

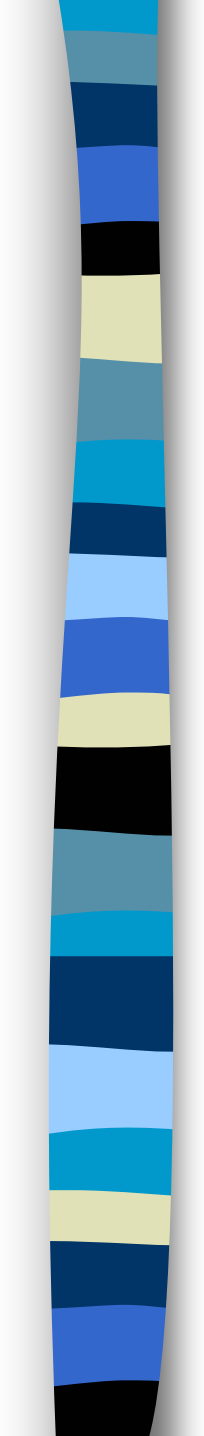


ALARO-0

Programme Item E1

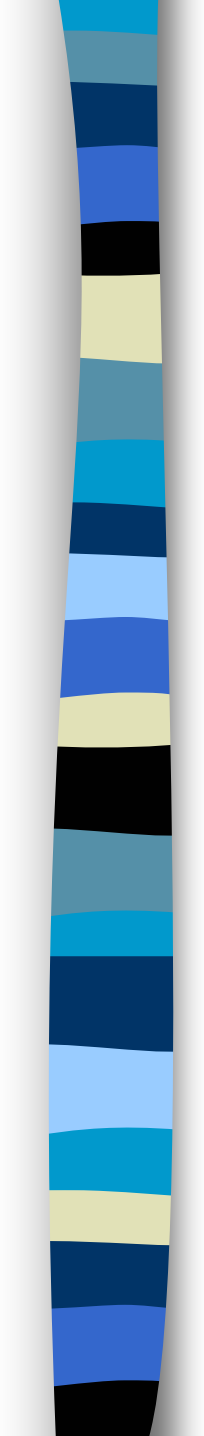
J.-F. Geleyn, ALADIN PM

TCA0, Radostovice, Czech Republic, 26-30/3/2007



Working group arrangements (1/3)

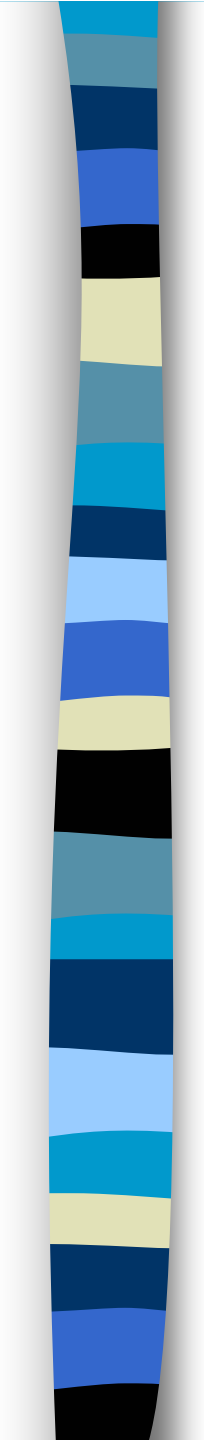
- Working Group A: (Martina => Bart), Christoph and Luc (***coding structures for modularity-flexibility and associated scientific constraints***)
- Working Group B: Jan, Filip and Joao (***rather convection-independent parameterisation issues***)
- Working Group C: Neva/Jure, (Siham) and Doina (***roughly speaking 3MT***)



Working group arrangements

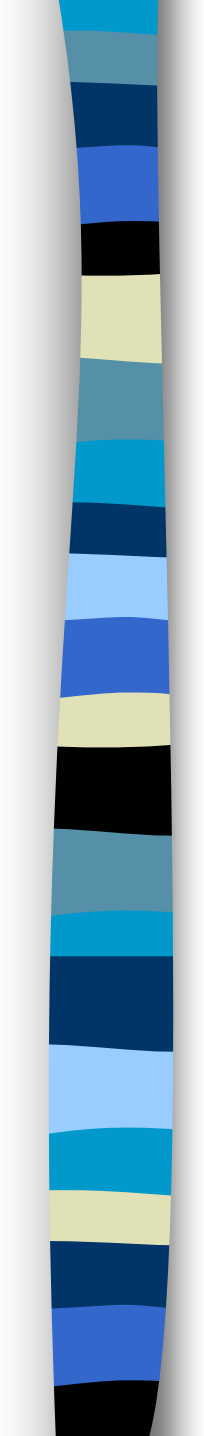
(2/3)

