Extending the functionality of "convection diagnostics" in ALADIN/ALARO/AROME

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

Part 3 (CW):
Report of LACE stay at ARSO, Ljubljana

24. 3. - 4. 4. 2014

1. Phasing into CY38T1

The results as described in Part1 and Part 2 of the joined report on "Extending the functionality of convection diagnostics in ALADIN/ALARO/AROME" were based on routines coded within the framework of CY36T1. In order to finalize the package and prepare it in a way to be included into the next common export version it was decided to phase the routines into the current export version of CY38T1. In the following, a list of routines and a short description of related namelist parameters is given.

a. ML CAPE

The following routines were modified to include another version for CAPE diagnostics in the model:

arpifs/fullpos/fpcica.F90
arpifs/module/yomcape.F90
arpifs/setup/sucape.F90
arpifs/fullpos/phymfpos.F90
arpifs/fullpos/endpos.F90
arpifs/fullpos/sufpc.F90

Activating the mixed layer CAPE computation is done by setting parameter NFPCAPE=6 in NAMFPC while asking for CFP2DF(1)='SURFCAPE.POS.F00',. In addition GCAPEPSD, defining the depth of the layer above model orography to be mixed, has to be set accordingly. Recommended values: GCAPEPSD=5000 (Pa) up to 10000, larger values would not make too much sense.

b. Storm motion vector, storm relative helicity and vertical shear

The following routines were updated to include these parameters in Fullpos:

arpifs/setup/suafn1.F90
arpifs/setup/suafn2.F90
arpifs/setup/suafn3.F90
arpifs/namelist/namafn.h
arpifs/namelist/namfpc.h
arpifs/module/yomafn.F90
arpifs/module/yomfpc.F90
arp/fullpos/sufpc.F90
arp/module/parfpos.F90
arpifs/fullpos/phymfpos.F90

To retrieve these fields, the following parameters have to be requested in NAMFPC:

```
CFP2DF(1)='SURFUSTROMMOTION'
CFP2DF(2)='SURFVSTORMMOTION',
CFP2DF(3)='SURFDEEPLAYERSHR',
CFP2DF(4)='SURFLOWLEVELSHR ',
CFP2DF(5)='SURFHELICITY    '
```

c. Validation of code modification in CY38T1

In order to ensure the functionality of the described parameters, the modified code was tested and compared to CY36T1. Figures 1+2 show option NFPCAPE=6 for CY36T1 and CY38T1, results are (bit) identical.

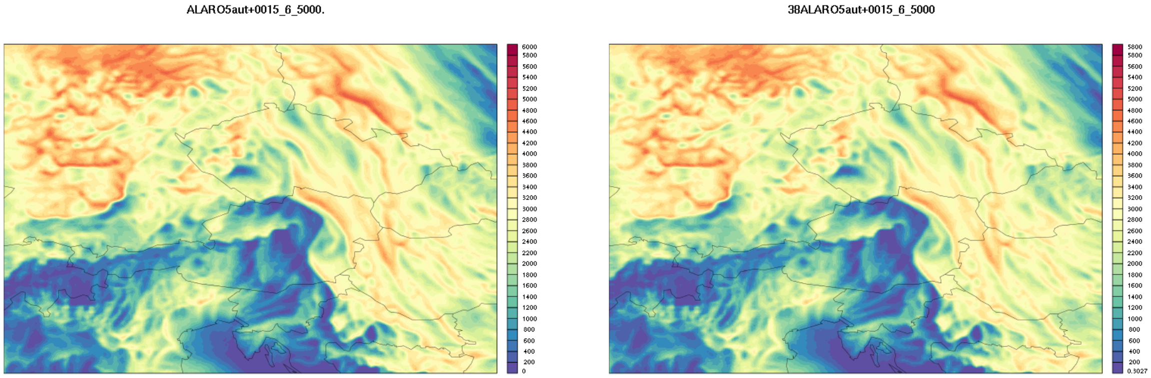


Figure 1+2: NFPCAPE=6 and GCAPEPSD=5000 computed with CY36T1 (left) and CY38T1 (right).

