Toulouse 01.11.2004 - 18.12.2004

MesoNH - Arome Upper Air Physics

Scientific supervisor: Sylvie Mallardel

Martina Tudor tudor@cirus.dhz.hr

Croatian HydroMeteorological Service

Zagreb

Contents

1 Numerical stability

1.1 Stiffness

In a meteorological model, we deal with a system of coupled equations that model procesess occuring at different rates. Such systems are stiff. They yield two sollutions, of which one is of a meteorological significance - giving slow (when compared to the time-step) evolution of an atmospheric state, and another solution that varies rapidly, causing oscillations around the slow solution. The best way to deal with a system of stiff equations is to solve them implicitly. However, as this is not possible for a non-linear set of equations in a meteorological model, other sollutions have to be sought. One sollution is to use a predictor-corrector scheme.

Although, one might think that such oscillations are acceptable, they produce a wrong forecast. One of the possibilities is an exaggerated rainfall (this example refers to Aladin, in any other model, consequences might be significantly different). The model gives a lot of rainfall in one time-step and very little or none in the next one. Accumulated over a period of time, the modelled amount of rainfall is exaggerated.

It was also noticed that these oscillations occur only in certain meteorological conditions, making the stability of the scheme situation dependant.

Analysis of the system of differential equations may reveal such weaknesess as stiffness, but it often relies on an analytic function approximating the solution and requires simplifications and assumptions in the equation system afterwards and it is usually solved on an uniformly spaced grid.

Such analysis of the whole system without any additional simplifications, assumptions and so forth on an unequally spaced grid is not only impractical, but impossible for a sophisticated system of equaitons.

It is possible to test a subset of equations, for example one parameterisation by one, to reveal if any of them might suffer from stiffness. The rapidly varying term is the one guilty for the oscillations. The idea is this term more active and, therefore, reveal itself.

If the timestep is reduced in a part of the model (one parameterisation) this part will produce some fluxes (if it is written in a flux form) different than in an original model (when the same time-step was used in the whole model). These fluxes are combined to produce a tendency using the time-step from the rest of the model. Such a treatement causes an amplification of the oscillations if the considered part of the model has a problem with stiffness. This method allows us to test a certain parameterisation as it interacts with the rest of the model (dynamics, horisontal diffusion, other parameterisations) on any grid in a real meteorological situation.

The set of parameterisations in the Arome model has been tested. The amplitude of the $2\Delta t$ oscillations is calculated from the fields of model variables on model levels in three consecutive time-steps as

$$\frac{1}{2}\left(\psi^{t+\Delta t}+\psi^{t-\Delta t}-2\psi^{t}\right)$$

for one level and for one model variable. As the temperature is affected by all of the cosidered parameterisations, only oscillations of the temperature will be shown. One should keep in mind that other model variables exhibit similar oscillation behaviour in the same areas. The oscillations are not confined close to the ground and stretch troughout the atmosphere.

Without predictor-corrector scheme, the sollution gives huge oscillations (in a reference run, without the test). Applying predictor-corrector, the oscillations calm down, but do not dissappear. The amplitude of oscillations reaches 1K (for temperature) which is a lot considering that the time-step is only 60 seconds. I was not able to run the tests without predictor-corrector, the model blew up even when run without the test when convection was used. The tests performed using predictor-corrector did not reveal any amplification of the oscillations.

(Tests) without predictor-corrector:

- Reference run of Arome blew up after 23 steps with 60 sec with nonhydrostatic dynamics and convection but without predictor-corrector.
- Stifness test on adjustment (without convection) non-hydrostatic, without predictor-corrector blew up in 58th time-step.
- Stifness test on turbulence (without convection) non-hydrostatic, without predictor-corrector blew up in 463th time-step.
- Stifness test on microphysics (without convection) non-hydrostatic, without predictor-corrector blew up in 57th time-step.

Tests with predictor-corrector:

adjustment - the time-step is set to half of its value prior to the call to ac_adjust and reset to its original value afterwards, also, all the variables containing Δt (prognostic values of model variables) are rescaled prior to the call to ac_adjust and reset to its original value accordingly.

$$\psi_{t+\Delta t} = \psi_{t+\Delta t} * 2 \qquad \mathit{and} \qquad \psi_{t+\Delta t} = \psi_{t+\Delta t} * \frac{1}{2}$$

for $\psi = \theta, r_v, r_c, r_r, r_i, r_s, r_g$.

turbulence - the time-step is set to half of its value prior to the call to ac_turb_mnh and reset to its original value afterwards, also, all the variables containing Δt (prognostic values of model variables) are rescaled prior to the call to ac_turb_mnh and reset to its original value accordingly.

$$\psi_{t+\Delta t} = \psi_{t+\Delta t} * 2 \qquad \mathit{and} \qquad \psi_{t+\Delta t} = \psi_{t+\Delta t} * \frac{1}{2}$$

for $\psi = u, v, w, TKE, \theta, r_v, r_c, r_r, r_i, r_s, r_g$.

microphysics - the time-step is set to half of its value prior to the call to ac_rain_ice and reset to its original value afterwards, also, all the variables containing Δt (prognostic values of model variables) are rescaled prior to the call to ac_rain_ice and reset to its original value accordingly.

$$\psi_{t+\Delta t} = \psi_{t+\Delta t} * 2 \qquad \mathit{and} \qquad \psi_{t+\Delta t} = \psi_{t+\Delta t} * \frac{1}{2}$$

for $\psi = u, v, w, TKE, \theta, r_v, r_c, r_r, r_i, r_s, r_g$ and accuulated precipitation.

1.2 Non-linear instability

Considering a problem in 1D in the vertical, non-linear instability occurs when a jump in the speed of propagation of the signal from one level to the next breaks the CFL condition. We may consider fluxes as vectors of propagation of the signal. Although this kind of instability is very rare, it is very unpleasant if the model blows up during the operational forecast. Non-linear instability occurs when

$$(M_{l+1} - M_l)\,\Delta t > \Delta p$$

where M_l and M_{l+1} represent mass fluxes through adjacent layers in the atmosphere. If the difference between the fluxes is too large - instability occurs. A parameterisation can be tested to non-linear instability by modifying local Δp in it. If the considered scheme is prone to the non-linear instability, the model should show signs of it in such test (eg. blow-up).

Tests with predictor-corrector:

turbulence - $\Delta z_{\tilde{l}}$ and $\rho_d J$ are reduced in ac_turb_mnh prior to the turbulence call and rescaled afterwards.

turbulence - $\Delta z_{\tilde{l}}$ and $\rho_d J$ are increased in ac_turb_mnh prior to the turbulence call and rescaled afterwards.

microphysics - Δz_l in ac_rain_ice and $\rho_d J$ in apl_arome are reduced prior to the turbulence call and rescaled afterwards.

microphysics - Δz_l in ac_rain_ice and $\rho_d J$ in apl_arome are increased prior to the turbulence call and rescaled afterwards.

1.3 Oscillations in Arome

As previously mentioned, there are oscillations in the reference run of Arome (different options). Since the time-step used in Arome is only 60 seconds, their amplitude of 1K (for temperature) is large enough to claim that these oscillations are significant. This is especially obvious when the application is run on lower resolution (5 and 7 km) with much longer time-steps. Since the amplification of the oscillations is the strongest for the non-hydrostatic run with convection without predictor-corrector (pure semi-implicit), there could be some interaction of the convection scheme with non-hydrostatic dynamics causing these oscillations.

The figures show $\frac{1}{2}\left(\psi^{t+\Delta t} + \psi^{t-\Delta t} - 2\psi^{t}\right)$ after 2 hours of integration. Some averaged value of this amplitude could be more valuable, but such output is not available.

The hydrostatic run on 2.5 km resolution has only some isolated points or small areas where the amplitude of oscillations could be significant. These may correspond to a grid-point storm if these oscillations persist for a longer period of time. The number of such areas increases when the convection scheme is on. The convection scheme can probably only cause problems on this resolution. The oscillations on the lowest model level seem to be well corellated with mountains. But, when we take a look higher in the atmosphere we may notice other problematic areas not so well connected to the areas of high mountains. I could only guess that there are more reasons than just one for these oscillations, one being connected to the orography and another more active higher in the atmosphere. This certainly does not limit the number of options, there could be more than two. These oscillations close to the surface seem unaffected by the predictor-corrector scheme (the figures ARE different).

It is expected that the amplitude of these oscillations would reduce if the model is run with much shorter time-step. This is true, the amplitude becomes lower as the time-step is reduced to 5 seconds. But, let's assume that the amplitude of the oscillations is reduced by the same factor as the time-step. This is not very crude if we imagine that the rate of change of a variable (tendency) calculated in one time-step does not depend on the length of the time-step. So, to make the amplitudes of the oscillations somparable between the runs with 60 and 5 seconds, the amplitudes of the second run were multiplied by 12. This reveals huge oscillations high in the atmosphere.

The model was also run on two lower resolutions, 5 and 7 km with timesteps of 120 and 300 seconds.

The hydrostatic runs do not show problematic spots. But do not be fooled, the isolines are drawn on different values. The values drawn at few points represent extremes and show that the amplitude extremes are lower than in 2.5 km run. When the model is run with non-hydrostatic dynamics, oscillation appear on the lowest model level that are mostly well correlated with mountains. These oscillations are significanly reduced by the predictor-corrector scheme. They remain unaffected by the 'bottom boundary condition' correction. This is not surprising since the BBC should deal with the problems on 1km resolution or less. All figures present the results of reference runs (without any manipulation of the time-step in any parameterisation).

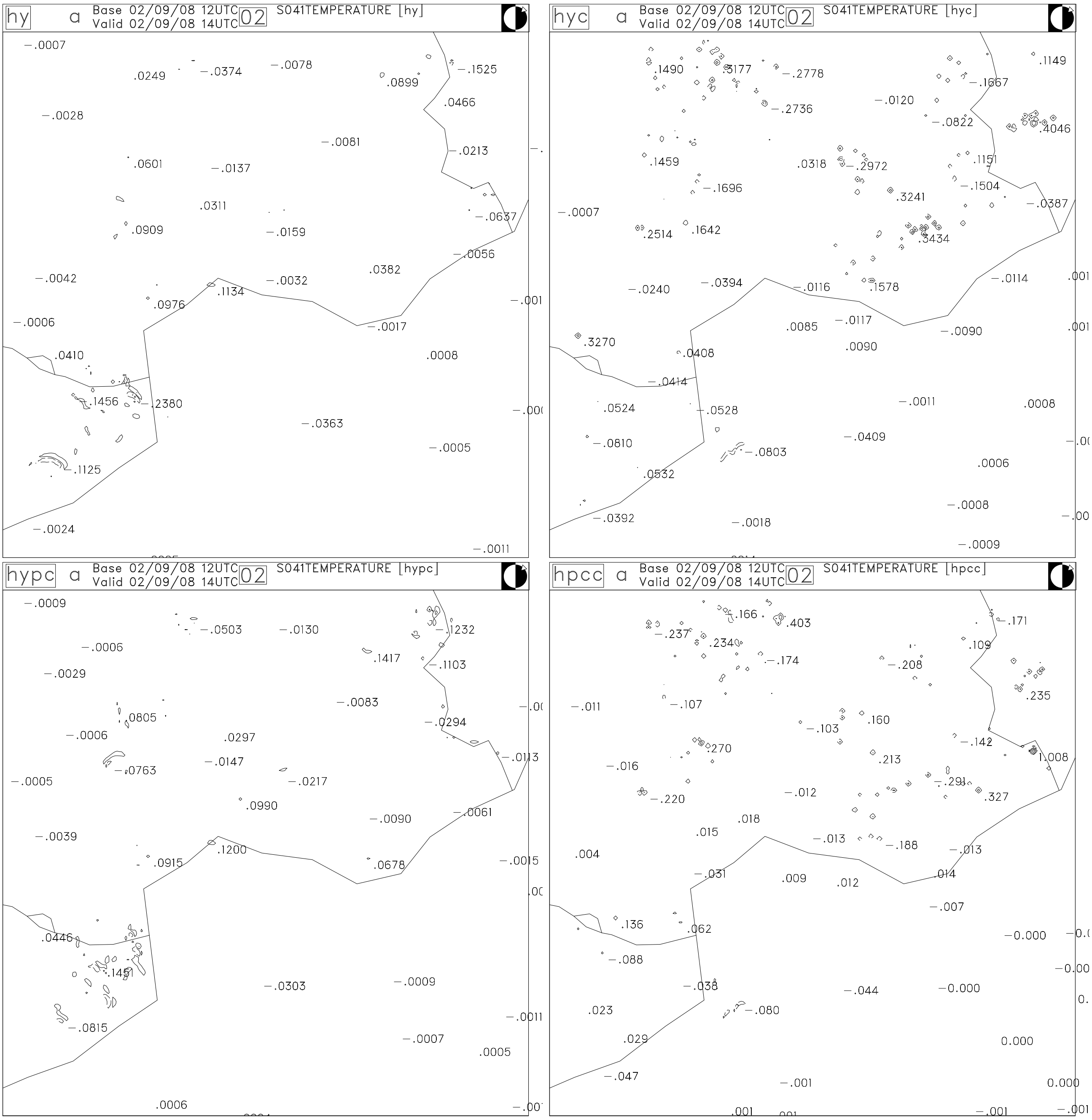


Figure 1: 2.5km hydrostatic, time-step 60 seconds, amplitude of the temperature oscillations after 2 hours of integration, pure semi-implicit (top row) and predictor-corrector (bottom row), without (left column) and with (right column) convection. The isolines are drawn for 0.05, 0.2, 0.5, 1 and 2K.

