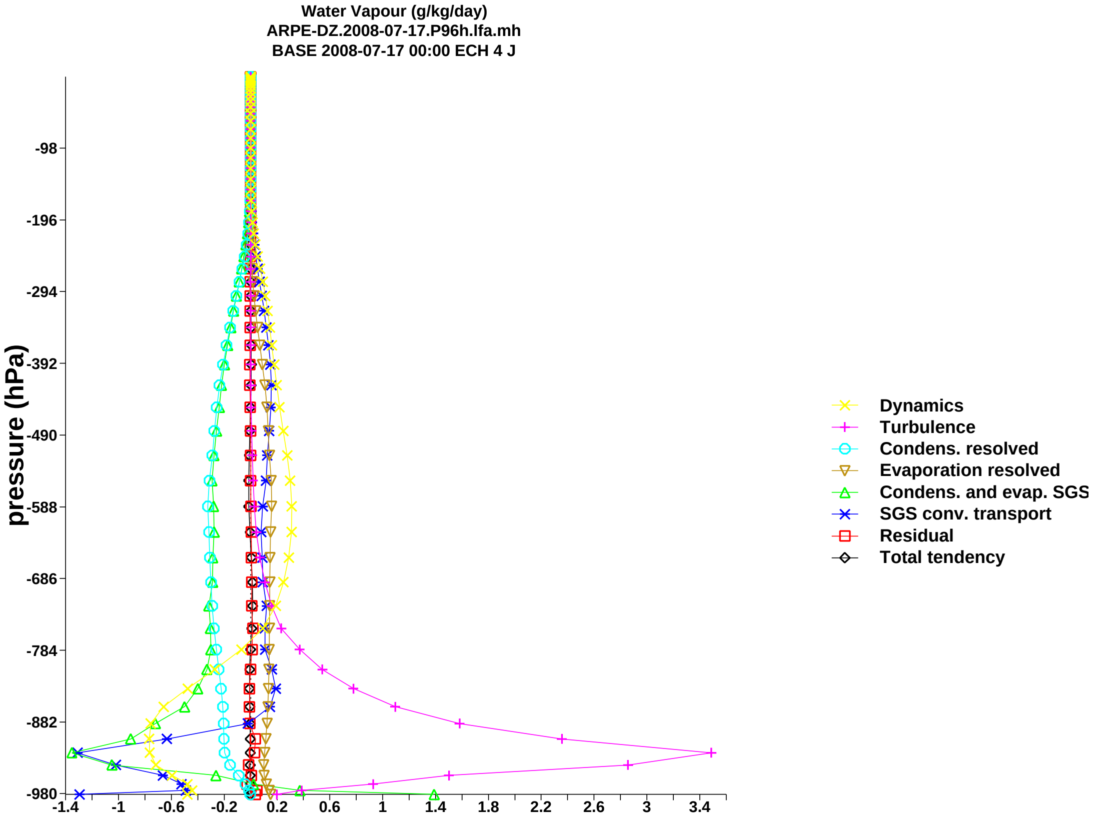
Diagnostics in Horizontal Domains (DDH)

Variables and budget equations, in horizontal mean

Arpege, Aladin and Arome models

Guide for users and developpers

July 24, 2008



Contents

Chapter 1

Main principles

This paper presents the diagnostics on horizontal domains (DDH) developed initally for the variable mesh of ARPEGE(global stretched model), and now also available for ALADIN (Limited Area Model) and AROME (Cloud Systems Resolving Model).

The main objective of the DDH tool is to provide, on user defined domains, the budget of prognostic variables of these models (momentum, temperature, water vapour, etc). The DDH tool is used by searchers and model developpers to understand the model's dynamical and physical interactions, thus contributing to the parameterization development process. The DDH are also used for other purposes, like getting mean model drifts with respect to analyses, or to extract model vertical profiles at given locations.

These diagnostics are made, on the one side, of logical functions which enable to manage several user defined horizontal domains depicting parts or the whole of the domain of integration, while reducing the number of scientific computations. On the other hand, they include the production of budget equations of the prognostic variables for domains such as zonal bands, rectangular areas, single vertical model columns, or the whole globe.

Each model point is described within the DDH sofware by its geographical position, a scale factor, the orientation of the geographical North vector, a mean value of each variable as well as some horizontal derivatives.

Each point is independent and can belong to different user-defined domains; any mean on several points makes a domain. The gathering of all points makes the global domain. A zonal mean is made on a grouping of points between two given latitudes, etc.

DDH therefore makes a double representation of domains: a first one, external, which operates groupings meaningful for users; that is a zonal band, or points within a given geographical area. The other one, internal, consists in the grouping of any other points, not even necessarily connex ones.

Information within a given domain is also categorized by DDH: for some parameters, an instantaneous value is required, for others, only a mean value in time is required. This implies the use of operators of horizontal means and of simple and linear mean values in time, commutative as often as possible.

1.1 User-defined horizontal domains

- whole globe,

- zonal bands of equal surfaces,
- limited domains, defined by either two or four corners, overlapping or not,
- isolated points, either inside or outside the above mentioned.

In practice, for the DDH software, limited domains and isolated points are the same kind.

1.2 User-defined outputs

- a print on standard output of budgets on any given domain, in vertical mean.
- the production of a file such as LFA which includes vertical profiles of mean horizontal parameters, eventually cumulated in time, on this ensemble of domains. Thus, the following files can be produced:
  - a file for the whole model domain,
  - a file containing zonal bands,
  - a file of limited domains and of isolated points,

1.3 Scientific content of diagnostics

- Budgets of mass / energy: balance of air and water masses, potential and internal energies, kinetic energy, momentum budget, entropy budget, can be activated from independent logical indicators.
- For surface, the DDH tool provides surface fluxes, but not soil budgets, as these soil budgets are given by the SURFEX software itself.

For more detail about budgets, see section (**??** p. **??**).

1.4 Output frequency

For DDH diagnostics, two independent control chains are available. One for printing, the other one for producing files. They enable to choose a time unit (time step or prediction time) and to generate output either at regular intervals or at specific irregular prediction times.

1.5 Internal representation of domains

The main principle of DDH is to allocate each model point to one and only one internal domain, and this even if this point belongs to several *user* domains. With this principle, scientific computations are made only once, thus saving computation time. Therefore, two ways to split the integration domain are used:

- the external (or user-defined) splitting: a given point can be at the same time, within the global domain, within a zonal band, and within one or more limited domains. Of course, final file or listing outputs are presented in this user-defined geometry, but one should try to make the least possible computations.

- the internal splitting, meaning the partition (in the sense of the ensemble theory) of the integration domain. Therefore, to each point is associated a single internal domain. The associated mask is set in the same way as the geographical positions' one, land/sea indicator, etc. To each user's domain corresponds a single set of internal domains. This set entirely defines the external domain.

1.6 Horizontal mean operator

It must be such that the average of the mean values of each sub ensemble gives the global mean value on the sphere (or on the entire model domain, in the case of a limited area model). Each point is given a weight which represents its geographical surface (but without dimension).

1.7 Computation time of DDH

Scientific computations of budgets are equivalent to a models's dynamics: advections and dynamical vertical fluxes are computed. As many intermediate results as possible are re-used (e.g. parametrized fluxes). The code tries to optimize to its best some aspects such as the vectorization of mean value in time. The code is respectful of the parallel treatment and results are satisfactorily similar, whether got in mono or multi task modes.

However, the success of the application of the principle 《one point, one internal domain》 leads to the fact, when a user calls an output requiring the entire set of points (e.g. from the global domain or from zonal means), complementary domains are almost free of computation time, only the cost of corresponding file or listing outputs comes into consideration.

Having said that, using the DDH code to produce diagnostics at one point only, requires a large computation time, as computation is done on all points anyway.

Chapter 2

Producing DDH files: geometry, user namelist

This chapter presents two aspects of the know-how: first of all what has to be declared in the user namelist to generate budgets and DDH files, and secondly the file structure.

2.1 Different kinds of horizontal domains

To each kind of horizontal domain (global, zonal band, limited area) is associated a logical indicator of activation, two output indicators (files and/or listing), and, for limited area domains, a geometrical identification longitude/latitude of the corners).

Recognized types:

- The global domain is activated if `LHDGLB` is true,
- Zonal bands are activated if `LHDZON` is true,
- Limited area domains or isolated points are activated if `LHDDOP` is true.

2.1.1 Global domain

- `LHDPRG`, true for printing.
- `LHDEFG`, true for producing a file.

2.1.2 Zonal bands

The total number of zonal bands is `NDHKD`. In case of file output, a single file will contain all the bands. In case of print, only one band will be printed; this band should be specified (`NDHZRP`).

To summarize

- `LHDPRZ` true induces the printing of the latitude band number `NDHZPR`,
- `LHDEFZ` true induces the writing of a file,
- `NDHKD` specifies the number of latitude bands.

The principle of it is to divide the real sphere into NDHKD bands of geographical latitudes of equal surfaces. Each band is identified by its index jkd ($1 \leq \mathtt{jkd} \leq \mathtt{NDHKD}$).

Let the zonal band between geographical latitudes $\theta_{g,jkd}$ and $\theta_{g,jkd+1}$, and notating

$$\mu_{jkd} = \sin\theta_{g,jkd},$$

the equality of the surfaces leads to create the following suite of the Northern boundary of the bands:

$$\mathtt{NDHKD} \times 2\pi a^2 \left(\mu_{jkd} - \mu_{jkd+1}\right) = 4\pi a^2,$$

with $\mu_1 = 1$. Therefore

$$\boxed{\mu_{jkd} = 1 - (jkd - 1)\,\frac{2}{\mathtt{NDHKD}}}$$

The latitude of the zonal band jkd is

$$\bar{\mu}_{jkd} = 1 - \left(jkd - \frac{1}{2}\right)\frac{2}{\mathtt{NDHKD}}$$

2.1.3 Limited areas

Several type of limited domains may be defined

- Type 1: an isolated point defined by its indexes (jlon, jgl),
- Type 2: a domain defined by its four corners identified by (geographical longitude, geographical latitude),
- Type 3: a domain defined by two opposite corners, identified by (geographical longitude, geographical latitude),
- Type 4: an isolated point defined by its geographical position.

