

**Extending the functionality of “convection diagnostics” in
the ALADIN/ALARO/AROME FullPos software**

Joined LACE report by Christoph Wittmann (CW) and Jure
Cedilnik (JC)

Part 1 (CW):

Report of LACE stay at Slovenian Environment Agency, Ljubljana
10.06.2013 – 21.06.2013

1. Mixed Layer CAPE

a. Methodology

The convective available potential energy (CAPE) is one important parameter to estimate the potential for deep convection from observed radiosoundings or pseudotemps created out of numerical weather prediction models. CAPE should give an estimate of the amount of energy available, once a given a parcels makes is way above the level of free convection. The way to compute CAPE is defined as the vertical integral of the local buoyancy of the rising parcel with respect to the environment (1). There exist different approaches to describe the characteristics of the parcel which is finally entering CAPE computation.

$$CAPE = \int_{z_f}^{z_n} g \left(\frac{T_{v,parcel} - T_{v,env}}{T_{v,env}} \right) dz \quad (1)$$

The current CAPE computation in the FullPos software offers the following possibilities:

- Lowest model level CAPE: Temperature and (specific) humidity of the lowest model level are used to characterize the parcel being lifted
- Screening level (2m) CAPE: A parcel characterized by 2m temperature and 2m humidity is lifted
- Most unstable parcel CAPE (MUCAPE): Within a defined layer depth (default: 300hPa above surface, namelist parameter GCAPEPSD), the parcel/level resulting in the largest CAPE value is searched

Another approach is the so called mixed layer CAPE: The parcel entering CAPE computation (starting from the surface) is characterized by the temperature/humidity which is valid for a well-mixed layer of typically 50-100hPa depth above the surface. This means that the layer is mixed by averaging potential temperature and specific humidity from the model surface to 50-100 hPa over the ground.

b. Implementation into Fullpos

CAPE is computed in the routines *fpcica.F90* (preparatory computations) and *fpcincape.F90* (vertical integration to compute CAPE and CIN). In order to implement a mixed layer cape the following routines were modified:

- *fpcica.f90*: add KCAPETYPE option 6 to *fpcica.F90* including a vertical averaging of Tpot and q within a defined layer over model surface (default: 100 hPa)
- *sufpc.f90*: allow NFPCAPE=6 in setup and abort in case the depth of the mixing layer is chosen too big (namelist parameter GCAPEPSD)
- *endpos.f90* + *phympos.f90*: add kappa as an argument given to *fpcica..F90* (needed to compute potential temperature)

Appendix A shows the relevant part of code being responsible for the computation of the mixed layer value entering the final computation in *fpcincape.F90*

c. Results

After technical validation of the new code fragments, the new CAPE version was tested for a representative case: 20.06.2013. This day can be characterized as the peak day of a heat wave in great parts of Central Europe showing quite extreme values for CAPE. The different CAPE version can be seen in Figures 1 – 6. It is obvious that the CAPE values computed with mixed layer versions (Figure 5, Figure 6) are in general lower than other, surface based versions. The code is ready for operational testing and validation.