- The evening working group sessions are of the mixed type, with also presentations about implementations' results. Neva is responsible for this part => if not yet done, see with her for the organisation schedule. **Plus the addition on 'time-schemes' in one afternoon.**
- There was intentionally no strong guideline for the preparatory (documentation) part of the WG part. There is none either for how the WG sessions will happen.
- Christoph and Luc (+Bart) are however advised to speed up preparation of their first session.



Working group arrangements (3/3)

- The exercise is meant to ultimately produce some special documentation about the specificities of ALARO-0. Obviously we are not at that stage, but it is not the main point now.
- We are here to UNDERSTAND, COMPARE, HARMONISE and IMAGINE a transversal documentation form that will be well shaped with respect to the reinforced 'scientific maintenance' new goals of ALARO-0.
- Let see empirically how we proceed to that along the six WG sessions.



Exercise sessions arrangements (1/3)

- These are also 'hours' meant for tutorial (in parallel => TV room, bond-fire place, etc.).
- People feeling they require tutorial (past the lectures) should contact Bart, who will try and dispatch the needs. There is no shame to do, so, for sure.
- Experienced people will make efforts to meet the tutorial demand. However we cannot guarantee to fulfil this to 100%.
- Otherwise the exercise sessions should be classical, but clearly '**scientific maintenance**' oriented.



Exercise sessions arrangements

(2/3)

- There will be five types of exercises:
 - 1) **algorithmic recognition**: 2 pieces of code and one basic 'document': the aim is to find the right one between the two codes and to explain why;
 - 2) **bug search**: in a single similar piece of code to the previous case; this time the declination is correct on the paper but intentionally ill-coded;
 - 3) **algorithmic anticipation**: like in the first case (2 codes), but the difference is situated in the consequences for stability or accuracy; thus no reference document provided;
 - 4) **results' interpretation**: cases study results made available, with in principle all necessary information available for the multi-source diagnostic of a weakness;
 - 5) **modularity (in 'passive mode')**: to create the equivalent of an existing code sub-item, starting from some non-ALARO-0 scientific and/or technical documentation.



Exercise sessions arrangements

(3/3)

- Each type of exercise will be practiced, however not on a very structured and coordinated basis; sorry for that from Martin and me.
- For the code exercises, encrypting was envisaged but there was no time for it. So please do not cheat. We are here to learn, not to do a beauty contest in debugging.
- What we are here to learn is a better methodology of scientific maintenance. Hopefully the exercises will help to it.
- Let us start with 'blank examples'!



Type N°1 mini-example (1/4)

- Routine, APLPAR.
- Document, governing equations paper Catry et al., 2007 => consistency between various thermodynamical processes contributing to the thermodynamic equation (see Lecture L02).
- Which of the two ensuing solutions is “equations’ compatible”.

Type N°1 mini-example (2/4)