Please note that, for the moment and for simplicity sake, affectation computations are based on computation of straight lines in space (λ, μ).

The management of either total or partial overlapping between domains calls for the notion of virtual plane. As a matter of fact, to each declared domain is associated a virtual plane. Thus, diagnostics DDH will know either

- that possible overlapping should be ignored and that diagnostics should refer to domains as they have been declared. To do so, domains must be located inside separate virtual planes,
- Or that overlapping is a way to modify the geometry of a domain already declared, therefore, diagnostics will refer to modified domains. To do so, domains must be affected to the same virtual plane, furthermore, the order of declaration is important. This will enable to create, in principle, any shapes of domains, from those elementary types.

Let us see a few examples.

Case 1.

All domains are disjoined (figure 2.1). The notion of virtual plane is needless. In practice, put all the domains in the same plane.

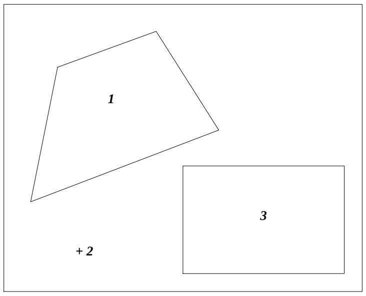


Figure 2.1: Case 1

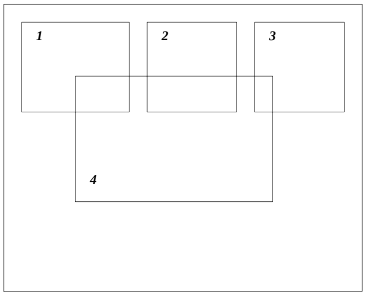


Figure 2.2: Case 2

Case 2.

Domain 4 overlaps domains 1, 2 and even 3 (figure 2.2). Complete results in each of these domains are required. In practice, allocate domains to distinct planes (figure 2.3).

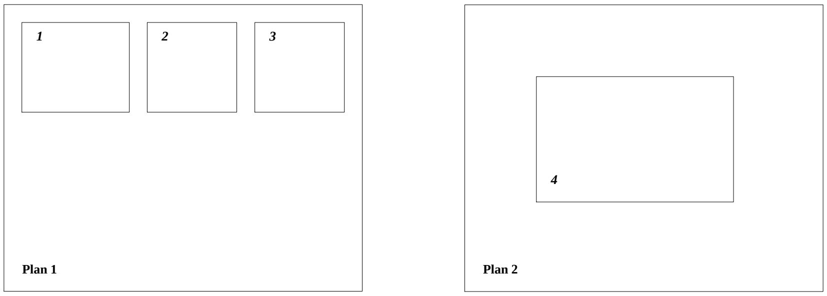


Figure 2.3: Case 2

Case 3.

Domains are embedded to up to three levels and complete results in each of them are required (figure 2.4). Allocate domains to distinct planes.

Case 4.

A domain is not a quadrilateral, another has the shape of a ring (figure 2.5). To get diagnostics in these domains, do the declarations in the same virtual plane, according to the numbers of figure 2.6.

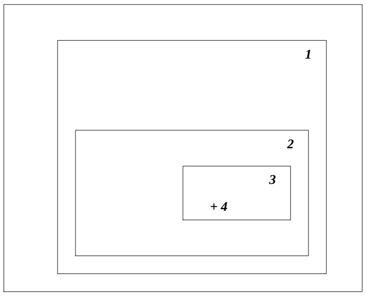


Figure 2.4: Case 3

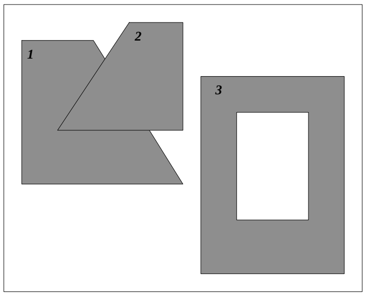


Figure 2.5: Case 4

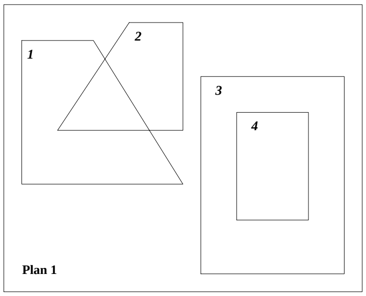


Figure 2.6: Case 4

Some remarks about declaring domains. First of all, there is no algorithmic limit to the number of overlapping levels. For practical reasons, the limit is given by `PARAMETER JPDXHPU`, which is (quite) easy to modify.

There is no need to execute a formal declaration of a virtual plane (hence its name). In practice and from the point of view of users, it is only a coordinate of a domain. Allocating a domain to the plane N creates the plane N. However, memory and time costs can be lessened if affectations are declared to successive planes (1, 2, 3, $\ldots$ and not 1, 3, 5, $\ldots$ which would leave planes 2 and 4 empty).

Case 4 shows how declarations are managed within a same plane. To each virtual plane is associated a mask of points distribution. The index of a point is the number of the domain in the plane, 0 if the point does not belong to any domain. The declaration of a domain of the kind 1 to 4 being read and verified, points will be allocated to it, independently of their indexes at that time, 0 or any other value.

Thus, let us go back to case 4. Quadrilateral 1 is declared as a guess for domain 1. Every single points included in this quadrilateral is given index 1. Then, quadrilateral 2 is declared and its points are given index 2, notably, those which previously had value 1. The shape of points《1》changes and takes the shape of the final domain 1. Conversely, to get ring 3, it is sufficient to declare its outer outline and to make a hole inside it and then to declare its inner outline. *Within any given virtual plane, a point is allocated solely to the last declared domain* which contains it.

To summarize:

Embedded or overlapping domains: allocate each domain to a distinct virtual plane. Require as many virtual planes as it is necessary.

Strange or punched domains: within the same virtual plane, distort and make holes in the successive sketches of the quadrilateral.

The management system can not guess by itself which of the two approaches you will use. Therefore, should you not pay any attention to the declarations, especially to the virtual coordinate, an error may occur.

Besides, let the absent-minded reader be reminded that, even if a point seems to be relevant for the four domains, diagnostics will be made only once. Virtual planes are given up by the software in favour of a unique distribution plane where each new possible intersection makes a new "internal" computation domain. Thus, case number 1 gives (without the horizontal means) four internal domains, case number 2 (always alone) gives seven internal domains, and so forth.

Let us now present what has to be declared in order to start the DDH diagnostics.

2.2 Declarations. The `NAMDDH` namelist

As always with ARPEGE, setting up options is done by a namelist. For these diagnostics, most items depend on `NAMDDH`. However, the control of output events depends on `NAMCT0` and `NAMCT1`. Some dimensions, presently coded as `PARAMETER` could be managed more flexibly through `NAMDIM`.

2.2.1 Declarations

`NAMDDH` regroups the main parameters controlling the diagnostics. Can be found logical indicators, some numerical parameters and also a table to declare possible limited domains.

2.2.1.1 Type of domains

- `LHDGLB:` global domain (diagnostics are produced if the indicator is true)
- `LHDZON:` zonal bands
- `LHDDOP:` limited domains and isolated points.

2.2.1.2 Variables to budgetise

- `LHDHKS:` budget of atmospheric mass, energy, momentum, relative humidity, soil budget.
- `LHDMCI:` budget of kinetic momentum
- `LHDENT:` budget of entropy

Should no domain be specified, no diagnostics are produced. Should a domain be specified, but no content specified, ARPEGE **is stopped**. The same goes if no output is requested.

2.2.1.3 Output on file or listing

- `LHDEFG:` write zonal bands diagnostics on file,
- `LHDEFZ:` write zonal bands diagnostics on listing,
- `LHDEFD:` write limited domains diagnostics on file,
- `LHDPRG:` write global diagnostics on listing,
- `LHDPRZ:` write zonal bands (a single band will be written) diagnostics on listing,
- `NDHZPR:` index of the latitude band whose budget will be printed (if `LHDPRZ` is true),
- `LHDPRD:` write limited domains diagnostics on listing,
- `LHDFIL:` the list of articles written in each DDH output file, will be written on listing.

2.2.1.4 Software maintenance, debugging mode

- `LHDLIST:` printing on listing in verbose mode,
- `LHDVRF:` verification mode, activating the budget computation in one point; the output is written on listing.
- `NVDHLO:` Index `JLON` of the verification point,
- `NVDHGL:` Index `JGL` of the verification point.

Futhermore, this point must be declared as a domain. For more details about this option, refer to pages **??** and following.

2.2.1.5 Control results reproductibility

- `LHDREP:` true if one wishes the results of the diagnostics to be reproducible bit to bit from a multitask run to the next one. This option, useful for data processing validation, is useless for scientific interpretation. Difference in «non reproducible» mode (`LHDREP` false) are not significant: to this day, no difference whatsoever, up to 10^{-10} in the relative way, has been noticed! The advantage of the recommended option `LHDREP = .FALSE.` is to make substantial savings in the occupation of the memory.

2.2.1.6 Number of zonal bands

- `NDHKD:` Number of zonal bands.