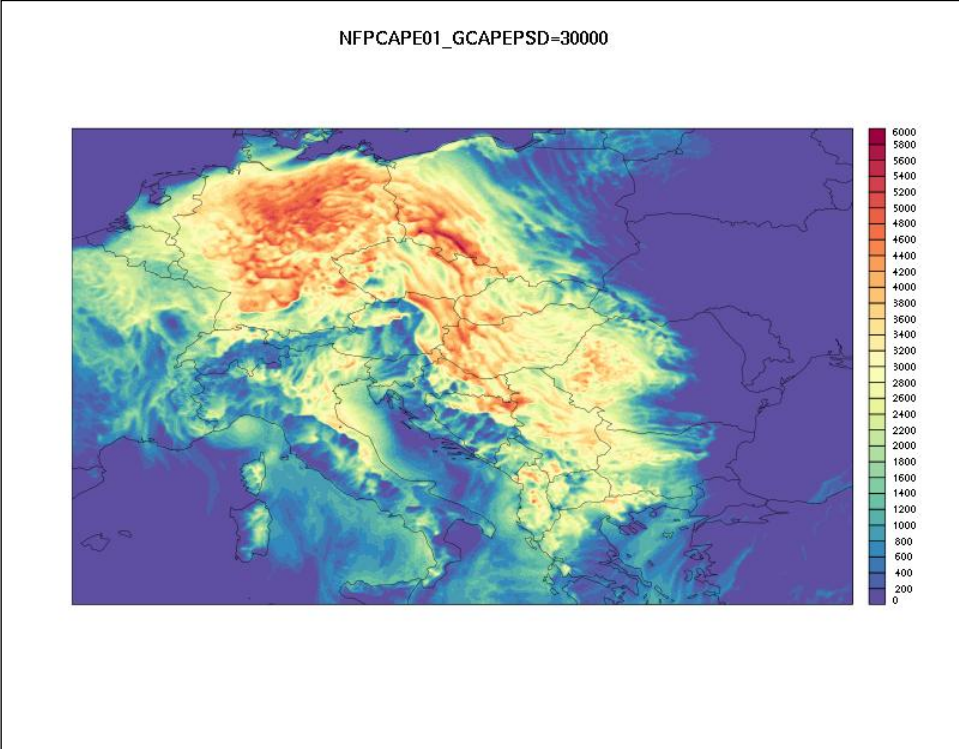


Figure 1: CAPE based on lowest model level for 201300620 15 UTC

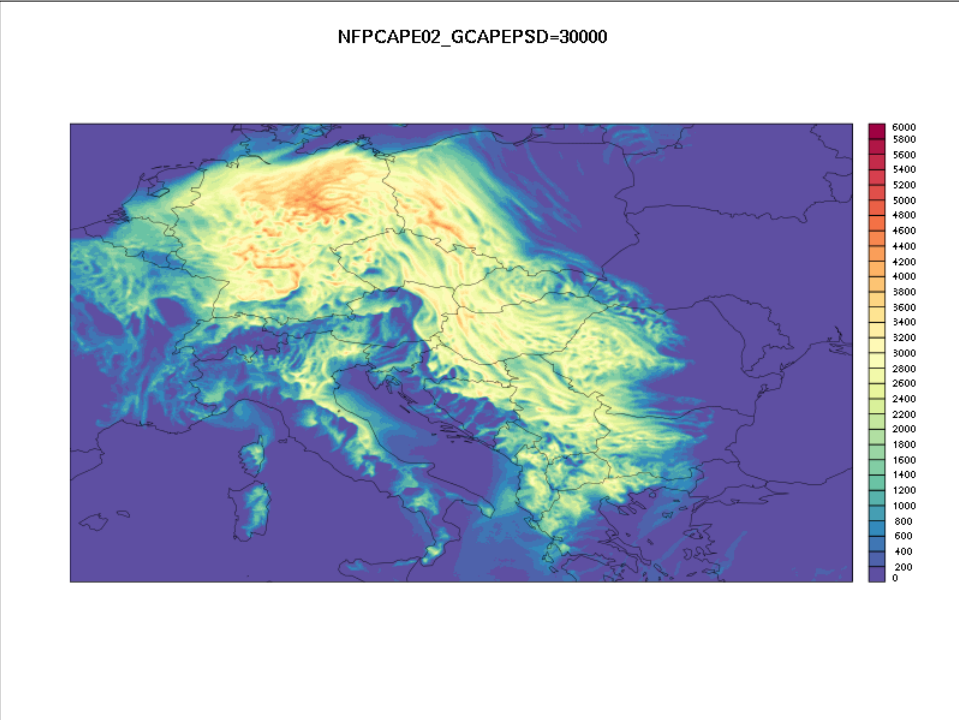


Figure 2: CAPE based on most unstable layer (or model level) for 201300620 15 UTC