2. Lightning diagnostics as a part of the convection package for Fullpos

a. Phasing to CY38T1 and small code modifications

As reported in the previous parts of this report, four different types of lightning diagnostics were tested and evaluated according their usability for AROME/ALARO. As a first step, the parameter “SURFFLASHDIAG” was introduced as new variable in Fullpos part (together with the parameters described in section 1) and phased into CY38T1. The routines which had to be modified/added:

arpifs/setup/suafn1.F90
arpifs/setup/suafn2.F90
arpifs/setup/suafn3.F90
arp/namelist/namafn.h
arp/namelist/namfpc.h

arpifs/module/yomafn.F90
arpifs/module/yomfpc.F90
arpifs/fullpos/phymfpos.F90
arp/fullpos/vpos.F90
arp/fullpos/sufpc.F90
arp/fullpos/fpdiagflash.F90
arp/fullpos/phymfpos.F90
arp/module/parfpos.F90

The following options can be chosen in NAMFPC to retrieve “lightening densities” via CFP2DF(1)='SURFFLASHDIAG':

- NMTFLASH=1: Lightning density derived through from vertical wind speed found within a (convective) cloud. [USE NOT YET RECOMENDED]
- NMTFLASH=2 : Lightning density derived from graupel flux in the mixed phase region of a (convective) cloud.
- NMTFLASH=3: Lightning density derived from the intregral of solid hydrometeors within a (convective) cloud.
- NMTFLASH=4: weighted combination of NMTFLASH=2 (recommended weight 0.95) and NMTFLASH=3 (rec. weight 0.05).

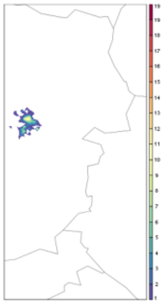
In addition to NMTFLASH, the following namelist parameters have been introduced:

- RFLASHQIL: Defining the critical threshold for cloud condensate (qi + ql) to be used for the criterion “inside a cloud” for NMTFLASH=1 and 3. Recommend value: 1.E-7
- RFLASHTMXPL: Defining the temperature used to identify the mixed phase region within a cloud. Recommended values. 258.16 (Kelvin).
- RFLASHTCRIT: [not yet used, reserved for other methods]
- RFLASHCAL see next section

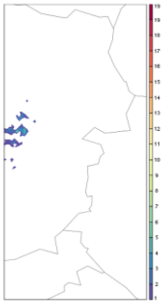
b. Calibration of Lightning density, NAMELIST parameter RFLASHCAL

While a first step or guess for a calibration of the lightning diagnostics were already reported in part 2 of the joined report, a more sophisticated method to calibrate the forecasted lightning density was now implemented during this stay. In order to allow a calibration of the simulated flash density, the observed lightning density has to be known on the same grid as the simulated one. This was realised by developing a software to fetch lightning data (categorized: cloud-cloud, cloud-ground and ground-cloud lightening and interpolate it to the model grid (in the present case AROME 2.5km grid used at ZAMG). The software was developed with the help of the remote sensing team at ZAMG.
Figures 3-12 show the observed lightning densities [flash / (5min * 2.5km**2)] for the period 20130809 13:00 – 22:00 UTC. As the implementation of a lightning diagnostic in Fullpos just allows us to compute instantaneous values out of the model (from historical files), the calibration intervals are rather short (5min or alternatively 1 model time step).

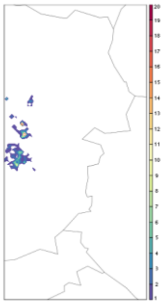
obs lightning densities 20130809 13:00 UTC [flashes/(5 min * 2.5km**2)]



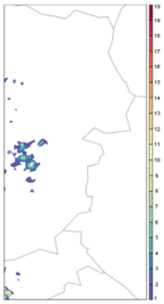
obs lightning densities 20130809 14:00 UTC [flashes/(5 min * 2.5km**2)]



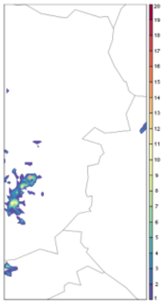
obs lightning densities 20130809 15:00 UTC [flashes/(5 min * 2.5km**2)]



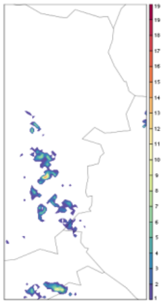
obs lightning densities 20130809 16:00 UTC [flashes/(5 min * 2.5km**2)]



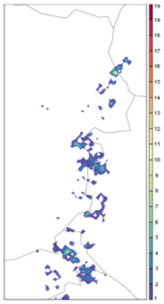
obs lightning densities 20130809 17:00 UTC [flashes/(5 min * 2.5km**2)]



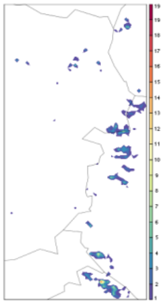
obs lightning densities 20130809 18:00 UTC [flashes/(5 min * 2.5km**2)]

