

DDH for AROME

Tomislav Kovacic

August 28, 2006

Contents

1	Introduction	4
2	Description of DDH	4
2.1	What is DDH?	4
2.2	Conservation equations	4
2.3	Vertical discretization and averaging	5
2.4	Classification of terms	5
2.5	Domains	6
3	Namelist	7
3.1	Namelist NAMDDH	7
3.1.1	Choosing domain type - turning on DDH	7
3.1.2	Scientific options	7
3.1.3	Outputs	7
3.1.4	Descriptions of DDH state	8
3.1.5	New switches - for AROME only	8
3.1.6	Limited domains	8
3.2	Namelist NAMPHY (AROME only)	9
3.3	Namelist NAMARPHY (AROME only)	9
3.4	Namelist NAMPARAR (AROME only)	10
3.5	Namelist NAMCT0 - control of output times	10
3.6	Namelist NAMOP	10
3.7	Namelist NAMGFL	10
4	Short description of DDH in ARPEGE/ALADIN	11
4.1	DDH subroutines	11
4.2	Data flow	11
4.3	Adding a new term	11
5	DDH in AROME	14
5.1	Microphysical process	14
5.2	Balance equations	15
5.2.1	Momentum	15
5.2.2	Turbulence kinetic energy	15
5.2.3	Enthalpy	16
5.2.4	Water vapour	16
5.2.5	Cloud water	17
5.2.6	Rain	17
5.2.7	Cloud ice	17
5.2.8	Snow	18
5.2.9	Graupel	18
5.3	Common Dynamics-Physics Interface CDPI	18
5.4	New module YOMPFT	18
5.5	Data flow	20

5.6	Setup	20
5.6.1	Subroutine SU0YOMA	20
5.6.2	Subroutine SU0YOMB	21
5.7	Entering physical fluxes and tendencies in DDH	21
5.8	Modified subroutines	22
5.8.1	Subroutine APL_AROME	22
5.8.2	Subroutine AROEND_BUDGET	22
5.8.3	Subroutine CPDYDDH	23
5.8.4	Subroutine CPG_DIA	23
5.8.5	Subroutine CPG	23
5.8.6	SU0PHY	23
5.8.7	Subroutine SUPHMNH	23
5.8.8	Subroutine SUNDH	24
5.8.9	Subroutine PPFIDH	25
5.9	Modified modules	25
5.9.1	Module YOMARPHY	25
5.9.2	Module YOMLDDH	25
5.9.3	Module YOMMNH	26
5.9.4	Module YOMPHY	26
5.10	New subroutines	27
5.10.1	Subroutine ARO_CPPHDDH	27
5.10.2	Subroutine ADDFT	27
5.10.3	Subroutine ARO_INIPFT	27
5.11	Adding a new term in DDH for AROME	28

1 Introduction

DDH (Diagnostics en Domaines Horizontaux) is an important tool which gives us an insight in the work of the model. In short it can be described as another way of getting outputs from the model. While in historical files prognostic variables are saved, in DDH files horizontally averaged variables and horizontally averaged and in time integrated terms of tendency equations are saved (see 2.4). During the model run DDH collects various terms of governing equations, puts them in flux form, integrates them and writes in the file at predefined times. DDH was first introduced in ARPEGE, in 1991, and recently it's usage in ALADIN was enabled. Now DDH is spread to AROME. Before talking about DDH for AROME we should recall that AROME consists of non hydrostatic ALADIN dynamics core and physical package from MesoNH. This brings new and different terms in governing equations compared to those in ALADIN. Therefore these new terms have to be introduced in DDH as well. For the time being this is done only for terms coming from physical parameterizations. Because of great difference in physical packages in ALADIN and AROME, physical fluxes in AROME are not just added to those in ALADIN but they are treated separately and in different way. Recently, common dynamics-physics interface was defined in concept ALARO and fluxes defined by it are introduced in DDH for AROME as well.

The aim of this paper is to explain the treatment of physical fluxes in DDH for AROME. First theoretical background of DDH is given. Then, namelists effecting DDH are explained in section 3. From the coding point of view DDH is described in sections 4 (ARPEGE/ALADIN) and 5 (AROME). To use DDH it is enough to read only sections 2 and 3. Detailed description of DDH for ARPAGE and ALADIN can be found in [2].

2 Description of DDH

2.1 What is DDH?

Diagnostics in horizontal domain means to check how good the conservation equations averaged over horizontal domain are fulfilled and how individual terms are effecting variables changing. This is done in two steps. First, during the model run horizontally averaged terms of conservation equations are saved in files. Then diagnostic is done using DDH tools, see [3]. This is a set of programs that perform various operations on data saved in DDH files. In DDH files data are written on all model vertical levels, so vertical profiles can be done for variables and conservation equation terms.

2.2 Conservation equations

DDH is based on model equations written in flux form. If χ is a model variable then the most general form of prognostic equation assumed in DDH written in the flux form is:

$$\frac{|c|}{\overline{\chi p \eta} = - \underbrace{\overline{\eta \chi \vec{v} p \eta}}_{(1)} - \underbrace{\overline{\eta \chi \dot{\eta} p \eta}}_{(2)} + \underbrace{\overline{S_d p \eta}}_{(3)} - \underbrace{g \overline{F_\varphi \eta}}_{(4)} - \underbrace{g \overline{S_\varphi G_\varphi \eta}}_{(5)}$$

where terms are divided in five kinds with following descriptions:

- 1. Horizontal divergence,
- 2. Vertical divergence,
- 3. Adiabatic source (pressure gradient force, ...),
- 4. Divergence of physical fluxes (turbulence, microphysics, ...),
- 5. Tendencies due to physical parameterizations (such terms appear in kinetic energy equation where divergence of physical fluxes is multiplied by velocity).

One characteristics of this equation is underlying assumption that physical processes give only vertical fluxes; no physical tendencies are expected.