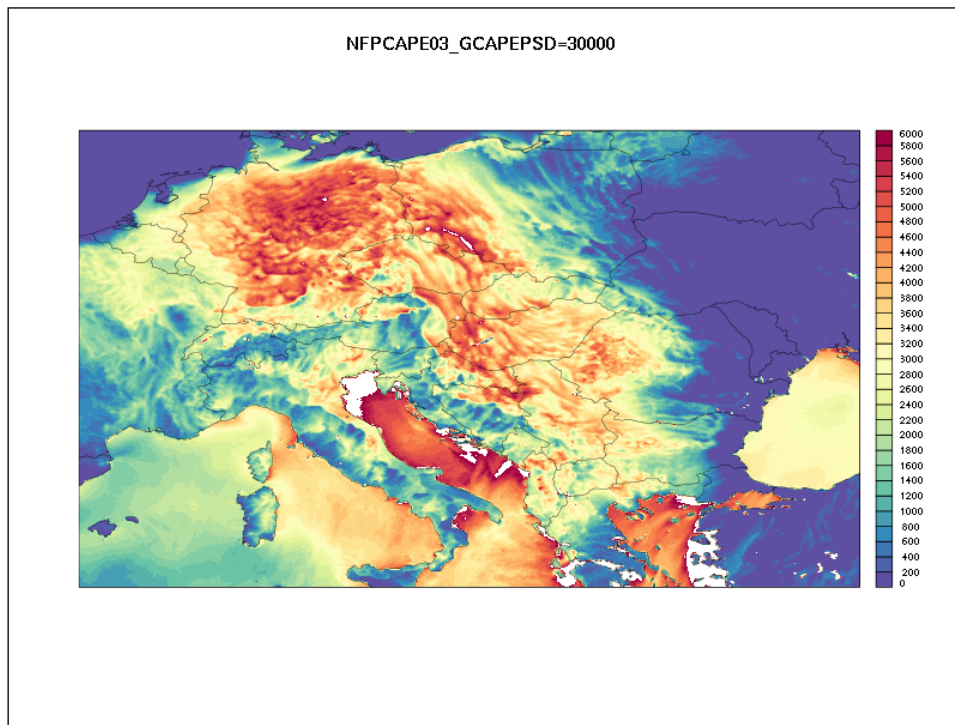


Figure 3: CAPE based on screening level (2m) (NFPCAPE=3) for 201300620 15 UTC

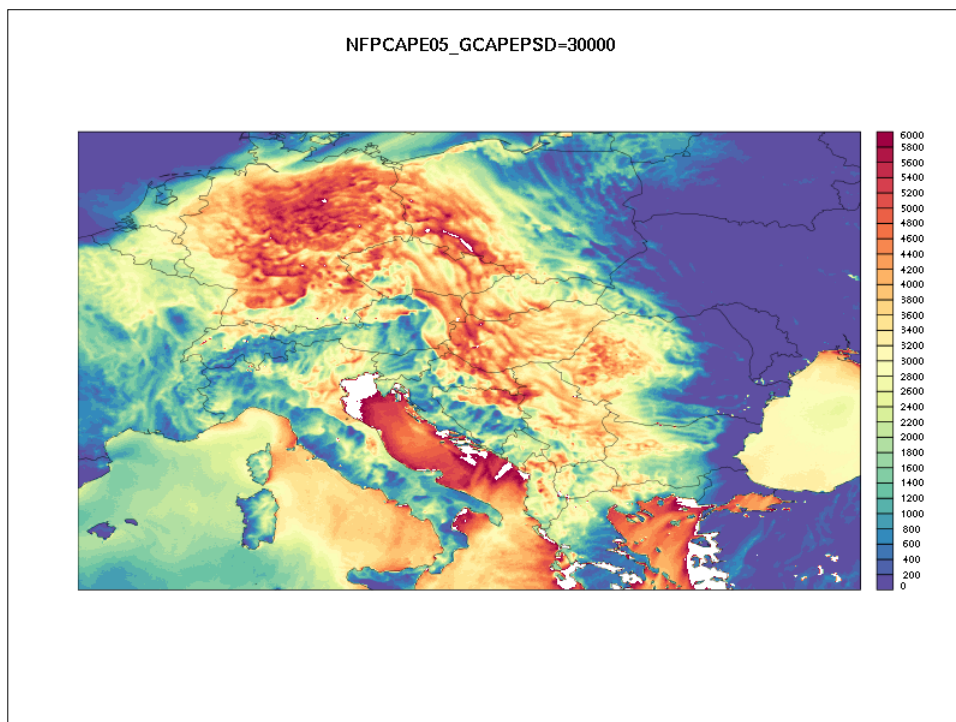


Figure 4: CAPE based most unstable layer including screening level (NFPCAPE=5, GCAPEPSD=30000.) for 201300620 15 UTC