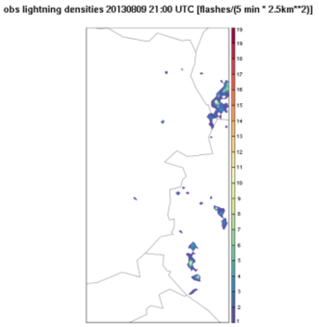


obs lightning densities 20130809 19:00 UTC [flashes/(5 min * 2.5km**2)]

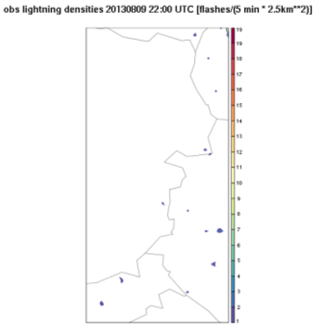


obs lightning densities 20130809 20:00 UTC [flashes/(5 min * 2.5km**2)]





obs lightning densities 20130809 21:00 UTC [flashes/(5 min * 2.5km**2)]

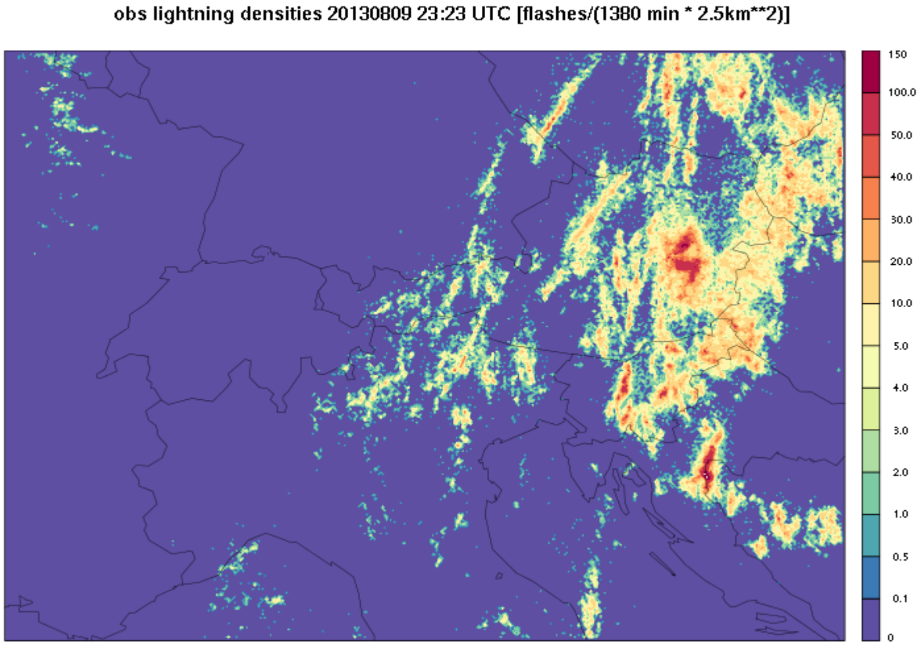


obs lightning densities 20130809 22:00 UTC [flashes/(5 min * 2.5km**2)]

For the first objective verification, lightning data for one day was used (20130809), the day with the highest number of detected lightning charges in convective season 2013 in Austria (see also figure 13). The following factors were computed:

- RFLASHCAL=6.0E4 for NMTFLASH=1
- RFLASHCAL=6.0E-1 for NMTFLASH=2

The calibration was done using by comparing the max and mean values of observed and simulated densities on a relatively small domain (as shown in Figures 3-12). Why choosing this domain? To have a meaningful first step towards a calibration it is necessary to use events with a sufficient correspondence between observed and forecasted precipitation – this is the case for this event.

