Hgrad parametrization in AROME

LACE stay report
Toulouse – Météo-France/CNRS,
05. Nov. 2018 – 30. Nov. 2018

Scientific supervisor: Rachel Honnert (CNRM UMR 3589, Météo-France/CNRS)

Report made by: Dávid Lancz (HMS – Hungarian Meteorological Service)

Contents:

1) Introduction

The Hgrad parametrization of turbulence (Moeng 2014, Verrelle et al. 2017) is an alternative approach to the turbulence beside the more common Kgrad method. In the Kgrad parametrization the vertical turbulent fluxes are computed from the vertical gradients:

$$\overline{w'\theta'_l} = \frac{-2}{3C_s} L \sqrt{e_{ref}} \frac{\partial \theta_l}{\partial z} \varphi_i \quad , \qquad (1)$$

$$\overline{w'r'_{np}} = \frac{-2}{3C_h} L \sqrt{e_{ref}} \frac{\partial r_{np}}{\partial z} \psi_i \quad , \qquad (2)$$

where θ_l and r_{np} are the conservative variables: liquid-ice potential temperature and total nonprecipitating water mixing ratio, $\overline{w'\theta'_l}$ and $\overline{w'r'_{np}}$ are their turbulent vertical fluxes and $\frac{\partial \theta_l}{\partial z}$ and $\frac{\partial r_{np}}{\partial z}$ are their vertical gradients. *L* is the Bougeault and Lacarrère mixing length, C_s and C_h are numeric constants, e_{ref} is the turbulent kinetic energy and φ_i and ψ_i are stability functions.

On the contrary to the Kgrad the Hgrad parametrization uses the horizontal gradients:

$$\overline{w'\theta'_l} = C_1 \left(\frac{\partial w}{\partial x} \frac{\partial \theta_l}{\partial x} + \frac{\partial w}{\partial y} \frac{\partial \theta_l}{\partial y} \right) \quad , \qquad (3)$$

$$\overline{w'r'_{np}} = C_2 \left(\frac{\partial w}{\partial x} \frac{\partial r_{np}}{\partial x} + \frac{\partial w}{\partial y} \frac{\partial r_{np}}{\partial y} \right) \quad , \qquad (4)$$

where C_1, C_2 are numeric constants and *x*, *y* are the zonal and meridional directions.

The aim of the Hgrad parametrization is to better treat the mixing inside the deep convection, thus the Hgrad parametrization only applies in grid-boxes where the altitude and the mixing ratio both reach the given threshold values.

The goal of the LACE stay was to implement the Hgrad parametrization into the AROME. The MesoNH already contains it and it was also tested (see Verrelle et al. 2017).

2) Modification

The AROME uses the physical parametrization package of the MesoNH but there are important differences between the two model. They have different type of vertical coordinate systems and while the AROME is a spectral model, the MesoNH computes the gradients in the grid-mesh. The 3 dimensional (3D) variables need to be converted before the computation of the physical parametrization and re-converted after it.

Inside the code on the level of the cpg.F90 routine is the mf_phys_prep.F90 routine and after it the mf_phys.F90. The mf_phys.F90 is responsible for the computation of the physics and the mf_phys_prep.F90 makes preliminary preparations for the mf_phys.F90.

The actual computation of the vertical turbulent fluxes happens in the turb_ver_thermo_flux.F90 routine. The way of the variables from the cpg.F90 to the urb_ver_thermo_flux.F90 routine is:

```
├ …
├ cpg.F90
└ …     ├ …
        ├ mf_phys_prep.F90
        ├ mf_phys.F90
        └ …     ├ …
                ├ apl_arome.F90
                └ …     ├ …
                        ├ aro_turb_mnh.F90
                        └ …     ├ …
                                ├ turb.F90
                                └ …     ├ …
                                        ├ turb_ver.F90
                                        └ …     ├ …
                                                ├ turb_ver_thermo_flux.F90
                                                └ …
```

The main modifications were made in mf_phys_prep.F90, aro_turb_mnh.F90 and turb_ver_thermo_flux.F90. In the other routines only minimal changes were made, in order to make them transfer the 3D variables from the level of the cpg.F90 to turb_ver_thermo_flux.F90.