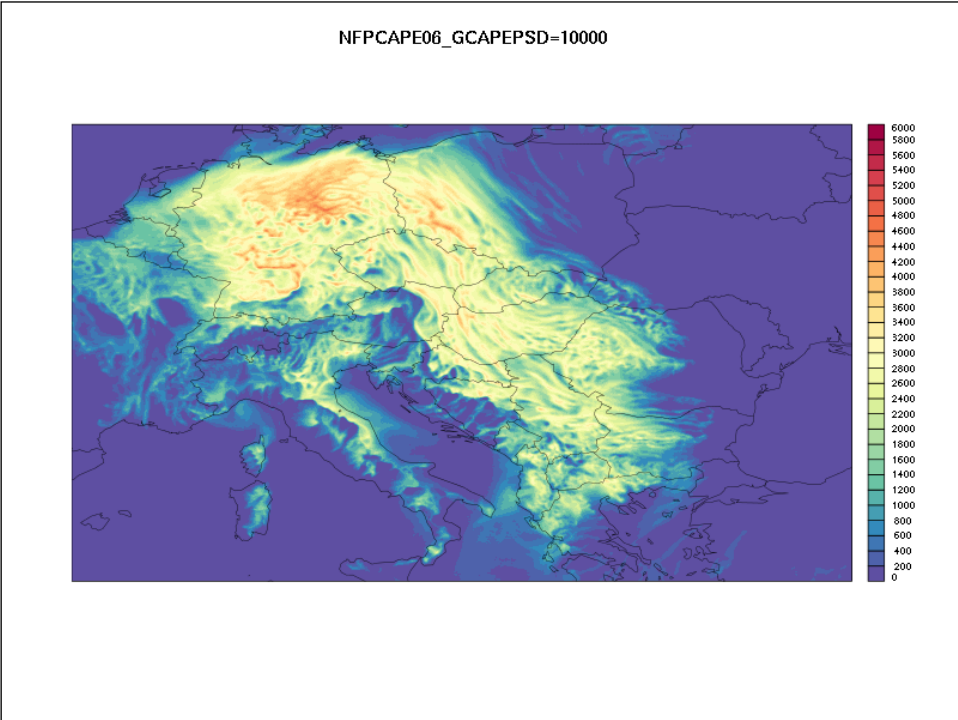


Figure 5: CAPE based on mixed layer (NFCAPE=6, GCAPEPSD=10000.) with depth 100 hPa for 201300620 15 UTC

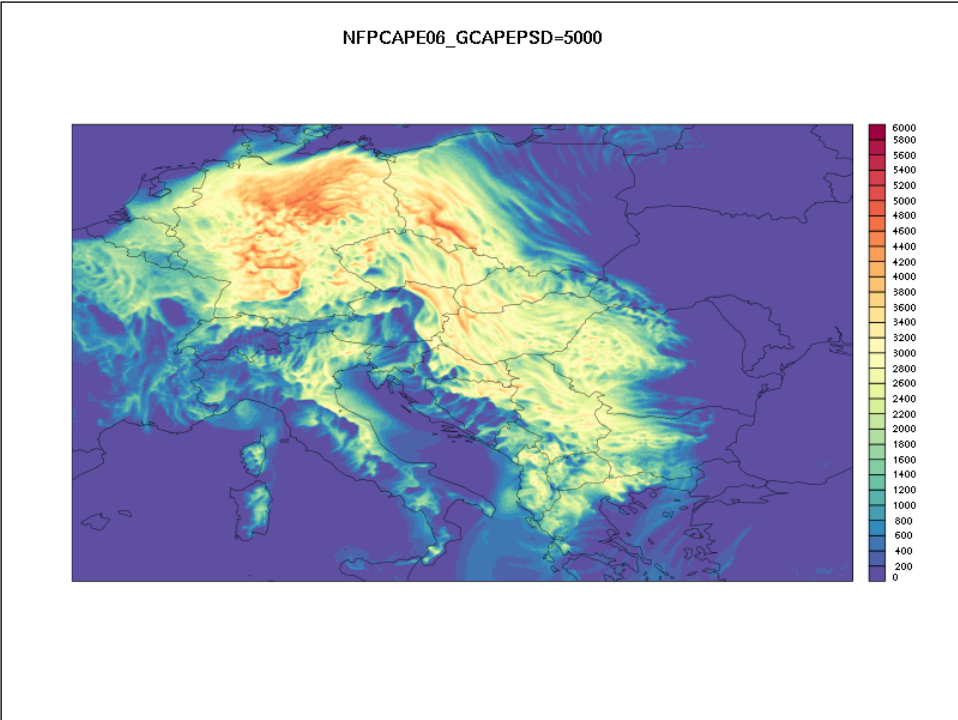


Figure 6: CAPE based on mixed layer (NFCAPE=6, GCAPEPSD=5000.) with depth 50 hPa for 201300620 15 UTC