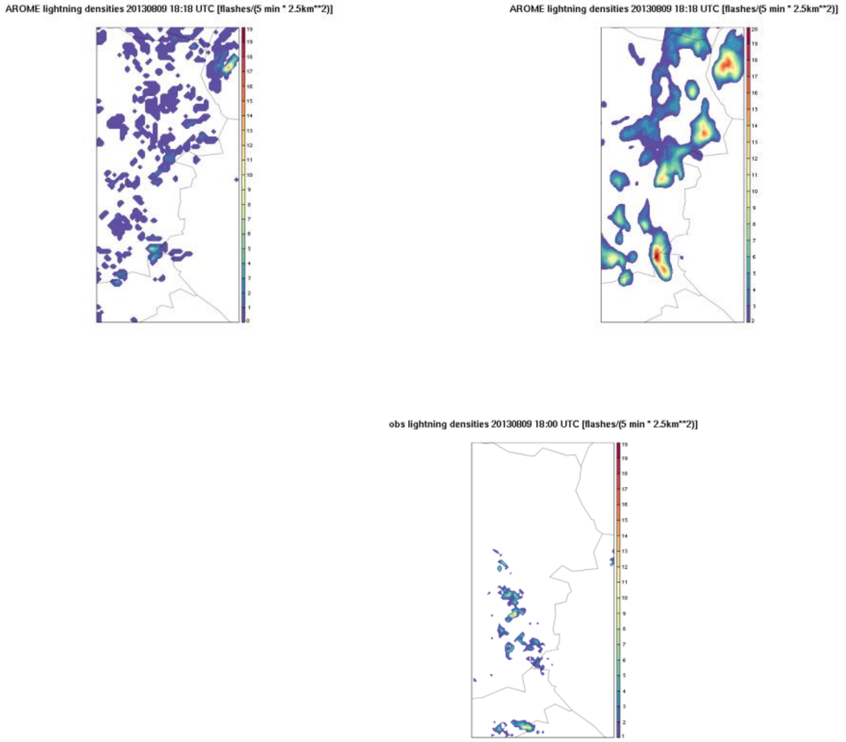


obs lightning densities 20130809 23:23 UTC [flashes/(1380 min * 2.5km**2)]

Figure 13: Observed lightning density for 201309 00 – 23 UTC (flashes / (23h ** 2.5km**2)

Figures 14-16 show the calibrated lightning densities for NMTFLASH=1 (left) and NMTFLASH=2 (middle) compared to the observed density (right). While the locations of the thunderstorms on this rather small (200x80km) is not sufficiently captured by the model, the maximum values of the simulated density are in rather good correspondence with the observed values after caöibration. In general the simulated thunderstorms are too much spread over the domain.

Figures 3-16 all show just few (5min) moments from a day with a large total number of lightnings. Using diagnosed lightnings from Fullpos just allows us to see simulated values in intervals given by the write-out steps of the model (usually hourly). It is of course more useful to have accumulated information about the simulated density for e.g. the whole day. Thus it was decided to code the parameters as a cumulated flux inside the model (with the option for post-processing later on) rather than continue with the calibration.


AROME lightning densities 20130809 18:18 UTC [flashes/(5 min * 2.5km**2)]
AROME lightning densities 20130809 18:18 UTC [flashes/(5 min * 2.5km**2)]
obs lightning densities 20130809 18:00 UTC [flashes/(5 min * 2.5km**2)]

Figures 14-17 AROME simulated lightning density for 20130809 18:00 UTC (flash/(5min* 2.5km**2) for NMTFLASH=1 (upper left) and NMTFLASH=2 (upper right). Observations shown on same grid (bottom)

3. Implementation of lightning density as a cumulated flux (CFU)

a. Technical implementation

Up to now, the authors of this report were not familiar with coding new cumulated fluxes inside the model. Thus, a distinct time had to be used to study relevant documentations and parts of the codes. The documentations found on GMAPdoc provided valuable information for that. The following routines were finally modified to include a new CFU field “SURFFLASHDENSITY” in the model:

arpifs/module/yomafn.F90
arpifs/module/yomcfu.F90
arpifs/module/parfpos.F90
arpifs/fullpos/hpos.F90
arpifs/fullpos/sufpcfu.F90
arpifs/setup/suafn1.F90
arpifs/setup/suafn2.F90
arpifs/setup/suafn3.F90
arpifs/setup/ptrgfu.F90
arpifs/setup/sucfu.F90
arpifs/adiab/cpg_dia.F90
arpifs/adiab/cpg.F90
arpifs/namelist/namafn.h
arpifs/namelist/namcfu.h
arpifs/dia/cpcfu.F90
arpifs/phys_dmn/mf_phys.F90
arpifs/phys_dmn/apl_arome.F90
arpifs/phys_dmn/diagflash.F90

At the time of writing this report the modified is technically working, but is not producing the expected output values. Thus, further technical validations have to follow. In order to activate the new CFU field switch LFLASH=.TRUE. has to be set in NAMCFU.