χ can be one of following: $1, \vec{v}, q_v, q_l, q_n, k, C_p, \vec{M}, s$

2.3 Vertical discretization and averaging

After discretizing conservation equation in vertical direction and applying horizontal averaging over chosen domain and time integrating from the start time to the recent time step, we get:

$$\overline{(\chi \delta p^H)}_{n \Delta t} - \overline{(\chi \delta p^H)}_0 = \delta t \sum_{jstep=0}^{nstep-1} [-\overline{\eta \chi \vec{v} \delta p^H} - \overline{\delta \chi \dot{\eta} p \eta^H} + \overline{S_d \delta p^H} - g \overline{\delta F_\varphi^H} - g \overline{S_\varphi \delta G_\varphi^H}]_{jstep}$$

This equation is valid at each model level ℓ . Operator $\delta \xi_\ell$ is defined as:

$$\delta \xi_\ell = \xi_{\tilde{\ell}} - \xi_{\tilde{\ell}-1}$$

where $\xi_{\tilde{\ell}}$ is the value of ξ at half levels $\tilde{\ell}$.

2.4 Classification of terms

In DDH data are divided in several classes or categories, these are:

- 1. Variables: $\delta t \sum_{jstep=0}^{nstep-1} \frac{1}{g} \overline{\chi \delta p^H}$,
- 2. Dynamical tendencies: $\delta t \sum_{jstep=0}^{nstep-1} \frac{\delta t}{g} \overline{\eta \chi \vec{v} \delta p^H}$ and $\delta t \sum_{jstep=0}^{nstep-1} \frac{\delta t}{g} \overline{S_d \delta p^H}$,

- 3. Dynamical fluxes: $\delta t \sum_{jstep=0}^{nstep-1} \frac{\delta t}{g} \overline{\chi \eta p \eta}^H$,
- 4. Physical fluxes: $\delta t \sum_{jstep=0}^{nstep-1} \delta t \overline{F}_\varphi^H$,
- 5. Physical tendencies: $\delta t \sum_{jstep=0}^{nstep-1} g \overline{S}_\varphi \delta \overline{G}_\varphi^H$.

Again, χ can be one of following: $1, \vec{v}, q_v, q_l, q_n, k, C_p, \vec{M}, s$

There are NFLEV variables and tendencies per domain because they are placed on full levels and NFLEV+1 fluxes, placed on the half levels. Besides, there are variables and fluxes at surface.

2.5 Domains

In DDH output files horizontally averaged data are saved. There are three kinds of horizontal domain over which averaging can be done. These are:

- **global domain:** the entire model domain,
- **zonal domain:** the Earth is divided in NDHKD zones of equal area,
- **local domain:** a part of model domain.

Outputs for each kind of domain go in separate files. All tree domain kinds can be defined at the same time. More over, several local domains can be defined at the same time, too. If local domains are overlapping they must be on different virtual plains, otherwise common points will belong only to domain which was defined first and the later domain will take its original shape without common points. If there are no overlapping domains virtual plain has always number 1. When we introduce a new domain we must put it on the virtual plain where it doesnt overlap with other domains. If there is no such virtual plain we must raise the number of virtual plains by one and put domain on this new virtual plain. To put domain on virtual plain means to put the ordinal number of virtual plain in domain's description, see 3.1.6.

There are fore local domain types, these are:

1. Point given by it's horizontal indices,
2. Quadrangle given by geographical coordinates of each corner (in clockwise direction),
3. Rectangle given by geographical coordinates (λ , θ) of opposite corners,
4. Point given by geographical coordinates (λ , θ).

More about local domains see at 3.1.6.

3 Namelists

3.1 Namelist NAMDDH

NAMDDH contains switches controlling DDH and BDEDDH array used for definition of limited domains.

3.1.1 Choosing domain type - turning on DDH

Following switches are used to determine domain kind. When switch is `.TRUE.` corresponding domain is used.

- LHDGLB: global domain ,
- LHDZON: zonal domain,
- LHDDOP: limited domains or points.

DDH will work only if at least one of these three switches is `.TRUE.`.

3.1.2 Scientific options

Scientific options give a possibility to chose equations for which diagnosis is done or which terms are saved in DDH files.

- LHDHKS: continuity equation, momentum equation, kinetic energy equation, continuity of relative humidity and water species, balance on soil,
- LHDMCI: angular momentum equation,
- LHDENT: entropy equation.

3.1.3 Outputs

Output of DDH data can be done either on printer or in file, or both.

- LHDEFG: writing in global domain file,
- LHDEFZ: writing in zonal band domain file,
- LHDEFD: writing in file with data from local domains and points (all in one file),
- LHDPRG: printing of global domain data,
- LHDPRZ: printing of zonal band data,
- NDHZPR: index of the zomal band domain for printing,
- LHDPRD: printing of horizontal means for limited domains,
- LHDFIL: printing article names for articles written in DDH output file.

If no output switch for chose domain type is set `.TRUE.` , program will abort.
Printing of output data must not be used in AROME.

3.1.4 Descriptions of DDH state

- LHDLIST: key for more extensive description of DDH state on standard output.

3.1.5 New switches - for AROME only

- LONLYVAR: if it is .TRUE. only horizontal means of variables are saved in DDH files,
- LHDORIGP: if it is .TRUE. horizontal means of fluxes or tendencies from original microphysical scheme are saved in DDH files. In AROME this is a microphysical scheme from Meso NH with five water species,
- LHDCDPI: if it is .TRUE. horizontal means of fluxes defined in Common Dynamics- Physics Interface (CDPI) are saved in DDH files, see 3.2.
If LONLYVAR and LHDCDPI is .FALSE. then LHDORIGP is set .TRUE. automatically.

3.1.6 Limited domains

BEDDDH is an REAL type array containing description of domains. It is declared as BEDDDH (10, JPDHNOX), where JPDHNOX is a PARAMETER determining a maximal number of limited domains. We see that one domain can be described with maximum of 10 real numbers.

The first two numbers are:

- BEDDDH(1, domain ordinal) = domain type,
- BEDDDH(2, domain ordinal) = virtual plain ordinal.