2. Lightning diagnostics

a. Methodology

In literature several approaches can be found to predict lightning threats with a numerical weather prediction model. In the current AROME/ALARO post-processing no lightning diagnostics is available. During the stay in Ljubljana first steps were undertaken to include some additional diagnostic to the current Fullpos version. The following methods were chosen to be evaluated according their usability for AROME/ALARO:

- Price and Rind (1992): Flash rate is estimated based on cloud top height or on maximum vertical velocity found within a convective updraft
- McCaul (2009): a) flash rate is estimated based on the graupel flux in the mixed phase region; b) flash rate is based on the integrated amount of solid hydrometeors in a convective cloud ; c) blended version of a) and b)

Appendix B shows the current status of the routine `fpdiagflash.F90`. In order to use it inside Fullpos the data flow has to be changed accordingly (hydrometeors, vertical velocity), etc. The following routines have been touched (work of Jure Cedilnik) to allow a call of `fpdiagflash` within fullpos:

```
arp/fullpos/vpos.F90
arp/fullpos/phymfpos.F90
arp/fullpos/sufpc.F90
arp/fullpos/fpdiagflash.F90
arp/module/yomfpc.F90
arp/module/yomafn.F90
arp/module/parfpos.F90
arp/namelist/namafn.h
arp/namelist/namfpc.h
arp/pp_obs/pos.F90
arp/setup/suafn1.F90
arp/setup/suafn2.F90
arp/setup/suafn3.F90
```

The data flow around the call to flash diagnostics has been coded and the first version of the `fpflashdiag` routine as well. The next steps are:

- technical validation of the data flow and input fields to `fpdiagflash` (not yet finished)
- scientific validation of the different options for the computation of flash rates for 1 or 2 case studies
- calibration of the different versions using forecasts and lightning data for a full convective season

References:

McCaul, E. W., Jr., S. J. Goodman, K. M. LaCasse, and D. J. Cecil, 2009: Forecasting lightning threat using cloud-resolving model simulations. *Wea. Forecasting*, 24, 709729.

Price, C., and D. Rind (1992), A simple lightning parameterization for calculating global lightning distributions, *J. Geophys. Res.*, 97(D9), 9919–9933, doi:10.1029/92JD00719.

APPENDIX A

relevant changes in routine fpcica.F90

```
!=====
ELSEIF (KCAPETYPE == 6) THEN
!=====

! Mixed layer cape (ML CAPE); parcel entering cape computation is created through mixing of
lowest GCAPEPSD Pa (usually 50-100hPa)
! the temperature used for averaging is the potential temperature

! so let's mix the pbl (compute average for Theta and q):
ILEVCOUNT(KST:KEND)=0
ZTHSUM(KST:KEND)=0.0_JPRB
ZQVSUM(KST:KEND)=0.0_JPRB
ZTEST(KST:KEND)=1.0_JPRB

ZUSRATM=1.0_JPRB/RATM

DO JLEV=KLEV,1,-1

  DO JLON=KST,KEND
    ! conditional statements (if below GCAPEPSD Pa)
    ILEVCOUNT(JLON)=ILEVCOUNT(JLON)+NINT(ZTEST(JLON))

    ! compute average of theta
    ZTHSUM(JLON)=ZTHSUM(JLON)+ZTEST(JLON)*PT(JLON,JLEV)*(PRP(JLON,JLEV)/PRP(JLON,KLEV))**(-
PKAP(JLON,JLEV))

    ! to compute average of qv
    ZQVSUM(JLON)=ZQVSUM(JLON)+ZTEST(JLON)*PQV(JLON,JLEV)

    ZTEST(JLON)=MAX(0._JPRB,SIGN(1._JPRB,GCAPEPSD-PRP(JLON,KLEV)&
& +PRP(JLON,JLEV)))

    ZP(JLON,JLEV)=PRP(JLON,JLEV)
    ZT(JLON,JLEV)=PT(JLON,JLEV)
    ZQV(JLON,JLEV)=PQV(JLON,JLEV)
  ENDDO

ENDDO

! build average:

DO JLON=KST,KEND

  ZP(JLON,KLEV+1)=PRCLS(JLON)
  ZT(JLON,KLEV+1)=ZTHSUM(JLON)/(ILEVCOUNT(JLON)*1.0_JPRB)
  ZQV(JLON,KLEV+1)=ZQVSUM(JLON)/(ILEVCOUNT(JLON)*1.0_JPRB)

ENDDO

ILEV=KLEV+1
CALL FPCINCAPE(KST,KEND,KPROMA,ILEV,ILEV,ZT,ZP,ZQV,PCAPE,PCIN,KLCL,KFCL,KEL)

! End of the test on KCAPETYPE
!=====
ENDIF
```