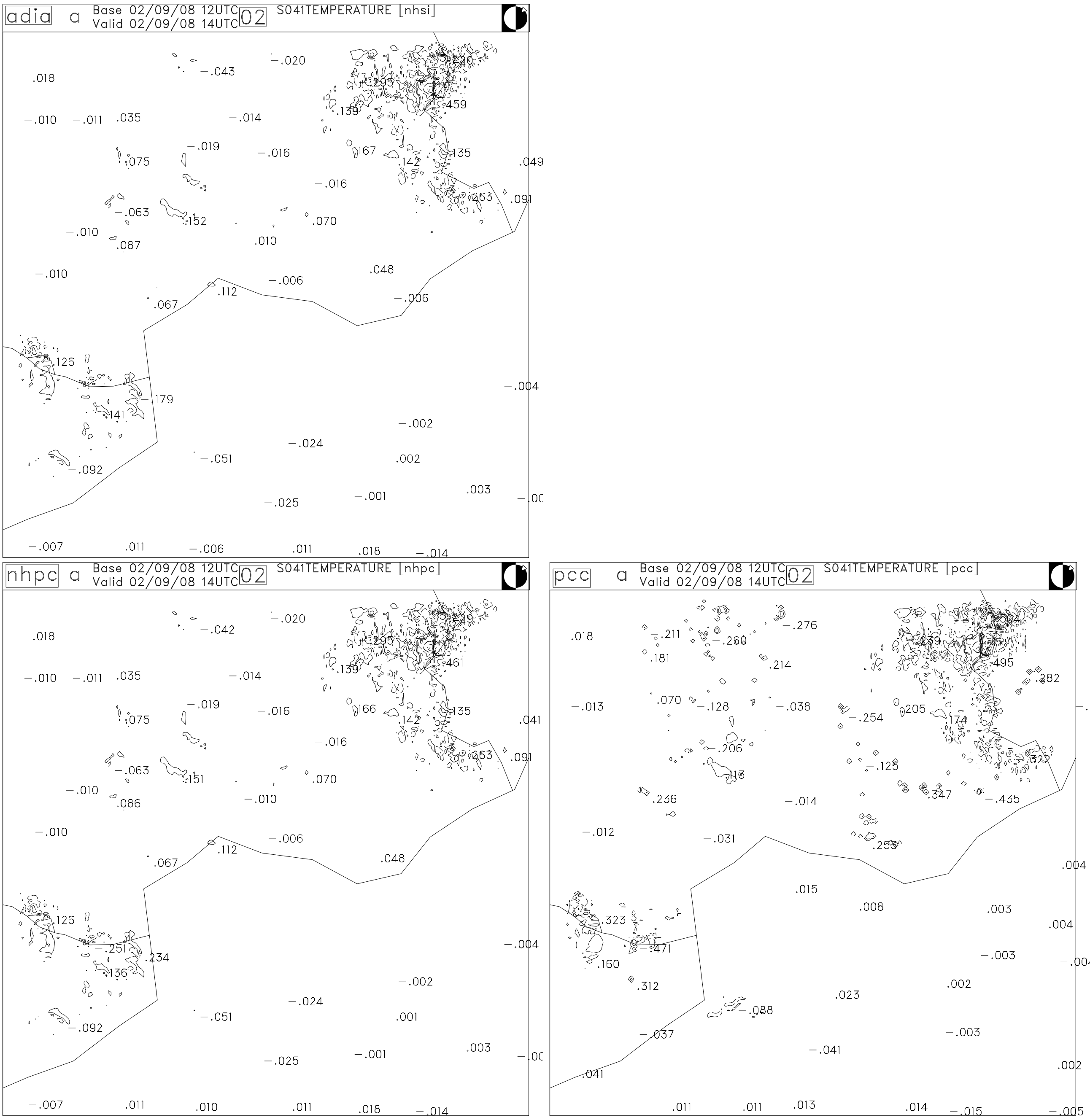


Figure 2: 2.5km nonhydrostatic, time-step 60 seconds, amplitude of the temperature oscillations after 2 hours of integration, pure semi-implicit (top row) and predictor-corrector (bottom row), without (left column) and with (right column) convection. The isolines are drawn for 0.05, 0.2, 0.5, 1 and 2K.

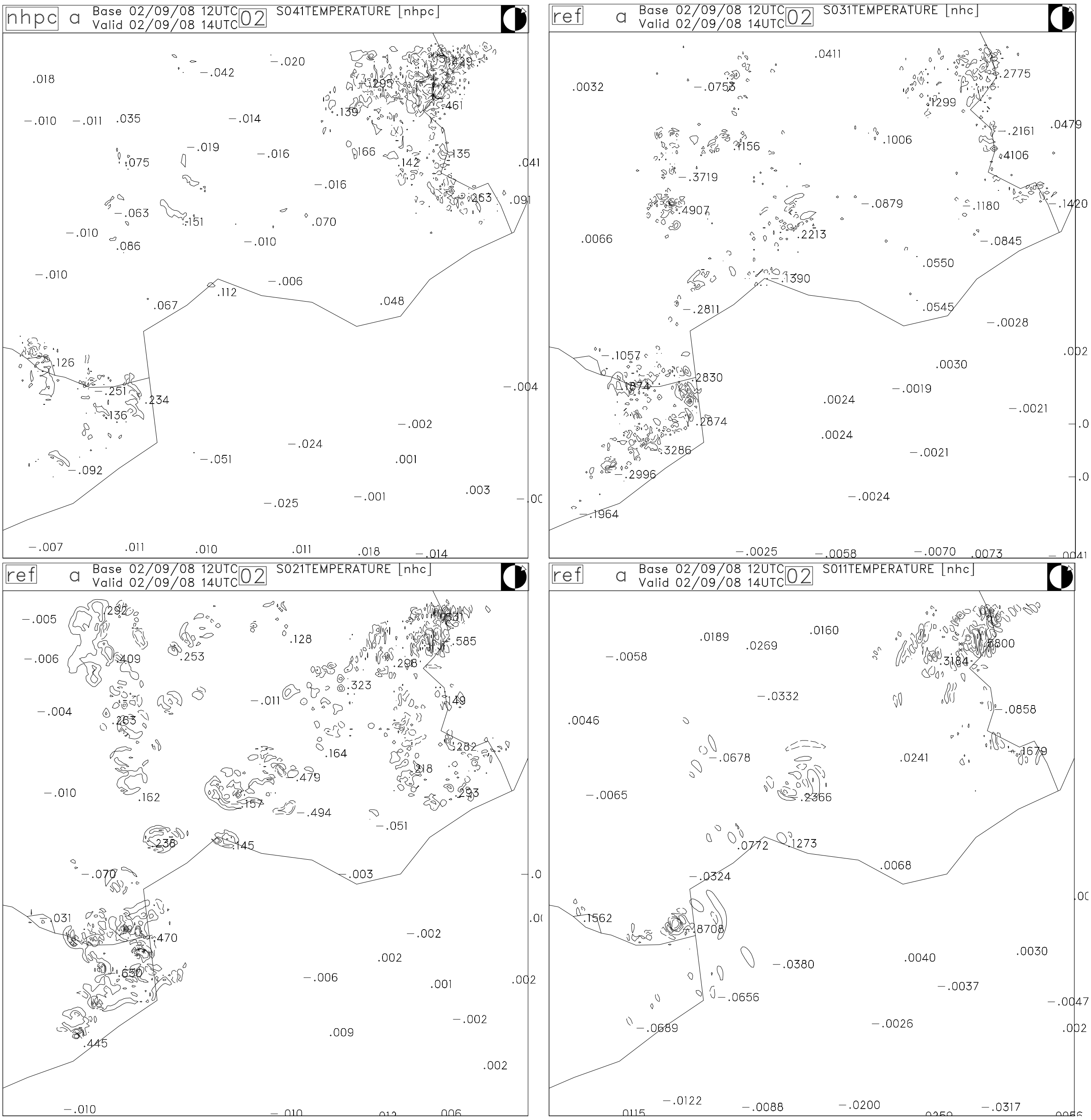

nhpc a Base 02/09/08 12UTC Valid 02/09/08 14UTC 02 S041TEMPERATURE [nhpc]
ref a Base 02/09/08 12UTC Valid 02/09/08 14UTC 02 S031TEMPERATURE [nhc]
ref a Base 02/09/08 12UTC Valid 02/09/08 14UTC 02 S021TEMPERATURE [nhc]
ref a Base 02/09/08 12UTC Valid 02/09/08 14UTC 02 S011TEMPERATURE [nhc]
.018
−.042
−.020
.299
.461
.139
.166
.142
.135
.263
.091
.041
.126
−.251
.234
.136
−.092
.4907
.3719
.2775
.2161
.4106
.2830
.2874
.3286
−.2996
−.1964
.292
.409
.585
.296
.323
.479
−.494
.470
.650
.445
.3184
.6800
.2366
.8708
.1562

Figure 3: 2.5km nonhydrostatic, time-step 60 seconds, amplitude of the temperature oscillations after 2 hours of integration, predictor-corrector, without convection, for the lowest model level no. 41 (top, left), level no. 31 (top, right), level 21 (bottom, left) and level 11 (bottom, right). The levels are approximately 17, 1600, 5600 and 13000 m above ground. The isolines are drawn for 0.05, 0.2, 0.5, 1 and 2K.

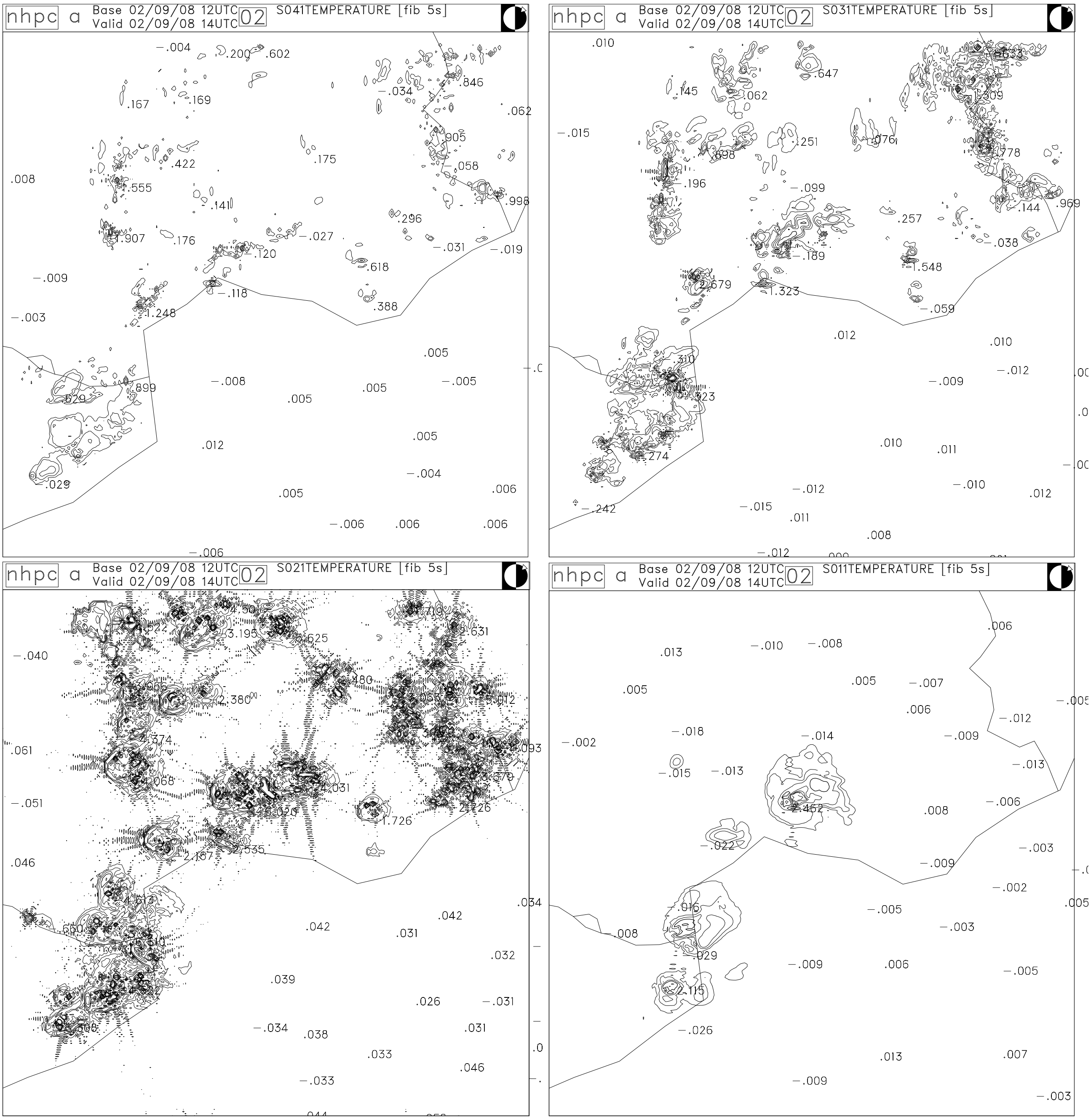


Figure 4: 2.5km nonhydrostatic, time-step 5 seconds, amplitude of the temperature oscillations after 2 hours of integration, predictor-corrector, without convection, for the lowest model level no. 41 (top, left), level no. 31 (top, right), level 21 (bottom, left) and level 11 (bottom, right). The amplitude is rescaled (multiplied by 12) to become comparable with the one for run with the 60 seconds time-step. The levels are approximately 17, 1600, 5600 and 13000 m above ground. The isolines are drawn for 0.05, 0.2, 0.5, 1 and 2K.

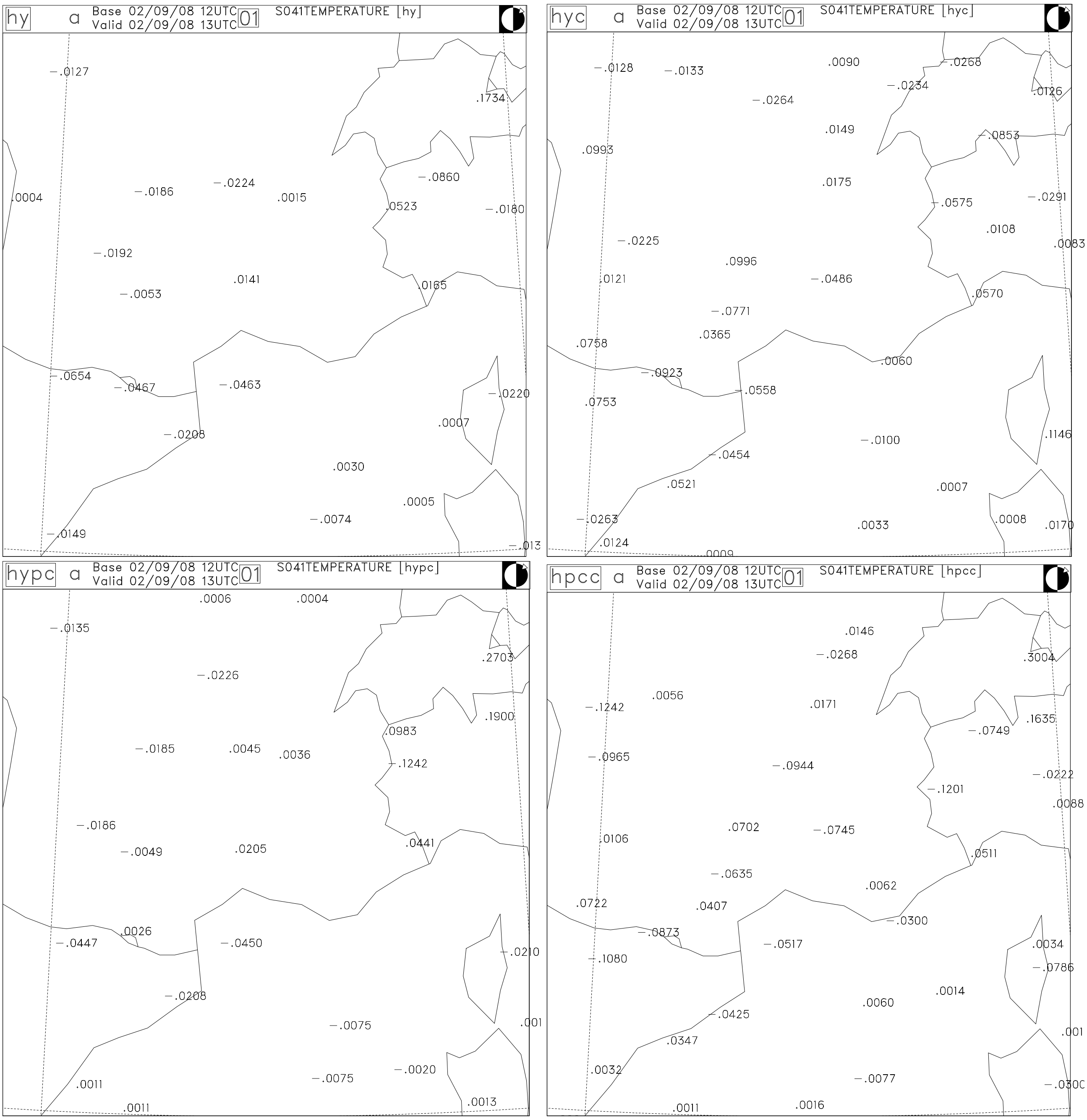


Figure 5: 5km hydrostatic, time-step 120 seconds, amplitude of the temperature oscillations after 2 hours of integration, pure semi-implicit (top row) and predictor-corrector (bottom row), without (left column) and with (right column) convection. The isolines are drawn for 1, 2.5, 5, 10 and 15 K.