The meaning of next eight numbers (indices 3 to 10) depends on domain type.

type 1

- BEDDDH(3, domain ordinal) = rjlon
- BEDDDH(4, domain ordinal) = rjgl

type 2

- BEDDDH(3, domain ordinal) = longitude of the first corner, in degrees, λ_1
- BEDDDH(4, domain ordinal) = latitude of the first corner, in degrees, θ_1
- (BEDDDH(5, -), BEDDDH(6, -)) = (λ_2 , θ_2)
- (BEDDDH(7, -), BEDDDH(8, -)) = (λ_3 , θ_3)
- (BEDDDH(9, -), BEDDDH(10, -)) = (λ_4 , θ_4)

type 3

- BDEDDH(3, domain ordinal) = longitude of the first corner, in degrees, λ_1
- BDEDDH(4, domain ordinal) = latitude of the first corner, in degrees, θ_1
- (BDEDDH(5, -), BDEDDH(6, -)) = (λ_3 , θ_3)

type 4

- BDEDDH(3, domain ordinal) = longitude λ in degrees
- BDEDDH(4, domain ordinal) = latitude θ in degrees

In this case diagnostics is done in the network point nearest to the given geographical coordinates.

3.2 Namelist NAMPHY (AROME only)

Already for some time there exists idea of an universal interface between dynamical part (adiabatic equations) and physical part (physical processes and parameterizations) of numerical models. Universal interface should allow usage of various physical packages in the model in the same manner. An proposal of such interface is given by [1] and we call it Common Dynamics-Physics Interface (CDPI). CDPI allows only some microphysics processes and they have a form of fluxes or pseudo-fluxes. All other processes are taken in to account through pseudo-fluxes between water vapour and water species taking a part in them. Microphysical scheme in AROME has much more processes than in CDPI and CPDI fluxes/pseudo-fluxes must be calculated from tendencies coming from microphysical processes in AROME. In AROME there is a possibility of slight difference from original definition of microphysical processes in CPDI, here they can take a form of tendencies (see 3.3) as well. A new logical switch is introduced to control calculation of CPDI fluxes/pseudo-fluxes.

- LPHCDPI: physical fluxes or tendencies for Common Dynamics-Physics Interface are calculated if it is .TRUE.

3.3 Namelist NAMARPHY (AROME only)

As mentioned earlier, CPDI processes can be calculated as fluxes or tendencies in AROME and the same holds for original processes.. This possibility is controlled by logical switch LBUFLUX.

- LBUFLUX if it is .TRUE. fluxes are calculated from physical tendencies saved in budget arrays. Only this option is available at the time being. Nothing will happen if we put it to .FALSE. but tendencies will be wrongly saved in DDH files.

3.4 Namelist NAMPARAR (AROME only)

One new logical key is added.

- `LAROBU_ENABLE`: key for budget calculations. It is `.TRUE.` if any of `LPHCDPI`, `LSDDH` and `LONLYVAR` is `.TRUE.`. In the namelist it is used only when we want to turn on saving in budget arrays if it is not done by other keys. This has sense only to test the work of budget analysis.

3.5 Namelist NAMCT0 - control of output times

Four variables from `NAMCT0` are used to control output frequency of DDH data. These are two integers of the form `NFRx` and two integer arrays having a form `NxTS(0:JPNPST)`, where `x` stays for `DHPD` or `DHFD`. `DHFD` is for writing in file and `DHPD` for printing.

`JPNPST` is defined in module `YOMCT0`.

Printing of output data must not be used in `AROME`.

There are three ways to define output time:

1. If `NFRx = n` ($n > 0$) and `NxTS(0) = 0`: output after every n time steps.
2. If `NFRx = n` ($n > 0$), `NxTS(0) = m` ($m > 0$) and `NxTS(i) = pi` ($p_i \geq 0$, $i \in 1, \dots, m$): output at every np_i time step.
3. If `NFRx = n` ($n > 0$), `NxTS(0) = -m` ($m > 0$) and `NxTS(i) = -pi` ($p_i \geq 0$, $i \in 1, \dots, m$): output at every np_i hour.

3.6 Namelist NAMOP

Only logical switch `LINC` is important for DDH. It must be in accordance with output times given in `NAMCT0`. Otherwise, it can happen that at several output times DDH files have the same name. In that case new data are overwriting older ones.

- `LINC`: if it is `.FALSE.` the number at the end of output file name is expressed in time steps, otherwise in hours.

3.7 Namelist NAMGFL

For dynamical tendencies horizontal derivatives of model variables are needed. Horizontal derivatives for water species and kinetic energy of turbulence are calculated only if attribute `LCDERS` is `.TRUE.` for `YQ_NL`, `YL_NL`, `YL_NL`, `YS_NL`, `YR_NL`, `YG_NL` and `YTKE_NL`.

4 Short description of DDH in ARPEGE/ALADIN

4.1 DDH subroutines

Subroutines important for DDH and their position relative to STEPO are shown on Fig. 1. Only subroutines important to understand implementation of DDH in AROME are shown. In subroutines APELPA and APL_AROME physical parameterizations are calculated. Later in CPG, within CPG_DIA, DDH subroutines CPDYNDH, CPPHDDH and CPCUDDH are called. Finally, at output time, subroutine POSDDH is called from STEPO, and for writing data in files PPFIDDH is called. Short and rather crude description of the most important DDH subroutines is following:

- CPDYNDH: saves variables and dynamical fluxes and tendencies in local array PDHCV. Data are saved for each point of each domain,
- CPPHDDH: saves physical fluxes and tendencies in local array PDHCV. Data are saved for each point of each domain,
- CPCUDDH: takes array PDHCV from CPG_DIA and calculates horizontal averages and performs summation in time,
- POSDDH: finishes calculation of averages and calls output subroutines,
- PPFIDDH: writes data in file.

More detailed description of DDH code is given in ...

4.2 Data flow

Only a part of the data flow needed to put physical fluxes and tendencies in DDH is described here, because this part is changed in AROME. As shown on Fig. 2 physical fluxes and tendencies are calculated in APLPAR, they are passed to MF_PHYS and through MF_PHYS to CPG. CPG passes them to CPG_DIA. On this way from, APLPAR into CPG_DIA data are not changed. DDH subroutines CPDYNDH and CPPHDDH called in CPG_DIA prepare data for saving in DDH. All other subroutines, after APLPAR, are just transporting data to CPDYNDH and CPPHDDH.

4.3 Adding a new term

Each new term must belong to one of five classes and must be entered under one of three scientific options (LHDHKS, LHDMCI, LHDENT). Besides these the term can belong to parameterization of soil processes, but this is not concerned here because we concentrate on the problem of adding new terms coming from parameterization of physical processes.