!=====

APPENDIX B

new routine to calculate flash rate diagnostics within Fullpos; still under development ;-)

```
SUBROUTINE FPDIAFLASH (KST, KEND, KPROMA, KLEV, &
                     & PQL, PQI, PQR, PQS, PQGH, PT, PDELP, PPFULL, PR, PVW, PLSM, PFLASH)

USE PARKIND1 , ONLY : JPIM, JPRB
USE YOMHOOK  , ONLY : LHOOK, DR_HOOK
USE YOMFPC   , ONLY : NMTFLASH

!***** FPDIAFLASH *****

!      PURPOSE:
!      -----
!      Compute flash rates (or lightning potential)

!      INTERFACE:
!      -----
!      *CALL FPDIAFLASH*

!      EXPLICIT ARGUMENTS:
!      -----
!      INPUT:
!      KST      : start of work
!      KEND     : end of work
!      KPROMA   : dimension of work
!      KLEV     : number of levels
!      PT       : temperature (K)
!      PQL      : cloud water (g/kg)
!      PQI      : cloud ice (g/kg)
!      PQS      : snow (g/kg)
!      PQR      : rain (g/kg)
!      PQGH     : graupel and hail (g/kg)
!      PWW      : vertical velocity
!      PDELP    : layer thickness (Pa)
!      PPFULL   : pressure on full levels (Pa)
!      PR       : gas constant (J/(kg K))
!      PVW      : vertical velocity (m/s)
!      PLSM     : land sea mask

!      OUTPUT:
!      PFLASH   : flash rate for given column

!      IMPLICIT ARGUMENTS:
!      -----
!      NONE

!      METHOD:
!      -----
!      different methods to compute flash rate can be chosen, dependent on
!      NMTFLASH:

!      = 0 Price and Rind (1992+)      : flash rate estimate based on (concevtive) cloud
top or max. vert. speed in conv. in column
!      = 1 McCaul et. al (2009) part 1: flash rate is based on upward flux of
precipitatinf ice hydrometeors (graupel [and hail if available])
!      in the mixed phase region
!      = 2 McCaul et. al (2009) part 2: flash rate is based on vertically integrated
amount of ice hydrometeors
!      = 3 McCaul et. al (2009) part 3: blending version 1 and 2

!      REFERENCE:
!      -----
!      NMTFLASH=0 : Price, C. and Rind, D.: A simple lightning parameterization for
calculating global lightning distributions, J. Geophys. Res., 97,
!      1992, +1993, +1994
```



```
! NMTFLASH=1-3: McCaul Jr, Eugene W., et al. "Forecasting lightning threat using cloud-
! resolving model simulations."
! Weather and Forecasting 24.3 (2009): 709-729.McCaul et al.
```

```
! AUTHOR:
! -----
! Christoph Wittmann *ZAMG*
```

```
! MODIFICATIONS:
! -----
```

```
! original version: June 2013
!
```

```
USE PARKIND1, ONLY: JPIM, JPRB
USE YOMHOOK, ONLY: LHOOK, DR_HOOK
USE YOMCST, ONLY: RTT
```

```
IMPLICIT NONE
```

```
INTEGER (KIND=JPIM), INTENT (IN) :: KST, KEND, KPROMA, KLEV
REAL (KIND=JPRB), INTENT (IN) ::
PQI (KPROMA, KLEV), PQL (KPROMA, KLEV), PQS (KPROMA, KLEV), PQR (KPROMA, KLEV)
REAL (KIND=JPRB), INTENT (IN) :: PQGH (KPROMA, KLEV), PT (KPROMA, KLEV), PDELP (KPROMA, KLEV)
REAL (KIND=JPRB), INTENT (IN) :: PPFULL (KPROMA, KLEV),
PR (KPROMA, KLEV), PVW (KPROMA, KLEV), PLSM (KPROMA)
REAL (KIND=JPRB), INTENT (OUT) :: PFLASH (KPROMA)
```

```
!local declarations
INTEGER (KIND=JPIM) :: JLon, JLEV
REAL (KIND=JPRB) :: ZHOOK_HANDLE
REAL (KIND=JPRB) :: ZINT (KPROMA), ZFLASH1 (KPROMA), ZFLASH2 (KPROMA), ZVW_MAX (KPROMA)
REAL (KIND=JPRB) :: ZTEST, ZTEST1, ZTEST2, ZTEST3, ZRESCAL, ZFINCAL
REAL (KIND=JPRB) :: ZRHOAIR, ZK1, ZK2, ZQILMIN, ZTMIX, ZVW