2.2.2 Declaration of limited domains

The declaration is made by filling in a double entry table `BDEDDH(10,JPDHNOX)`. `JPDHNOX` is a `PARAMETER` which gives the maximum number of possible limited domains. It goes together with `JPDHXPU`, maximum number of virtual planes. For each domain, its type, its virtual plane and indications dependant on the type are given. Therefore

`BDEDDH`(1, domain number) $=$ type

`BDEDDH`(2, domain number) $=$ virtual plane

For type 1, point given by its indexes

`BDEDDH`(3, domain number) $=$ rjlon

`BDEDDH`(4, domain number) $=$ rjgl

For type 2, quadrilateral given by its four corners

`BDEDDH`(3, domain number) $=$ Longitude of corner #1, in degrees, λ_1

`BDEDDH`(4, domain number) $=$ Latitude of corner #1, in degrees, θ_1

(`BDEDDH`(5, -), `BDEDDH`(6, -)) $= (\lambda_2, \theta_2)$

(`BDEDDH`(7, -), `BDEDDH`(8, -)) $= (\lambda_3, \theta_3)$

(`BDEDDH`(9, -), `BDEDDH`(10, -)) $= (\lambda_4, \theta_4)$

In order to specify a domain, one must comply with the following constraints:

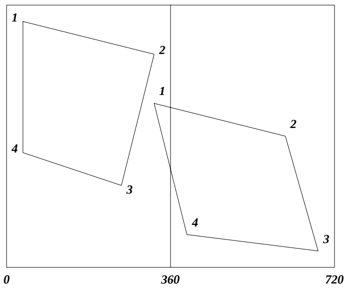


Figure 2.7: Constraints for quadrilateral declarations

- For a domain which does not intersect the Greenwich meridian

$$-1 \leq \mu_i = \sin\theta_i \leq 1\,, \quad 0 \leq \lambda_i \leq 360^\circ$$

and

$$\lambda_2 > \lambda_1\,, \quad \mu_3 < \mu_2\,, \quad \lambda_3 > \lambda_4\,, \quad \mu_1 > \mu_4$$

- For a domain which intersects the Greenwich meridian, the same order in the declaration of corners applies, but furthermore, we have

$$(\lambda_1 \leq 360^\circ \text{ ou } \lambda_4 \leq 360^\circ) \quad \text{and} \quad (\quad \lambda_2 > 360^\circ \text{ ou } \lambda_3 > 360^\circ)$$

For type 3, rectangular domain given by two opposite corners

BDEDDH(3, domain number)	$=$	Longitude of corner #1, in degrees, λ_1
BDEDDH(4, domain number)	$=$	Latitude of corner #1, in degrees, θ_1
(BDEDDH(5, -), BDEDDH(6, -))	$=$	(λ_3, θ_3)

This means that, with the same constraints as above, one declares only corners 1 and 3. Implicitly

$$(\lambda_2 = \lambda_3, \mu_2 = \mu_1) \quad \text{et} \quad (\lambda_4 = \lambda_1, \mu_4 = \mu_3)$$

For type 4, points given by their geographical position

BDEDDH(3, domain number)	$=$	longitude λ_g in degrees
BDEDDH(4, domain number)	$=$	latitude θ_g in degrees

In that case, diagnostics will be made on the closest grid point, using the spatial Euclidian metric in the (λ, μ) space.

Allocation of points to the domains

The point whose geographical coordinate is (λ_g, μ_g) is allocated to a type 2 or 3 domain in the following conditions.

Domain which does not intersect the Greenwich meridian ($\lambda_2 \leq 2\pi$ et $\lambda_3 \leq 2\pi$)

$$\mu_g \leq \frac{\mu_2 - \mu_1}{\lambda_2 - \lambda_1}\lambda_g + \frac{\mu_1\lambda_2 - \mu_2\lambda_1}{\lambda_2 - \lambda_1}$$

$$\lambda_g \leq \frac{\lambda_3 - \lambda_2}{\mu_3 - \mu_2}\mu_g + \frac{\mu_3\lambda_2 - \mu_2\lambda_3}{\mu_3 - \mu_2}$$

$$\mu_g \geq \frac{\mu_4 - \mu_3}{\lambda_4 - \lambda_3}\lambda_g + \frac{\mu_3\lambda_4 - \mu_4\lambda_3}{\lambda_4 - \lambda_3}$$

$$\lambda_g \geq \frac{\lambda_1 - \lambda_4}{\mu_1 - \mu_4}\mu_g + \frac{\mu_1\lambda_4 - \mu_4\lambda_1}{\mu_1 - \mu_4}$$

Domain overlapping $\lambda = 0$ The same tests must be made on ($\lambda_g + 2\pi$, μ_g) for any point so that $\lambda_g \leq \max(\lambda_3 - 2\pi, \lambda_2 - 2\pi)$.

2.2.3 Default values

By default, all logical indicators are `FALSE` and dimensions are set to zero. By default, no DDH is done.

Here is an example of a namelist which activates diagnostics in all types of domains. 6 zonal bands are requested. 6 limited domains are declared in 3 virtual planes. Domain 2 is intersecting the Greenwich meridian.

```
NAMDDH
LHDGLB = .TRUE.,  LHDZON = .TRUE.,  LHDDOP = .TRUE.,  LHDHKS = .TRUE.,
LHDMCI = .FALSE., LHDENT = .FALSE., LHDPRG = .TRUE.,  LHDPRD = .TRUE.,
LHDPRZ = .  TRUE. NDHZPR = 3,
LHDEFG = .TRUE.,  LHDEFZ = .TRUE.,  LHDEFD = .TRUE.,  LHDLIST = .TRUE.,
NDHKD = 6,        LHDVRF = .TRUE.,  NVDHLO = 5,       NVDHGL = 29,
BDEDDH(1,1) = 2., BDEDDH(2,1) = 1., BDEDDH(3,1) = 250.,BDEDDH(4,1) = 45.,
BDEDDH(5,1) = 440.,BDEDDH(6,1) = 80.,BDEDDH(7,1) = 360.,BDEDDH(8,1) = 15.,
BDEDDH(9,1) = 215.,BDEDDH(10,1) = 35.,
BDEDDH(1,2) = 2., BDEDDH(2,2) = 2., BDEDDH(3,2) = 125.,BDEDDH(4,2) = 20.,
BDEDDH(5,2) = 350.,BDEDDH(6,2) = 75.,BDEDDH(7,2) = 360.,BDEDDH(8,2) = -5.,
BDEDDH(9,2) = 85.,BDEDDH(10,2) = -20.,
BDEDDH(1,3) = 3., BDEDDH(2,3) = 1., BDEDDH(3,3) = 30.,BDEDDH(4,3) = 10.,
BDEDDH(5,3) = 245.,BDEDDH(6,3) = -30.,
BDEDDH(1,4) = 3., BDEDDH(2,4) = 2., BDEDDH(3,4) = 0., BDEDDH(4,4) = 90.,
BDEDDH(5,4) = 359.9,BDEDDH(6,4) = 85.,
BDEDDH(1,5) = 1., BDEDDH(2,5) = 1., BDEDDH(3,5) = 10.,BDEDDH(4,5) = 1.,
BDEDDH(1,6) = 4., BDEDDH(2,6) = 2., BDEDDH(3,6) = 20.,BDEDDH(4,6) = -70.,
```

2.3 Output occurrence control

As for other ARPEGE output, regular output frequency may be given in time step. One may also fill in a table which will enable to make irregular outputs.
The control tables are `NxTS(0:JPNPST)` with `x=DHFG` for file outputs of the global domain, `x=DHFZ` for file outputs of zonal band domains, `x=DHFD` for file outputs of limited area domains, and `x=DHP` for printed outputs.
Time units are `NFRx`.
Tables and units are initialized through `NAMCT0`.

Three kinds of outputs are possible

1. If `NFRx`$= n$ $(n > 0)$ and `NxTS(0)=`0: output every n time steps.

2. If `NFRx`$= n$ $(n > 0)$, `NxTS(`0`)=`m $(m > 0)$ and `NxTS(`i`)=`p_i $(p_i \geq 0$, $i \in \{1, \dots, m\})$: output at time steps np_i. In this case it will be preferrable to set `LINC=.FALSE.` in namelist `NAMOPH`, in order to force the date units in output file names to be in time steps (rather than in hours).

3. If `NFRx`$= n$ $(n > 0)$, `NxTS(`0`)=`$-m$ $(m > 0)$ and `NxTS(`i`)=`$-p_i$ $(p_i \geq 0$, $i \in \{1, \dots, m\})$: output at hours np_i.

As a matter of fact, these outputs are only possible at these time steps. They are actually produced if, in addition, `N1x = 1`.

These parameters belong to `MODULE/YOMCT1/`, initialized in `SU1YOM` with the namelist `NAMCT1`. They are set to zero, if either diagnostics `DDH` are not activated or if no file or no listing are requested. This cancels the corresponding output, whatever the content of `NAMCT0` may be.

2.4 Identification of domains in the code and in the outputs

The user-type identification of domains (`BDEDDH`) is transformed in a simpler form, for use by the internal part of the DDH software. Here are indicated the identification conventions which are used internally by the DDH. To each domain is associated a descriptor of 11 words.

- Words 1 and 2 are the coordinates of the domain : virtual plane and number in the plane,
- Words 3 to 10 are mostly geographical information type dependant,
- Word 11 is the kind of domain.

DOMAIN IDENTIFIER										
1	2	3	4	5	6	7	8	9	10	11
Virtual plane	number in this plane	Type-dependent information								Type
plan	number	λ_g	μ_g	r_{jlon}	r_{jgl}					Type 1. Point
plan	number	λ_g	μ_g	r_{jlon}	r_{jgl}	λ_g user value	μ_g user value			Type 4. Point
plan	number	λ_1	μ_1	λ_2	μ_2	λ_3	μ_3	λ_4	μ_4	Type 2. Quadrilateral
plan	number	λ_1	μ_2	λ_2	μ_2=μ_1	λ_3=λ_2	μ_3	λ_4=λ_1	μ_4=μ_3	Type 3. Rectangle
0	0									Type 5. Globe
0	jkd	ndhkd	$\bar{\mu}_{jkd}$							Type 6. Zonal band

Non allocated values are initialized to zero. Longitudes are in radian. For points and limited domains, this information is kept in table `FNODDH(11,JPDHNOX)` of the MODULE `/YOM1DDH/`. For the globe and zonal bands, the information is really useful only during the output.