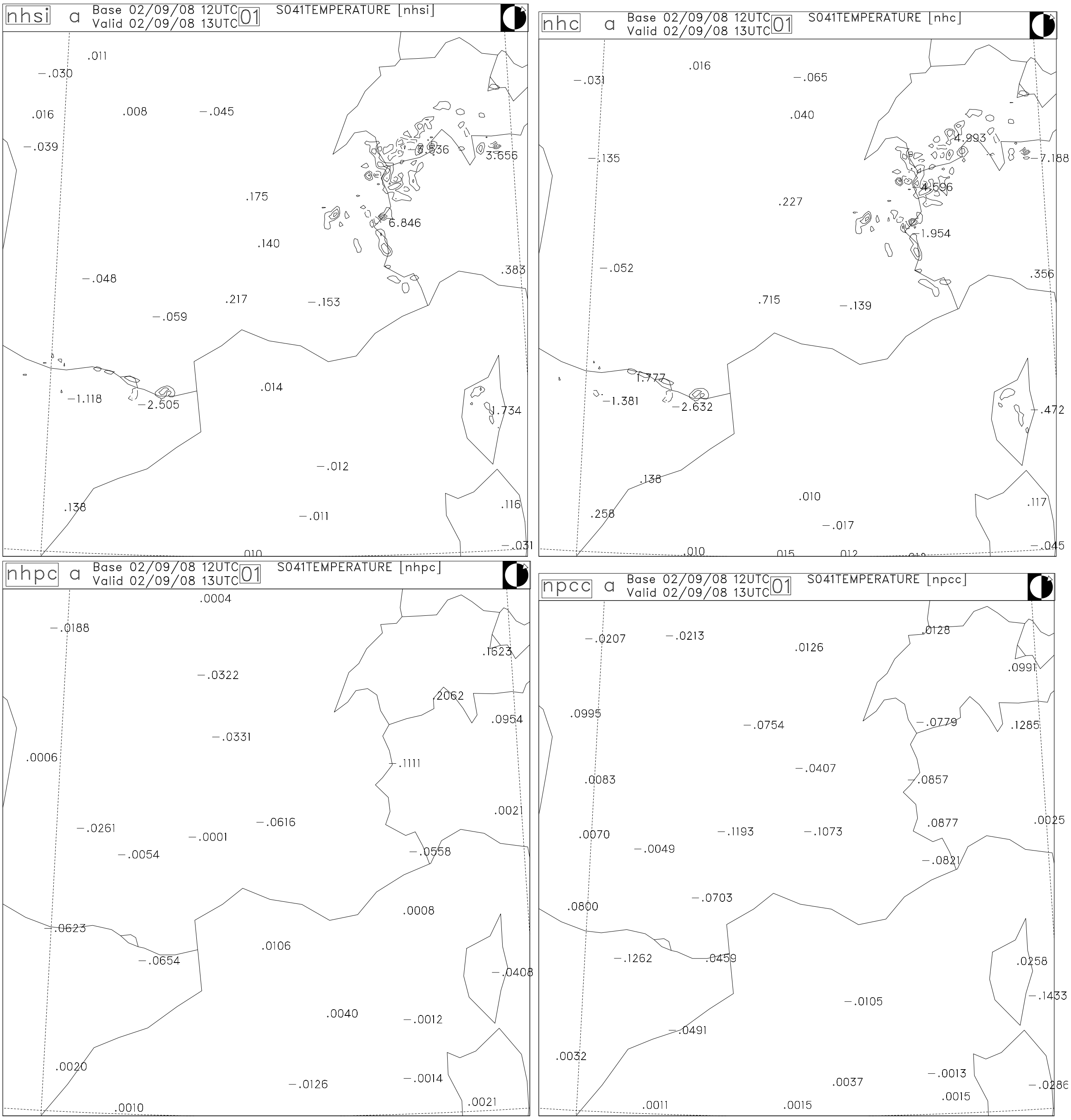


Figure 6: 5km nonhydrostatic, time-step 120 seconds, amplitude of the temperature oscillations after 2 hours of integration, pure semi-implicit (top row) and predictor-corrector (bottom row), without (left column) and with (right column) convection The isolines are drawn for 1, 2.5, 5, 10 and 15 K.

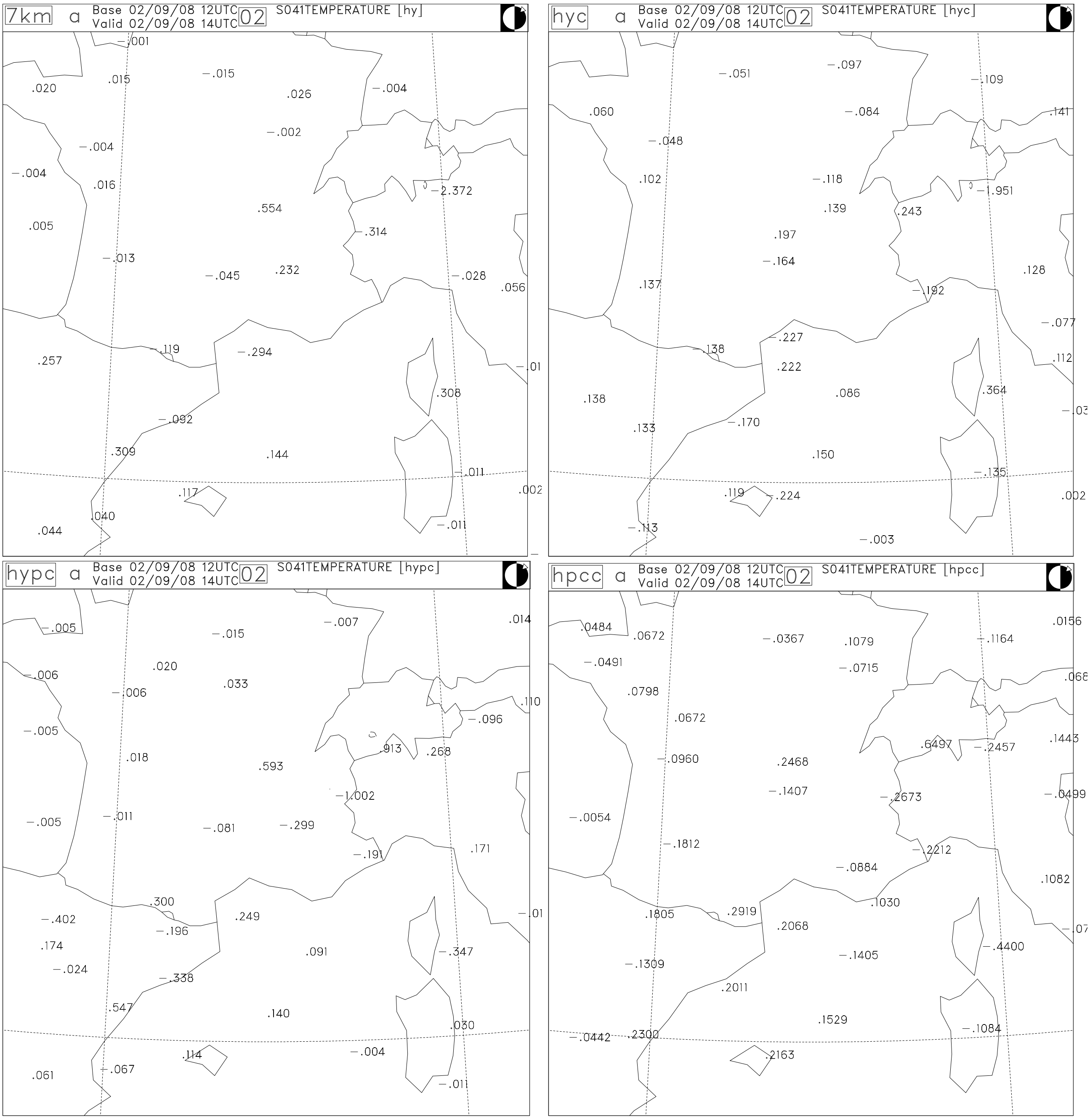


Figure 7: 7km hydrostatic, time-step 300 seconds, amplitude of the temperature oscillations after 2 hours of integration, pure semi-implicit (top row) and predictor-corrector (bottom row), without (left column) and with (right column) convection. The isolines are drawn for 1, 2.5, 5, 10 and 15 K.