4. Outlook

The decision to move the parameter “lightning density” from pure Fullpos to the model itself (call to routine diagflash.F90 in apl_arome for AROME) will open new possibilities to apply extensive calibration on the simulated values, as cumulated values can be used, not only instantanous values. The software to perform the calibration is now ready (at ZAMG) to be used on any grid. The calibration together with the finalization of technical validation of the new code will now be the next target.At the moment the implementation is done for AROME, but afterwards there is no reason to implement it also for ALARO (through a call from aplpar to diagflash.F90). As we need vertical wind speed and graupel content, having prognostic graupel in ALARO would be necessary. For vertical wind speed, the use of updraught velocity (together with upraught mash fraction) from 3MT might be a good choice to identify convective cells in ALARO serving as an input for the lightning diagnostic.

APPENDIX A
List of routines (including modification) to add additonal CFU “SURFFLASHDENSITY”

arpifs/module/yomafn.F90
```
TYPE(FPDSPHY) :: GFP_CFLSH !Flash density
```

arpifs/setup/suafn1.F90
```
GFP_CFLSH=GFP_SU(-9999,IG,12,0,0,0,LLF,LLT,'SURFFLASHDENSITY',CL)
```

arpifs/setup/suafn2.F90:
```
CALL CTRL_GFP(I,IMAX,GFP_CFLSH,GFP_PHYDS)
```

arpifs/setup/suafn3.F90
```
CALL PRINT_GFP('Flash densities..................','GFP_CFLSH',GFP_CFLSH)
```

arpifs/namelist/namafn.h
```
,GFP_CFLSH
```

arpifs/module/yomcfu.F90
```
! LFLASH = .T.   ===> FLASH DENSITY
LOGICAL :: LFLASH
```

arpifs/setup/ptrgfu.F90
```
INTEGER(KIND=JPIM) :: MFLSH  ! Flash density
```

arpifs/setup/sucfu.F90
```
USE YOMAFN LFLSH
USE PTRGFU MFLSH
NAMELIST/NAMAFN/ &
..GFP_CFLSH,..
LFLASH = .FALSE.
LFLASH =LFLASH .AND.LCUMFU

MFLSH=0
IF (LFLASH) THEN
  CALL NEXTYPE1(GFP_CFLSH%CLNAME(1:4),GFP_CFLSH%CLNAME(5:),0,MFLSH)
ENDIF

  WRITE(UNIT=KULOUT,FMT='('' LSTRD  = '',L2,'' LSTRC  = '',L2 &
   & ,'' LSTRT  = '',L2,'' LFPLC  = '',L2,'' LFPLCG  = '',L2 &
   & ,'' LFPLCH = '',L2,'' LFPLS  = '',L2,'' LFPLSG  = '',L2 &
   & ,'' LFPLSH = '',L2,'' LFR    = '',L2,'' LFSOIL = '',L2 &