Moreover, the properties of he domains are the ones declared by the user (except for 4): a way to show that a domain has then been deformed or punched is still lacking.

2.5 Logical structure of output files

For a given date, files contain a suite of domains. For each of them, a suite of profiles and soil variables can be found. Three files can be produced: global, zonal, limited area. These files are physically written with the `LFA` software (Jean-Marcel Piriou), if LHDLFA is true, and in pseudo-GRIB format, if LHDLFA is false.

2.5.1 File names

Global	`DHFGLeeee+nnnn`
Zonal bands	`DHFZOeeee+nnnn`
Limited area domains	`DHFDLeeee+nnn`

`eeee`: the first 4 letters of the name of the experiment,

`nnnn`: output date in hour or time step, according to the logical indicator `LINC` from namelist `NAMOPH`.

2.5.2 Articles giving information about dimensions and dates

Article 1.

The first physical article `'INDICE EXPERIENCE'` contains the name of the experiment.

Article 2.

Article `'DATE'` (11 mots).

1 . Year,

2 . Month,

3 . Day,

4 . Hour,

5 . Minute, date of integration start.

6 . 1 if forecast range is in hours, 2 if forecast range is in days,

7 . Forecast range,

8 . 0,

9 . 10, except maybe at the beginning,

10 . Number of cumulated values,

11 . 0.

Article 3.

Article `'DOCFICHIER'` (17 words).

1. File type:

   1 limited area domains,

   5 global domain,

   6 zonal bands.

2. 0 if `LHDHKS` is false, 1 if true,
3. 0 if `LHDMCI` is false, 1 if true,
4. 0 if `LHDENT` is false, 1 if true,
5. `NSTEP`, current time step value,
6. `NFLEV`, number of levels. Length of variable profiles or variable tendencies. The length of the flux profiles is `NFLEV+1`,
7. `NDHCV`, total number of vertical profiles for each domain,
8. `NDHCS`, total number of soil fields,
9. `NDHVV`, number of variable profiles at a given time. The file contains 2 instantaneous variables: the initial one and that of current time step.
10. `NDHFVD`, number of《dynamical》fluxes or tendencies in vertical profiles,
11. `NDHFVP`, number of《physical》fluxes or tendencies in vertical profiles,
12. `NDHVS`, number of instantaneous soil variables,
13. `NDHFSD`, number of soil《dynamical》fluxes,
14. `NDHFSP`, number of soil《physical》fluxes,
15. number of domains in the file:

    1 for the globe,

    `NDHKD` for zonal bands,

    `NDHNOM` for limited area domains.

16. number of "free" soil variables: these variables are used at ECMWF for diagnostics such as $10\,m$ winds, roughness, etc.
17. number of "free" soil fluxes.

Article 4.

Article `'ECHEANCE'` forecast range in seconds (1 word).

2.5.3 Articles giving information about the type of domains

For each domain, there is an identification article 'DOCDnnnn', where `nnn` is the name of the domain. This article is made of 11 words whose content has been described page 21.

2.5.4 Articles giving information about scientific fields

The last part of this documentation will be about the definition of each field in each option as well as the name of this field. Here, we only will indicate how the name of articles are constituted.

The name of articles takes the form `nnntvvssssssssss`, with

`nnn:` number of the domain in the file. `nnn` varies from 1 to `DOCFICHIER(17)`.

`t:` type of field contained in the article:

- `V:` variable profile, length `NFLEV`,
- `T:` tendency profile, length `NFLEV`,
- `F:` flux profile, length `NFLEV+1`,
- `S:` soil data, length: cf. page 42.

`vv:` physical variable written in this file article:

- `PP:` pressure,
- `QV:` specific water vapour content ,
- `UU:` zonal momentum,
- `VV:` merional momentum,
- `KK:` kinetic energy,
- `CT:` enthalpy,
- `EN:` entropy,
- `M1:` angular momentum,
- `EP:` potential energy ($\Phi = g\,z$).

The next 10 characters (suffix) make the field specific name. However, some general rules do also apply: for variables given as profiles (whose name is therefore `VV`, the instant must be indicated

`ssssssssss = 0`	variable at initial time step,
`ssssssssss = 1`	variable at current time step.

Some suffixes crop up quite frequently

`ssssssssss = DIVFLUHOR`	for terms of the kind $\mathrm{div}_\eta\left(\chi\vec{v}\,\frac{\partial p}{\partial\eta}\right)$
`ssssssssss = FLUVERTDYN`	for terms $\chi\dot{\eta}\,\frac{\partial p}{\partial\eta}$,
`ssssssssss = FLUDUAPLUI`	for terms $\delta_m F_p \chi$.

Chapter 3

Budget equations and horizontal mean

This chapter is about budget equations and discretization, in space and time.

3.1 Generic budget equation

Let χ be a variable of the model whose budget is required. The generic form of the χ budget may be written

$$\boxed{\frac{\partial}{\partial t}\left(\chi\,\frac{\partial p}{\partial \eta}\right) = \; -\underbrace{\operatorname{div}_\eta\left(\chi\vec{v}\,\frac{\partial p}{\partial \eta}\right)}_{(1)} - \underbrace{\frac{\partial}{\partial \eta}\left(\chi\dot{\eta}\,\frac{\partial p}{\partial \eta}\right)}_{(2)} + \underbrace{S_d\,\frac{\partial p}{\partial \eta}}_{(3)} - \underbrace{g\,\frac{\partial F_\varphi}{\partial \eta}}_{(4)} - \underbrace{gS_\varphi\,\frac{\partial G_\varphi}{\partial \eta}}_{(5)}}$$

To estimate the budgets, it has been decided to systematically work on the extensive scales $\chi\,\dfrac{\partial p}{\partial \eta}$, that is to say, $\chi_\ell \delta p_\ell$ for the discrete value in the layer ℓ. In practice, the application of the vertical discretization leads this equation to

$$\boxed{\frac{\partial}{\partial t}\,(\chi\,\delta p) = \; -\operatorname{div}_\eta\,(\chi\vec{v}\,\delta p) - \delta\left(\chi\dot{\eta}\,\frac{\partial p}{\partial \eta}\right) + S_d\delta p - g\,\delta F_\varphi - gS_\varphi\,\delta G_\varphi}$$

where every term is indexed by ℓ, index of the layer of the model for which this equation means something/makes sense/. The operator $\delta\xi_\ell$ is

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

where $\xi_{\tilde{\ell}}$ takes the value of ξ at the interlayer $\tilde{\ell}$.

3.1.1 Term 1. Divergence of horizontal fluxes at the boundaries of the domain

In order to be computed, this term needs to know the χ gradient. For the initial conditions this will not always be the case. This term will not be complete, especially every time when χ depends on the momentum (momentum itself, kinetic energy, angular momentum, etc). Whenever possible,

$$-\,\chi\left(\delta p\,\mathrm{div}\,\vec{v}+\delta B\vec{v}\,\vec{\nabla}\pi\right)-\delta p\,\vec{v}\,\vec{\nabla}\chi$$

is computed. The first term can always be computed. Term 1 must be null in global mean. For a band of latitude, it gives the value of the divergence of the meridian flux χ.

3.1.2 Term 2. Divergence of the adiabatic vertical flux

This term will be treated as a flux: the horizontal mean of the quantity will be kept

$$\left(\chi\dot{\eta}\,\frac{\partial p}{\partial\eta}\right)_{\tilde{\ell}}$$

As in the discretization of vertical advection terms,

$$\boxed{\left(\chi\dot{\eta}\,\frac{\partial p}{\partial\eta}\right)_{\tilde{\ell}}=\frac{1}{2}\left(\chi_{\ell}+\chi_{\ell+1}\right)\left(\dot{\eta}\,\frac{\partial p}{\partial\eta}\right)_{\tilde{\ell}}\qquad\text{pour}\quad\tilde{\ell}=0,\cdots,NFLEV}$$

will be computed.
The vertical speed $\dot{\eta}\,\partial p/\partial\eta$ is computed by `GPCTY` and modified by the lower boundary conditions.

3.1.3 Term 3. Adiabatic source term

Some terms of this kind can be deduced from the dynamical code. For example, the potential and the internal budget express the term called conversion term

$$\left[S_d\,\frac{\partial p}{\partial\eta}\right]_{c_pT}\;=\;-\,\frac{1}{g}\,\vec{v}\;\left[\vec{\nabla}\Phi+\frac{RT}{p}\,\vec{\nabla}p\right]\,\frac{\partial p}{\partial\eta}$$

Some similar terms are to be found in the kinetic energy budget. They, as term 1, are in the «tendencies» category, expressed at ℓ levels.

3.1.4 Term 4. Physical fluxes divergence term

Physical fluxes F_{φ_ℓ} are horizontally averaged as such. The enthalpy flux due to precipitations is the only tricky one. Formally, the following form is assumed

$$F_{\varphi_{c_pT}}=L\left(\eta,T\right)F_{\varphi_q}^{precip}\left(\eta\right)$$

,

some assuptions need to be introduced, like

$$L\left(T_{\tilde{\ell}}\right)=L\left(\frac{1}{2}\left(T_{\ell}+T_{\ell+1}\right)\right)$$

where L is an effective specific enthalpy of phase change.

3.1.5 Terme 5. Tendency term due to physical parametrizations

Such terms occur in the energy budgets, e.g. the dissipation term

$$\vec{v}\,\frac{\partial F^{tur+conv}_{\varphi_{\vec{v}}}}{\partial \eta}$$

or in the entropy budget

$$\frac{1}{T}\,\frac{\partial F^{ray}_{\varphi_T}}{\partial \eta}$$

.

These terms are computed, using variables at time t. Budget terms gather into three categories:

- variables, from the $\frac{\partial}{\partial t}\chi\delta p$ term,
- tendencies, at model levels (such as $\mathrm{div}\,(\chi\delta p\,\vec{v})$),
- fluxes, at the inter-layers (δF_χ).