REAL (KIND=JPRB) :: ZPRFACZL, ZPRPOTZL, ZPRFACWL, ZPRPOTWL
REAL (KIND=JPRB) :: ZPRFACZS, ZPRPOTZS, ZPRFACWS, ZPRPOTWS
```

```
!!!! do we need a cloud free counter??? to avoid that we consider regions that we should not?
```

```
IF (LHOOK) CALL DR_HOOK ('FPDIAGFLASH', 0, ZHOOK_HANDLE)
! -----
```

```
ZINT (:)=0.0_JPRB
ZQILMIN=1.E-6_JPRB
ZVW_MAX (:)=100._JPRB ! max wind speed (m/s)
```

```
!used for Price and Rind:
!-----
```

```
! FAC .. factor or pot ... exponent; Z or W ... based on cloud top z or vert. velocity W ; L
... over land or S over sea
```

```
! over land:
```

```
ZPRFACZL=3.44E-5_JPRB ! just used in the case we base our flashes on ztop
ZPRPOTZL=4.9_JPRB ! just used in the case we base our flashes on ztop
ZPRFACWL=5.0E-6_JPRB
ZPRPOTWL=4.54_JPRB
```

```
!over seas (following modification from Michalon (1999))
```

```
ZPRFACZS=6.57E-4_JPRB ! just used in the case we base our flashes on ztop
ZPRPOTZS=4.9_JPRB ! just used in the case we base our flashes on ztop
ZPRFACWS=1.11E-6_JPRB
ZPRPOTWS=4.54_JPRB ! check values again !!!!!!!
```

```
!calibration factor (f(grid_size))
ZRESCAL=1.0_JPRB
```

```
! used for McCaul:
!-----
```

```
ZTMIX=RTT-15._JPRB !
ZK1=0.042_JPRB
ZK2=0.2_JPRB
```

```

! overall calivration:
ZFINCAL=1.0_JPRB ! could be namelist variable???

! 0. Method based on Price and Rind (1992+)
! -----

IF (NMTFLASH == 0) THEN

write(*,*) "dummy statement 0 ",NMTFLASH
! we have to find the maximum vertical velocity in a (convective) updraught

DO JLEV=KLEV,1,-1
DO JLON=KST,KEND

! we should verify that we are inside a cloud !!!!
ZTEST1=MAX(0.0_JPRB,SIGN(1.0_JPRB,PQI(JLON,JLEV)+PQL(JLON,JLEV)-ZQILMIN))
ZTEST1=1._JPRB !!!!
! remember max vertical speed
!! ZVW_MAX(JLON)=MAX(PVW(JLON,JLEV)*ZTEST1,ZVW_MAX(JLON))
!! ZVW_MAX(JLON)=MAX(PQI(JLON,JLEV)+PQL(JLON,JLEV),ZVW_MAX(JLON))
ZVW_MAX(JLON)=MAX(PQL(JLON,JLEV),-10.0_JPRB)

ENDDO
ENDDO

DO JLON=KST,KEND

!! ZFLASH1(JLON)=PLSM(JLON)*(ZPRFACWL*ZVW_MAX(JLON)**ZPRPOTWL)+&
!! &(1._JPRB-PLSM(JLON))*(ZPRFACWS*ZVW_MAX(JLON)**ZPRPOTWS)

ZFLASH1(JLON)=ZVW_MAX(JLON)
! calibration due to horizontal grid-size?

ENDDO

ENDIF

IF (NMTFLASH == 1 .OR. NMTFLASH == 3) THEN

write(*,*) "dummy statement 1",NMTFLASH

ZINT(:)=0.0_JPRB

! first: find level closest to -15 degree isotherm inside a cloud
DO JLEV=KST+2,KLEV,1
DO JLON=KST,KEND

! test 1: inside a cloud? , test 2: temperture below -15 , test 3: temperatur above -15
! test= 1+2+3 should garantie that we've just passed the -15 degree isotherm inside a
cloud
ZTEST1=MAX(0.0_JPRB,SIGN(1.0_JPRB,PQI(JLON,JLEV)+PQL(JLON,JLEV)-ZQILMIN))
ZTEST2=MAX(0.0_JPRB,SIGN(1.0_JPRB,PT(JLON,JLEV-1)-ZTMIX))
ZTEST3=MAX(0.0_JPRB,SIGN(1.0_JPRB,ZTMIX-PT(JLON,JLEV)))
ZTEST=ZTEST1*ZTEST2*ZTEST3

! if all test are successful we save the level (just below -15) inside a cloud we need
! vertical velocity and graupe+hail content
! there for we take avaeage from level below and above isotherm
! and multiply with vertical velocity (average from level below and above isotherm)
ZVW=0.5_JPRB*(PVW(JLON,JLEV-1)+PVW(JLON,JLEV))
!the following line should be active max 1 time during vertical loop
ZINT(JLON)=ZINT(JLON)+ZTEST*0.5_JPRB*(PQGH(JLON,JLEV-1)+PQGH(JLON,JLEV))*ZVW

ENDDO
ENDDO

DO JLON=KST,KEND
ZFLASH1(JLON)=ZK1*ZINT(JLON)
ENDDO

ENDIF

IF (NMTFLASH == 2 .OR. NMTFLASH == 3) THEN

