{g \Delta t} (P_1^l q_s + P_3^l \mathsf{ZACONL}').$$

V. Collection (ACCOLL), called 2 times

Call of ACCOLL is done twice:

1. seeded part
2. non-seeded part

Combined collection rates: ZCOLNL, ZCOLNI, ZCOLRL, ZCOLRI

Update of precipitation fluxes (including graupel)

$$P_l = P_l + \frac{\Delta p}{g \Delta t} P_3^l (\mathrm{ZCOLRL} + \mathrm{ZCOLRI}),$$

$$P_s = P_s + \frac{\Delta p}{g \Delta t} P_3^s (\mathrm{ZCOLNL} + \mathrm{ZCOLNI}),$$

$$P_g = P_g + \frac{\Delta p}{g \Delta t} P_3^l (\mathrm{ZZCOLN}),$$

VI. Evaporation and melting (ACEVMEL), called 2 times

Call of ACEVMEL is done twice

- for evaporation/sublimation of precipitation in the “clear sky part”
- for melting/freezing of precipitation species in the “cloudy part”

Resulting evaporation/melting increments: ZEVAR, ZEVAN, ZFONT

Update of pecipitation fluxes:

$$P_l = \max\left(0, P_l - \frac{\Delta p}{g\Delta t} P_3^l(\text{ZEVAR} - \text{ZFONT})\right),$$

$$P_s = max\left(0, P_s - \frac{\Delta p}{g\Delta t} P_3^s(\text{ZEVAN} + \text{ZFONT})\right),$$

VII. Final computation of fluxes for actual layer

Final computations of fluxes (and pseudo-fluxes) for

- condensation: P_{lc} (PFCSQL) and P_{sc} (PFCSQN)
- autoconversion: P_{la} (PFASL) and P_{sa} (PFASN)
- evaporation: P_{le} (PFESL) and P_{se} (PFESN)
- precipitation: P_{l} (PFPLSL) and P_{s} (PFPLSN)
- graupel: P_{g} (ZFSGRPL)

Graupel-effect

- graupel flux exists just locally in the routine
- no direct effect on any prognostic quantitiy

Influence on:

- fall speed ratio snow/rain ZEVGSL (graupel is falling faster than snow)
- Collection efficiencies

- the sedimentation probabilities used for graupel are the ones for the liquid phase

Recomputation of fall speed

- Initial (precipitation dependent) fall speeds computed at the beginning of the sedimentation part are based on the precipitation fluxes from the layer above

→ recomputation of fall speeds (PFALLL, PFALLN) using actual fluxes (corrected according to autoconversion, ...) is done:

$$\omega_l = \Omega^r \left(\frac{P_l'}{\rho^4}\right)^{\frac{1}{6}}$$

$$\omega_s = \Omega^r \left(\frac{P_s'}{\rho^4}\right)^{\frac{1}{6}} \frac{1}{\text{ZEVGSL}}$$

Preparations for the next layer

VIII. Prepare fluxes at the top of the next layer (LRNUMX)

LRNUMX=.TRUE.:
Maximum overlap of adjacent Clouds and random positioning of cloudy air seperated by clear air

"Cloud" is devided into:

- seeded part
- non-seeded part
- clear air covered with precipitation

Computation of proportions (cloud/clear) and connected fluxes

LRNUMX=.FALSE.
Random cloud overlap

- proportions and conected fluxes become equal