One shows below the discretization processs, on the simplified budget equation

$$\boxed{\frac{\partial}{\partial t}\left(\frac{1}{g}\,\chi\delta p\right) = \left(\frac{1}{g}\,\chi\delta p\right)\,\mathrm{tend} - \delta F_\chi}$$

which shows three categories. The goal of DDH diagnostics is to give information on the mean budget on an horizontal domain $\mathcal{D}$ (surface $S_\mathcal{D}$):

$$\frac{1}{S_\mathcal{D}}\,\frac{\partial}{\partial t}\iint_\mathcal{D}\frac{1}{g}\,\chi\delta p\,d\sigma = \frac{1}{S_\mathcal{D}}\iint_\mathcal{D}\left(\frac{1}{g}\,\chi\delta p\right)\,\mathrm{tend}\,d\sigma - \frac{1}{S_\mathcal{D}}\iint_\mathcal{D}\delta F_\chi\,d\sigma$$

Some terms, such as the effect of the horizontal diffusion, cannot be diagnosed by DDH: horizontal diffusion is computed in spectral mode, is not converted into grid-point space, and thus unavailable for DDH.

3.2 Horizontal mean

Let $[\,\chi\,]_G$ be the global mean. We have

$$[\,\chi\,]_G = \frac{1}{4\pi a^2}\int_0^{2\pi}\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}}\chi\,a^2\,\cos\theta_g\,d\theta_g\,d\lambda_g = \frac{1}{4\pi}\int_0^{2\pi}\int_{-1}^{1}\chi\,d\mu_g\,d\lambda_g$$

For a given truncation, a precise quadrature of this relation reads

$$[\,\chi\,]_G = \sum_{k=1}^{K}\frac{1}{J(k)}\sum_{j=1}^{J(k)}\varpi_k\chi_{j,k} \quad \text{with} \quad K \geq \frac{3N+1}{2}$$

where N is the triangular truncation, ϖ_k is Gauss weight and $J(k)$ the number of points on the circle of latitude k.

Thus, k refers to the latitudes of the gaussian grid and j to the longitudes. On the stretched sphere the scale factor is a function of the spherical harmonics:

$$[\chi]_G = \frac{1}{S_G}\sum_{k=1}^{K}\sum_{j=1}^{J(k)} \chi_{j,k}\,\frac{\varpi_k}{J(k)m_{j,k}^2}$$

now with

$$S_G = \sum_{k=1}^{K}\sum_{j=1}^{J(k)} \frac{\varpi_k}{J(k)m_{j,k}^2} \neq 1$$

Therefore, the weight of each point will be assimilated as, for the present case, a non dimensional «area element» $\sigma_{j,k}$

$$\boxed{\sigma_{j,k} = \frac{\varpi_k}{J(k)m_{j,k}^2}}$$

Surface of domain $\mathcal{D}$:

$$S_{\mathcal{D}} = \sum_{(j,k)\in\mathcal{D}} \sigma_{j,k}$$

The horizontal mean of parameter χ on $\mathcal{D}$ is written

$$\boxed{\left[\frac{1}{S_{\mathcal{D}}}\iint_{\mathcal{D}} \chi\, d\sigma\right] = [\chi]_{\mathcal{D}} = \frac{1}{S_{\mathcal{D}}}\sum_{(j,k)\in\mathcal{D}} \chi_{j,k}\sigma_{j,k}}$$

With this definition, a division of the globe in D domains $\mathcal{D}$ is such that

$$[\chi]_G = \frac{1}{S_G}\sum_{d=1}^{D} [\chi]_d\, S_d$$

The user domain $\mathcal{D}$ is divided into one or several internal domains $\mathcal{D}_i$. For a multitask run on P processors, isolated P terms for every internal domain are computed, which means,

$$[\chi]_{\mathcal{D}_i}\, S_{\mathcal{D}_i} = \sum_{p=1}^{P}\Big(\sum_{(j,k)\in\mathcal{D}_i(p)} \chi_{j,k}\sigma_{j,k}\Big)$$

XX

where $\mathcal{D}(p)$ are the points belonging to $\mathcal{D}$ treated in the p task.

If the domain is cut in P parts, identical in mono or multi tasks, the mean being a simple sum (products are made in each under task p), results become easily reproducible. To summarize, the output mean on the domain $\mathcal{D}$ called for the user is

$$[\chi]_{\mathcal{D}} = \frac{1}{S_{\mathcal{D}}}\sum_{\substack{\mathcal{D}_i\\ \cup\mathcal{D}_i=\mathcal{D}}}\left[\sum_{p=1}^{P}\Big(\sum_{(j,k)\in\mathcal{D}_i(p)} \chi_{j,k}\sigma_{j,k}\Big)\right]$$

Where the two most external $\sum$ signs can very easily switch over. The algorithm is thus the following:

- Parallel computations of the necessary quantities $\chi_{j,k}$ on every points, initial computation of $\sigma_{j,k}$ and transit through the physico-dynamical interface.
- ≪condensation≫ of results for a p task in parts of the internal domains it manages. Note that this non vectorisable operation is nevertheless field and level independent: it is therefore along this direction that vectorization will take place.
- Synthesis for output needs only (i. e. from time to time) of partial sums on user's domains.

3.3 Temporal discretization

The typical budget equation

$$\int_0^{nstep\times\delta t} \frac{\partial}{\partial t}\,\frac{1}{g}\,\chi\delta p\,dt = \int_0^{nstep\times\delta t}\left[\left(\frac{1}{g}\,\chi\delta p\right)\ \mathrm{tend} - \delta F_\chi\right] dt$$

is integrated by the diagnostics `DDH` as follow

$$\boxed{\left(\frac{1}{g}\,\chi\delta p\right)^{nstep} - \left(\frac{1}{g}\,\chi\delta p\right)^{0} = \delta t \sum_{jstep=0}^{nstep-1}\left[\left(\frac{1}{g}\,\chi\delta p\right)\ \mathrm{tend} - \delta F_\chi\right]^{jstep}}$$

Where δt stands for `TSTEP`, the nominal time step, and $nstep$ the number of the current time step. Tendencies and fluxes cumulated in time must be stopped at the tendencies and at the fluxes of the time step preceding the output moment `NSTEP`. Tendencies which are computed by `CPG` when the grid-point variable is known must not be added before the output of results. This adds an important constraint to the parallel treatment.

In practice, one uses two arrays: one with initial variables and variables cumulated in time up to `NSTEP-1`, and another with values at `NSTEP` and variables cumulated in time up to `NSTEP` (which leads to the variable d'état at `NSTEP+1`).

In short, the typical budget equation for a layer ℓ and a discretized domain $\mathcal{D}$

$$[\,\xi\,]_{\ \mathcal{D}} = \frac{1}{S_{\mathcal{D}}} \sum_{(j,k)\in\mathcal{D}} \xi_{(j,k)}\sigma_{(j,k)}$$

with

$$S_{\mathcal{D}} = \sum_{(j,k)\in\mathcal{D}} \sigma_{(j,k)} \qquad \sigma_{(j,k)} = \frac{\varpi_k}{J(k)m^2_{j,k}}$$

$$\boxed{\begin{aligned}&\left[\frac{1}{g}\,\chi\,\delta p\right]^{\ell}_{\mathcal{D}}(t = NSTEP\times\delta t) - \left[\frac{1}{g}\,\chi\,\delta p\right]^{\ell}_{\mathcal{D}}(t=0) = \delta t \sum_{n=0}^{nstep-1}\left[\left(\frac{1}{g}\,\chi\,\delta p\right)\ \mathrm{tend}\right]^{\ell}_{\mathcal{D}}(n) + \\ &\qquad\qquad \delta t\Big(\sum_{n=0}^{nstep-1}\ [\,F_\chi\,]^{\tilde{\ell}-1}_{\mathcal{D}}(n) - \sum_{n=0}^{nstep-1}\ [\,F_\chi\,]^{\tilde{\ell}}_{\mathcal{D}}(n)\Big)\end{aligned}}$$

and the vertical mean budget

$$\boxed{\begin{aligned}
&\sum_{\ell=1}^{NFLEV}\left[\frac{1}{g}\,\chi\,\delta p\right]_{\mathcal{D}}^{\ell}(t = NSTEP \times \delta t) - \sum_{\ell=1}^{NFLEV}\left[\frac{1}{g}\,\chi\,\delta p\right]_{\mathcal{D}}^{\ell}(t = 0) = \\
&\delta t\sum_{\ell=1}^{NFLEV}\sum_{n=0}^{nstep-1}\left[\left(\frac{1}{g}\,\chi\,\delta p\right)\ \mathrm{tend}\right]_{\mathcal{D}}^{\ell}(n) + \delta t\Big(\sum_{n=0}^{nstep-1}[\,F_\chi\,]_{\mathcal{D}}^{0}(n) - \sum_{n=0}^{nstep-1}[\,F_\chi\,]_{\mathcal{D}}^{NFLEV}(n)\Big)
\end{aligned}}$$

Chapter 4

Budget and diagnostics

This chapter presents budget equations which are the first application of DDH. It gives information about the content of output files.

4.1 Dry air mass budget

Budget equation

$$\frac{\partial r_\eta}{\partial t} = -\operatorname{div}_\eta (r_\eta \vec{v}) - \frac{\partial}{\partial \eta} (r_\eta \dot{\eta}) + \delta_m \frac{\partial F_p}{\partial \eta}$$

where

- $r_\eta = -\frac{1}{g} \dfrac{\partial p}{\partial \eta}$.
- $\left(\dot{\eta} \dfrac{\partial p}{\partial \eta} \right)_{\eta=0} = 0 \qquad \left(\dot{\eta} \dfrac{\partial p}{\partial \eta} \right)_{\eta=1} = \delta_m g E.$
- $F_p = F_p^{conv-l} + F_p^{conv-n} + F_p^{stra-l} + F_p^{stra-n}$.
- $\delta_m = 0$: masse conserved, $\delta_m = 1$: variable mass.