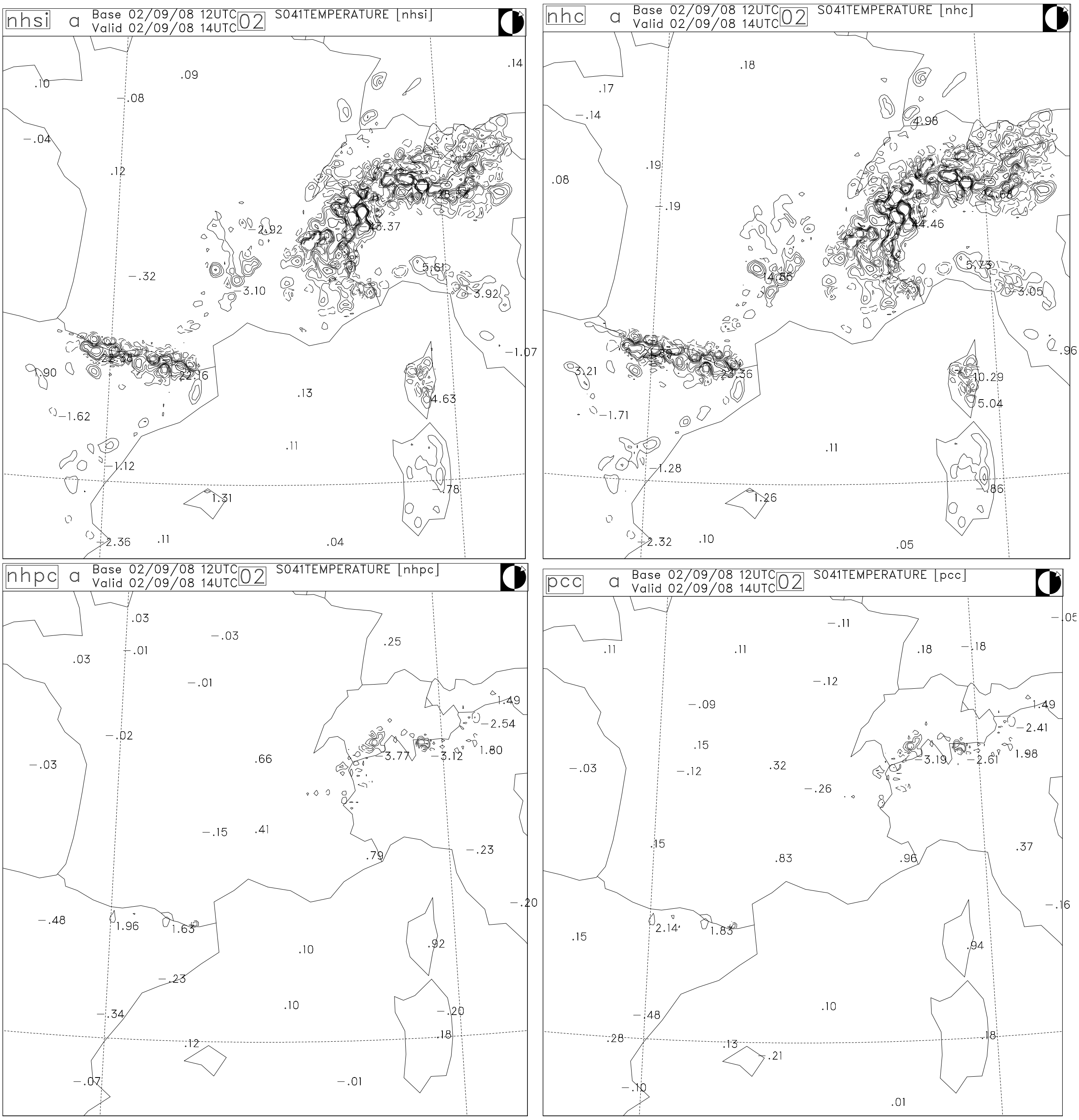


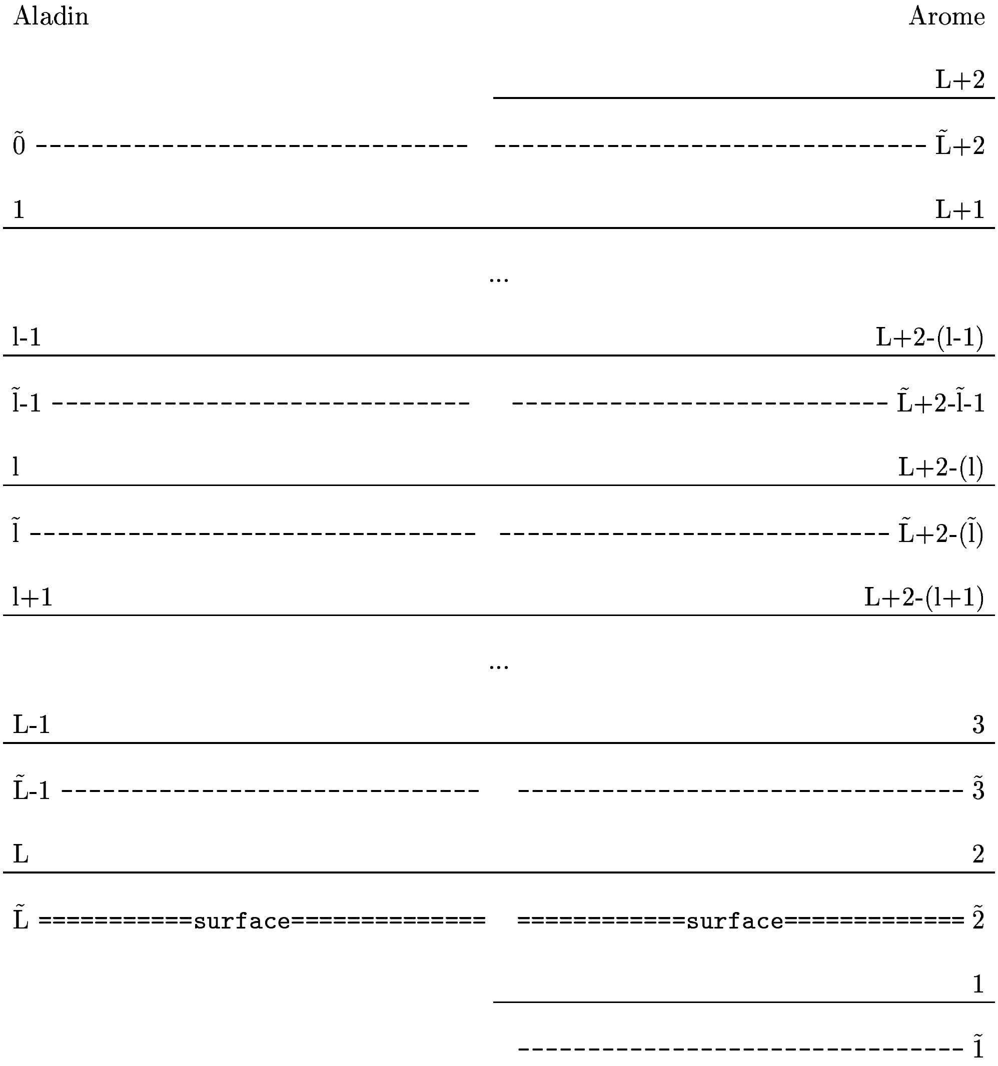
Figure 8: 7km nonhydrostatic, time-step 300 seconds, amplitude of the temperature oscillations after 2 hours of integration, pure semi-implicit (top row) and predictor-corrector (bottom row), without (left column) and with (right column) convection. The isolines are drawn for 1, 2.5, 5, 10 and 15 K.

2 Arome

This part of the document attempts to describe the upper-air physics of MesoNH that has been implemented into the model Arome with the addition of convection. Since there were plenty of unclear issues for the author, the document contains many questions and concerns. Inside physics the model variables are the three velocity components: u, v and w, potential temperature, turbulent kinetic energy, moist variables: water vapour, cloud water, cloud ice, rain, snow and graupel. Hydrostatic pressure and true pressure are also used. Vertical coordinate is z.

2.1 Levels in the vertical

If you come from the Aladin to Arome world you might find yourself upside-down. There are two more levels and one more level interface. The physics computations are done inside a vertical column (with an exception of 3D turbulence, but only 1D turbulence is used in Arome, so far) from bottom usually IKB to the top IKE. The additional levels contain model variables that are either equal to the values on the adjacent level or zero (for the moist variables).



The transformation of variables, including the level reversal, takes place in the apl_arome subroutine.

3 Upper-air physics call

The surface physics is called separately, outside cpg.

3.1 apl_arome calling tree

This is how the calling tree of apl_arome looks like. Some routines are ommited from the tree (eg. fmwrit) since they are not important. A few subroutines are described later in the document in more detail.

```
apl_arome
    ↪ ac_adjust
        ↪ ice_adjust
            ↪ condensation
    ↪ gprcp
    ↪ gpgeo
    ↪ actqsat
    ↪ radact
    ↪ suozon
    ↪ acradin
        ↪ recmwf
            ↪ radslw
                ↪ lw
                ↪ sw
        ↪ rfmr
            ↪ radslw15
                ↪ lw15
                ↪ sw15
    ↪ radheat
    ↪ bri2acconv
        ↪ ac_conv_mnh
            ↪ ini_convpar
            ↪ deep_convection
                ↪ convect_trigger_funct
                    ↪ convect_satmixratio
                ↪ convect_updraft
                    ↪ convect_condens
                    ↪ convect_mixing_funct
                ↪ convect_tstep_pref
                ↪ convect_downdraft
                    ↪ convect_satmixratio
                ↪ convect_precip_adjust
                ↪ convect_closure
                    ↪ convect_closure_thrvlcl
                    ↪ convect_satmixratio
                    ↪ convect_closure_adjust
                ↪ ch_convect_scavenging
                ↪ convect_chem_transport
            ↪ ini_convpar_e1
            ↪ deep_convection
            ↪ ini_convpar
            ↪ deep_convection
            ↪ deep_convection
            ↪ ini_convpar_shal
            ↪ shallow_convection
```

```
                ↪ convect_trigger_shal
                    ↪ convect_satmixratio
                ↪ convect_updraft_shal
                    ↪ convect_condens
                    ↪ convect_mixing_funct
                ↪ convect_closure_shal
                    ↪ convect_closure_thrvlcl
                    ↪ convect_satmixratio
                    ↪ convect_closure_adjust
                ↪ convect_chem_transport
↪ ac_turb_mnh
    ↪ turb
        ↪ bl89
        ↪ rmc01
        ↪ prandtl
        ↪ turb_ver
            ↪ turb_ver_thermo_flux
                ↪ tridiag
            ↪ turb_ver_dyn_flux
                ↪ tridiag_wind
            ↪ turb_ver_sv_flux
                ↪ tridiag
            ↪ turb_ver_sv_corr
        ↪ turb_hor_split - for 3D turbulence
        ↪ tke_eps_sources
            ↪ tridiag_tke or tridiag
            ↪ les_mean_subgrid
        ↪ second_mnh
        ↪ les_mean_subgrid
↪ ac_rain_ice
    ↪ rain_ice
        ↪ rain_ice_sedimentation
        ↪ rain_ice_nucleation
        ↪ rain_ice_slow
        ↪ rain_ice_warm
        ↪ rain_ice_fast_rs
        ↪ rain_ice_fast_rg
        ↪ rain_ice_fast_ri
```

3.2 Description of the subroutines in the calling tree

ac_adjust - computes the microphysical sources related to the resolved clouds and precipitation. Computes the real absolute pressure, negative values of the current guess of all mixing ratio are removed, sources are transformed in physical tendencies. After calling to microphysical routines, the physical tendencies are switched back to prognostic variables.

ice_adjust - computes the fast microphysical sources through a saturation ajustement procedure in case of mixed-phase clouds.

condensation - diagnose cloud fraction, liquid and ice condensate mixing ratios based on the large-scale fields of temperature, water vapor and possibly liquid and solid condensate, the conserved quantities r_t and h_l are constructed and then fractional cloudiness, liquid and solid condensate are diagnosed.

gprcp - Computes Cp, R and R/Cp from Q.

gpgeo - Computes half and full level geopotential height gz. Laplace relation writes: $\frac{d(gz)}{d\pi} = -\frac{RT}{p} = -\frac{RT}{\pi}\frac{\pi}{p}$ where: gz is the geopotential height. π is the hydrostatic pressure. p is the total pressure including non-hydrostatic effects. R is the air constant (including moisture effects). T is the temperature. Integrating the Laplace equations yields the following discretisation for gz.

- gz at interlayer $\tilde{l}$: $gz_{\tilde{l}} = gz_{surf} + \sum_{k=L}^{l+1} \frac{\pi_k}{p_k} R_k T_k \delta_k$
- gz at layer l: $gz_l = gz_{\tilde{l}} + \frac{\pi_l}{p_l} R_l T_l \alpha_l$

actqsat - computation of the saturation point and wet-bulb characteristics.

radact - computes distribution of aerosols and ozone.

suozon - initialization of the ozone vertical profile.

acradin - non-standard fortran buffer routine, it is called only every NRADFR time-steps.

recmwf - Meteo France radiation interface to the ECMWF radiation scheme.

radlsw - interface to the ECMWF long-wave and short-wave radiation schemes.

lw - computes long-wave radiation fluxes.

sw - computes short-wave radiation fluxes.

rfmr - non-standard fortran buffer routine.

radlsw15 - interface to the ECMWF long-wave and short-wave radiation schemes.

lw15 - computes long-wave radiation fluxes.

sw15 - computes short-wave radiation fluxes.

radheat - computes temperature changes due to radiation.

bri2acconv - Bridge to MNH convection call (deep and shallow).

ac_conv_mnh - is an interface routine to call the parametrisation of convection (deep and shallow) of the MNH physical package.

ini_convpar - initialize the constants stored in modules. Since ordinary convection, ensemble convection and shallow convection use the same parameters but with different values, the initialization procedure has to be called each time before the call to the corresponding subroutine.

deep_convection - determine the convective tendencies. The routine first prepares all necessary grid-scale variables. The final deep convection tendencies are then computed by calls of different subroutines. Convective columns in the model domain are selected by calling TRIGGER_FUNCT. Then, memory for the convection updraft and downdraft variables is allocated and the grid scale variables are gathered in convective arrays. The updraft and downdraft computations are done level by level starting at the bottom and top of the domain, respectively. All computations are done on MNH thermodynamic levels. The depth of the current model layer k is defined by $\Delta p_k = p_{k-1} - p_k$.

convect_trigger_funct - Determine convective columns as well as the cloudy values of θ,and q_v at the lifting condensation level (LCL). Computations are done at every model level starting from bottom. The use of masks allows optimisation of the inner loops (horizontal loops). What we look for is the undermost unstable level at each grid point. Determine highest necessary loop test layer, enter loop for convection test, we exit the trigger test when the center of the mixed layer is more than 3500 m above soil level. Construct a mixed layer of at least 60 hPa (XZPBL), use an empirical direct solution (Bolton formula) to determine temperature and pressure at LCL (Note: the adiabatic saturation temperature is not equal to the dewpoint temperature). Correct ZTLCL in order to be completely consistent with MNH saturation formula. If ZRVLCL = PRVMIX is oversaturated set humidity and temperature to saturation values. Determine vertical loop index at the LCL and DPL, estimate height and environmental θ_v at LCL, check to see if cloud is bouyant, compute grid scale vertical velocity perturbation term, compute parcel vertical velocity at LCL and look for parcel that produces sufficient cloud depth. The cloud top is estimated as the level where the CAPE is smaller than a given value (based on vertical velocity eq.).

convect_satmixratio - Compute vapor saturation mixing ratio over liquid water and to return values for L_v L_s and C_{ph}.

convect_updraft - determine updraft properties (mass flux, thermodynamics, precipitation) Computations are done at every model level starting from bottom. The use of masks allows to optimise the inner loops (horizontal loops). Compute undilute updraft θ_e for CAPE computations Bolton (1980) formula. define accurate enthalpy for updraft, set updraft properties between DPL and LCL, estimate condensate, L_v L_i, C_{ph} and θ_v at level k+1, compute square of vertical velocity using entrainment at level k, update total precipitation: $dr_r = (r_c + r_i) * exp(-rate * dz)$ update r_c, r_i, enthalpy, r_w for precipitation. Compute entrainment and detrainment using conservative variables adjusted for precipitation (not for entrainment). Compute critical mixed fraction by estimating unknown $T^{mix} r_c^{mix}$ and r_i^{mix} from enthalpy h^{mix} and r_w^{mix}. We determine the zero crossing of the linear curve evaluating the derivative using ZMIXF=0.1. Compute final midlevel values for entr. and detrainment after call of distribution function. If the calculated detrained mass flux is greater than the total updraft mass flux, or vertical velocity is negative, all cloud mass detrains at previous model level, exit updraft calculations - CTL is attained. Compute CAPE for undilute ascent using θ_e and θ_{es} instead of θ_v. This estimation produces a significantly larger value for CAPE than the actual one. Compute final values of updraft mass flux, enthalpy, r_w at level k+1. Adjust mass flux profiles, detrainment rates, and precipitation fallout rates to reflect linear decrease in mass flux between the ETL and CTL. Set mass flux and entrainment in the source layer, linear increase throughout the source layer, if cloud thickness is smaller than 3 km, no convection is allowed (Note: For technical reasons, we stop the convection computations in this case and do not go back to TRIGGER_FUNCT to look for the next unstable LCL which could produce a thicker cloud.).

convect_condens - Compute temperature cloud and ice water content from enthalpy and r_w and to return values for L_v, L_s and C_{ph}. Condensate is extracted iteratively.

convect_mixing_funct - Determine the entrainment and detrainment rate by evaluating the area under the distribution function determined by KMF; KMF=1 - gaussian, KMF=2 - triangular distribution function (not yet coded). The integration interval is limited by the critical mixed fraction PMIXC.

convect_tstep_pref - determine the convective advection time step PTIMEC as a function of the mean ambient wind as well as the precipitation efficiency as a function of wind shear and cloud base height. Mean precipitation efficiency is used to compute rainfall.

convect_downdraft - determine downdraft properties (mass flux, thermodynamics). Computations are done at every model level starting from top. The use of masks allows to optimise the inner loops (horizontal loops). Determine the LFS by looking for minimum of environmental saturated θ_e. Determine the mixed fraction using environmental and updraft values of θ_e at LFS. Estimate the effect of melting on the downdraft. Initialize humidity at LFS as a saturated mixture of updraft and environmental air. Determine the DBL by looking for level where the envir. θ_{es} at the LFS corrected by melting effects becomes larger than environmental value. Define mass flux and entrainment/detrainment rates at LFS, downdraft detrainment is assumed to occur in a layer of 60 hPa, determine top level IDDT of this layer, enter loop for downdraft computations and make a first guess of initial downdraft mass flux. In the downdraft computations we use θ_{es} instead of enthalpy as it allows to better take into account evaporation effects. As the downdraft detrainment rate is zero apart from the detrainment layer, we just compute enthalpy downdraft from θ_{es} in this layer. Calculate total downdraft evaporation. rate for given mass flux (between DBL and IDDT). Determine wet bulb temperature at DBL from θ_e. The iteration algoritm is similar to that used in routine CONVECT_CONDENS. Sum total downdraft evaporation rate. No evaporation if actual humidity is larger than specified one. If downdraft does not evaporate any water for specified relative humidity, no downdraft is allowed.