To add a new term, e.g. when new parameterization is used, following changes in the program have to be done:

Figure 1: Position of subroutines producing physical fluxes/tendencies and DDH subroutines during time step

Figure 2: Data flow for DDH in ARPEGE/ALADIN.

1. Changing pointers
This is done in SUNDDH. Number of terms in the category to which the new term belongs must be raised by one.
2. Add a description of the new term
In YOMTDDH a description of the new term must be added as a comment. This is done only for documentation.
3. Putting a new term in DDH arrays This is done by addition of code in CPDYDDH or CPDYDDH. There, a new term must be saved either in PDHCV or PDHCS. New lines of code look like following:

```

C      n.- CALCULATIONS FOR CHOUSEN CHATEGORY
          IDHCV=pointer to the starting position of category

C      IF (scientific option) THEN
          DO n000 JLEV=1,number of levels
          DO n001 JROF=1,KPROF
C      term description 1
          PDHCV(JROF,JLEV,IDHCV+1)= expression that calculates the term
C      term description 2
          PDHCV(JROF,JLEV,IDHCV+2)= expression that calculates the term
          ...
C      term description n
          PDHCV(JROF,JLEV,IDHCV+n)= expression that calculates the term
here we put a new term
          PCHCV(JROF,JLEV,IDHCV+n+1)= expression that calculates the term
n001 CONTINUE
n000 CONTINUE
          IDHCV=IDHCV+n+1 (pointer for the next category is raised by one)
          ENDIF
          etc

```

4. Adjusting output In subroutine PPFIDH following lines must be added for each new term.

```

WRITE(CLNOMA,5000) KNUM,'category','variable',10 character long name
CALL LFIECR(ICOREP,NPODDH,CLNOMA,PDDHCV(0 ou 1,IDHCV+n+1,length)

```

Starting index for the next category must be raised by one for each new term. IDHCV=IDHCV+n+1

5. Adjusting output on printer. This is done in PPEDDH. For AROME this adjustment is not done.

5 DDH in AROME

5.1 Microphysical process

In AROME physical parameterizations from MesoNH are used. The greatest difference compared to parameterizations used in ALADIN, in respect of number and nature of fluxes and tendencies, appears in microphysics. Microphysical processes in AROME with corresponding tendencies/fluxes are given in table 1. Each column belongs to one microphysical species; tendencies/fluxes written in it change it's species amount. In the last column are tendencies/fluxes that change enthalpy due to microphysical processes. On output from MesoNH subroutines only tendencies are available but later they are changed to fluxes if LBUFLUX is .TRUE..

Microphysical process	vapour	cloud wa- ter	rain	cloud ice	snow	graupel	enthalpy
water vapour adjustment		$F_{q_l}^{cdepi}$		$F_{q_i}^{cdepi}$			F_h^{cdepi}
heterogeneous nucleation	$F_{q_v}^{henu}$			$F_{q_i}^{henu}$			F_h^{henuv}
homogeneous nucleation		$F_{q_l}^{hon}$		$F_{q_i}^{hon}$			F_h^{honl}
spontaneous freezing			$F_{q_r}^{sfrz}$			$F_{q_a}^{sfr}$	F_h^{sfr}
deposition on snow	$F_{q_v}^{deps}$				$F_{q_s}^{dep}$		F_h^{deps}
collection of ice on snow				$F_{q_i}^{agg}$	$F_{q_s}^{agg}$		
auto-conversion of ice to snow				$F_{q_i}^{auto}$	$F_{q_s}^{auto}$		
deposition on graupel	$F_{q_v}^{depg}$					$F_{q_a}^{dep}$	F_h^{depg}
auto-conversion of cloud water		$F_{q_l}^{auto}$	$F_{q_r}^{auto}$				
accretion		$F_{q_l}^{accr}$	$F_{q_r}^{accr}$				
rain evaporation	$F_{q_v}^{reva}$		$F_{q_r}^{reva}$				F_h^{reva}
riming by cloud droplets		$F_{q_l}^{rim}$			$F_{q_s}^{rim}$	$F_{q_a}^{rim}$	F_h^{rim}
collection of raindrops			$F_{q_r}^{accs}$		$F_{q_s}^{accs}$	$F_{q_a}^{accs}$	F_h^{accs}
melting of aggregates					$F_{q_s}^{cmel}$	$F_{q_a}^{cmel}$	
contact freezing			$F_{q_r}^{cfrz}$	$F_{q_i}^{cfrz}$		$F_{q_a}^{cfrz}$	F_h^{cfrz}
wet growth		$F_{q_l}^{wetg}$	$F_{q_r}^{wetg}$	$F_{q_i}^{wetg}$	$F_{q_s}^{wetg}$	$F_{q_a}^{wetg}$	F_h^{wetg}
dry growth		$F_{q_l}^{dryg}$	$F_{q_r}^{dryg}$	$F_{q_i}^{dryg}$	$F_{q_s}^{dryg}$	$F_{q_a}^{dryg}$	F_h^{dryg}
melting of graupel			$F_{q_r}^{mltg}$			$F_{q_a}^{mltg}$	F_h^{mltg}
melting of cloud ice		$F_{q_l}^{mlti}$		$F_{q_i}^{mlti}$			F_h^{mlti}
Bergeron-Findeisen effect		$F_{q_l}^{berfi}$		$F_{q_i}^{berfi}$			F_h^{berfi}

Table 1: Microphysical fluxes.

5.2 Balance equations

In this section are given fluxes from physical parameterizations in AROME by conservation equations where they appear. Fluxes are presented by names in DDH files and symbols. Physical process is described by superscript in symbolic name of flux. Superscript starting with tur means turbulence, neg stays for correction of negative values, conv is for convection parameterization and ray is for radiation. The meaning of other superscripts can be found in table 5.1. In subscripts are symbols of prognostic variables.