File output

`VPP0`	$\dfrac{1}{g}\,\delta p$ (t=0)	
`VPP1`	$\dfrac{1}{g}\,\delta p$ (t=NSTEP δt)	
`TPPDIVFLUHOR`	$-\dfrac{\delta t}{g}\,\operatorname{div}_\eta (\vec{v}\,\delta p)$	cumulated
`FPPFLUVERTDYN`	$\dfrac{\delta t}{g}\,\dot{\eta}\,\dfrac{\partial p}{\partial \eta}$	cumulated
`FPPSUMFPL`	$\delta_m \delta t F_p$	cumulated

If χ^* stands for a quantity modified by the mass exchange and if by χ the initial quantity, then

$$\left(\dot{\eta}\,\frac{\partial p}{\partial \eta}\right)^*_{\tilde{\ell}} = \left(\dot{\eta}\,\frac{\partial p}{\partial \eta}\right)_{\tilde{\ell}} + \delta_m g\,\left[B_{\tilde{\ell}}\left(F_{p\tilde{L}} + E\right) - F_{p\tilde{\ell}}\right]$$

One should have at the lower limit

$$F^{tur}_{q\tilde{L}} = E\,(1 - \delta_m q_v)$$

Furthermore

$$\left(\frac{\omega}{p}\right)^*_{\ell} = \left(\frac{\omega}{p}\right)_{\ell} - \delta_m g \frac{1}{\delta p_\ell}\,\left[\alpha_\ell\,\delta F_p + \ln\left(\frac{p_{\tilde{\ell}}}{p_{\tilde{\ell}-1}}\right) F_{p\tilde{\ell}-1}\right]$$

$$\left(\frac{\partial \pi}{\partial t}\right)^* = \frac{\partial \pi}{\partial t} - \delta_m g\left(E + F_{p\tilde{L}}\right)$$

4.2 Water mass budget

Lagrangian equation

$$r_\eta \frac{\partial q_\psi}{\partial t} = T^{diff-hor}_{q_\psi} + \frac{\partial F_{q_\psi}}{\partial \eta} - \delta_m q_\psi \frac{\partial F_p}{\partial \eta}$$

Budget equation

$$\frac{\partial (r_\eta q_\psi)}{\partial t} = -\,\mathrm{div}_\eta\,(r_\eta q_\psi \vec{v}) - \frac{\partial}{\partial \eta}\,(r_\eta q_\psi \dot{\eta}) + \frac{\partial F_{q_\psi}}{\partial \eta}$$

where

- $q_\psi = q_v$, q_l ou q_n.
- $r_\eta = -\frac{1}{g}\dfrac{\partial p}{\partial \eta}$.
- $F_{q_v} = F_c^{conv-l} + F_c^{conv-n} + F_c^{stra-l} + F_c^{stra-n} + F^{tur}_{q_v} + F^{tur-conv}_{q_v}$.
- $F_{q_l} = F_p^{conv-l} + F_p^{stra-l} - F_c^{conv-l} - F_c^{stra-l} + F^{tur}_{q_l} + F^{tur-conv}_{q_l}$.
- $F_{q_n} = F_p^{conv-n} + F_p^{stra-n} - F_c^{conv-n} - F_c^{stra-n} + F^{tur}_{q_n} + F^{tur-conv}_{q_n}$.
- $F^{tur}_{q_\psi}$ contains the correction $F^{q<0}_{q_\psi}$ of the negative water values created eventually by the dynamics.

The tendency term due to the horizontal diffusion is missing from the budget equation: this diffusion is done in spectral mode, the information is thus not accessible to the DDH grid-point diagnostics.

File output

FQTPRECICOL	$\delta t F_p^{conv-l}$	cumulated	
FQTPRECICON	$\delta t F_p^{conv-n}$	cumulated	
FQTPRECISTL	$\delta t F_p^{stra-l}$	cumulated	
FQTPRECISTN	$\delta t F_p^{stra-n}$	cumulated	
FQTCONDECOL	$\delta t F_c^{conv-l}$	cumulated	if LHDQLN=.TRUE.
FQTCONDECON	$\delta t F_c^{conv-n}$	cumulated	if LHDQLN=.TRUE.
FQTCONDESTL	$\delta t F_c^{stra-l}$	cumulated	if LHDQLN=.TRUE.
FQTCONDESTN	$\delta t F_c^{stra-n}$	cumulated	if LHDQLN=.TRUE.
VQV0	$\frac{1}{g}\, q_v \delta p$ (t=0)		
VQV1	$\frac{1}{g}\, q_v \delta p$ (t=NSTEP)		
TQVDIVFLUHOR	$-\frac{\delta t}{g}\ \mathrm{div}_\eta\left(q_v \delta p\, \vec{v}\right)$	cumulated	
FQVFLUVERTDYN	$\frac{\delta t}{g}\, q_v \dot{\eta}\, \frac{\partial p}{\partial \eta}$	cumulated	
FQVTUR	$\delta t F_{q_v}^{tur}$	cumulated	
FQVTURCONV	$\delta t F_{q_v}^{tur-conv}$	cumulated	
FQVTURQNEGAT	$\delta t F_{q_v}^{q<0}$	cumulated	
VQL0	$\frac{1}{g}\, q_l \delta p$ (t=0)		if LHDQLN=.TRUE.
VQL1	$\frac{1}{g}\, q_l \delta p$ (t=NSTEP)		if LHDQLN=.TRUE.
TQLDIVFLUHOR	$-\frac{\delta t}{g}\ \mathrm{div}_\eta\left(q_l \delta p\, \vec{v}\right)$	cumulated	if LHDQLN=.TRUE.
FQLFLUVERTDYN	$\frac{\delta t}{g}\, q_l \dot{\eta}\, \frac{\partial p}{\partial \eta}$	cumulated	if LHDQLN=.TRUE.
FQLTUR	$\delta t F_{q_l}^{tur}$	cumulated	if LHDQLN=.TRUE.
FQLTURCONV	$\delta t F_{q_l}^{tur-conv}$	cumulated	if LHDQLN=.TRUE.
FQLTURQNEGAT	$\delta t F_{q_l}^{q<0}$	cumulated	if LHDQLN=.TRUE.
VQN0	$\frac{1}{g}\, q_n \delta p$ (t=0)		if LHDQLN=.TRUE.
VQN1	$\frac{1}{g}\, q_n \delta p$ (t=NSTEP)		if LHDQLN=.TRUE.
TQNDIVFLUHOR	$-\frac{\delta t}{g}\ \mathrm{div}_\eta\left(q_n \delta p\, \vec{v}\right)$	cumulated	if LHDQLN=.TRUE.
FQNFLUVERTDYN	$\frac{\delta t}{g}\, q_n \dot{\eta}\, \frac{\partial p}{\partial \eta}$	cumulated	if LHDQLN=.TRUE.
FQNTUR	$\delta t F_{q_n}^{tur}$	cumulated	if LHDQLN=.TRUE.
FQNTURCONV	$\delta t F_{q_n}^{tur-conv}$	cumulated	if LHDQLN=.TRUE.
FQNTURQNEGAT	$\delta t F_{q_n}^{q<0}$	cumulated	if LHDQLN=.TRUE.

4.3 Momentum budget

Budget equation

$$\frac{\partial}{\partial t}\left(\frac{1}{g}\,\frac{\partial p}{\partial \eta}\,\vec{v}\right) = -\,\frac{1}{g}\,\vec{v}\,\left[\mathrm{div}_{\eta}\left(\frac{\partial p}{\partial \eta}\vec{v}\right) + \frac{\partial}{\partial \eta}\left(\dot{\eta}\frac{\partial p}{\partial \eta}\right)\right] \;-\frac{1}{g}\,\frac{\partial p}{\partial \eta}\left(\vec{v}\cdot\vec{\nabla}\right)\vec{v} - \frac{1}{g}\,\frac{\partial p}{\partial \eta}\,f\vec{k}\times\vec{v}$$
$$-\,\frac{1}{g}\,\frac{\partial p}{\partial \eta}\left(\vec{\nabla}\Phi + RT\,\vec{\nabla}\ln p\right) - \delta_m\frac{\partial F_p\vec{v}}{\partial \eta} - \frac{\partial}{\partial \eta}\left(\vec{F}_v^{tur} + \vec{F}_v^{tur-conv} + \vec{F}_v^{rel} + \vec{F}_v^{meso}\right)$$

where $\vec{v} = (u, v)$ represents the real wind projected in the local geographic coordinates (u positive towards the East, v positive towards the North) and

- $\vec{F}_v^{tur}$ is the turbulent flux,
- $\vec{F}_v^{tur-conv}$ is the convective transport,
- $\vec{F}_v^{rel}$ is the momentum flux due to gravity wave drag.

In the model ($u^{*\prime}$, $v^{*\prime}$) are expressed on the transformed sphere. One must, therefore, go back to the real wind (from transformed sphere to real sphere, then modification by the scale factor) before making a rotation, given by

$$\begin{pmatrix} \vec{e} \\ \\ \vec{n} \end{pmatrix} = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \\ \sin\alpha & \cos\alpha \end{pmatrix} \begin{pmatrix} \vec{i}\,' \\ \\ \vec{j}\,' \end{pmatrix}$$

where ($\vec{e}$, $\vec{n}$) stands for the local geographical vectors. In Arpege $\cos\alpha$ GNORDM $\sin\alpha$ GNORDL. Likewise for gradients.

Pressure force: the value written on file is

$$-\,\frac{1}{g}\,\frac{\partial p}{\partial \eta}\,\left[f\vec{k}\times\vec{v} + \vec{\nabla}\Phi + RT\,\vec{\nabla}\ln p\right]$$

File output

All wind components (variables, fluxes, tendencies) are relative to the true geographical sphere.