The Hgrad parametrization needs horizontal gradients. To provide them, to the code of the mf_phys_prep.F90 were added a part, where these gradients are computed. The gradients of *wg* (*w* – vertical velocity, *g* – grav. acceleration) were already available in mf_phys_prep.F90, but the gradients of liquid-ice potential temperature (θ_l) had to be prepared. The problem with this was that in the current code, the horizontal gradients of the water variables are not available (hopefully this can change in some years). Due to this fact the horizontal gradient of potential temperature was provided to the Hgrad parametrization (as it was only a dry case). This was done by using the equations of the potential temperature (θ) and gradient of potential temperature (here only the *x* component):

$$\theta = T(p_0/p)^{\kappa} \quad , \qquad (5)$$

$$\frac{\partial\theta}{\partial x} = \frac{\partial T}{\partial x}(p_0/p)^{\kappa} - \kappa T p_0^{\kappa}\left(\frac{\partial p}{\partial x}\right)^{-(\kappa+1)} = p_0^{\kappa}\left(\frac{\partial T}{\partial x}p^{-\kappa} - \left(\frac{\partial p}{\partial x}\right)^{-(\kappa+1)}\kappa T\right) \quad , \qquad (6)$$

where p_0=1000 hPa, *T* is the temperature, *p* is the hydrostatic pressure and κ is the Poisson constant – the ratio of the gas constant (*R*) to the specific heat capacity of dry air at constant pressure (C_{pd}).

If the horizontal gradients of the mixing ratios were available, the computation of the gradient of the liquid-ice potential temperature would be possible by the equation:

$$\frac{\partial\theta_l}{\partial x} \approx \frac{\partial\theta}{\partial x} - \frac{L_v}{C_{pd}}\frac{\partial r_l}{\partial x} - \frac{L_s}{C_{pd}}\frac{\partial r_i}{\partial x} \quad , \qquad (7)$$

where L_v (L_s) is the latent heat of vaporization (sublimation) and r_l (r_i) is the mixing ratio of liquid water (ice).

Converting horizontal gradient components of scalar function *A* between *s=s(x,y,z,t)* and *z=z(x,y,s,t)* vertical coordinate systems is possible through the equation (Kasahara 1974):

$$\left(\frac{\partial A}{\partial c}\right)_s = \left(\frac{\partial A}{\partial c}\right)_z + \frac{\partial A}{\partial z}\left(\frac{\partial z}{\partial c}\right)_s \quad , \qquad (8)$$

where *c* can be *x*, *y* or *t*. Thus the used equations in the code were

$$\left.\frac{\partial(wg)}{\partial x}\right|_z = \left.\frac{\partial(wg)}{\partial x}\right|_\sigma - \frac{\partial\varphi}{\partial x}\frac{\Delta(wg)}{\Delta\varphi} \qquad (9)$$

and

$$\left.\frac{\partial\theta}{\partial x}\right|_z = \left.\frac{\partial\theta}{\partial x}\right|_\sigma - \frac{\partial\varphi}{\partial x}\frac{\Delta\theta}{\Delta\varphi} \quad , \qquad (10)$$

where φ is the geopotential.

1) Actual code modifications in **mf_phys_prep.F90**:

Some variables:

```
PKAP0(YDGEOMETRY%YRDIM%NPROMA,YDGEOMETRY%YRDIMV%NFLEVG) !KAPPA at t, full lev
PRE0L(YDGEOMETRY%YRDIM%NPROMA) !surf pres
PRE0M(YDGEOMETRY%YRDIM%NPROMA) !surf pres
ZTI(YDGEOMETRY%YRDIM%NPROMA,0:YDGEOMETRY%YRDIMV%NFLEVG) !temp. at t, half lev
ZKAP0H(YDGEOMETRY%YRDIM%NPROMA,0:YDGEOMETRY%YRDIMV%NFLEVG) !KAPPA at t, half lev
ZTHH(YDGEOMETRY%YRDIM%NPROMA,0:YDGEOMETRY%YRDIMV%NFLEVG) !potential temp (th) at t, half lev
ZPRESL(YDGEOMETRY%YRDIM%NPROMA,YDGEOMETRY%YRDIMV%NFLEVG) !zonal comp. of "grad temp" at t, full lev
ZPRESM(YDGEOMETRY%YRDIM%NPROMA,YDGEOMETRY%YRDIMV%NFLEVG) !meridian comp. of "grad temp" at t, full
lev
ZTHGRLF(YDGEOMETRY%YRDIM%NPROMA,YDGEOMETRY%YRDIMV%NFLEVG) !zonal comp. of "grad th" at t, full lev
ZTHGRMF(YDGEOMETRY%YRDIM%NPROMA,YDGEOMETRY%YRDIMV%NFLEVG) !meridian comp. of "grad th" at t, full
lev
ZTHGRLLF(YDGEOMETRY%YRDIM%NPROMA,YDGEOMETRY%YRDIMV%NFLEVG) !zonal comp. of "grad th_l" at t, full
lev
ZTHGRMLF(YDGEOMETRY%YRDIM%NPROMA,YDGEOMETRY%YRDIMV%NFLEVG) !meridian comp. of "grad th_l" at t, full
lev
```

Computation of potential temperatures:

```
  ZTI(KST:KEND,0)=PGMV(KST:KEND,1,YT0%MT)
  ZKAP0H(KST:KEND,0)=PKAP0(KST:KEND,1)
  ZTI(KST:KEND,NFLEVG)=PGMV(KST:KEND,NFLEVG,YT0%MT)
  ZKAP0H(KST:KEND,NFLEVG)=PKAP0(KST:KEND,NFLEVG)
DO JLV=1,NFLEVG-1
  ZTI(KST:KEND,JLV)=(PGMV(KST:KEND,JLV,YT0%MT)+PGMV(KST:KEND,JLV+1,YT0%MT))*0.5_JPRB
  ZKAP0H(KST:KEND,JLV)=(PKAP0(KST:KEND,JLV)+PKAP0(KST:KEND,JLV+1))*0.5_JPRB
ENDDO

DO JLV=0,NFLEVG
  ZTHH(KST:KEND,JLV)= RATM**ZKAP0H(KST:KEND,JLV) * ZTI(KST:KEND,JLV) * PRE0(KST:KEND,JLV)**(-
ZKAP0H(KST:KEND,JLV))
ENDDO
```

Computation of potential temperature gradients:

```
DO JLV=1,NFLEVG
  ZVBF=(YDGEOMETRY%YRVAB%VBH(JLV-1)+YDGEOMETRY%YRVAB%VBH(JLV))*0.5_JPRB
  ZPRESM(KST:KEND,JLV) = ZVBF*PRE0M(KST:KEND)
  ZPRESL(KST:KEND,JLV) = ZVBF*PRE0L(KST:KEND)
ENDDO
```

```
DO JLV=1,NFLEVG
  ZTHGRLF(KST:KEND,JLV)= RATM**PKAP0(KST:KEND,JLV) * ( &
 & PGMV(KST:KEND,JLV,YT0%MTL) * PRE0F(KST:KEND,JLV)**(-PKAP0(KST:KEND,JLV)) - &
 & ZPRESL(KST:KEND,JLV)**(-PKAP0(KST:KEND,JLV)-1.0_JPRB) * PKAP0(KST:KEND,JLV) *
PGMV(KST:KEND,JLV,YT0%MT))
  ZTHGRMF(KST:KEND,JLV)= RATM**PKAP0(KST:KEND,JLV) * ( &
 & PGMV(KST:KEND,JLV,YT0%MTM) * PRE0F(KST:KEND,JLV)**(-PKAP0(KST:KEND,JLV)) - &
 & ZPRESM(KST:KEND,JLV)**(-PKAP0(KST:KEND,JLV)-1.0_JPRB) * PKAP0(KST:KEND,JLV) *
PGMV(KST:KEND,JLV,YT0%MT))
ENDDO
```

