ICE3 in ALARO & AROME

A report on modularity of ICE3 to ALARO

Meral SEZER
(TSMS)

Supervised by
Jean-François GELEYN
(Meteo-France & CHMI)

13-24 September 2010

1. Introduction

The central routine for microphysical computations is APLMPHYS inside ALARO. It represents the algoritmic bridge between the physics-dynamics interface's handling of APLPAR and CPTEND_NEW on the one side and microphysical computations of ACACON, ACCOLL and ACEVMEL on the other side. ACACON should handle the in cloud processes related to on-precipitating species (autoconversion type including WBF). ACCOLL should group processes concerning the mechanical growth of precipitating species (collection type). ACEVMEL should group processes describing mass changes of precipitating species, excluding mechanical growth (evaporation type).
The computations of the AROME microphysics is based on MESO-NH ICE3 code.
It is planned to extract microphysical computations from MESO-NH ICE3 (the microphysical code is RAIN_ICE) so that each process in the code is modularized in a way being called from ACACON, ACCOLL and ACEVMEL.

2. AROME vs. ALARO-0 microphysics

2.1 Main Differences

Although the microphysical routines of AROME (RAIN_ICE) and ALARO-0 (APLMPHYS) are similar microphysical exchanges, their followed paths are quite different. Below is a list of the main differences between AROME microphysics code RAIN_ICE and ALARO microphysics code APLMPHYS.

- In ALARO-0, the microphysical interactions act on precipitation fluxes. In AROME, the interactions are computed on the rain/snow quantities.
- In ALARO, microphysical processes are grouped in a separate stand alone subroutines. In AROME, each process are called 'in block'.

- In ALARO, sedimentation takes place on the source or sink precipitation after each process. In AROME, sedimentation is computed only once.
- In ALARO, there are 4 prognostic quantities (graupel is a diagnostic type). In AROME, there are 5 prognostic moist quantities.
- In ALARO, moist quantities are represented by mass ratios with respect to total mass. AROME uses mass ratios with respect to the mass of dry air.
- In ALARO, it has 8 possible mass exchanges. In AROME, it has about 30 possible exchanges.
- In ALARO, each layer is divided in four parts for the rain and snow ratios according to the geometric overlap of the cloudy and precipitating area's. In AROME, all mass ratios are homogeneous over the entire layer.
- In ALARO, the various species are treated in bulk and only sedimentation assumes a Marshall-Palmer distribution for rain and snow. In AROME, each process is based on a gamma law distribution. (the species are treated in bulk for computing the tendency.)

2.2 Microphysical Processes in ALARO-0 and AROME

Below is given microphysics processes in APLMPHYS and their equivalent form in RAIN_ICE and processes which are available only in RAIN_ICE. Also the processes in RAIN_ICE are categorized considering the subroutines ACACON, ACCOLL and ACEVMEL in APLMPHYS.

Processes in ALARO :

1. Autoconversion (ACACON) : q_l -> q_r and q_i -> q_s

   both processes are available inside RAIN_ICE : RCAUTR (r_c -> r_r)

   and RIAUTS (r_i -> r_s).

2. WBF (ACACON) : q_l -> q_s

   In RAIN_ICE : RCBERI (r_c -> r_i).

3. Collection (ACCOLL) : q_l -> q_r , q_l -> q_s , q_i -> q_r , q_i -> q_s .

   In RAIN_ICE : RCACCR (r_c -> r_r) , RCRIMSS (r_c -> r_s + r_g) ,

   RIAGGS (r_i -> r_s).

4. Evaporation (ACEVMEL) : q_r -> q_v and q_s -> q_v .

   In RAIN_ICE : RREVAV (r_r -> r_v).

5. Melting/freezing (ACEVMEL) : q_s -> q_r and q_r -> q_s .

   In RAIN_ICE : RSCVMG and RGMLTR (r_s -> r_g -> r_r) , RRWETG

   (r_r -> r_g).

Additional processes in AROME :

1. Ice nucleation (condensation) : RVHENI (r_v -> r_i).

   Condensation in ALARO-0 is done outside APLMPHYS.