VUU0	$\frac{1}{g}\,u\delta p$ (t=0)	
VVV0	$\frac{1}{g}\,v\delta p$ (t=0)	
VUU1	$\frac{1}{g}\,u\delta p$ (t=NSTEP δt)	
VVV1	$\frac{1}{g}\,v\delta p$ (t=NSTEP δt)	
TUUDIVFLUHOR	$-\frac{\delta t}{g}\,u\,\mathrm{div}_{\eta}\,(\delta p\vec{v})$	cumulated
TVVDIVFLUHOR	$-\frac{\delta t}{g}\,v\,\mathrm{div}_{\eta}\,(\delta p\vec{v})$	cumulated

The wind tendency due to advection of wind by itself is missing!

TUUFFVGADPSG	$\delta t \left(\frac{f}{g}\, v\delta p - \frac{\delta p}{g}\left(\frac{\partial \Phi}{\partial x} + RT\,\frac{\partial \ln p}{\partial x}\right)\right)$	cumulated
TVVFFUGADPSG	$-\,\delta t \left(\frac{f}{g}\, u\delta p + \frac{\delta p}{g}\left(\frac{\partial \Phi}{\partial y} + RT\,\frac{\partial \ln p}{\partial y}\right)\right)$	cumulated
FUUFLUVERTDYN	$\frac{\delta t}{g}\, u\dot\eta\, \frac{\partial p}{\partial \eta}$	cumulated
FVVFLUVERTDYN	$\frac{\delta t}{g}\, v\dot\eta\, \frac{\partial p}{\partial \eta}$	cumulated
FUUFLUDUAPLUI	$\delta t \delta_m F_p u$	cumulated
FVVFLUDUAPLUI	$\delta t \delta_m F_p v$	cumulated
FUUTUR	$\delta t F_u^{tur}$	cumulated
FVVTUR	$\delta t F_v^{tur}$	cumulated
FUUTURCONV	$\delta t F_u^{tur-conv}$	cumulated
FVVTURCONV	$\delta t F_v^{tur-conv}$	cumulated
FUUONDEGREL	$\delta t F_u^{rel}$	cumulated
FVVONDEGREL	$\delta t F_v^{rel}$	cumulated
FUUMESO	$\delta t F_u^{meso}$	cumulated
FVVMESO	$\delta t F_v^{meso}$	cumulated

4.4 Kinetic energy budget

Budget equation

$$\frac{\partial}{\partial t}\left(\frac{1}{g}\,k\,\frac{\partial p}{\partial \eta}\right) = -\,\frac{1}{g}\,k\ \mathrm{div}_\eta\left(\vec v\,\frac{\partial p}{\partial\eta}\right) - \frac{1}{g}\,\frac{\partial}{\partial\eta}\left(k\dot\eta\,\frac{\partial p}{\partial\eta}\right) - \frac{1}{g}\,\frac{\partial p}{\partial \eta}\,\vec v\cdot\vec\nabla k$$

$$-\,\frac{1}{g}\,\frac{\partial p}{\partial\eta}\,\vec v\left(\vec\nabla\Phi + RT\,\vec\nabla\ln p\right) - \delta_m\frac{\partial k F_p}{\partial \eta} - \vec v\,\frac{\partial}{\partial\eta}\left(\vec F_v^{tur}+\vec F_v^{tur-conv}+\vec F_v^{rel}+\vec F_v^{meso}\right)$$

The work due to pressure gradient force is precisely known by LAGGRAD. However, the advection term cannot be diagnosed at initial time.

File output

VKK0	$\frac{1}{g}\,k\delta p$ (t=0)	
VKK1	$\frac{1}{g}\,k\delta p$ (t=NSTEP δt)	
TKKDIVFLUHOR	$-\frac{\delta t}{g}\,k\ \mathrm{div}_\eta\left(\delta p\vec v\right)$	cumulated (the advection term is missing!
TKKCONVERSI1	$-\frac{\delta t}{g}\,\delta p\vec v\,\left(\vec\nabla\Phi + RT\,\vec\nabla\ln p\right)$	cumulated
FKKFLUVERTDYN	$\frac{\delta t}{g}\,k\,\dot\eta\,\frac{\partial p}{\partial\eta}$	cumulated
FKKFLUDUAPLUI	$\delta t\delta_m k\, F_p$	cumulated
TKKDISSIPTUR	$-\delta t\vec v\,\delta\vec F_v^{tur}$	cumulated
TKKDISSIPCONV	$-\delta t\vec v\,\delta\vec F_v^{tur-conv}$	cumulated
TKKDISSIPGREL	$-\delta t\vec v\,\delta\vec F_v^{rel}$	cumulated

TKKDISSIPMESO	$-\delta t \vec{v}\, \delta \vec{F}_v^{meso}$	cumulated

4.5 Enthalpy budget

Two types of enthapy equations used in ARPEGE: the DDH use the budget type (4.3), the CPTEND routine uses the eulerien type in h (4.2). The corresponding lagrangian equation in T is given for information in (4.1).

Lagrangian equation in T

$$
\begin{aligned}
r_\eta c_p \frac{dT}{dt} &= r_\eta RT \frac{\omega}{p} + \frac{\partial F_h}{\partial \eta} - L_{v>l}(T) \frac{\partial F_c^l}{\partial \eta} - L_{v>n}(T) \frac{\partial F_c^n}{\partial \eta} \\
&+ F_p^l \frac{\partial}{\partial \eta} \left\{ T \left[c_l - c_{p_a}(1-\delta_m) \right] \right\} + F_p^n \frac{\partial}{\partial \eta} \left\{ T \left[c_n - c_{p_a}(1-\delta_m) \right] \right\} \\
&- T \left[(c_{p_v} - c_{p_a}) \frac{\partial (F_{q_v}^{tur} + F_{q_v}^{tur-conv})}{\partial \eta} + (c_l - c_{p_a}) \frac{\partial (F_{q_l}^{tur} + F_{q_l}^{tur-conv})}{\partial \eta} + (c_n - c_{p_a}) \frac{\partial (F_{q_n}^{tur} + F_{q_n}^{tur-conv})}{\partial \eta} \right] \\
&+ \delta_m F_p \frac{\partial (\Phi + \frac{u^2+v^2}{2})}{\partial \eta} - \vec{v} \cdot \frac{\partial \vec{F}_v^{phys}}{\partial \eta}
\end{aligned}
\tag{4.1}
$$

Eulerian equation in $h = c_p T + \Phi + \frac{u^2+v^2}{2}$

$$
\begin{aligned}
r_\eta \frac{\partial h}{\partial t} &= -r_\eta \left(\vec{v} \cdot \vec{\nabla} \right) h - r_\eta \dot{\eta} \frac{\partial h}{\partial \eta} + r_\eta RT \frac{\omega}{p} \\
&+ \frac{\partial}{\partial \eta} \left\{ F_h + F_{hp} + F_p^l T \left[c_l - c_{p_a}(1-\delta_m) \right] + F_p^n T \left[c_n - c_{p_a}(1-\delta_m) \right] - \delta_m c_p T F_p \right\} \\
&+ \delta_m F_p \frac{\partial h}{\partial \eta}
\end{aligned}
\tag{4.2}
$$

One assumes $\dfrac{\partial \Phi}{\partial t} = 0$.

Budget equation

$$
\begin{aligned}
\frac{\partial}{\partial t} (r_\eta c_p T) &= -\operatorname{div}_\eta (r_\eta c_p T \vec{v}) - \frac{\partial}{\partial \eta} (r_\eta c_p T \dot{\eta}) + r_\eta RT \frac{\omega}{p} \\
&+ \frac{\partial}{\partial \eta} \left\{ F_h + F_{hp} + F_p^l T \left[c_l - c_{p_a}(1-\delta_m) \right] + F_p^n T \left[c_n - c_{p_a}(1-\delta_m) \right] \right\} \\
&+ \delta_m F_p \frac{\partial (\Phi + \frac{u^2+v^2}{2})}{\partial \eta} - \vec{v} \cdot \frac{\partial \vec{F}_v^{phys}}{\partial \eta}
\end{aligned}
\tag{4.3}
$$

where

- $r_\eta = -\frac{1}{g} \dfrac{\partial p}{\partial \eta}$.
- $F_h = F_h^{sol} + F_h^{ther} + F_h^{meso} + F_s^{tur} + F_s^{tur-conv}$.

- $F_s^{tur-conv}$ is the subgrid-scale transport of dry static energy $s = c_p T + \Phi$, due to deep convection.
- $F_{hp} = F_{hp}^l + F_{hp}^n$.
- $F_{hp}^l = F_{hp}^{conv-l} + F_{hp}^{stra-l}$.
- $F_{hp}^n = F_{hp}^{conv-n} + F_{hp}^{stra-n}$.
- $F_{hp}^{conv-l} = - \left[L_{v>l}(T) + (c_l - c_{p_v})T \right] F_c^{conv-l}$.
- $F_{hp}^{conv-n} = - \left[L_{v>n}(T) + (c_n - c_{p_v})T \right] F_c^{conv-n}$.
- $F_{hp}^{stra-l} = - \left[L_{v>l}(T) + (c_l - c_{p_v})T \right] F_c^{stra-l}$.
- $F_{hp}^{stra-n} = - \left[L_{v>n}(T) + (c_n - c_{p_v})T \right] F_c^{stra-n}$.
- F_h^{sol} et F_h^{ther} are the solar and infra-red fluxes.
- $\vec{F}_v^{phys} = \vec{F}_v^{tur} + \vec{F}_v^{tur-conv} + \vec{F}_v^{rel} + \vec{F}_v^{meso}$.
- $c_p = c_{p_a} q_a + c_{p_v} q_v + c_l q_l + c_n q_n$.