■ First code version

```
DO JLEV=KTDIA,KLEV
DO JLON=KIDIA,KFDIA
  ZQX1=ZQR(JLON,JLEV)-ZIPOI(JLON,JLEV)*(0.0_JPRB &
    & -(PFPFPL(JLON,JLEV)-PFPFPL(JLON,JLEV-1)) &
    & +(PFPLSL(JLON,JLEV)-PFPLSL(JLON,JLEV-1)) &
    & +(PFPEVPL(JLON,JLEV)-PFPEVPL(JLON,JLEV-1)) )
  ZQR(JLON,JLEV)=MAX(0.0_JPRB,ZQX1)
  ZDQR=MAX(0.0_JPRB,ZQX1)-ZQX1
  ZFCQRNG(JLON,JLEV)=ZFCQRNG(JLON,JLEV-1)-ZDQR*ZPOID(JLON,JLEV)
  PFCQRNG(JLON,JLEV)=PFCQRNG(JLON,JLEV)+ZFCQRNG(JLON,JLEV)

  ZQX1=ZQS(JLON,JLEV)- etc.
  ZDQS=MAX(0.0_JPRB,ZQX1)-ZQX1
  etc.
  ZDQC=ZDQR+ZDQS

  ZDFCQL=ZFCQL(JLON,JLEV)-ZFCQL(JLON,JLEV-1) &
    & - PFCSQL(JLON,JLEV)+PFCSQL(JLON,JLEV-1) &
    & - PFCCQL(JLON,JLEV)+PFCCQL(JLON,JLEV-1)
  ZFCQLDM(JLON,JLEV)=ZFCQLDM(JLON,JLEV-1)+ZDFCQL
  ZDFCQI= etc.

  ZQL(JLON,JLEV)=ZQL(JLON,JLEV)-ZIPOI(JLON,JLEV)*( &
    & (PFPFPL(JLON,JLEV)-PFPFPL(JLON,JLEV-1)) &
    & -ZDFCQL )
  ZQI(JLON,JLEV)= etc.

  PFCSQL(JLON,JLEV)=PFCSQL(JLON,JLEV)+ZFCQLDM(JLON,JLEV)
  PFCSQN(JLON,JLEV)=PFCSQN(JLON,JLEV)+ZFCQIDM(JLON,JLEV)
ENDDO
ENDDO
```

Type N°1 mini-example (3/4)

■ Second code version

```
DO JLEV=KTDIA, KLEV
  DO JLON=KIDIA, KFDIA
    ZQX1=ZQR(JLON, JLEV) - ZIPOI(JLON, JLEV) * (0.0_JPRB &
      & - (PFPPPL(JLON, JLEV) - PFPPPL(JLON, JLEV-1)) &
      & + (PFPLSL(JLON, JLEV) - PFPLSL(JLON, JLEV-1)) &
      & + (PFPEVPL(JLON, JLEV) - PFPEVPL(JLON, JLEV-1)) )
    ZQR(JLON, JLEV) = MAX(0.0_JPRB, ZQX1)
    ZDQR = MAX(0.0_JPRB, ZQX1) - ZQX1
    ZFCQRNG(JLON, JLEV) = ZFCQRNG(JLON, JLEV-1) - ZDQR * ZPOID(JLON, JLEV)
    PFCQRNG(JLON, JLEV) = PFCQRNG(JLON, JLEV) + ZFCQRNG(JLON, JLEV)

    ZQX1=ZQS(JLON, JLEV) - etc.
    ZDQS = MAX(0.0_JPRB, ZQX1) - ZQX1
    etc.
    ZDQC = ZDQR + ZDQS

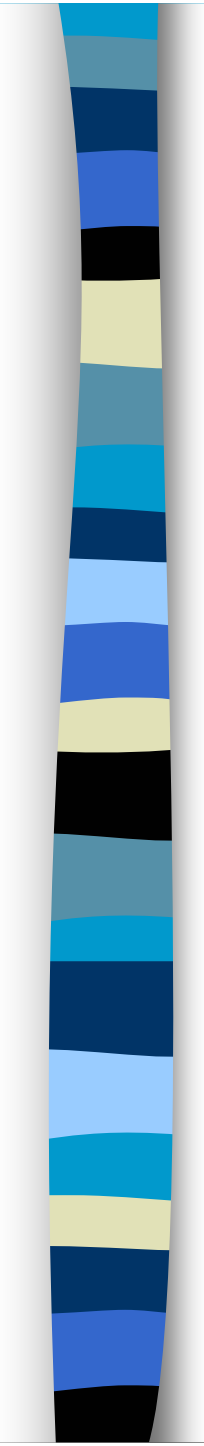
    ZDFCQL = ZFCQL(JLON, JLEV) - ZFCQL(JLON, JLEV-1) &
      & - PFCSQL(JLON, JLEV) + PFCSQL(JLON, JLEV-1) &
      & - PFCCQL(JLON, JLEV) + PFCCQL(JLON, JLEV-1)
    ZFCQLDM(JLON, JLEV) = ZFCQLDM(JLON, JLEV-1) + ZDFCQL
    ZDFCQI = etc.

    ZQL(JLON, JLEV) = ZQL(JLON, JLEV) - ZIPOI(JLON, JLEV) * ( &
      & (PFPPPL(JLON, JLEV) - PFPPPL(JLON, JLEV-1)) &
      & - ZDFCQL )
    ZQI(JLON, JLEV) = etc.

ZT(JLON, JLEV) = ZT(JLON, JLEV) - ZIPOI(JLON, JLEV) / PCP(JLON, JLEV) * &
& (0.0_JPRB - ZLHV(JLON, JLEV) * ZDFCQL - ZLHS(JLON, JLEV) * ZDFCQI)

    PFCSQL(JLON, JLEV) = PFCSQL(JLON, JLEV) + ZFCQLDM(JLON, JLEV)
    PFCSQN(JLON, JLEV) = PFCSQN(JLON, JLEV) + ZFCQIDM(JLON, JLEV)
  ENDDO
ENDDO
```