2. Deposition : RVDEPS (r_v -> r_s), RVDEPG (r_v -> r_g)

   can be implemented in ACEVMEL as additional (inverse) evaporation pseudo-flux contribution.

3. Rain accretion on aggregates : RRACCSS (r_r + r_s -> r_s), RRACCSG

   (r_r + r_s -> r_g) , RSACCRG (r_r + r_s -> r_g) can be implemented in ACEVMEL as freezing.

4. Rain contact freezing: RRCFRIG(r_i + r_r -> r_g) and RICFRRG

   (r_i + r_r -> r_g) can be implemented in ACCOLL.

5. Dry growth : RCDRYG(r_c + r_g -> r_g) and RIDRYG(r_i + r_g -> r_g)

   can be implemented in ACCOLL. RRDRYG(r_r + r_g -> r_g) and

   RSDRYG (r_s + r_g -> r_g) can be imlemented in ACEVMEL additional pseudo-flux contributions.

6. Wet growth : RIWETG($r_i + r_g \rightarrow r_r$) can be implemented in ACCOLL. RSWETG ($r_s + r_g \rightarrow r_g$) and RRWETG ($r_r + r_g \rightarrow r_g$) can be implemented in ACEVMEL as additional contributions.
7. Cloud ice melting : RIMLTC ($r_i \rightarrow r_c$) can be implemented in ACACON.

3. Some modifications on RAIN_ICE

In order to use AROME microphysics in APMLPHYS, some modifications are needed in RAIN_ICE and APLMPHYS. Prognostic treatment of graupel and the associated additional fluxes are the main extension in new APLMPHYS with a prognostic/diagnostic switch. In AROME side, it is crucial that each process in RAIN_ICE is rewritten as a subroutine. These subroutines are stand alone and a single microphysical process. For computing the pseudo-fluxes to each process , the fluxes of the involved species should be output.
In this stay, the processes of RAIN_ICE are studied to extract as a subroutine for modularization to APLMPHYS. In the appendix , there are given some examples of the subroutines in RAIN_ICE.

4. Summary

In order to be able to call AROME microphysics in ALARO-0 , there are three unavoidable conditions:

1. The routine RAIN_ICE should become a calling sequence of subroutines, each treating a single microphysical process.
2. These subroutines should be single level, stand alone and have no model characteristic computations.
3. A solution for the graupel compatibility inside ALARO-0 must be found.

Appendix: Examples for subroutines in RAIN_ICE.

There are two examples about conversion of two microphysical process as a subroutine in RAIN_ICE which are studied in this stay.