   & ,'' LNEBTT = '',L2,'' LFSF   = '',L2,'' LNEBPAR= '',L2 &
   & ,'' LAMIP  = '',L2,'' LRAYS  = '',L2,'' LFRRC  = '',L2 &
   & ,'' LFDUTP = '',L2,'' LMOON  = '',L2,'' LRAYD  = '',L2 &
   & ,'' LFLASH = '',L2)')&
   & LSTRD,LSTRC,LSTRT,LFPLC,LFPLCG,LFPLCH,LFPLS,LFPLSG,LFPLSH,&
   & LFR,LFSOIL,LNEBTT,LFSF,LNEBPAR,LAMIP,LRAYS,LFRRC,LFDUTP,LMOON,LRAYD,&
   & LFLASH

  WRITE(UNIT=KULOUT,FMT='('' MSTRDU = '',I2,'' MSTRDV = '',I2 &
   & ,'' MSTRCU = '',I2,'' MSTRCV = '',I2,'' MFDICQ = '',I2,'' MFDICS = '',I2 &
   & ,'' MSTRTU = '',I2,'' MSTRTV = '',I2,'' MFDITQ = '',I2,'' MFDITS = '',I2 &
   & ,'' MFPLCL = '',I2,'' MFPLSL = '',I2,'' MFPLCN = '',I2,'' MFPLSN = '',I2 &
   & ,'' MFCCQL = '',I2,'' MFCCQN = '',I2,'' MFCSQL = '',I2,'' MFCSQN = '',I2 &
   & ,'' MFRSO  = '',I2,'' MFRTH  = '',I2,'' MFNEB  = '',I2,'' MWS    = '',I2 &
   & ,'' MSNS   = '',I2,'' MQTOT  = '',I2,'' MFNEBT = '',I2,'' MOZONT = '',I2 &
   & ,'' MSPRES = '',I2,'' MFCL   = '',I2,'' MFCS   = '',I2,'' MFDISH = '',I2 &
   & ,'' MFTOPH = '',I2,'' MQICE  = '',I2,'' MQLI   = '',I2,'' MFRSOC0= '',I2 &
   & ,'' MFRTHC0= '',I2,'' MFRSOPS= '',I2,'' MFRSOPT= '',I2,'' MFRSOLU= '',I2 &
   & ,'' MFRSODS= '',I2,'' MFRTHDS= '',I2,'' MFCLL  = '',I2,'' MFCLN  = '',I2 &
   & ,'' MFEVL  = '',I2,'' MFEVN  = '',I2,'' MFONTE = '',I2,'' MFCHSP = '',I2 &
   & ,'' MFEVV  = '',I2,'' MFLWSP = '',I2,'' MFTR   = '',I2,'' MRUISL = '',I2 &
   & ,'' MRUISP = '',I2,'' MRUISS = '',I2,'' MNEBCON= '',I2,'' MNEBHAU= '',I2 &
   & ,'' MNEBMOY= '',I2,'' MFGEL  = '',I2,'' MFGELS = '',I2,'' MNEBBAS= '',I2 &
   & ,'' MFRSOC1= '',I2,'' MFRTHC1= '',I2,'' MFDUTP = '',I2,'' MFLSH= '',I2 &
   & )') &
   & MSTRDU ,MSTRDV ,MSTRCU ,MSTRCV ,MFDICQ ,MFDICS ,MSTRTU ,MSTRTV ,&
   & MFDITQ ,MFDITS ,MFPLCL ,MFPLSL ,MFPLCN ,MFPLSN ,MFCCQL ,MFCCQN ,&
   & MFCSQL ,MFCSQN ,MFRSO  ,MFRTH  ,MFNEB  ,MWS    ,MSNS   ,MQTOT  ,&
   & MFNEBT ,MOZONT ,MSPRES ,MFCL   ,MFCS   ,MFDISH ,MFTOPH ,MQICE  ,&
   & MQLI   ,MFRSOC0,MFRTHC0,MFRSOPS,MFRSOPT,MFRSOLU,MFRSODS,MFRTHDS,&
   & MFCLL  ,MFCLN  ,MFEVL  ,MFEVN  ,MFONTE ,MFCHSP ,MFEVV  ,MFLWSP ,&
   & MFTR   ,MRUISL ,MRUISP ,MRUISS ,MNEBCON,MNEBHAU,MNEBMOY,MFGEL  ,&
   & MFGELS ,MNEBBAS,MFRSOC1,MFRTHC1,MFDUTP,MFLSH
```