Conversion between z and mass-based hybrid pressure terrain-following vertical coordinates systems:

```
! d(wg)/dx_z
PGRADH_PHY(KST:KEND,JLV,5)=PGWFL(KST:KEND,JLV)        &
 & -PHI0FL(KST:KEND,JLV)*ZDELTA_MWGF(KST:KEND,JLV)/ZDELTA_PHIF(KST:KEND,JLV)
! d(wg)/dy_z
PGRADH_PHY(KST:KEND,JLV,6)=PGWFM(KST:KEND,JLV)        &
 & -PHI0FM(KST:KEND,JLV)*ZDELTA_MWGF(KST:KEND,JLV)/ZDELTA_PHIF(KST:KEND,JLV)
! d(theta_l)/dx_z !WRONG not theta_l
PGRADH_PHY(KST:KEND,JLV,7)=ZTHGRLF(KST:KEND,JLV) &
 & -PHI0FL(KST:KEND,JLV)*ZDELTA_MTHF(KST:KEND,JLV)/ZDELTA_PHIF(KST:KEND,JLV)
! d(theta_l)/dy_z !WRONG not theta_l
PGRADH_PHY(KST:KEND,JLV,8)=ZTHGRMF(KST:KEND,JLV) &
  &-PHI0FM(KST:KEND,JLV)*ZDELTA_MTHF(KST:KEND,JLV)/ZDELTA_PHIF(KST:KEND,JLV)
```

2) Actual code modifications in **aro_turb_mnh.F90**:

Transforms from AROME arrays to MesoNH arrays:

```
DO JTR=1,4
CALL ADD_BOUNDS(1,KLON,KLEV,1,PHGRAD(:,:,:,JTR),ZHGRAD(:,:,:,JTR))
ENDDO
```

3) Actual code modifications in **turb_ver_thermo_flux.F90**:

New variables:

```
  REAL, DIMENSION(SIZE(PTHLM,1),SIZE(PTHLM,2),SIZE(PTHLM,3))  :: ZF_NEW, ZMG_COEFF
  REAL, DIMENSION(SIZE(PTHLM,1),SIZE(PTHLM,2),SIZE(PTHLM,3))  :: ZRWTHL, ZRWRNP
  REAL, DIMENSION(SIZE(PTHLM,1),SIZE(PTHLM,2),SIZE(PTHLM,3))  :: ZCLD_THOLD
  REAL, DIMENSION(SIZE(PZZ,1),SIZE(PZZ,2),SIZE(PZZ,3))  :: ZALT

  REAL, DIMENSION(:,:,:,:)   ,INTENT(IN) :: PHGRAD
  REAL, DIMENSION(:)         ,INTENT(IN) :: POROG
```

Conditions:

```
IF ( KRRL >= 1 ) THEN
  IF ( KRRI >= 1 ) THEN
    ZCLD_THOLD(:,:,:)  = PRM(:,:,:,2)  + PRM(:,:,:,4)
  ELSE
    ZCLD_THOLD(:,:,:)  = PRM(:,:,:,2)
  END IF
END IF
```

Computation of the new flux and tendency (conservative potential temperature):

```
XDUMMY4=2000.
XDUMMY5=422500.
XDUMMY6=422500.
XDUMMY8=-1.0
LDUMMY8=.TRUE.

IF (LDUMMY8) THEN
ZF_NEW (:,:,:)= XDUMMY5*((PHGRAD(:,:,:,1)*PHGRAD(:,:,:,3)/XG+            &
                         PHGRAD(:,:,:,2)*PHGRAD(:,:,:,4))/XG)
END IF

…

ZRWTHL(:,:,:)= PRHODJ(:,:,:)*(PTHLP(:,:,:)-PTHLM(:,:,:))/PTSTEP

IF (LDUMMY8) THEN
DO JK=KKU,KKA
  ZALT(:,1,JK)=PZZ(:,1,JK)-POROG(:)/XG
END DO

WHERE ( (ZCLD_THOLD >= XDUMMY8 ) .AND. ( ZALT(:,:,:) >= XDUMMY4 ) )
ZRWTHL(:,:,:) = -PRHODJ(:,:,:)*ZF_NEW(:,:,:)
END WHERE
END IF

PRTHLS(:,:,:)= PRTHLS(:,:,:)  + ZRWTHL(:,:,:)

…

IF (LDUMMY8) THEN
WHERE ( (ZCLD_THOLD >= XDUMMY8 ) .AND. ( ZALT(:,:,:) >= XDUMMY4 ) )
ZFLXZ(:,:,:)      = ZF_NEW(:,:,:)
END WHERE
END IF
```