```

```

write(*,*) "dummy statement 2  ",NMTFLASH
ZINT(:)=0.0_JPRB

! we need to integrate solid hydrometeors inside a cloud
DO JLEV=KLEV,1,-1
DO JLON=KST, KEND
! if sum of cloud condensate exceeds a certain threshold we assume that we're inside a
cloud
ZTEST1=MAX(0.0_JPRB,SIGN(1.0_JPRB,PQI(JLON,JLEV)+PQL(JLON,JLEV)-ZQILMIN))
ZRHOAIR=PPFULL(JLON,JLEV)/(PR(JLON,JLEV)*PT(JLON,JLEV))
ZINT(JLON)=ZINT(JLON)+ZTEST1*ZRHOAIR*PDELP(JLON,JLEV)*&
&(PQI(JLON,JLEV)+PQS(JLON,JLEV)+PQGH(JLON,JLEV))
ENDDO
ENDDO

DO JLON=KST,KEND
ZFLASH2(JLON)=ZK2*ZINT(JLON)
ENDDO

ENDIF ! end of NTMFLASH

IF (NMTFLASH == 3 ) THEN

write(*,*) "dummy statement 3  ",NMTFLASH

! McCaul version 3 is simply a weighted combination of version 1 and 2 (-> McCaul 2009)
DO JLON=KST,KEND
ZFLASH1(JLON)=0.95_JPRB*ZFLASH1(JLON)+0.05_JPRB*ZFLASH2(JLON)
ENDDO

ENDIF

! finally (calibrate?) and store the data in global scope

DO JLON=KST,KEND
PFLASH(JLON)=ZFLASH1(JLON)*ZFINCAL
ENDDO

IF (LHOOK) CALL DR_HOOK('FPDIAGFLASH',1,ZHOOK_HANDLE)

END SUBROUTINE FPDIAGFLASH

```