Example 1: In the RAIN_ICE code, the process for the computing the autoconversion of r_c for r_r production is given below as a subroutine.

```
SUBROUTINE RCAUTR ( KIDIA,KFDIA,KLON,&
 !-------------------------------------------------------------------------
 ! - INPUT LOGICAL
 ! - INPUT
 &  HSUBG_AUCV,PRHOAIR,PPART, PSIGMA_RC, &
 ! - INPUT/OUTPUT .
 & PQLST, PQRST, &
 ! - OUTPUT  .
 & PZW )
!**** *RCAUTR* - CALCULS D'AUTO-CONVERSION.

!     Sujet.
!     ------
!     - COMPUTATION OF  AUTO-CONVERSION LIQUIDE => RAIN
!**   Interface.
!     ----------
!        *CALL* *RCAUTR*
! -   ARGUMENTS D'ENTREE.
!     -------------------
! - NOM DES PARAMETRES DE DIMENSIONNEMENT DE LA PHYSIQUE.


!     Externes.
!     ---------

!     Methode.
!     --------

!     Auteur.
!     -------

!     Modifications
!     --------------
!------------------------------------------------------------------------

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

USE YOMPHY4 , ONLY : RTIMAUTC, RCRIAUTC

!------------------------------------------------------------------------

IMPLICIT NONE

INTEGER(KIND=JPIM),INTENT(IN)  :: KLON
INTEGER(KIND=JPIM),INTENT(IN)  :: KIDIA
INTEGER(KIND=JPIM),INTENT(IN)  :: KFDIA
```

```fortran
CHARACTER(LEN=4), INTENT(IN)   :: HSUBG_AUCV
REAL(KIND=JPRB) , INTENT(IN)   :: PPART(KLON),PRHOAIR(KLON),PSIGMA_RC(KLON)
REAL(KIND=JPRB) , INTENT(INOUT):: PQLST(KLON),PQRST(KLON)
REAL(KIND=JPRB) , INTENT(OUT)  :: PZW(KLON)
!-------------------------------------------------------------------------

INTEGER(KIND=JPIM) :: JLON

REAL(KIND=JPRB)    :: ZCRIAUTC,ZRTMIN,ZPART, ZEPS1,ZQRST, &
         &           ZQLST,ZZW1,ZZW2,ZZW3,ZZW4,ZTEST1,ZTEST2


REAL(KIND=JPRB)   :: ZHOOK_HANDLE

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


ZRTMIN = 1.0E-20_JPRB
ZEPS1  = 1.E-10_JPRB


 IF ( HSUBG_AUCV == 'CLFR' ) THEN

    DO JLON=KIDIA,KFDIA
       ZPART=MAX(0.0_JPRB,PPART(JLON))
       ZQRST=MAX(0.0_JPRB,PQRST(JLON))
       ZQLST=MAX(0.0_JPRB,PQLST(JLON))
       PZW(JLON) =RTIMAUTC*MAX( ZQLST/(MAX(ZEPS1,ZPART))- &
                      & RCRIAUTC/PRHOAIR(JLON),0.0_JPRB)
       PZW(JLON) = MIN( ZQLST,(ZPART*PZW(JLON)))
       PQLST(JLON) = PQLST(JLON) - PZW(JLON)
       PQRST(JLON) = PQRST(JLON) + PZW(JLON)
    ENDDO

 ELSEIF ( HSUBG_AUCV == 'SIGM') THEN

     DO JLON=KIDIA,KFDIA
          ZQLST=MAX(0.0_JPRB,PQLST(JLON))
          ZCRIAUTC = RCRIAUTC / PRHOAIR(JLON)
          ZZW1 =PQLST(JLON)-ZCRIAUTC-PSIGMA_RC(JLON)
          ZZW2 =ABS(PQLST(JLON)-ZCRIAUTC)-PSIGMA_RC(JLON)
          ZZW3 = MIN( ZQLST,RTIMAUTC*(ZQLST-ZCRIAUTC))
          ZZW4 = MIN( ZQLST,RTIMAUTC*(ZQLST+PSIGMA_RC(JLON)-ZCRIAUTC)**2 &
                    &     /(4.0_JPRB*PSIGMA_RC(JLON)))
          ZTEST1=MAX(0.0_JPRB,SIGN(1.0_JPRB,ZZW1))
          ZTEST2=MIN(0.0_JPRB,SIGN(1.0_JPRB,ZZW2))
          PZW(JLON)=ZZW3*ZTEST1+ZZW4*ZTEST2
          PQLST(JLON) = PQLST(JLON) - PZW(JLON)
          PQRST(JLON) = PQRST(JLON) + PZW(JLON)
     ENDDO

  ELSE
      DO JLON=KIDIA,KFDIA
        ZQLST=MAX(0.0_JPRB,PQLST(JLON))
        ZCRIAUTC = RCRIAUTC / PRHOAIR(JLON)
        PZW(JLON) = MIN( ZQLST,RTIMAUTC*MAX( ZQLST- ZCRIAUTC, 0.0_JPRB) )
        PQLST(JLON) = PQLST(JLON) - PZW(JLON)
        PQRST(JLON) = PQRST(JLON) + PZW(JLON)
     ENDDO
```

```
 ENDIF

IF (LHOOK) CALL DR_HOOK('RCAUTR',1,ZHOOK_HANDLE)
END SUBROUTINE RCAUTR
```