5.2.1 Momentum

$$\begin{aligned} \text{FUUTUR} & F_u^{tur} \\ \text{FVVTUR} & F_v^{tu} \\ \text{FVWTUR} & F_w^{tur} \end{aligned}$$

5.2.2 Turbulence kinetic energy

$$\begin{aligned} \text{FTETURB} & F_{tke}^{tur} \\ \text{FTEDYPRO} & F_{tke}^{tur-prod-dyn} \\ \text{FTETERMPRO} & F_{tke}^{tur-prod-term} \\ \text{FTEDISS} & F_{tke}^{tur-diss} \end{aligned}$$

5.2.3 Enthalpy

FCTNEGC1	F_h^{negc1}
FCTCDEPI	F_h^{cdipi}
FCTVCONV	F_h^{conv}
FCTVTURB	F_h^{tur}
FCTDISSTUR	$F_h^{tur-diss}$
FCTNEGC	F_h^{negc}
FCTHENUI	F_h^{henuv}
FCTHON	F_h^{honl}
FCTSFR	F_h^{sfr}
FCTDEPS	F_h^{deps}
FCTDEPG	F_h^{depg}
FCTREVA	F_h^{reva}
FCTRIM	F_h^{rim}
FCTACCS	F_h^{accs}
FCTCFRZ	F_h^{cfrz}
FCTWETG	F_h^{wetg}
FCTDRYG	F_h^{dryg}
FCTMLTG	F_h^{mltg}
FCTMLTI	F_h^{mlti}
FCTBERFI	F_h^{berfi}
FCTRAYSOL1	$F_h^{raysol1}$
FCTRAYTER1	$F_h^{rayter1}$

5.2.4 Water vapour

FQVNEGC1	$F_{q_v}^{negc1}$
FQVDEPI	$F_{q_v}^{dipi}$
FQVVCONV	$F_{q_v}^{conv}$
FQVVTURB	$F_{q_v}^{tur}$
FQVNEGC	$F_{q_v}^{negc}$
FQVHENUI	$F_{q_v}^{henu}$
FQVDEPS	$F_{q_v}^{deps}$
FQVDEPG	$F_{q_v}^{depg}$
FQVREVA	$F_{q_v}^{reva}$

5.2.5 Cloud water

FQLNEGC1	$F_{q_l}^{negc1}$
FQLCDEPI	$F_{q_l}^{cdepi}$
FQLVCONV	$F_{q_l}^{conv}$
FQLVTURB	$F_{q_l}^{tur}$
FQLNEGC	$F_{q_l}^{negc}$
FQLHON	$F_{q_l}^{hon}$
FQLAUTO	$F_{q_l}^{autor}$
FQLACCR	$F_{q_l}^{accr}$
FQLRIMS	$F_{q_l}^{rim}$
FQLWETG	$F_{q_l}^{wetg}$
FQLDRYG	$F_{q_l}^{dryg}$
FQLMLTI	$F_{q_l}^{mlti}$
FQLBERFI	$F_{q_l}^{berfi}$

5.2.6 Rain

FQRNEGC	$F_{q_r}^{negc}$
FQRSEDI	$F_{q_r}^{p}$
FQRSFR	$F_{q_r}^{sfrz}$
FQRAUTO	$F_{q_r}^{autor}$
FQRACCL	$F_{q_r}^{accr}$
FQRREVA	$F_{q_r}^{reva}$
FQRACCS	$F_{q_r}^{accs}$
FQRCFRZ	$F_{q_r}^{cfrz}$
FQRWETG	$F_{q_r}^{wetg}$
FQRDRYG	$F_{q_r}^{dryg}$
FQRMLTG	$F_{q_r}^{mltg}$

5.2.7 Cloud ice

FQINEGC1	$F_{q_i}^{negc1}$
FQICDEPI	$F_{q_i}^{cdepi}$
FQICONV	$F_{q_i}^{conv}$
FQITURB	$F_{q_i}^{tur}$
FQINEGC	$F_{q_i}^{negc}$
FQISEDI	$F_{q_i}^{p}$
FQIHENU	$F_{q_i}^{henu}$
FQIHON	$F_{q_i}^{hon}$
FQIAGGS	$F_{q_i}^{agg}$
FQIAUTS	$F_{q_i}^{autoi}$
FQICFRZ	$F_{q_i}^{cfrz}$
FQIWETG	$F_{q_i}^{wetg}$
FQIDRYG	$F_{q_i}^{dryg}$
FQIMLT	$F_{q_i}^{mlti}$
FQIBERFI	$F_{q_i}^{berfi}$

5.2.8 Snow

FQSNEGC	$F_{q_s}^{negc}$
FQSSEDI	F_{sp}
FQSDEPS	$F_{q_s}^{dep}$
FQSAGGS	$F_{q_s}^{agg}$
FQSAUTS	$F_{q_s}^{autoi}$
FQSRIM	$F_{q_s}^{rim}$
FQSACC	$F_{q_s}^{accs}$
FQSCMEL	$F_{q_s}^{cmel}$
FQSWETG	$F_{q_s}^{wetg}$
FQSDRYG	$F_{q_s}^{dryg}$

5.2.9 Graupel

FQGNEGC	$F_{q_g}^{negc}$
FQGSEDI	F_{gp}
FQGSFR	$F_{q_g}^{sfr}$
FQGDEPG	$F_{q_g}^{dep}$
FQGRIM	$F_{q_g}^{rim}$
FQGACC	$F_{q_g}^{accs}$
FQGCMEL	$F_{q_g}^{cmel}$
FQGCFRZ	$F_{q_g}^{cfrz}$
FQGWETG	$F_{q_g}^{wetg}$
FQGDYRG	$F_{q_g}^{dryg}$
FQGMLT	$F_{q_g}^{mltg}$

5.3 Common Dynamics-Physics Interface CDPI

FQVPL1	P'_l
FQVPI1	P'_i
FQLPL2	P''_l
FQIPI2	P''_i
FQRPL3	P'''_l
FQSPI3	P'''_i
FQGPG3	P'''_g
FQRPR0	P_r
FQIPI0	P_i
FQSPS0	P_s
FQGPG0	P_g

5.4 New module YOMPHT

Procedure of adding a new term in DDH in ARPEGE/ALADIN is already described. Mainly, for each new term one line of program code must be added in CPDYDDH or CPDYDDH and two lines in PPFIDH. In AROME a lot of new

fluxes/tendencies are introduced, and many old ones are not used. Adding new lines for each new term would make a code quite unreadable. For this reason and to avoid caring fluxes/tendencies through subroutines a new module YOMPHFT is introduced.