convect_satmixratio - Compute vapor saturation mixing ratio over liquid water and to return values for L_v L_s and C_{ph}.

convect_precip_adjust - Adjust up- and downdraft mass fluxes to be consistent with the mass transport at the LFS given by the precipitation efficiency relation. The total mass transported from the updraft to the down- draft at the LFS must be consistent with the three water budget terms. Downdraft evaporation rate at the DBL. The evaporation rate in downdraft must be consistent with precipitation efficiency relation. Some preliminar computations for downdraft = total precipitation rate. The precipitation is evaluated in a layer thickness DP=XUSRDPTH=165 hPa above the LCL. The difference between updraft precipitation and downdraft precipitation (updraft supply rate) is used to drive the downdraft through evaporational cooling. Total amount of precipitation that falls out of the up- draft between the LCL and the LFS. Condensate transfer from up to downdraft at LFS. Increase the first guess downdraft mass flux to satisfy precipitation efficiency relation. If downdraft does not evaporate any water at the DBL for the specified relative humidity, or if the corrected mass flux at the LFS is positive no downdraft is allowed. Increase updraft mass flux, mass detrainment rate, and water substance detrainment rates to be consistent with the transfer of the estimated mass from the up- to the downdraft at the LFS.

convect_closure - determine the final adjusted (over a time step PTIMEC) environmental values of θ_l, r_w, r_c, r_i The final convective tendencies can then be evaluated in the main routine DEEP_CONVECT by (PTHC-PTH)/PTIMEC. Computations are done at every model level starting from bottom. The use of masks allows to optimise the inner loops (horizontal loops). Compute fractional time step. For stability or mass conservation reasons one must split full time step PTIMEC.

convect_closure_thrvlcl - determine the thermodynamic properties at the new lifting condensation level LCL. Construct a mixed layer as in TRIGGER_FUNCT.

convect_satmixratio - Compute vapor saturation mixing ratio over liquid water and to return values for L_v L_s and C_{ph}.

convect_closure_adjust - Uses closure adjustment factor to adjust mass flux using the factor PADJ computed in CONVECT_CLOSURE and to modify precipitation efficiency when necessary. The computations are similar to routine CONVECT_PRECIP_ADJUST.

ch_convect_scavenging - (I could not find the subroutine anywhere. I guess it is not used in Arome).

convect_chem_transport - Compute modified chemical tracer values due to convective event. Identical to the computation of the conservative variables in the main deep convection code.

ini_convpar_e1 - initialize the convective constants stored in modules with modifications for ensemble run. The initialization is the same as in ini_convpar, but XCRAD has different values; in ordinary run XCRAD=1500 and in the ensemble run XCRAD=500.

deep_convection - (as above, different arguments)

ini_convpar - reinitialize convection parameters.

deep_convection - (as above, different arguments)

deep_convection - (as above, different arguments)

ini_convpar_shal - initialize the convective constants stored in modules for shallow convection. The initialization is the same as in ini_convpar, but XCRAD=50 and other parameters have different values than in the deep convection parameterisation.

shallow_convection - determine tendencies due to shallow convection. The routine first prepares all necessary grid-scale variables. The final shallow convection tendencies are computed calling different subroutines. Shallow convection columns in the model domain are selected through the call of routine TRIGGER_FUNCT_SHAL. Then, memory for the convection updraft and downdraft variables is allocated and the grid scale variables are gathered in convective arrays. The updraft and downdraft computations are done level by level starting at the bottom and top of the domain, respectively. All computations are done on MNH thermodynamic levels. The depth of the current model layer k is defined by $\Delta p_k = p_{k-1} - p_k$.

convect_trigger_shal - procedure is similar to the deep convection one.

convect_updraft_shal - procedure is similar to the deep convection one.

convect_closure_shal - procedure is similar to the deep convection one.

convect_chem_transport - same as for the deep convection.

ac_turb_mnh - compute the turbulence sources and the TKE evolution for the Arome model.

turb - compute the source terms in the evolution equations due to the turbulent mixing. The source term is computed as the divergence of the turbulent fluxes. The cartesian fluxes are obtained by a one and a half order closure, based on a prognostic equation for the Turbulence Kinetic Energy (TKE). The metric coefficients are recovered from the grid knowledge. The system is closed by prescribing a turbulent mixing length. Different choices are available for this length.

- HTURBLEN='BL89' the Bougeault and Lacarrere algorithm is used.
- HTURBLEN='DELT' the mixing length is given by the mesh size depending on the model dimensionality, this length is limited with the ground distance.
- HTURBLEN='DEAR' the mixing length is given by the mesh size depending on the model dimensionality, this length is limited with the ground distance and also by the Deardorff mixing length in the stable cases.
- HTURBLEN='KEPS' the mixing length is deduced from the TKE dissipation, which becomes a prognostic variable of the model (Duynkerke formulation).

The conservative variables are computed along with Lv/Cp. The turbulent Prandtl numbers are computed from the resolved fields and TKE. The sources associated to the vertical turbulent fluxes are computed with a temporal scheme allowing a degree of implicitness given by PIMPL, varying from PIMPL=0 (purely explicit scheme) to PIMPL=1 (purely implicit scheme). PIMPL is hardcoded to 1 in Arome. The sources associated to the horizontal fluxes are computed with a purely explicit temporal scheme. These sources are only computed when the turbulence parameterization is 2D or 3D (HTURBDIM='3DIM'). The sources for TKE are computed, along with the dissipation of TKE if HTURBLEN='KEPS'.

bl89 - HTURBLEN='BL89' the Bougeault and Lacarrere algorithm is used. The mixing length is given by the vertical displacement from its original level of an air particule having an initial internal energy equal to its TKE and stopped by the buoyancy forces.

rmc01 - is called only if ORMC01 that is hardcoded to false in ac_turb_mnh. It modifies the mixing and dissipative length near the SBL. The mixing length is given by the mesh size depending on the model dimensionality, this length is limited with the distance to the ground.

prandtl - compute the Redelsperger numbers and then get the turbulent Prandtl and Schmidt numbers: for the heat fluxes - $\phi_3 = 1/Prandtl$ for the moisture fluxes - $\psi_3 = 1/Schmidt$.

turb_ver - compute the vertical turbulent fluxes of the prognostic variables and give back the source terms to the main program. In the case of large horizontal meshes, the divergence of these vertical turbulent fluxes represent the whole effect of the turbulence but when the three-dimensionnal version of the turbulence scheme is activated (CTURBDIM="3DIM"), these divergences are completed in the next routine TURB_HOR. An arbitrary degree of implicitness has been implemented for the temporal treatment of these diffusion terms. The vertical boundary conditions are as follows:

- at the bottom, the surface fluxes are prescribed at the same as the other turbulent fluxes
- at the top, the turbulent fluxes are set to 0.

It should be noted that the condensation has been implicitely included in this turbulence scheme by using conservative variables and computing the subgrid variance of a statistical variable s indicating the presence or not of condensation in a given mesh. 1D type calculations are made; The vertical turbulent fluxes are computed in an off-centered implicit scheme (a Crank-Nicholson type with coefficients different than 0.5), which allows to vary the degree of implicitness of the formulation. The different prognostic variables are treated one by one. The contributions of each turbulent fluxes are cumulated into the tendency PRvarS, and into the dynamic and thermal production of TKE if necessary.

turb_ver_thermo_flux - the thermodynamical fields are considered. Only the turbulent fluxes of the conservative variables (θ_l and R_{np} stored in PRx(:,:,:,1)) are computed. Note that the turbulent fluxes at the vertical boundaries are given either by the soil scheme for the surface one (at the same instant as the others fluxes) and equal to 0 at the top of the model. The thermal production is computed by vertically averaging the turbulent flux and multiply this flux at the mass point by a function E_θ or E_{moist}, which preform the transformation from the conservative variables to the virtual potential temperature. The variance of the statistical variables indicating presence or not of condensation, is determined in function of the turbulent moments of the conservative variables and its square root is stored in PSIGS. This information will be completed in the horizontal turbulence if the turbulence dimensionality is not equal to "1DIM".

tridiag - give a field PVARP at t+1, by solving an implicit tridiagonal system obtained by the discretization of the vertical turbulent diffusion. It should be noted that the degree of implicitness can be varied (PIMPL parameter, hardcoded to 1 in ac_turb_mnh) and the sources of evolution other than the turbulent diffusion can be taken into account through the PSOURCE field.

turb_ver_dyn_flux - the wind components are considered. The surface flux $\langle u'w' \rangle$ is computed from the value of the surface fluxes computed in axes linked to the orography (i", j", k"): i" is parallel to the surface and in the direction of the maximum slope, j" is also parallel to the surface and in the normal direction of the maximum slope, k" is the normal to the surface. In order to prevent numerical instability, the implicit scheme has been extended to the surface flux regarding to its dependence in function of U. The dependence in function of the other components introduced by the different rotations is only explicit. The turbulent fluxes are used to compute the dynamical production of TKE. For the last TKE level (located at PDZZ(:,:,IKB)/2 from the ground), a harmonic extrapolation from the dynamical production at PDZZ(:,:,IKB) is used to avoid an evaluation of the gradient of U in the surface layer. The same steps are repeated but for the y direction (v component) and a diagnostic computation of the W variance is performed.

tridiag_wind - give a field PVARP at t+1, by

turb_ver_sv_flux - the turbulent fluxes for the passive scalar variables are computed by the same way as the conservative thermodynamical variables.

tridiag - give a field PVARP at t+1, by

turb_ver_sv_corr - compute the subgrid Sv2 and SvThv terms. Functions E_θ and E_{moist} allow to compute the coefficients for the turbulent correlation between any variable and the virtual potential temperature from its correlations with the conservative potential temperature and the humidity conservative variable: $A'\theta'_v = E_\theta A'\theta'_l + E_{moist} A' R'_{np}$.

turb_hor_splt - for 3D turbulence, not used in Arome.

tke_eps_sources - compute the sources necessary for the evolution of the turbulent kinetic energy and its dissipation if necessary. The vertical turbulent flux is computed in an off-centered implicit scheme (a Crank-Nicholson type with coefficients different than 0.5), which allows to vary the degree of implicitness of the formulation. In high resolution, the horizontal transport terms are also calculated, but explicitly. The evolution of the dissipation as a variable is made if the parameter HTURBLEN is set equal to KEPS. The same reasoning made for TKE applies.

tridiag_tke - give a field PVARP at t+1, by

second_mnh - used only in LES if LLES_CALL, hardcoded to FALSE in ini_turb.

les_mean_subgrid - used only in LES if LLES_CALL, hardcoded to FALSE in ini_turb.

ac_rain_ice - compute the microphysical sources related to the resolved clouds and precipitation it computes the real absolute pressure, negative values of the current guess of all mixing ratio are removed. This is done by a global filling algorithm based on a multiplicative method (Rood, 1987), in order to conserved the total mass in the simulation domain. Sources are transformed in physical tendencies, by removing the multiplicative term $\rho_d J$. After calling to microphysical routines, the physical tendencies are switched back to prognostic variables.

rain_ice - compute the slow microphysical sources which can be computed explicitly. Timestep is split into KSPLITR sub-time-steps for sedimntation. The autoconversion computation follows Kessler (1969). The sedimentation rate is computed with a time spliting technique and an upstream scheme, written as a difference of non-advective fluxes. This source term is added to the future instant (split-implicit process). The other microphysical processes are evaluated at the central instant (split-explicit process): autoconversion, accretion and rain evaporation. These last 3 terms are bounded in order not to create negative values for the water species at the future instant. The processes are calculated on a subset of grid-points - only where the considered processes actually take place - where any r_j is bigger than the predefined minimum value. This subroutine contains and calls the following subroutines:

rain_ice_sedimentation - the prodecure is done in KSPLITR steps. Sedimentation is calculated for rain, snow, graupel and nominally pristine ice, but cloud ice field is used in the code (although there is another variable containing pristine ice, but at time t). It is calculated only for the points where the forecast value of any of the precipitable species (or cloud ice) is bigger than the prescribed minimum.

rain_ice_nucleation - it is computed only for the subset of points where $T < T_{tt}$. Compute heterogeneous nucleation source (RVHENI): first the cloud ice concentration is computed and theni the cloud ice and water vapour mixing ratios r_i and r_v are updated.

rain_ice_slow - compute the homogeneous nucleation source: RCHONI, RRHONG, compute the deposition, aggregation and autoconversion sources, the thermodynamical function $A_i(T, P)$ and the c'_j (in the ventilation factor), compute the riming-conversion of r_c for r_i production: RCAUTI, compute the deposition on r_s: RVDEPS, compute the aggregation on r_s: RIAGGS, compute the autoconversion of r_i for r_s production: RIAUTS and compute the deposition on r_g: RVDEPG.

rain_ice_warm - compute the autoconversion of r_c for r_r production: RCAUTR, compute the accretion of r_c for r_r production: RCACCR and compute the evaporation of r_r: RREVAV.

rain_ice_fast_rs - cloud droplet riming of the aggregates, riming of the small sized aggregates, riming-conversion of the large sized aggregates into graupel, rain accretion onto the aggregates, raindrop accretion on the small sized aggregates, raindrop accretion-conversion of the large sized aggregates into graupel, conversion-melting of the aggregates.

rain_ice_fast_rg - rain contact freezing, compute the dry growth case, accretion of aggregates on the graupel, accretion of raindrops on the graupel, compute the wet growth case, select wet or dry case, melting of the graupel.

rain_ice_fast_ri - first the cloud ice is melted if $T > T_{tt}$, aterwards the Bergeron-Findeisen effect is calculated: RCBERI. This effect describes the evaporation of cloud droplets and depostion to the cloud ice in the mixed phase cloud due to $e_{si}(T) < e_{sw}(T)$.

4 Subroutine apl_arome

Subroutine apl_arome reverts the levels upside-down, transforms the variables and calls interfaces to the MesoNH parameterisations.