Example 2:In the RAIN_ICE code , the microphysical process: RCACCR (computation of the accertion of r_c for r_r production) is below as a subroutine.

```
SUBROUTINE RCACCR ( KIDIA,KFDIA,KLON,&
 !-------------------------------------------------------------------------
 ! - INPUT LOGICAL
 ! - INPUT
 & PRHOAIR,    &
 ! - INPUT/OUTPUT .
 & PQLST, PQRST, &
 ! - OUTPUT  .
 & PZW )
!**** *RCACCR* - CALCULS DE COLLECTION

!     Sujet.
!     ------
!     - COMPUTATION THE ACCRETION OF LIQUID  TO FORM LIQUID PRECIPITATION
!**   Interface.
!     ----------
!        *CALL* *RCACCR*
! -   ARGUMENTS D'ENTREE.
!     -------------------
! - NOM DES PARAMETRES DE DIMENSIONNEMENT DE LA PHYSIQUE.
!     Externes.
!     ---------
!     Methode.
!     --------
!     Auteur.
!     -------
!     Modifications
!     --------------
!-------------------------------------------------------------------------

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

USE YOMPHY4 , ONLY  :  REXCACCR, RCEXVT, RRTMIN, RCCR, RCR, &
                    & RALPHAR, RNUR, RDR,RBR,RLBEXR,RAR, &
                    & RTHVREFZ2
USE YOMCST, ONLY  : RPI,RATM,RD,RG   , RV   , RTT  , RPI  ,&
  & RCS  , RCW  , RCPV , RLVTT, RLSTT, RETV , RALPW, RALPS,&
  & RALPD, RBETW, RBETS, RBETD, RGAMW, RGAMS, RGAMD
USE MOD_GAMMA
!-------------------------------------------------------------------------

IMPLICIT NONE

INTEGER(KIND=JPIM),INTENT(IN)    :: KLON
INTEGER(KIND=JPIM),INTENT(IN)    :: KIDIA
INTEGER(KIND=JPIM),INTENT(IN)    :: KFDIA
REAL(KIND=JPRB), INTENT(IN)      :: PRHOAIR(KLON)
REAL(KIND=JPRB), INTENT(INOUT)   :: PQLST(KLON)
REAL(KIND=JPRB), INTENT(INOUT)   :: PQRST(KLON)
REAL(KIND=JPRB), INTENT(OUT)     :: PZW(KLON)
```

```
!------------------------------------------------------------------------

INTEGER(KIND=JPIM) :: JLON

REAL(KIND=JPRB) ::GEN_GAMMA
REAL(KIND=JPRB) ::ZRHO00,ZGAMMA1,ZGAMMA2,ZLBDAR
REAL(KIND=JPRB) ::ZQLST,ZQRST,ZFCACCR,ZLBR
REAL(KIND=JPRB) ::ZHOOK_HANDLE


# include "fcttrm.h"

!------------------------------------------------------------------------
IF (LHOOK) CALL DR_HOOK('RCACCR',0,ZHOOK_HANDLE)
!------------------------------------------------------------------------
ZRHO00=RATM/(RD*RTHVREFZ2)
ZGAMMA1=GEN_GAMMA(RNUR+(RDR+2._JPRB)/RALPHAR)/GEN_GAMMA(RNUR)
ZGAMMA2=GEN_GAMMA(RNUR+RBR/RALPHAR)/GEN_GAMMA(RNUR)

ZFCACCR=(RPI/4.0_JPRB)*RCCR*RCR*(ZRHO00**RCEXVT)*ZGAMMA1
ZLBR= ( RAR*RCCR*ZGAMMA2)**(-RLBEXR)
     DO JLON=KIDIA,KFDIA

      ZQLST = MAX(0._JPRB, PQLST(JLON))
      ZQRST = MAX(0._JPRB, PQRST(JLON))
      ZLBDAR = ZLBR*PRHOAIR(JLON)*MAX(ZQRST,RRTMIN(3))**RLBEXR
      PZW(JLON)= MIN( ZQLST,  &
               &  ZFCACCR*ZQLST*ZLBDAR**REXCACCR*PRHOAIR(JLON)**(-RCEXVT))
      PQLST(JLON)= PQLST(JLON)- PZW(JLON)
      PQRST(JLON)= PQRST(JLON) + PZW(JLON)
    ENDDO

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


END SUBROUTINE RCACCR
```