Type N°1 mini-example (3/4)

- 
- The second code version is correct. Indeed, when doing the cascading approach, the process catalogued in the ‘auto-conversion’ part of the equations set (true auto-conversion and collection) have two meanings:
 - going from cloud to precipitating species => no thermodynamical impact, only a ‘dynamical’ one;
 - For Wegener-Bergeron-Findeisen process, for collection of cloud liquid water by snow and of cloud ice water by rain, there is an associated implicit phase change before the change of species-type => there is also a thermodynamical impact.
 - This has to be reflected in the cascade.

Type N°2 mini-example (1/3)

- Routine, ACCOEFK
- Goal of the development: make the computation of the shallow convection correction to the Richardson Number
$$\frac{L \cdot \partial (q - q_{sat}(T)) / \partial \Phi}{C_p}$$

go (implicitly) from:

So-called 'non-conservative' variables:

$$s = C_p \cdot T + \Phi, \quad q_v$$

to

So-called 'moist-conservative' variables:

$$s_L = C_p \cdot T + \Phi - L_v \cdot q_l - L_s \cdot q_i, \quad q_t = q_v + q_l + q_i$$

- Author: anonymous, by definition!

Type N°2 mini-example (2/3)

- Relevant code in ACCOEFK

```
DO JLEV=KTDIA,KLEV
  DO JLON=KIDIA,KFDIA
    ZQL=PQL(JLON,JLEV)
    ZQI=PQI(JLON,JLEV)
    ZTC=PT(JLON,JLEV)-(FOLH(PT(JLON,JLEV),0.0_JPRB)*ZQL &
      & + FOLH(PT(JLON,JLEV),1.0_JPRB)*ZQI)
    ZDSE(JLON,JLEV)=PCP(JLON,JLEV)*ZTC+PAPHIF(JLON,JLEV)
    ZEW=FOEW(ZTC,ZDELTA)      The computation of ZDELTA is omitted
    ZEPS=ZEW/PAPRSF(JLON,JLEV)
    ZQSAT(JLON,JLEV)=FOQS(ZEPS)
  ENDDO
ENDDO
ELSE
  DO JLEV=KTDIA,KLEV
    DO JLON=KIDIA,KFDIA
      ZQSAT(JLON,JLEV)=PQSAT(JLON,JLEV)
      ZDSE(JLON,JLEV)=PCP(JLON,JLEV)*PT(JLON,JLEV)+PAPHIF(JLON,JLEV)
    ENDDO
  ENDDO
```

Type N°2 mini-example (3/3)

■ Relevant code in ACCOEFK

```
DO JLEV=KTDIA,KLEV
  DO JLON=KIDIA,KFDIA
    ZQL=PQL(JLON,JLEV)
    ZQI=PQI(JLON,JLEV)
    ZTC=PT(JLON,JLEV)-(FOLH(PT(JLON,JLEV),0.0_JPRB)*ZQL &
      & + FOLH(PT(JLON,JLEV),1.0_JPRB)*ZQI)/PCP(JLON,JLEV) !!!!!
    ZDSE(JLON,JLEV)=PCP(JLON,JLEV)*ZTC+PAPHIF(JLON,JLEV)
    ZEW=FOEW(ZTC,ZDELTA)      The computation of ZDELTA is omitted
    ZEPS=ZEW/PAPRSF(JLON,JLEV)
    ZQSAT(JLON,JLEV)=FOQS(ZEPS)
  ENDDO
ENDDO
ELSE
  DO JLEV=KTDIA,KLEV
    DO JLON=KIDIA,KFDIA
      ZQSAT(JLON,JLEV)=PQSAT(JLON,JLEV)
      ZDSE(JLON,JLEV)=PCP(JLON,JLEV)*PT(JLON,JLEV)+PAPHIF(JLON,JLEV)
    ENDDO
  ENDDO
```