1. Initialisation,
   - if LTWOTL ZDT=PDT/2 (half the time step is used in MesoNH subroutines since they were designed for 3TL).
   - for three-time-level, if KSTEP=0 ZDT=PDT, and ZDT=PDT/2 otherwise. (I am confused here since I am used to ZDT=PDT/2 for the 0 KSTEP and ZDT=PDT otherwise in the three-time-level scheme.)
   - Reverse level interfaces and trasform geopotentiel to z and add the additional interlayer:

     $$z_{\tilde{l}} = \frac{1}{g}\phi_{\tilde{L}+2-\tilde{l}} \qquad z_{\tilde{1}} = 2z_{\tilde{2}} - z_{\tilde{3}}$$

   - Reverse levels and trasform geopotentiel to z and add the additional levels:

     $$z_l = \frac{1}{g}\phi_{L+2-l} \qquad z_1 = \frac{3}{2}z_{\tilde{2}} - \frac{1}{2}z_{\tilde{3}} \qquad z_{L+2} = z_{L+1} + z_{\tilde{L}+2} - z_{\tilde{L}+1}$$

   - initialize the specific mass of dry air to be used for conversion from q_j to r_j.

     $$q_d = 1 - q_v - q_c - q_r - q_i - q_s - q_g$$

   - Initialize density of dry air (reference density) $q_d\rho = \rho_d$

     $$\rho_{drefl} = \frac{p_{L+2-l}}{R_{L+2-l}T_{L+2-l}q_{dL+2-l}} \qquad \rho_{dref1} = \rho_{dref2} \qquad \rho_{drefL+2} = \rho_{drefL+1}$$

   - Initialization of the dry densiti multiplied by Jacobian

     $$\rho_d J_l = \frac{\Delta p_{L+2-l}}{g} \qquad \rho_d J_1 = \rho_d J_2 \qquad \rho_d J_{L+2} = \rho_d J_{L+1}$$

   - initialize Exner function

     $$\pi_l = \left(\frac{p_{L+2-l}}{p_{00}}\right)^{\frac{R_d}{c_{pd}}} \qquad \pi_1 = \pi_2 \qquad \pi_{L+2} = \pi_{L+1}$$

   - initialize the model variables - reverse levels and add the two extra levels used in MesoNH.

     $$\psi_l = \psi_{L+2-l} \qquad \psi_1 = \psi_2 \qquad \psi_{L+2} = \psi_{L+1} \qquad for \quad \psi = p, u, v, w, TKE, \sigma$$

     for pressure, the three wind components, turbulent kinetic energy and σ.
   - initialize the potential temperature

     $$\theta_l = \frac{T_{L+2-l}}{\pi_l} \qquad \theta_1 = \theta_2 \qquad \theta_{L+2} = \theta_{L+1}$$

   - initialize prognostic moist variables

     $$r_{jl} = \frac{q_{jL+2-l}}{q_{dL+2-l}} \qquad r_{j1} = 0 \qquad r_{jL+2} = 0 \qquad for \quad j = v, c, r, i, s, g$$

     for water vapour, cloud water, cloud ice, rain, snow and graupel.
   - initialize reference potential temperature

     $$\theta_{vrefl} = \frac{\theta_l\left(1 + r_{vl}\frac{R_v}{R_d}\right)}{1 + r_{vl} + r_{cl} + r_{rl} + r_{il} + r_{sl} + r_{gl}} \qquad \theta_{vref1} = \theta_{vref2} \qquad \theta_{vrefL+2} = \theta_{vrefL+1}$$

- initialization of the prognostic variables (sources)

$$\psi_l^{t+\Delta t} = \frac{\psi_{L+2-l}^t}{\Delta t} \qquad \psi_1^{t+\Delta t} = \psi_2^{t+\Delta t} \qquad \psi_{L+2}^{t+\Delta t} = \psi_{L+1}^{t+\Delta t} \qquad for \quad \psi = u, v, w, TKE$$

- initialization of prognostic potential temperature

$$\theta_l^{t+\Delta t} = \frac{\theta_l^t}{\Delta t}$$

- initialization of the prognostic values of moist variables.

$$r_{jl}^{t+\Delta t} = \frac{r_{jl}^t}{\Delta t} \qquad for \quad j = v, c, r, i, s, g$$

2. Adjustment

   These operations are performed only if LMICRO.

- The values of prognostic potential temperature θ and r_j moist variables are stored to temporary arrays before calling adjustment.
- CALL AC_ADJUST - returns updated values of prognostic potential temperature and moist species mixing ratios and computes cloudiness used by radiation scheme.
- cloudiness is turned upside-down to serve as input to the radiation scheme (radheat).

$$CF_l^{t+\Delta t} = NEB_{L+2-l}$$

- t values of potential temperature are updated

$$\theta_l^t = \theta_l^{t+\Delta t}\Delta t \qquad \theta_1^t = \theta_2^{t+\Delta t}\Delta t \qquad \theta_{L+2}^t = \theta_{L+1}^{t+\Delta t}\Delta t$$

- t values of moist variables are updated

$$r_{jl}^t = r_{jl}^{t+\Delta t}\Delta t \qquad r_{j1}^t = r_{j2}^{t+\Delta t}\Delta t \qquad r_{jL+2}^t = r_{jL+1}^{t+\Delta t}\Delta t \qquad for \quad j = v, c, r, i, s, g$$

- recompute the specific mass of dry air to be used for conversion from r_j to q_j.

$$q_{dl} = \frac{1}{1 + r_{vl} + r_{cl} + r_{rl} + r_{il} + r_{sl} + r_{gl}}$$

- compute q_j from r_j (update variables after adjustment)

$$q_{jl}^t = r_{jl}^t q_{dl} \qquad for \quad j = v, c, r, i, s, g$$

- update tendency of temperature (it was 0 up to this point)

$$\left(\frac{\partial T}{\partial t}\right)_{phyl} = \left(\frac{\partial T}{\partial t}\right)_{phyl} + \left(\theta_{L+2-l}^{t+\Delta t} - \theta_{L+2-l}^{tsave}\right)\pi_{L+2-l}$$

where θ_{L+2-l}^{tsave} is $\theta_{L+2-l}^{t+\Delta t}$ before adjustment and equal to $\frac{\theta_{L+2-l}^t}{\Delta t}$

- update tendency of moist variables (it was 0 up to this point)

$$\left(\frac{\partial q_j}{\partial t}\right)_{phyl} = \left(\frac{\partial q_j}{\partial t}\right)_{phyl} + \left(r_{jL+2-l}^{t+\Delta t} - r_{jL+2-l}^{tsave}\right)q_{dl}$$

where r_j^{tsave} is $r_j^{t+\Delta t}$ before adjustment and equal to $\frac{r_j^t}{\Delta t}$

- CALL GPRCP that calculates C_p, R and κ.
- compute temperature

$$T_l^t = \theta_{L+2-l}^t \pi_{L+2-l}$$

- compute dry air reference density

$$\rho^t_{drefl} = \frac{p^t_{L+2-l}}{R^t_{L+2-l}T^t_{L+2-l}q^t_{dL+2-l}} \qquad \rho^t_{dref1} = \rho^t_{dref2} \qquad \rho^t_{drefL+2} = \rho^t_{drefL+1}$$

- CALL GPGEO - computes half and full level geopotentiel from new T and R.
- recompute height of level interfaces

$$z_{\tilde{l}} = \frac{1}{g}\phi^t_{\tilde{L}+2-\tilde{l}} \qquad z_{\tilde{1}} = z_{\tilde{2}} - z_{\tilde{3}}$$

- recompute height of levels

$$z_l = \frac{1}{g}\phi^t_{L+2-l} \qquad z_1 = \frac{3}{2}z_2 - \frac{1}{2}z_3 \qquad z_{L+2} = z_{L+1} + z_{\tilde{L+2}} - z_{\tilde{L+1}}$$

3. Radiation

   These actions are performed only if LRAYFM.

   - CALL ACTQSAT
   - initialization of moist variables used in the ECMWF radiation scheme

$$q_{ice} = max\left(0, \frac{q_i}{1 - q_i - q_c - q_r - q_g - q_s}\right)$$

$$q_{liq} = max\left(0, \frac{q_c}{1 - q_i - q_c - q_r - q_g - q_s}\right)$$

$$q_{vap} = max\left(0, \frac{q_v}{1 - q_i - q_c - q_r - q_g - q_s}\right)$$

   These q_{ice}, q_{liq} and q_{vap} are different than q_i, q_c and q_v.

   - Introduce albedo, emissivity and surface temperature from the surface fileds array.
   - this part is not called each time-step to save CPU. If module of KSTEP and radiation frequency computations is zero: Initialize half-level temperature and aerosols.
     CALL RADACT
     CALL SUOZON
     CALL ACRADIN
     save shortwave surface fluxes endif.

   - CALL RADHEAT - radiative flux calculations performed every time-step.

   - Surface IR flux is saved to be used in the surface scheme.
   - Tendency of temperature is updated with the contribution from radiation

$$\left(\frac{\partial T}{\partial t}\right)_{phyl} = \left(\frac{\partial T}{\partial t}\right)_{phyl} + \left(\frac{\partial T}{\partial t}\right)_{radl}$$

4. Convection

   - initialize temporary variables used in convection
   - The values of prognostic potential temperature θ and r_j moist variables are stored to temporary arrays before calling convection.
   - CALL BRI2ACCONV

- update tendency of temperature

$$\left(\frac{\partial T}{\partial t}\right)_{phyl} = \left(\frac{\partial T}{\partial t}\right)_{phyl} + \left(\frac{\partial T}{\partial t}\right)_{cnvL+2-l}$$

- update tendency of water vapour

$$\left(\frac{\partial q_v}{\partial t}\right)_{phyl} = \left(\frac{\partial q_v}{\partial t}\right)_{phyl} + \left(\frac{\partial r_v}{\partial t}\right)_{cnvL+2-l} q_{dl}$$

- update tendency of cloud water

$$\left(\frac{\partial q_c}{\partial t}\right)_{phyl} = \left(\frac{\partial q_c}{\partial t}\right)_{phyl} + \left(\frac{\partial r_c}{\partial t}\right)_{cnvL+2-l} q_{dl}$$

- update tendency of cloud ice

$$\left(\frac{\partial q_i}{\partial t}\right)_{phyl} = \left(\frac{\partial q_i}{\partial t}\right)_{phyl} + \left(\frac{\partial r_i}{\partial t}\right)_{cnvL+2-l} q_{dl}$$

- update prognostic mixing ration of water vapour

$$r_{vl}^{t+\Delta t} = r_{vl}^{t+\Delta t} + \left(\frac{\partial r_v}{\partial t}\right)_{cnvl}$$

- update prognostic mixing ration of cloud water

$$r_{cl}^{t+\Delta t} = r_{cl}^{t+\Delta t} + \left(\frac{\partial r_c}{\partial t}\right)_{cnvl}$$

- update prognostic mixing ration of cloud ice

$$r_{il}^{t+\Delta t} = r_{il}^{t+\Delta t} + \left(\frac{\partial r_i}{\partial t}\right)_{cnvl}$$

- update prognostic mixing ration of potential temperature

$$\theta_l^{t+\Delta t} = \theta_l^{t+\Delta t} + \left(\frac{\partial T}{\partial t}\right)_{cnvl} \left(\frac{p_0}{p_{L+2-l}}\right)^{\frac{R_d}{c_{pd}}}$$

- store instantaneous precipitation rates at surface for rain

$$R_r = R_r + \left(\frac{\partial R}{\partial t}\right)_{cnv} - \left(\frac{\partial R_s}{\partial t}\right)_{cnv}$$

- store instantaneous precipitation rates at surface for snow

$$R_s = R_s + \left(\frac{\partial R_s}{\partial t}\right)_{cnv}$$

5. Turbulence

   Turbulence is called if LTURB.

   - initialize temporary variables used in convection
   - The values of prognostic wind components u, v and w, turbulent kinetic energy TKE, potential temperature θ and moist variables r_j are stored to temporary arrays before calling turbulence.
   - CALL AC_TURB_MNH

- update

$$\sigma_l = \sigma_{L+2-l}$$

- update tendencies of wind components and TKE

$$\left(\frac{\partial \psi}{\partial t}\right)_{phyl} = \left(\frac{\partial \psi}{\partial t}\right)_{phyl} + \left(\psi^{t+\Delta t}_{jL+2-l} - \psi^{tsave}_{jL+2-l}\right) \qquad for \qquad \psi = u, v, w, TKE$$

where ψ^{tsave}_{L+2-l} is $\psi^{t+\Delta t}_{L+2-l}$ before turbulence.

- update tendency of temperature

$$\left(\frac{\partial T}{\partial t}\right)_{phyl} = \left(\frac{\partial T}{\partial t}\right)_{phyl} + \left(\theta^{t+\Delta t}_{L+2-l} - \theta^{tsave}_{L+2-l}\right) \pi_{L+2-l}$$

where θ^{tsave}_{L+2-l} is $\theta^{t+\Delta t}_{L+2-l}$ before turbulence.

- update tendency of moist variables

$$\left(\frac{\partial q_j}{\partial t}\right)_{phyl} = \left(\frac{\partial q_j}{\partial t}\right)_{phyl} + \left(r^{t+\Delta t}_{jL+2-l} - r^{tsave}_{jL+2-l}\right) q_{dl}$$

where r^{tsave}_j is $r^{t+\Delta t}_j$ before turbulence. q_d is not recomputed before this to include contribution from convection because physics is parallel, not sequential.

6. Microphysics

Microphyisics is called if LMICRO.

- The values of prognostic potential temperature θ and r_j moist variables are stored to temporary arrays before calling microphysics.
- the accumulated surface precipitation (input from the surface scheme) is divided by two before calling AC_RAIN_ICE only to be multipiled afterwards.

$$R^a_r = \frac{1}{2} R^a_r \qquad R^a_s = \frac{1}{2} R^s_r \qquad R^a_g = \frac{1}{2} R^g_r$$

- CALL AC_RAIN_ICE
- the accumulated surface precipitation is multiplied by two

$$R^a_r = 2R^a_r \qquad R^a_s = 2R^s_r \qquad R^a_g = 2R^g_r$$

- update tendency of temperature

$$\left(\frac{\partial T}{\partial t}\right)_{phyl} = \left(\frac{\partial T}{\partial t}\right)_{phyl} + \left(\theta^{t+\Delta t}_{L+2-l} - \theta^{tsave}_{L+2-l}\right) \pi_{L+2-l}$$

where θ^{tsave}_{L+2-l} is $\theta^{t+\Delta t}_{L+2-l}$ before microphysics.

- update tendencies of moist variables

$$\left(\frac{\partial q_j}{\partial t}\right)_{phyl} = \left(\frac{\partial q_j}{\partial t}\right)_{phyl} + \left(r^{t+\Delta t}_{jL+2-l} - r^{tsave}_{jL+2-l}\right) q_{dl}$$

where r^{tsave}_j is $r^{t+\Delta t}_j$ before microphysics. Why q_d is not recomputed before this? OK, if it is assumed that the physics schemes conserve mass then q_d should not change but?

5 Subroutine ac_adjust

Subroutine ac_adjust computes the microphysical sources related to the resolved clouds and precipitation. Computes the real absolute pressure, negative values of the current guess of all mixing ratio are removed, sources are transformed in physical tendencies. After calling to microphysical routines, the physical tendencies are switched back to prognostic variables.