Computation of the new flux and tendency (conservative mixing ratio):

```
IF (LDUMMY8) THEN
ZF_NEW (:,:,:)= XDUMMY6*((PHGRAD(:,:,:,1)*PHGRAD(:,:,:,3)/XG+            &
                         PHGRAD(:,:,:,2)*PHGRAD(:,:,:,4))/XG)
ENDIF

…

ZRWRNP (:,:,:)= PRHODJ(:,:,:)*(PRP(:,:,:)-PRM(:,:,:,1))/PTSTEP

IF (LDUMMY8) THEN
DO JK=KKU,KKA
  ZALT(:,1,JK)=PZZ(:,1,JK)-POROG(:)/XG
END DO

WHERE ( (ZCLD_THOLD >= XDUMMY8 ) .AND. ( ZALT(:,:,:) >= XDUMMY4 ) )
ZRWRNP (:,:,:) =  -PRHODJ(:,:,:)*ZF_NEW(:,:,:)
END WHERE
END IF

PRRS(:,:,:,1) = PRRS(:,:,:,1) + ZRWRNP (:,:,:)

…

IF (LDUMMY8) THEN
  WHERE ( (ZCLD_THOLD >= XDUMMY8 ) .AND. ( ZALT(:,:,:) >= XDUMMY4 ) )
   ZFLXZ(:,:,:)      = ZF_NEW(:,:,:)
  END WHERE
END IF
```

3) Experiments

The Hgrad parametrization were planed to be tested on a case with heavy precipitation in the Mediterranean sea on 17th of September 2014 (Fig. 1). Unfortunately the implementation of the Hgrad parametrization was not successful, the “segmentation fault” error occurred and it was not fixed before the end of the stay.

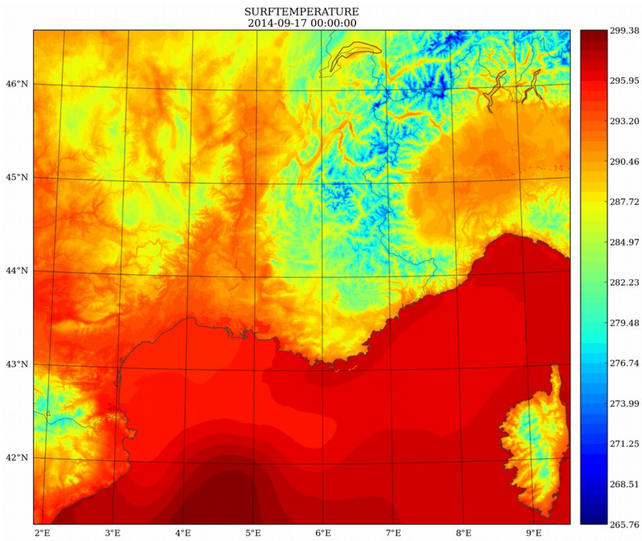

Fig. 1) The domain of the planed experiments

4) Summary

Hgrad is a turbulence parametrization which uses the horizontal gradients and showed promising results in the treatment of turbulence inside convective clouds (Verrelle et al. 2017). It was implemented into the AROME model, but unfortunately the “segmentation fault” error occurred during the testing and this problem was not solved before the end of the stay. Further developments are required.

6) Acknowledgments

Many thanks to Rachel Honnert for her help during my stay.

7) References

AKIRA KASAHARA (1974): *Various Vertical Coordinate Systems Used for Numerical Weather Prediction.* Mon. Wea. Rev. **102**, 509-522

ANTOINE VERRELLE, DIDIER RICARD and CHRISTINE LAC (2017): *Evaluation and Improvement of Turbulence Parameterization inside Deep Convective Clouds at Kilometer-Scale Resolution.* Mon. Wea. Rev. **145**, 3947-3967

CHIN-HOH MOENG 1984: *A large-eddy-simulation model for the study of planetary boundary-layer turbulence.* J. Atmos. Sci., **41**, 2052–2062