Type N°3 mini-example (1/3)

- Routine ACEVMEL
- It computes the evaporation and melting/freezing effects on the falling precipitations.
- It must be verified that it does not lead to any stupid solution for the fluxes:
 - The sum of snow evaporation (EVAN) and algebraic 'melting' (FONT) should not cancel the the snow flux;
 - The difference of rain evaporation (EVAR) and algebraic 'melting' (FONT) should not cancel the the rain flux.

Type N°3 mini-example (2/3)

■ Relevant code in ACEVMEL

! SECURITIES.

DO JLON=KIDIA,KFDIA

ZFONTLP=PLPLSN(JLON)/(PPOID(JLON))

ZFONTLM=-PLPLSL(JLON)/(PPOID(JLON))

(A1) PFONT(JLON)=*MAX*(*MIN*(PFONT(JLON),ZFONTLP),ZFONTLM) **OR**

(A2) PFONT(JLON)=*MIN*(*MAX*(PFONT(JLON),ZFONTLP),ZFONTLM) **??**

PEVAR(JLON)=PEVAR(JLON)-PFONT(JLON)

PEVAN(JLON)=PEVAN(JLON)+PFONT(JLON)

(B1) PTEST(JLON)=1.0_JPRB-MAX(0.0_JPRB,SIGN(1.0_JPRB,PEVAN(JLON)+ZFONTLM))&
&*MAX(0.0_JPRB,SIGN(1.0_JPRB,PEVAR(JLON)-ZFONTLP)) **OR**

(B2) PTEST(JLON)=1.0_JPRB-MAX(0.0_JPRB,SIGN(1.0_JPRB,PEVAR(JLON)+ZFONTLM))&
&*MAX(0.0_JPRB,SIGN(1.0_JPRB,PEVAN(JLON)-ZFONTLP)) **??**

PEVAR(JLON)=MIN(PEVAR(JLON),-ZFONTLM)

PEVAN(JLON)=MIN(PEVAN(JLON),ZFONTLP)

PEVAR(JLON)=PEVAR(JLON)+PFONT(JLON)

PEVAN(JLON)=PEVAN(JLON)-PFONT(JLON)

ENDDO

Type N°3 mini-example (3/3)

■ Relevant code in ACEVMEL

```
! SECURITIES.
DO JLON=KIDIA,KFDIA
  ZFONTLP=PLPLSN(JLON)/(PPOID(JLON))
  ZFONTLM=-PLPLSL(JLON)/(PPOID(JLON))

  PFONT(JLON)=MAX(MIN(PFONT(JLON),ZFONTLP),ZFONTLM)
  ! PFONT(JLON)=MIN(MAX(PFONT(JLON),ZFONTLP),ZFONTLM)

  PEVAR(JLON)=PEVAR(JLON)-PFONT(JLON)
  PEVAN(JLON)=PEVAN(JLON)+PFONT(JLON)

  ! PTEST(JLON)=1.0_JPRB-MAX(0.0_JPRB,SIGN(1.0_JPRB,PEVAN(JLON)+ZFONTLM))&
  ! &*MAX(0.0_JPRB,SIGN(1.0_JPRB,PEVAR(JLON)-ZFONTLP))
  PTEST(JLON)=1.0_JPRB-MAX(0.0_JPRB,SIGN(1.0_JPRB,PEVAR(JLON)+ZFONTLM))&
  &*MAX(0.0_JPRB,SIGN(1.0_JPRB,PEVAN(JLON)-ZFONTLP))

  PEVAR(JLON)=MIN(PEVAR(JLON),-ZFONTLM)
  PEVAN(JLON)=MIN(PEVAN(JLON),ZFONTLP)
  PEVAR(JLON)=PEVAR(JLON)+PFONT(JLON)
  PEVAN(JLON)=PEVAN(JLON)-PFONT(JLON)
ENDDO
```



Conclusion

Enjoy the next two lectures (L04 & L05), after that the unpredictable part of the Training Course starts!