arpifs/dia/cpcfu.F90

```
USE MFLSH
REAL(KIND=JPRB)   ,INTENT(IN)    :: PFLASH(NPROMA)

ELSEIF (IADD == MFLSH) THEN
      ZGFU(KST:KND,IFLD)=ZGFU(KST:KND,IFLD)+PFLASH(KST:KND)*TSTEP
```

arpifs/adiab/cpg_dia.F90
input, call to CPCFU

arpifs/adiab/cpg.F90

```
REAL(KIND=JPRB) :: ZFLASH(NPROMM)                    ! cf. PFLASH in MF_PHYS

CALL CPG_DIA(CDCONF,IST,IEND,ISTC,IENDC,ISTGLO,IDDHI, &
         & LLDIAB,LLADNMI,LLFSTEP,LLRESET,NMEANSTEPS,NXGSTTS,&
         & YRGSGEOM(IBL),YRCSGEOM(IBL),ZOROGL(1,ILL),ZOROGM(1,ILL), &
         & PGMV(1,1,1,IBL),GFL(:,:,:,IBL), &
         & ZPRE0(1,0,ILL),ZPRE0F(1,1,ILL),ZXYB0(1,1,1,ILL), &
         & ZPHI0(1,0,ILL),ZPHIF0(1,1,ILL),ZPHI0FL(1,1,ILL),ZPHI0FM(1,1,ILL), &
         & ZRCP0(1,1,1,ILL),ZCTY0(1,0,1,ILL),PB2(1,MSLB2VVEL,IBL),ZKENE0(1,1,ILL), &
         & ZRT0L(1,1,ILL),ZRT0M(1,1,ILL),ZPRE0L(1,ILL),ZPRE0M(1,ILL), &
         & ZNHPRE0F(1,1,ILL),ZNHPRE0H(1,0,ILL), &
         & ZQCHA0L(1,1,ILL),ZQCHA0M(1,1,ILL), &
         & ZVCRS0(1,0,1,ILL),ZVCW0F(1,1,ILL), &
         & ZRDLR0(1,0,1,ILL), &
         & SD_VF(:,:,IBL),SP_SG(:,:,IBL),SP_SB(:,:,:,IBL),SP_RR(:,:,IBL),&
         & ZDHSF, &
         & ZALB,ZCAPE,ZCLCC,ZCLCH,ZCLCL,ZCLCM,ZCLCT,ZCLPH,ZCT, &
         & ZDIFCQ,ZDIFCQN,ZDIFCQL,ZDIFCS,ZDIFTQ,ZDIFTQN,ZDIFTQL,ZDIFTS, &
         & ZFCCQL,ZFCCQN,ZFCHSP,ZFCLL,ZFCLN,ZFCQNNG,ZFCQLNG,ZFCQNG,ZFCS, &
         & ZFCSQL,ZFCSQN,ZFEVL,ZFEVN,ZFEVV,ZFGEL,ZFGELS,ZFLWSP,ZFONTE, &
         & ZFPLCL,ZFPLCN,ZFPLCG,ZFPLCH,ZFPLSL,ZFPLSN,ZFPLSG,ZFPLSH,&
         & ZFRMH,ZFRSO,ZFRSOC,ZFRSODS,ZFRSOLU, &
         & ZFRSOPS,ZFRSOPT,ZFRTH,ZFRTHC,ZFRTHDS,ZFTR,ZGZ0,ZGZ0H,ZNEB, &
         & ZQICE(1,1,ILL),ZQLI(1,1,ILL),ZQS(1,ILL), &
         & ZRH,ZRHCLS,ZRUISL,ZRUISP,ZRUISS, &
         & ZSTRCU,ZSTRCV,ZSTRDU,ZSTRDV,ZSTRTU,ZSTRTV, &
         & ZUCLS,ZVCLS,ZTCLS,ZQCLS,ZUGST,ZVGST, &
         & ZMOCON,ZFDIS, &
         & ZFHPCL,ZFHPCN,ZFHPSL,ZFHPSN,ZFHSCL,ZFHSCN,ZFHSSL,ZFHSSN, &
         & ZFEPFP, ZFCMPCQ, ZFCMPSN, ZFCMPSL, &
         & ZFPFPSL,ZFPFPSN,ZFPFPCL,ZFPFPCN,ZFPEVPSL,ZFPEVPSN, &
         & ZFPEVPCL,ZFPEVPCN,ZGPAR(1,1,ILL), &
         & ZFCQRNG, ZFCQSNG,ZDIAGH,ZFLASH, &
         & GMVTNDSI_DDH(1,1,1,IBL),GMVTNDHD_DDH(1,1,1,IBL),&
         & GFLTNDHD_DDH(1,1,1,IBL),ZATND(1,1,1,ILL),&
         & ZDHCV, ZDRNSHF)

!$OMP& PRIVATE (ZFEPFP, ZFCMPCQ, ZFCMPSN, ZFCMPSL, ZFCQRNG, ZFCQSNG, ZDIAGH, ZFLASH) &


        CALL MF_PHYS &
         & (CDCONF,IBL,IGPCOMP,IST,IEND,IGL1,IGL2,IGL3,IGL4,IOFF,ISTGLO, &
         & LDRETCFOU,LDWRTCFOU0,LLADNMI,ZDTPHY, &
         & YRCSGEOM(IBL),YRGSGEOM(IBL),YROROG(IBL),ZOROGL(1,ILL),ZOROGM(1,ILL), &
         & RCUCONVCA(IOFF),RNLCONVCA(IOFF), &