In this module a data type, TYPE_APFT, is defined to carry a description of data.

```

TYPE TYPE_APFT
  CHARACTER(LEN=1)  :: CFT
  CHARACTER(LEN=2)  :: CVAR
  CHARACTER(LEN=10) :: CNAME
END TYPE TYPE_APFT

```

The most important in module YOMPHFT are two arrays declared as:

```

TYPE(TYPE_APFT), ALLOCATABLE :: YAPFT(:)    !descriptions of fluxes & tendencies
REAL(KIND=JPRB), ALLOCATABLE :: APFT(:, :, :) ! array with fluxes & tendencies

```

Dimensions of APFT are:

- 1. dimension: horizontal extend,
- 2. dimension: vertical extend,
- 3. dimension: NAPHFT (described later).

A dimension of YAPFT is NAPHFT; description in YAPFT must be at the same index value as the third index of corresponding flux/tendency in APFT.

In APFT data at each point are sorted so that for each variable it's value at start time and output time and terms appearing in conservation equation for one variable stay one after another. The order in which variables enter array APFT is given in 5.10.3. Number of variables is given with:

```

INTEGER(KIND=JPIM) :: NPROGVAR

```

Terms that belong to each conservation equation can be find in APFT by virtue of two arrays, defined as follows:

```

INTEGER(KIND=JPIM), ALLOCATABLE :: MJJ1(:) !first index in APFT for variable 'i'
INTEGER(KIND=JPIM), ALLOCATABLE :: MJJ2(:) !last index in APFT for variable 'i'

```

Numbers of fluxes/tendencies in APFT and YAPFT may differ, because APFT can be used for saving CDPI fluxes when DDH is not active, and are given with:

```

INTEGER(KIND=JPIM) :: NAPHFT !number of fluxes/tendencies in APFT
INTEGER(KIND=JPIM) :: NDDHFT !number of fluxes/tendencies in YAPFT

```

Physical fluxes can be put in the form needed for Common Dynamics-Physics Interface (CDPI). Number of CDPI fluxes and their positions in APFT (third index) are defined with following integers:

```

INTEGER(KIND=JPIM)  :: NCDPIPHFT ! number of fluxes/tendencies in CDPI,
INTEGER(KIND=JPIM)  :: NQVQL1   ! positions of Pl'
INTEGER(KIND=JPIM)  :: NQVQI1   ! positions of Pi'
INTEGER(KIND=JPIM)  :: NQLQR2   ! positions of Pl''
INTEGER(KIND=JPIM)  :: NQIQS2   ! positions of Pi''
INTEGER(KIND=JPIM)  :: NQRQV3   ! positions of Pl'''
INTEGER(KIND=JPIM)  :: NQSQV3   ! positions of Pi'''
INTEGER(KIND=JPIM)  :: NQGQV3   ! positions of Pg'''
INTEGER(KIND=JPIM)  :: NQR0     ! positions of Pr - flux of rain
INTEGER(KIND=JPIM)  :: NQIO     ! positions of Pi - flux of cloud ice
INTEGER(KIND=JPIM)  :: NQSO     ! positions of Ps - flux of snow
INTEGER(KIND=JPIM)  :: NQGO     ! positions of Pg - flux of graupel

```

Each physical parameterizations can be deactivated that changes number of physical fluxes. Position of fluxes in APFT is than changed and for some fluxes position in APFT array must be known for programming reasons. Position of these fluxes are given by following integers:

- for positions of momentum turbulent fluxes

```

INTEGER(KIND=JPIM)  :: NUUTURFT ! third index in APFT for U turb. flux
INTEGER(KIND=JPIM)  :: NVVTURFT ! third index in APFT for V turb. flux

```

- for positions of convective fluxes

```

INTEGER(KIND=JPIM)  :: NQLCONV ! third index in APFT for convective rain
INTEGER(KIND=JPIM)  :: NQNCONV ! third index in APFT for convective snow

```

5.5 Data flow

Due to usage of module YOMPHFT data flow for DDH data in AROME is simplified compared to ARPEGE/ALADIN. Physical fluxes and tendencies are saved in array APFT within subroutine APL_AORME and used in subroutine CPPHDDH. This is shown on Fig. 3.

5.6 Setup

To take in to account for all changes done in AROME to implement DDH some changes in setup subroutines where needed.

5.6.1 Subroutine SU0YOMA

In SU0PHY new control variables LPHCDPI and NPHY are initialized. In SUNDDH following some other new control variables are assigned values (see 5.8.8).

Figure 3: Data flow for DDH in ARPEGE/ALADIN.

5.6.2 Subroutine SU0YOMB

In SUPHMNH called in SUPHY control variable LAROBU_ENABLE is initialized. Subroutines ARO_SUBBUDGET and AROINI_BUDGET are called to initialize budget arrays and saving physical tendencies in budget arrays within NMH subroutines. Also, ARO_INIAPFT is called to do a part of initializations needed for DDH in AROME (see 5.10.3).

5.7 Entering physical fluxes and tendencies in DDH

The most of physical parameterizations in AROME are calculated in MesoNH subroutines. The general layout, in AROME, of implementation of MesoNH subroutine and getting tendencies out of it is given on Fig. 4. Each MesoNH subroutine is called from an interface subroutine which itself is called in APL_AORME. Within MesoNH subroutine a subroutine BUDGET is called to save physical tendencies in so called budget arrays. These arrays can't be reached from APL_AORME but to reach them an interface subroutine, called AROEND_BUDGET, is called at the end of APL_AORME. It puts physical fluxes or tendencies in array APFT, which is passed to it as an actual argument.

Figure 4: Getting physical tendencies from MesomN subroutine and saving in APFT.

5.8 Modified subroutines

5.8.1 Subroutine APL_AROME

Before any calculations of physical parameterizations in APL_AROME, subroutine ARO_STARTBU is called to prepare budget arrays for saving tendencies (in MNH tendency is calculated for each physical process).