1. some dimensioning and options hardcoding preliminary computations. IKB that should be the bottom of the atmosphere is $IKB = 1{+}JPVEXT$ increased by $JPVEXT$ meaning that we do not start calculations from the bottom of the atmosphere but some level above if $JPVEXT \neq 0$. The top of the atmosphere IKE is determined by $IKE = size(z,3) - JPVEXT$ meaning that the uppermost level corresponds to the uppermost level of z shifted downwards by $JPVEXT$. $HCLOUD$ is hardcoded to $ICE3$, $HRAD$ to $NONE$ and $HTURBDIM$ to $1DIM$.

2. some computations of local relevant variables:

$$T = \theta\pi_{ref} \qquad L_v = L_{VTT}{+}(c_{pv}{-}c_l)(T{-}T_{TT}) \qquad L_s = L_{STT}{+}(c_{pv}{-}c_i)(T{-}T_{TT}) \qquad c_{ph} = c_{pd}{+}c_{pv}2\Delta t r_v^{t+\Delta t}$$

3. negative values of the forecast values of the precipitation species $r_j^{t+\Delta t}$ for j=3,5,6,7 are removed applying simple mass correction. However, this correction is not applied if $HCLOUD = ICE3$ (as hardcoded in Arome) so this correction is not used.

4. And

   - Negative values of prognostic cloud ice are corrected and prognostic values water vapour and potential temperature are modified accordingly. The corrections are applied only where $r_i^{t+\Delta t} < 0$.

   $$r_v^{t+\Delta t} = r_v^{t+\Delta t} + r_i^{t+\Delta t} \qquad \theta^{t+\Delta t} = \theta^{t+\Delta t} - r_i^{t+\Delta t}\frac{L_s}{c_{ph}}\frac{1}{\pi_{ref}} \qquad r_i^{t+\Delta t} = 0$$

   - Negative values of prognostic cloud water are corrected and prognostic values of water vapour and potential temperature are modified accordingly. The corrections are applied only where $r_c^{t+\Delta t} < 0$.

   $$r_v^{t+\Delta t} = r_v^{t+\Delta t} + r_c^{t+\Delta t} \qquad \theta^{t+\Delta t} = \theta^{t+\Delta t} - r_c^{t+\Delta t}\frac{L_v}{c_{ph}}\frac{1}{\pi_{ref}} \qquad r_c^{t+\Delta t} = 0$$

   - Negative values of prognostic water vapour are corrected and prognostic values of cloud water and potential temperature are modified accordingly. The corrections are applied only where $r_v^{t+\Delta t} < 0$ and $r_c^{t+\Delta t} > 0$.

   $$r_v^{t+\Delta t} = r_v^{t+\Delta t} + r_c^{t+\Delta t} \qquad \theta^{t+\Delta t} = \theta^{t+\Delta t} - r_c^{t+\Delta t}\frac{L_v}{c_{ph}}\frac{1}{\pi_{ref}} \qquad r_c^{t+\Delta t} = 0$$

   - Negative values of prognostic water vapour are corrected and prognostic values of cloud ice and potential temperature are modified accordingly. The corrections are applied only where $r_v^{t+\Delta t} < 0$ and $r_i^{t+\Delta t} > 0$ if $KRR > 3$.

   $$ZC = min(-r_v^{t+\Delta t}, r_i^{t+\Delta t}) \qquad r_v^{t+\Delta t} = r_v^{t+\Delta t}{+}ZC \qquad \theta^{t+\Delta t} = \theta^{t+\Delta t}{-}ZC\frac{L_s}{c_{ph}}\frac{1}{\pi_{ref}} \qquad r_i^{t+\Delta t} = r_i^{t+\Delta t}{-}ZC$$

5. CALL ICE_ADJUST

NOTE: There is stil a possibility that there are some places where r_v,r_c or r_i would have negative values. r_v can become negative after first two corrections and can not be restored to get the positive value by the other two corrections since then $r_c = 0$ and $r_i = 0$. In correction 3 all cloud water is evaporated to correct for negative r_v but in correction 4 only part of the cloud ice is sublimated - just enough to restore r_v to zero (unless there is not enough r_i). The 3^{rd} correction could give $r_v > r_{vs}$ mixing ratio of water vapor larger than saturated and $r_c = 0$. The 4^{th} correction gives $r_i > 0$ presence of cloud ice when $r_v = 0$ there is no water vapour.

6 Subroutine bri2acconv

This routine is a bridge to the MesoNH deep and shallow convection called only if LKFBCONV.

- Convection time-step ZDTCONV is set equal to the physics time-step TSPHY from YOMPHY2.
- gridsize is determined (depending on LELAM key),
- temperature is reversed $ZT_l = PT_{L+2-l}$, $ZT_1 = ZT_2$ and $ZT_{L+2} = ZT_{L+1}$.
- CALL AC_CONV_MNH

6.1 Subroutine ac_conv_mnh

This routine is an interface to call the MesoNH parameterisations of deep and shallow convection.

- determine size of arrays and calculation loops, initialization, allocation
- chemical species are put to zero,
- if LDEEP: CALL INI_CONVPAR
- CALL DEEP_CONVECTION these and similar calls are repeated in a case of an ensemble convection run,
- if LSHALLOW: CALL INI_CONVPAR_SHAL
- CALL SHALLOW_CONVECTION these and similar calls are repeated in
- add deep and shallow convection tendencies of T, r_v, r_c and r_i

$$\left(\frac{\partial\psi}{\partial t}\right)_{cnv} = \left(\frac{\partial\psi}{\partial t}\right)_{deepcnv} + \left(\frac{\partial\psi}{\partial t}\right)_{shalcnv} \qquad for \qquad \psi = T, r_v, r_c \quad and \quad r_i$$

- determine total and solid surface precipitation rate,

$$R_{total} = R_{liquid} + R_{solid} \qquad R_{solid} = R_{solid}$$

  the right hand side variables are passed out of the routine as arguments, but afterwards, in apl_arome $R_{liquid} = R_{total} - R_{solid}$ so perhaps this part is not needed?

- calculate convective mass flux for subgrid condensation (PMF),

$$M_{tot} = M_{up,deepcnv} + M_{down,deepcnv} + M_{up,shalcnv}$$

  total mass flux is the sum of updraft and downdraft mass fluxes from deep convection and updraft flux from shallow convection.

- diagnostics (if LDIAGCONV),

$$M_d = M_{down,deepcnv}$$

  total downdraft flux is equal to the total deep convection downdraft flux.

$$M_u = M_{up,deepcnv} + M_{up,shalcnv}$$

  total updraft flux is equal to the som of total deep convection updraft flux and shallow convection updraft mass flux.

- calculate CAPE etc.

7 Subroutine ac_turb_mnh

Subroutine ac_turb_mnh computes the turbulent sources and the evolution of TKE.

- initialization of few parameters: number of small steps for turbulence KSPLIT is set to 1, turbulence HTURBDIM is set to 1DIM and HTURBLEN is set to BL89.

$$\Delta z_{\tilde{l}} = z_l - z_{l-1} \tag{1}$$

- multiply guess $t + \Delta t$ values by $\rho_d J$ to obtain source terms of MesoNH.

$$\psi^{t+\Delta t} = \psi^{t+\Delta t} \rho_d J \tag{2}$$

  for $\psi = u, v, w, \theta, TKE, r_v, r_c, r_r, r_i, r_s, r_g$ and passive scalars.

- CALL TURB

- divide guess $t + \Delta t$ values by $\rho_d J$

$$\psi^{t+\Delta t} = \frac{\psi^{t+\Delta t}}{\rho_d J} \tag{3}$$

  for $\psi = u, v, w, \theta, TKE, r_v, r_c, r_r, r_i, r_s, r_g$ and passive scalars.

7.1 Subroutine turb

Subroutine turb computes source terms due to the turbulent mixing as the divergence of the turbulent fluxes obtained by one and half order closure based on a prognostic equation for turbulent kinetic energy (TKE). The system is closed by turbulent mixing length that can be calculated in few different ways in MesoNH, but in Arome BL89 is used.

- initial computations: size of computational array, explicitness $a_{expl} = 1 - a_{impl}$, timesetep is divided by two if CCONF=START in the first time-step, but in Arome CCONF is hardcoded to be always RESTA.

- computation of mixing length depends on the selected scheme:

  - if BL98 - CALL BL98

  - if DELT - mixing length is determined by vertical resolution

$$L_l = z_{\tilde{l}+1} - z_{\tilde{l}}$$

  - if DEAR - mixing length is first computed from vertical resolution

$$L_l = z_{\tilde{l}+1} - z_{\tilde{l}}$$

    and afterwards it is limited by stability according to Deardorff formulation

$$\theta_v = \theta \frac{1 + \frac{R_v}{R_d} r_v}{1 + r_v + r_c + r_r + r_i + r_s + r_g}$$

    where $\frac{1}{2}\left(\frac{\theta_{v\,l+1} - \theta_{v\,l}}{\Delta z_{l+1}} + \frac{\theta_{v\,l} - \theta_{v\,l-1}}{\Delta z_l}\right)\frac{g}{\theta_{vref}} > 0$

$$L = min\left(L, 0.76\left(\frac{TKE}{\frac{1}{2}\left(\frac{\theta_{v\,l+1} - \theta_{v\,l}}{\Delta z_{l+1}} + \frac{\theta_{v\,l} - \theta_{v\,l-1}}{\Delta z_l}\right)\frac{g}{\theta_{vref}}}\right)^{\frac{1}{2}}\right)$$

  - if BLKR - mixing lenth is first set to 100 and afterwards (the initialization with 100 is actually not neccesary)

$$L_l = \alpha \frac{l_0\left(\frac{1}{2}(z_{\tilde{l}+1} + z_{\tilde{l}}) - z_b\right)}{l_0 + \alpha\left(\frac{1}{2}(z_{\tilde{l}+1} + z_{\tilde{l}}) - z_b\right)}$$

    where $\alpha = \frac{1}{2}^{-\frac{3}{2}}$ and $l_0 = 100$.

- if KEPS K-ϵ mixing length is calculated

$$L_l = C_d \frac{TKE^{\frac{3}{2}}}{\epsilon}$$

– dissipative lenths is first set equal to the mixing length an afterwards corrected in the surface layer (only?) by Monin-Obukhov if ORMC01 (this is hardcoded to false for Arome).

- c_{ph} and Exner function π are computed

$$c_{ph} = c_{pd} + c_{pv}r_v + c_l r_c + c_l r_r + c_i r_i + c_i r_s + c_i r_g \qquad \pi = \left(\frac{p}{p_0}\right)^{\frac{R_d}{c_{pd}}}$$

- conservative variables are computed:
- compute temperature at time t from θ at time t

$$T = \theta \left(\frac{p}{p_0}\right)^{\frac{R_d}{c_{pd}}}$$

- compute $\frac{L_v}{c_{ph}}$ at time t

$$C_{pex} = \frac{L_v}{c_{ph}} = \frac{L_{VTT} + (c_{pv} - c_l)(T - T_{TT})}{c_{pd} + c_{pv}r_v + c_l r_c + c_l r_r + c_i r_i + c_i r_s + c_i r_g}$$

- compute saturation vapour pressure

$$e_s = \psi_3 = exp\left(\alpha_w - \frac{\beta_w}{T} - \gamma_w alogT\right)$$

- compute saturated mixing ratio for water vapour at time t

$$r_{vs} = \psi_3 = \frac{\epsilon e_s}{p - e_s}$$

where $\epsilon = \frac{XMV}{XMD}$.

- compute derivative of saturated mixing ratio for water vapour at time t

$$r'_{vs} = \phi_3 = \left(\frac{\beta_w}{T} - \gamma_w\right)\frac{r_{vs}}{T}\left(1 + \frac{r_{vs}}{\epsilon}\right)$$

- compute A_{moist}

$$A_{moist} = \frac{0.5}{1 + r'_{vs}\frac{L_v}{c_{ph}}}$$

- compute A_θ

$$A_\theta = A_{moist}\left(\frac{p}{p_0}\right)^{\frac{R_d}{c_{pd}}}(r_{vs} - r_v)\frac{\frac{L_v}{c_{ph}}}{1 + r'_{vs}\frac{L_v}{c_{ph}}}\left[r_{vs}\left(1 + \frac{r_{vs}}{\epsilon}\right)\left(-2\frac{\beta_w}{T} + \gamma_w\right)\frac{1}{T^2} + r'_{vs}\left(1 + 2\frac{r_{vs}}{\epsilon}\right)\left(\frac{\beta_w}{T} - \gamma_w\right)\frac{1}{T} - r'_{vs}\right]$$

- compute $\frac{L_v}{c_{ph}}\frac{1}{\pi_{ref}}$

$$C_{pex} = \frac{C_{pex}}{\left(\frac{p}{p_0}\right)^{\frac{R_d}{c_{pd}}}}$$

- compute conservative (liquid) potential temperature at time t

$$\theta_l = \theta_l - \frac{L_v}{c_{ph}}\frac{1}{\pi_{ref}}r_c$$

- compute total moisture at time t that will be subject to the turbulent diffusion

$$r_v = r_v + r_c$$

- compute conservative (liquid) potential temperature at time $t + \Delta t$

$$\theta_l^{t+\Delta t} = \theta_l^{t+\Delta t} - \frac{L_v}{c_{ph}} \frac{1}{\pi_{ref}} r_c^{t+\Delta t}$$

- compute sources for total moisture (variable at $t + \Delta t$) that will enter turbulent diffusion

$$r_v^{t+\Delta t} = r_v^{t+\Delta t} + r_c^{t+\Delta t}$$

why there is no r_i anywhere?

- CALL PRANDTL - computes turbulent Prandtl and Schmidt numbers
- computations of proportionality coefficients between wind and stress and fluxes tangent to the surface.

$$C_d = -\left(\frac{F_u^2 + F_v^2}{u_1^2 + v_1^2}\right)^{\frac{1}{2}}$$

where F_u and F_v are surface fluxes of wind components.

- compute fluxes tangent to the surface

$$\tau_{11} = \tau_{22} = \tau_{33} = \frac{2}{3}\left[\left(1 + \frac{z_{b+1} - z_b}{\Delta z_{b+1} + \Delta z_b}\right) TKE_b - \frac{1}{2} TKE_{b+1}\right]$$

- CALL TURB_VER - compute source terms due to the vertical turbulent fluxes
- CALL TKE_EPS_SOURCES - compute sources of turbulent prognostic variables: TKE and dissipation.
- restore the non-conservative variables

$$r_v = r_v - r_c$$
$$r_v^{t+\Delta t} = r_v^{t+\Delta t} - r_c^{t+\Delta t}$$
$$\theta_l = \theta_l + \frac{L_v}{c_{ph}} \frac{1}{\pi_{ref}} r_c$$
$$\theta_l^{t+\Delta t} = \theta_l^{t+\Delta t} + \frac{L_v}{c_{ph}} \frac{1}{\pi_{ref}} r_c^{t+\Delta t}$$

7.2 Subroutine turb_ver

- initialization to zero of thermal and dynamical TKE production terms
- CALL TURB_VER_THERMO_FLUX - compute turbulent fluxes of θ_l and r_v.
- CALL TURB_VER_DYN_FLUX - compute turbulent fluxes of u, v and w.
- CALL TURB_VER_SV_FLUX - compute turbulent fluxes of passive scalars.
- CALL TURB_VER_SV_CORR - compute correction to the previous.