         & PGMV(1,1,1,IBL),PGMVS(1,1,IBL),GFL(:,:,:,IBL),&
         & ZW0_PHY,ZW0L_PHY,ZW0M_PHY,ZRCP0(1,1,1,ILL),ZPHI0(1,0,ILL),ZPHIF0(1,1,ILL), &
         & ZPREHYD0_PHY,ZPREHYD0F_PHY,ZPRE0_PHY,ZPRE0F_PHY,ZXYB0_PHY, &
         & ZVCRS0(1,0,YYTVC0%M_VCASRSH,ILL), &
         & ZW9_PHY,ZRCP9(1,1,1,ILL),ZPHI9(1,0,ILL),ZPHIF9(1,1,ILL), &
         & ZPREHYD9_PHY,ZPREHYD9F_PHY,ZPRE9_PHY,ZPRE9F_PHY,ZXYB9_PHY, &
         & ZKOZO(1,1,ILL), &
         & ZGPSDT2D(1,1,IBL), &
         & ZCTY0(1,0,1,ILL), &
         & ZSLBUF1AU(1,1,ILL),PB2(1,1,IBL),&
         & PGMVT1(1,1,1,IBL),PGMVT1S(1,1,IBL),PGFLT1(1,1,1,IBL), &
         & SP_SB(:,:,:,IBL),SP_SG(:,:,IBL),SP_RR(:,:,IBL),&
         & SD_VF(:,:,IBL),SD_VP(:,:,IBL),SD_VV(:,:,IBL),SD_VH(:,:,IBL), &
         &
SD_VA(:,:,IBL),SD_VC(:,:,IBL),SD_VD(:,:,IBL),SD_SFL(:,:,IBL),SD_SFO(:,:,IBL),&
         & EMTD(1,1,IBL), EMTU(1,1,IBL), TRSW(1,1,IBL),RMOON(1,IBL),&
         & ZGPAR(1,1,ILL) ,ZDHSF, &
         & ZALB,ZCAPE,ZCLCC,ZCLCH,ZCLCL,ZCLCM,ZCLCT,ZCLPH,ZCT, &
         & ZDIFCQ,ZDIFCQN,ZDIFCQL,ZDIFCS,ZDIFTQ,ZDIFTQN,ZDIFTQL,ZDIFTS, &
         & ZFCCQL,ZFCCQN,ZFCHOZ,ZFCHSP,ZFCLL,ZFCLN,ZFCQNNG,ZFCQLNG,ZFCQNG, &
         & ZFCS,ZFCSQL,ZFCSQN,ZFEVL,ZFEVN,ZFEVV,ZFGEL,ZFGELS,ZFLWSP,ZFONTE, &
         & ZFPLCL,ZFPLCN,ZFPLCG,ZFPLCH,ZFPLSL,ZFPLSN,ZFPLSG,ZFPLSH,&
         & ZFRMH,ZFRSO,ZFRSOC,ZFRSODS,ZFRSOLU, &
         & ZFRSOPS,ZFRSOPT,ZFRTH,ZFRTHC,ZFRTHDS,ZFTR,ZGZ0,ZGZ0H,ZNEB, &
         & ZQCLS,ZQICE(1,1,ILL),ZQLI(1,1,ILL),ZQS(1,ILL), &
         & ZRH,ZRHCLS,ZRUISL,ZRUISP,ZRUISS, &
         & ZSTRCU,ZSTRCV,ZSTRDU,ZSTRDV,ZSTRMU,ZSTRMV,ZSTRTU,ZSTRTV, &
         & ZDIFCQLC,ZDIFCQIC,ZFIMCC, &
         & ZFEDQLC,ZFEDQIC,ZFEDQRC,ZFEDQSC,ZFCNEGQLC,ZFCNEGQIC,ZFCNEGQRC,ZFCNEGQSC, &
         & ZUCLS,ZVCLS,ZTCLS,ZUGST,ZVGST, &
         & ZMOCON,ZFDIS, &
         & ZFHPCL,ZFHPCN,ZFHPSL,ZFHPSN,ZFHSCL,ZFHSCN,ZFHSSL,ZFHSSN, &
         & ZFEPFP, ZFCMPCQ, ZFCMPSN, ZFCMPSL,&
         & ZTENDU,ZTENDV, &
         & ZQSOL(1,ILL), &
         & ZFPFPSL,ZFPFPSN,ZFPFPCL,ZFPFPCN,ZFPEVPSL,ZFPEVPSN, &
         & ZFPEVPCL,ZFPEVPCN,ZDRNSHF, &
         & ZFCQRNG, ZFCQSNG, ZDIAGH, ZFLASH)


arpifs/phys_dmn/mf_phys.F90
input …
declaration
2x call apl_arome


arpifs/phys_dmn/apl_arome.F90
input,
use yomcfu LFLASH
declaration

call to diagflash


arpifs/phys_dmn/diagflash.F90

arpifs/module/parfpos.F90
increase CFU fields + 1

arpifs/fullpos/sufpcfu.F90
USE YOMAFN ,GFP_CFLSH
USE LFLASH
ELSEIF (CFPCFU(JFLD) == GFP_CFLSH%CLNAME) THEN
      LFLASH=.TRUE.

Arpifs/fullpis/hpos.F90
USE MFLSH
USE GFP_CFLSH

ELSEIF (IC == GFP_CFLSH%ICOD) THEN
      PFPBUF1(IST:IEND,IFLD)=ZGFU(IST:IEND,MFLSH)
```