To put two radiation fluxes in APFT this to lines are added at the end of radiation parameterizations:

```
APFT(JLON, JLEV, MJJ2(4)-1) = ZFRSO(JLON, JLEV, 1)
APFT(JLON, JLEV, MJJ2(4))   = ZFRTH(JLON, JLEV, 1)
```

Convection parameterization fluxes are not saved in budget arrays within MNH subroutines but this is done in subroutine *ARO_CONVBU* added at the end of convection calculations. *ARO...*

At the end of all physical parameterizations calculations fluxes/tendencies are saved in the array APFT by call to subroutine AROEND_BUDGET.

5.8.2 Subroutine AROEND_BUDGET

AROEND_BUDGET is an interface subroutine between ARPAGE/ALADIN code and MNH code. It's purpose is to save physical fluxes/tendencies in array APFT. MNH subroutines save them in budget arrays as tendencies. If LDFLUX

is `.TRUE.`, fluxes are calculated from tendencies and saved in `APFT`, else, tendencies are saved. If `(LDAR OCDPI.OR.LDHDCDPI)` is `.TRUE.`, `CDPI` fluxes and pseudo-fluxes are calculated and saved in `APFT`.

5.8.3 Subroutine `CPDYDDH`

Dynamical fluxes and tendencies were not changed for `AROME` only new variables are added. Difference in number and nature of variables in `ARPEGE/ALADIN` and `AROME` is treated by `SLECT CASE` statement like following:

```
SELECT CASE (NPHY)
  CASE(JPHYARO)
    DO JLEV = 1, KFLEV
      DO JROF=KSTART,KPROF
        - variables for AROME are saved in PDHCV
      ENDDO
    ENDDO
  CASE DEFAULT
    DO JLEV = 1, KFLEV
      DO JROF=KSTART,KPROF
        IF(LHDQLN) THEN
          - variables for ARPEGE/ALADIN are saved in PDHCV
        ENDDO
      ENDDO
    END SELECT
```

5.8.4 Subroutine `CPG_DIA`

A call of subroutine `ARO_CPPHDDH` (see 5.10.1) is added. It is called in case that `AROME` physics is used. A new argument `PGPAR` is added to get surface temperature and humidity (see 5.8.5) for `AROME`.

5.8.5 Subroutine `CPG`

`ZGPAR` is added in call of `CPG_DIA`

5.8.6 `SUOPHY`

Tree new variables used for control are assigned values these are `LPHCDPI`, `NPHY` and `LBUFLUX`.

5.8.7 Subroutine `SUPHMNH`

This subroutine is used in `AROME` for setup of `MNH` physical package. For `DDH` some changes were needed in it. First, `LAROBU_ENABLE` is assigned as:

```
LAROBU_ENABLE=LAROBU_ENABLE.OR.(LSDDH.AND..NOT.LONLYVAR).OR.LPHCDPI
```

Then budget arrays are prepared for usage, if LAROBU_ENABLE is .TRUE..

```
CALL ARO_SUBBUDGET(ILON,ILEV,ZTSTEP)
CALL AROINI_BUDGET(LAROBU_ENABLE,CLUOUT,KULOUT,ZTSTEP,ISV,IRR,&
& CLRAD,CLDCONV,CLTURB,CLTUREDIM, CLCLOUD)
```

Finally variables and arrays needed for DDH in AROME (mainly those from module YOMPHFT) are initialized.

```
CALL ARO_INIAPFT(LAROBU_ENABLE)
```

5.8.8 Subroutine SUNDDH

This subroutine must be changed when new terms are entered in DDH. Namelist NAMDDH is read in this subroutine so we are obliged to give default values to tree new variables.

```
LONLYVAR = .TRUE.
LHDORIGP = .TRUE.
LHDCDPI = .FALSE.
```

Other variables that must be changed are:

```
NHDQLNVA ! total number of variables for water
NHDQLNTD ! number of dynamical tendencies for water when LHDHKS is .TRUE.
NHDQLNFD ! number of dynamical fluxes for water when LHDHKS is .TRUE.
NHDQLNFP ! number of physical fluxes for water when LHDHKS is .TRUE.
NHDQLNTP ! number of physical tendencies for water when LHDHKS is .TRUE.
NDHVHK   ! total number of variables when LHDHKS is .TRUE.
NDHAHKD  ! total number of dynamical tendencies when LHDHKS is .TRUE.
NDHBHKD  ! total number of dynamical fluxes when LHDHKS is .TRUE.
NDHAHKP  ! total number of physical fluxes when LHDHKS is .TRUE.
NDHBHKP  ! total number of physical tendencies when LHDHKS is .TRUE.
```

This is done in SELECT CASE sentences like this one:

```
SELECT CASE (NPHY)
CASE(JPHYARO)
- assignments for AROME
CASE DEFAULT
- assignments for ARPEGE/ALADIN
END SELECT
```

Additionally, NHDQLNFP is changed if LHDCDPI is .TRUE..

5.8.9 Subroutine PPFIDH

New variables are added in the way described in 4.3 but within a SELECT CASE statement like this:

```
SELECT CASE (NPHY)
CASE(JPHYARO)
- AROME variables
CASE DEFAULT
- ARPEGE/ALADIN variables
END SELECT
```

Physical fluxes are saved in file in the following way:

```
SELECT CASE (NPHY)
CASE(JPHYARO)
  IF(.NOT.LONLYVAR) THEN
    DO JI= 1, NDDHFT
      WRITE (CLNOMA,5000) KNUM,YAPFT(JI)%CFT,YAPFT(JI)%CVAR,YAPFT(JI)%CNAME
      CALL WRIFDH (ICOREP,NPODDH,CLNOMA,PDDHCV(0,IDHCV+JI),NFLEVG+1)
    ENDDO
  ENDIF
CASE DEFAULT
- ARPEGE/ALADIN fluxes
END SELECT
```

The advantage of DO loop is that for AROME this subroutine doesn't have to be changed when new terms are introduced in equations.