7.3 Subroutine tke_eps_sources

Subroutine tke_eps_sources computes TKE and its dissipation.

- compute the effective diffusion coefficient

$$K_{eff} = L^t \sqrt{TKE^t}$$

- compute explicit TKE sources

$$DP_b = DP_b \left(1 + \frac{D_{zzb+1}}{D_{zzb}}\right)$$

  is the dynamical production term. If e-ϵ closure is used - if KEPS

$$source = \frac{TKE^{t+\Delta t}}{\rho_d J} - \frac{TKE^t}{2\delta t} + DP + TP - \epsilon$$

  otherwise

$$source = \frac{TKE^{t+\Delta t}}{\rho_d J} - \frac{TKE^t}{2\delta t} + DP + TP - \delta_{expl} C_d \frac{\sqrt{TKE^t}}{L_{eps}} TKE^t$$

- determine some upperdiagonal terms

$$PA = -2\delta t C_{et} MZM(K_{eff}) MZM(\rho_d J) \frac{1}{D_{zz}}$$

- call TRIDIAG or TRIDIAG_TKE

$$TKE^{t+\Delta t} = TKE^{t+\Delta t} \frac{\rho_d J}{2\delta t}$$

- if KEPS - e-ϵ closure is used we have to solve the dissipation equation

$$source = \frac{\epsilon^{t+\Delta t}}{\rho_d J} - \frac{\epsilon^t}{2\delta t} + \left[C_{dp}(DP + TP) - C_{dd}\epsilon^t\right] \frac{\epsilon^t}{TKE^t}$$

  negative values of TKE are corrected: where $TKE < TKE_{min}$ $source = \frac{\epsilon_{min} - \epsilon^t}{2\Delta t}$

- call TRIDIAG

$$\epsilon^{t+\Delta t} = \epsilon^{t+\Delta t} \frac{\rho_d J}{2\delta t}$$

7.4 Subroutine tridiag

First, the right hand side of the implicit equation is computed.

$$ZY_k = \psi_k^t + 2\Delta t \psi_k^{t+\Delta t} - \frac{\delta_{expl}}{\rho_d J_k} \left(\psi_{k-1}^t PA_k - \psi_k^t \left(PA_k - PA_{k+1}\right) + \psi_{k+1}^t PA_{k+1}\right) \tag{4}$$

where psi_k^t is the variable at t-dt (t), $\psi_k^{t+\Delta t}$ the supplementary sources of psi_k^t (and not $psi_k^t \rho_d J_k$) and the right term on the right is the explicit part of the vertical turbulent diffusion.

$$ZY_b = \psi_b^t + 2\Delta t \psi_b^{t+\Delta t} - \frac{\delta_{expl}}{\rho_d J_b} \left(\psi_b^t + \psi_{b+1}^t\right) PA_{b+1} \tag{5}$$

For the first level, only the upper part is considered, the lower one is replaced by the turbulent surface flux (taken into account in the $\psi_b^{t+\Delta t}$ term).

$$ZY_e = \psi_e^t + 2\Delta t \psi_e^{t+\Delta t} + \frac{\delta_{expl}}{\rho_d J_e} \left(\psi_e^t + \psi_{e-1}^t\right) PA_e \tag{6}$$

For the last level, only the lower part is considered, the upper one is replaced by the turbulent flux which is taken equal to 0 (taken into account in the $\psi_e^{t+\Delta t}$ term). Then, the classical tridiagonal algorithm is used to invert the implicit operator. Its matrix is given by:

$$\begin{pmatrix}
b_b & c_b & 0 & 0 & 0 & \dots & 0 & 0 & 0 \\
a_{b+1} & b_{b+1} & c_{b+1} & 0 & 0 & \dots & 0 & 0 & 0 \\
0 & a_{b+2} & b_{b+2} & c_{b+2} & 0 & \dots & 0 & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & \dots & 0 & a_k & b_k & c_k & 0 & \dots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & 0 & \dots & a_{e-1} & b_{e-1} & c_{e-1} \\
0 & 0 & 0 & 0 & 0 & \dots & 0 & a_e & b_e
\end{pmatrix}$$

b and e represent the first and the last inner mass levels of the model. The coefficients are:

$$a_k = \frac{\delta_{impl}}{\rho_d J_k} PA_k \tag{7}$$

$$b_k = 1 - \frac{\delta_{impl}}{\rho_d J_k} PA_k - \frac{\delta_{impl}}{\rho_d J_k} PA_{k+1} \tag{8}$$

$$c_k = \frac{\delta_{impl}}{\rho_d J_k} PA_{k+1} \tag{9}$$

for all $k \neq b$ or e

$$b_b = 1 - \frac{\delta_{impl}}{\rho_d J_b} PA_{b+1} \tag{10}$$

$$c_b = \frac{\delta_{impl}}{\rho_d J_b} PA_{b+1} \tag{11}$$

(discretization of the upper part of the implicit operator)

$$b_e = 1 - \frac{\delta_{impl}}{\rho_d J_e} PA_e \tag{12}$$

$$a_e = \frac{\delta_{impl}}{\rho_d J_e} PA_e \tag{13}$$

(discretization of the lower part of the implicit operator)
The matrix is solved in two loops. First bottom to top:

$$\beta_b = 1 - \frac{\delta_{impl}}{\rho_d J_b} PA_{b+1} \tag{14}$$

$$\psi_b^t = \frac{Y_b}{\beta_b} \tag{15}$$

$$\gamma_k = \frac{\delta_{impl}}{\rho_d J_k} \frac{PA_k}{\beta_{k-1}} \tag{16}$$

$$\beta_k = 1 - \frac{\delta_{impl}}{\rho_d J_k} \left(PA_k \left(1 + \gamma_k\right) + PA_k\right) \tag{17}$$

$$\psi_k = \frac{1}{\beta_k} \left(Y_k - \frac{\delta_{impl}}{\rho_d J_k} PA_k \psi_{k-1}\right) \tag{18}$$

$$\gamma_e = \frac{\delta_{impl}}{\rho_d J_{e-1}} \frac{PA_e}{\beta_{e-1}} \tag{19}$$

$$\beta_e = 1 - \frac{\delta_{impl}}{\rho_d J_e} PA_e \left(1 + \gamma_e\right) \tag{20}$$

$$\psi_e = \frac{1}{\beta_e} \left(Y_e - \frac{\delta_{impl}}{\rho_d J_e} PA_e \psi_{e-1}\right) \tag{21}$$

and then from top to bottom

$$\psi_k = \psi_k - \gamma_{k+1} \psi_{k+1} \tag{22}$$

the marginal points are prescribed $\psi_{b-1} = \psi_b$ and $\psi_{e+1} = \psi_e$.

7.5 Subroutine tridiag_wind

Works the same way as tridiag. The bottom coefficient b_b is different.

$$b_b = 1 - \frac{\delta_{impl}}{\rho_d J_b} PA_{b+1} - \delta_{impl} C_d \tag{23}$$

The bottom β coefficient in the bottom to top loop is calculated differently.

$$\beta_b = 1 - \delta_{impl} \left(\frac{PA_{b+1}}{\rho_d J_b} + C_d 2\Delta t \right) \tag{24}$$

This subroutine is used only for u and v, there is no turbulent diffusion of the w component.

7.6 Subroutine tridiag_tke

Here ψ is turbulent kinetic energy. Matrix contains additional term linked to implicit dissipation. The right hand side of the implicit equation is computed in the same way as for θ_l and r_v. The same tridiagonal algorithm is used to invert the implicit operator. Its matrix is the same as for θ_l and r_v. The coefficients are the same, except for b_k:

$$b_k = 1 + \frac{\delta_{impl}}{\rho_d J_k} \epsilon_k - \frac{\delta_{impl}}{\rho_d J_k} PA_k - \frac{\delta_{impl}}{\rho_d J_k} PA_{k+1} \tag{25}$$

for all $k \neq b$ or e

$$b_b = 1 + \frac{\delta_{impl}}{\rho_d J_b} \epsilon_b - \frac{\delta_{impl}}{\rho_d J_b} PA_{b+1} \tag{26}$$

(discretization of the upper part of the implicit operator)

$$b_e = 1 + \frac{\delta_{impl}}{\rho_d J_e} \epsilon_e - \frac{\delta_{impl}}{\rho_d J_e} PA_e \tag{27}$$

(discretization of the lower part of the implicit operator)
The matrix is solved in two loops as before. Only the β coefficients in the bottom to top loop are calculated differently.

$$\beta_b = 1 + \frac{\delta_{impl}}{\rho_d J_b} \left(\epsilon_b - PA_{b+1} \right) \tag{28}$$

$$\beta_k = 1 + \frac{\delta_{impl}}{\rho_d J_k} \left(\epsilon_k - PA_k \left(1 + \gamma_k \right) + PA_k \right) \tag{29}$$

$$\beta_e = 1 + \frac{\delta_{impl}}{\rho_d J_e} \left(\epsilon_e PA_e \left(1 + \gamma_e \right) \right) \tag{30}$$

The loop from top to bottom is the same as for other variables. The marginal points are prescribed in the same way.

8 Subroutine ac_rain_ice

Subroutin ac_rain_ice computes the microphysical sources related to the resolved clouds and precipitation it computes the real absolute pressure, negative values of the current guess of all mixing ratio are removed. This is done by a global filling algorithm based on a multiplicative method (Rood, 1987), in order to conserve the total mass in the simulation domain. Sources are transformed in physical tendencies, by removing the multiplicative term $\rho_d J$. After calling to microphysical routines, the physical tendencies are switched back to prognostic variables.

1. some dimensioning, options hardcoding and preliminary computations. IKB that should be the bottom of the atmosphere is $IKB = 1+JPVEXT$ increased by $JPVEXT$ meaning that we do not start calculations from the bottom of the atmosphere but some level above if $JPVEXT \neq 0$. The top of the atmosphere IKE is determined by $IKE = size(z,3) - JPVEXT$ meaning that the uppermost level corresponds to the uppermost level of z shifted downwards by $JPVEXT$. $HCLOUD$ is hardcoded to $ICE3$.
2. some computations of local relevant variables:

$$T = \theta\pi_{ref} \qquad L_v = L_{VTT}+(c_{pv}-c_l)(T-T_{TT}) \qquad L_s = L_{STT}+(c_{pv}-c_i)(T-T_{TT}) \qquad c_{ph} = c_{pd}+c_{pv}2\Delta t r_v^{t+\Delta t}$$

3. negative values of the forecast values of the precipitation species $r_j^{t+\Delta t}$ for j=3,5,6,7 are removed applying simple mass correction. However, this correction is not applied if $HCLOUD = ICE3$ (as hardcoded) so this correction is not used.
4. And
   - Negative values of prognostic cloud ice are corrected and prognostic values water vapour and potential temperature are modified accordingly. The corrections are applied only where $r_i^{t+\Delta t} < 0$.

   $$r_v^{t+\Delta t} = r_v^{t+\Delta t} + r_i^{t+\Delta t} \qquad \theta^{t+\Delta t} = \theta^{t+\Delta t} - r_i^{t+\Delta t}\frac{L_s}{c_{ph}}\frac{1}{\pi_{ref}} \qquad r_i^{t+\Delta t} = 0$$

   - Negative values of prognostic cloud water are corrected and prognostic values of water vapour and potential temperature are modified accordingly. The corrections are applied only where $r_c^{t+\Delta t} < 0$.

   $$r_v^{t+\Delta t} = r_v^{t+\Delta t} + r_c^{t+\Delta t} \qquad \theta^{t+\Delta t} = \theta^{t+\Delta t} - r_c^{t+\Delta t}\frac{L_v}{c_{ph}}\frac{1}{\pi_{ref}} \qquad r_c^{t+\Delta t} = 0$$

   - Negative values of prognostic water vapour are corrected and prognostic values of cloud water and potential temperature are modified accordingly. The corrections are applied only where $r_v^{t+\Delta t} < 0$ and $r_c^{t+\Delta t} > 0$.

   $$r_v^{t+\Delta t} = r_v^{t+\Delta t} + r_c^{t+\Delta t} \qquad \theta^{t+\Delta t} = \theta^{t+\Delta t} - r_c^{t+\Delta t}\frac{L_v}{c_{ph}}\frac{1}{\pi_{ref}} \qquad r_c^{t+\Delta t} = 0$$

   - Negative values of prognostic water vapour are corrected and prognostic values of cloud ice and potential temperature are modified accordingly. The corrections are applied only where $r_v^{t+\Delta t} < 0$ and $r_i^{t+\Delta t} > 0$ if $KRR > 3$.

   $$ZC = min(-r_v^{t+\Delta t}, r_i^{t+\Delta t}) \qquad r_v^{t+\Delta t} = r_v^{t+\Delta t}+ZC \qquad \theta^{t+\Delta t} = \theta^{t+\Delta t}-ZC\frac{L_s}{c_{ph}}\frac{1}{\pi_{ref}} \qquad r_i^{t+\Delta t} = r_i^{t+\Delta t}-ZC$$

5. CALL RAIN_ICE

NOTE: There is stil a possibility that there are some places where r_v,r_c or r_i would have negative values. r_v can become negative after first two corrections and can not be restored to get the positive value by the other two corrections since then $r_c = 0$ and $r_i = 0$. In correction 3 all cloud water is evaporated to correct for negative r_v but in correction 4 only part of the cloud ice is sublimated - just enough to restore r_v to zero (unless there is not enough r_i). In correction 3 all cloud water is evaporated to correct for negative r_v but in correction 4 only part of the cloud ice is sublimated - just enough to restore r_v to zero (unless there is not enough r_i). The 3rd correction could give $r_v > r_{vs}$ mixing ratio of water vapor larger than saturated and $r_c = 0$. The 4th correction gives $r_i > 0$ presence of cloud ice when $r_v = 0$ there is no water vapour.

9 Remarks

Parameters controlling deep and shallow convection (and ensemble deep convection) have to be initialized before each call to the corresponding subroutine because parameters have same name but different value. It would be more transparent if the parameters would be initialized only once and have different names in different parameterisations. Subroutine rain_ice contains subroutines rain_ice_sedimentation, rain_ice_nucleation, rain_ice_slow, rain_ice_warm, rain_ice_fast_rs, rain_ice_fast_rg and rain_ice_fast_ri.