5.9 Modified modules

5.9.1 Module YOMARPHY

In ARPEGE/ALADIN is supposed that the outputs of physical parameterizations are fluxes, in MNH these are always tendencies. To give a choice of either fluxes or tendencies a new switch is introduced. See for details in 3.3.

```
LOGICAL :: LBUFLUX
!   LBUFLUX : If TRUE fluxes are calculated in AROEND_BUDGET,
!             if FALSE, tendencies remain
```

5.9.2 Module YOMLDDH

Three switches are added to control the content of output file. LONLYVAR gives a possibility to save either only variables or variables and all other terms. Other two are connected to possibility of having microphysical fluxes/tendencies in

form needed for CDPI (see 3.2). It is possible to save microphysical fluxes/tendencies as they are calculated in physical package, as in CDPI or both (see 5.9.4).

```

LOGICAL :: LONLYVAR ! if .TRUE. only variables are saved
LOGICAL :: LHDORIGP ! if .TRUE. original fluxes from microphysical parameterizations
                    ! are saved
LOGICAL :: LHDCDPI ! if .TRUE. CDPI fluxes are saved

```

5.9.3 Module YOMMNH

In this module one logical variable, three arrays and one parameter are added for control of saving in data budget arrays and retrieving them.

```

LOGICAL :: LAROBU_ENABLE ! when it is .TURE. tendencies are saved in budget arrays

```

(see 3.4)

```

! for budgets and DDH, number of processes in budget arrays
INTEGER(KIND=JPIM),ALLOCATABLE :: NBUPROC(:)
INTEGER(KIND=JPIM),ALLOCATABLE :: NJBUDG1(:)
INTEGER(KIND=JPIM),ALLOCATABLE :: NJBUDG2(:)
! number of budgets used in AROME, without passive scalars
INTEGER(KIND=JPIM), PARAMETER :: JPAROBUD = 12

```

5.9.4 Module YOMPHY

Two new control variables and tree parameters are added in YOMPHY. LPHCDPI controls calculation of CDPI fluxes/tendencies (but not output, see). In various parts of the code there was a need to branch because of possibility to use several physical packages. The best way to do it is by SLECT CASE statement. For this reason a new variable NPHY is introduced. It is a code of physical package used in the model and it can take values defined with tree new parameters each for one physical package of those used in ARPEGE/ALADIN/AROME. When new package is introduced in the model, a new parameter must be defined.

```

LOGICAL :: LPHCDPI
INTEGER(KIND=JPIM)  :: NPHY
! Values that NPHY can obtain:
INTEGER(KIND=JPIM), PARAMETER  :: JPHYEC = 1 ! for ECMWF physics
INTEGER(KIND=JPIM), PARAMETER  :: JPHYMF = 2 ! for MF physics
INTEGER(KIND=JPIM), PARAMETER  :: JPHYARO = 3 ! for AROME physics

```

5.10 New subroutines

5.10.1 Subroutine ARO_CPPHDDH

ARO_CPPHDDH is used in AROME instead of CPPHDDH used in ARPEGE/ALADIN. It has the same purpose as CPPHDDH, to calculate and save physical fluxes and tendencies in PDHCV. For the time being only fluxes are saved, and this is done in DO loop:

```
DO JPHFT= 1, NDDHFT
  DO JLEV = 0, KFLEV
    DO JROF=KSTART,KPROF
      PDHCV(JROF, JLEV, JPHFT+IDHCV) = APFT(JROF, JLEV, JPHFT)
    ENDDO
  ENDDO
ENDDO
```

5.10.2 Subroutine ADDET

This subroutine adds a description of a new term in array YAPFT. It is used in ARO_INIPFT.

5.10.3 Subroutine ARO_INIPFT

In this subroutine memory is allocated for all allocatable arrays from module YOMAPFT (5.4) and values are assigned to all variables and arrays from YOMAPFT. Descriptions in YAPFT are sorted by variable. The order of variables is following:

1. U: x-component of horizontal velocity
2. V: y-component of horizontal velocity
3. W: vertical velocity
4. EP: turbulence kinetic energy
5. θ : potential temperature
6. q_v : specific ratio of water vapour
7. q_l : specific ratio of cloud water
8. q_i : specific ratio of cloud ice
9. q_r : specific ratio of rain
10. q_s : specific ratio of snow
11. q_g : specific ratio of graupel

Physical fluxes/tendencies for one variable are in the same order as in budget arrays that means in the order of execution of processes. Process in MesoNH means anything between: physical process, parameterisation, dynamical terms and all other terms in equations even some mathematical procedures like filtering.

Following main tasks are done in ARO_INIPFT:

1. Calculating a number of fluxes/tendencies and allocations
2. Descriptions of fluxes/tendencies are saved in array YAPFT. At the same time first and last index in YAPFT for each variable are saved in arrays MJJ1 and MJJ2.
3. Matching APFT with budget arrays; the first and the last index in budget arrays are saved in NJBUDG1 and NJBUDG2.

If common interface is used positions of CPDI fluxes are determined.

5.11 Adding a new term in DDH for AROME

If we want to add a new variable, dynamical flux or tendency and physical tendency (as it is defined in DDH) the procedure is the same as in ARPEGE/ALADIN (4.3). The procedure for physical fluxes is changed because there are much more changes in them when physical package is changed than in other categories. Intention is to do changes only in ARO_INIPFT but but some changes must be still done in SUNDDH. The procedure is following:

- A. Adding a variable, dynamical flux or tendency or physical tendency: the same procedure as in 4.3
- B. Adding a physical flux
 1. Do changes in subroutine SUNDDH (see 5.8.8),
 2. Add a new term in subroutine ARO_INIPFT. First NAPHFT and NDDHFT must be changed. Than, description of the new term must be put at the right place in YAPFT (see 5.10.3). Besides, changes in NBUPROC, NJBUDG1 and NJBUDG2 may be needed.

References

- [1] Catry, B., J.-F. Geleyn, M. Tudor, P. Benard, A. Trojakova, 2006: Flux/conservative thermodynamic equations in a mass-weighted framework. *Tellus*, 1-2.
- [2] Piriou, J.-M., 2002: Diagnostics en Domaines Horizontaux. (ALADIN documentation on web, http://www.cnrm.meteo.fr/gmapdoc/article.php3?id_article=19).

- [3] Piriou, J.-M., 2002: The use of DDH in ARPEGE - ALADIN. (ALADIN documentation on web, http://www.cnrm.meteo.fr/gmapdoc/article.php3?id_article = 19).