File output

`VCT0`	$\frac{1}{g}\, c_p T \delta p$ (t=0)	
`VCT1`	$\frac{1}{g}\, c_p T \delta p$ (t= NSTEP δt)	
`TCTDIVFLUHOR`	$-\frac{\delta t}{g}\ \mathrm{div}_\eta \left(c_p T \delta p\, \vec{v}\, \right)$	cumulated
`TCTCONVERSI2`	$\frac{\delta t}{g}\, RT\delta p(\omega/p)$	cumulated
`TCTCONVERSI3`	$-\delta_m \delta t F_p \delta\Phi$	cumulated
`FCTFLUVERTDYN`	$\frac{\delta t}{g}\, c_p T \dot{\eta}\, \frac{\partial p}{\partial \eta}$	cumulated
`FCTTUR`	$\delta t\, F_s^{tur}$	cumulated
`FCTTURCONV`	$\delta t\, F_s^{tur-conv}$	cumulated
`FCTRAYSOL1`	$\delta t\, F_h^{sol}$	cumulated
`FCTRAYTER1`	$\delta t\, F_h^{ther}$	cumulated
`FCTMESO`	$\delta t\, F_h^{meso}$	cumulated
`FCTPRECISTL`	$-\delta t\, F_{hp}^{stra-l}$	cumulated
`FCTPRECISTN`	$-\delta t\, F_{hp}^{stra-n}$	cumulated
`FCTPRECICOL`	$-\delta t\, F_{hp}^{conv-l}$	cumulated
`FCTPRECICON`	$-\delta t\, F_{hp}^{conv-n}$	cumulated
`FCTPRECCSCOL`	$-\delta t\, F_p^{conv-l} T \left[c_l - c_{p_a}(1-\delta_m) \right]$	cumulated
`FCTPRECCSCON`	$-\delta t\, F_p^{conv-n} T \left[c_n - c_{p_a}(1-\delta_m) \right]$	cumulated
`FCTPRECCSSTL`	$-\delta t\, F_p^{stra-l} T \left[c_l - c_{p_a}(1-\delta_m) \right]$	cumulated
`FCTPRECCSSTN`	$-\delta t\, F_p^{stra-n} T \left[c_n - c_{p_a}(1-\delta_m) \right]$	cumulated

4.6 Angular momentum budget

The angular momentum is defined by

$$\vec{M} = \vec{r} \times (\vec{\Omega} \times \vec{r} + \vec{v})$$

where $\vec{r}$ stands for the position vector, $\vec{v}$ the horizontal wind and $\vec{\Omega}$ the Earth pulsation. In Cartesian coordinate, the kinetic moment reads

$$\begin{aligned}
\vec{M} &= M_1 \vec{i} + M_2 \vec{j} + M_3 \vec{k} \\
&= (av \sin\lambda - a(u + a\Omega\cos\theta) \sin\theta \cos\lambda) \vec{i} \\
&\quad +(-av \cos\lambda - a(u + a\Omega\cos\theta) \sin\theta \sin\lambda) \vec{j} \\
&\quad +a(u + a\Omega\cos\theta) \cos\theta \vec{k}
\end{aligned}$$

with

- $(\vec{i}, \vec{j}, \vec{k})$: $\vec{k}$ along the earth rotation axis, $\vec{i}$ et $\vec{j}$ in the equatorial plane. This base does not rotate with the earth: the position of the Greenwich meridian at `00 UTC` on the day of the beginning of the integration of the model sets $\vec{i}$. $\vec{j}$ is got by an equatorial rotation of $90°$ towards the East of $\vec{i}$, and of $\vec{k} = \vec{i} \times \vec{j}$.
- u and v zonal and meridional wind.
- a earth radius.
- θ and λ latitude and longitude.

Budget equation

$$\frac{\partial}{\partial t}\left(r_\eta \vec{M}\right) = -\vec{M}\left[\mathrm{div}_\eta\left(r_\eta \vec{v}_a\right) + \frac{\partial}{\partial \eta}\left(r_\eta \dot{\eta}\right)\right] - r_\eta\left(\vec{v}_a \cdot \vec{\nabla}\right)\vec{M} + \delta_m \vec{M}\frac{\partial F_p}{\partial \eta} + r_\eta \vec{r} \times \vec{\alpha}$$

avec

- $r_\eta = -\frac{1}{g}\frac{\partial p}{\partial \eta}$
- $\vec{v}_a = \vec{\Omega} \times \vec{r} + \vec{v}$
- $\vec{\alpha} = \vec{\Omega} \times (\vec{\Omega} \times \vec{r}) - \frac{1}{r_\eta}\left[\Phi\vec{\nabla}r_\eta + \frac{1}{g}\frac{\partial}{\partial \eta}\left(\Phi\vec{\nabla}p\right)\right] - \vec{\nabla}\Phi + \frac{1}{r_\eta}\left[\frac{\partial \vec{F}_v}{\partial \eta} + \delta_m F_p \frac{\partial \vec{v}}{\partial \eta}\right]$
- $\vec{F}_v = \vec{F}_v^{tur} + \vec{F}_v^{tur-conv} + \vec{F}_v^{rel}$

File output

`VA10`	$\frac{\delta p}{g} M_1$ (t=0)
`VA20`	$\frac{\delta p}{g} M_2$ (t=0)
`VA30`	$\frac{\delta p}{g} M_3$ (t=0)
`VA11`	$\frac{\delta p}{g} M_1$ (t=NSTEP δt)
`VA21`	$\frac{\delta p}{g} M_2$ (t=NSTEP δt)
`VA31`	$\frac{\delta p}{g} M_3$ (t=NSTEP δt)

TA1DIVFLUHOR	$-\frac{\delta t}{g}\, M_1 \left[\mathrm{div}_\eta\, (\vec{v}\delta p) + a\Omega \cos\theta\, \delta B\, \frac{\partial \pi}{\partial x} \right]$	cumulated
TA2DIVFLUHOR	$-\frac{\delta t}{g}\, M_2 \left[\mathrm{div}_\eta\, (\vec{v}\delta p) + a\Omega \cos\theta\, \delta B\, \frac{\partial \pi}{\partial x} \right]$	cumulated
TA3DIVFLUHOR	$-\frac{\delta t}{g}\, M_3 \left[\mathrm{div}_\eta\, (\vec{v}\delta p) + a\Omega \cos\theta\, \delta B\, \frac{\partial \pi}{\partial x} \right]$	cumulated
FA1FLUVERTDYN	$\delta t \frac{\delta p}{g}\, M_1 \dot{\eta}$	cumulated
FA2FLUVERTDYN	$\delta t \frac{\delta p}{g}\, M_2 \dot{\eta}$	cumulated
FA3FLUVERTDYN	$\delta t \frac{\delta p}{g}\, M_3 \dot{\eta}$	cumulated
TA1ADJUST	$\frac{\delta t}{g} a\, [- \left(\delta p \frac{\partial RT}{\partial y} + RT \delta B \frac{\partial \Phi}{\partial y} \right) \sin(\lambda + \Omega t)$ $+ \left(\delta p \frac{\partial RT}{\partial x} + RT \delta B \frac{\partial \Phi}{\partial x} \right) \cos(\lambda + \Omega t) \sin\theta]$	cumulated
TA2ADJUST	$\frac{\delta t}{g} a\, [- \left(\delta p \frac{\partial RT}{\partial y} + RT \delta B \frac{\partial \Phi}{\partial y} \right) \cos(\lambda + \Omega t)$ $+ \left(\delta p \frac{\partial RT}{\partial x} + RT \delta B \frac{\partial \Phi}{\partial x} \right) \sin(\lambda + \Omega t) \sin\theta]$	cumulated
TA3ADJUST	$-\frac{\delta t}{g} a \left(\delta p \frac{\partial RT}{\partial x} + RT \delta B \frac{\partial \Phi}{\partial x} \right) \cos\theta$	cumulated
TA1NONAX	$\delta t \frac{\delta p}{g} a^2 \Omega^2 \sin\theta \cos\theta \sin(\lambda + \Omega t)$	cumulated
TA2NONAX	$\delta t \frac{\delta p}{g} a^2 \Omega^2 \sin\theta \cos\theta \cos(\lambda + \Omega t)$	cumulated
FA1GRAV	$-\frac{\delta p}{g} a \left[\sin\theta \cos(\lambda + \Omega t) \frac{\partial \Phi}{\partial x} + \sin(\lambda + \Omega t) \frac{\partial \Phi}{\partial y} \right]$	cumulated
FA2GRAV	$-\frac{\delta p}{g} a \left[\sin\theta \sin(\lambda + \Omega t) \frac{\partial \Phi}{\partial x} + \cos(\lambda + \Omega t) \frac{\partial \Phi}{\partial y} \right]$	cumulated
FA3GRAV	$\frac{\delta p}{g} a \cos\theta \frac{\partial \Phi}{\partial x}$	cumulated
FA1FLUDUAPLUI	$\delta t\, \delta_m\, F_p\, M_1$	cumulated
FA2FLUDUAPLUI	$\delta t\, \delta_m\, F_p\, M_2$	cumulated
FA3FLUDUAPLUI	$\delta t\, \delta_m\, F_p\, M_3$	cumulated
TA1TUR	$\delta t\, a\, \left[F_v^{tur} \sin(\lambda + \Omega t) - F_u^{tur} \sin\theta \cos(\lambda + \Omega t) \right]$	cumulated
TA2TUR	$\delta t\, a\, \left[F_v^{tur} \cos(\lambda + \Omega t) - F_u^{tur} \sin\theta \sin(\lambda + \Omega t) \right]$	cumulated
TA3TUR	$\delta t\, a\, F_u^{tur} \cos\theta$	cumulated
TA1TURCONV	$\delta t\, a\, \left[F_v^{tur-conv} \sin(\lambda + \Omega t) - F_u^{tur-conv} \sin\theta \cos(\lambda + \Omega t) \right]$	cumulated
TA2TURCONV	$\delta t\, a\, \left[F_v^{tur-conv} \cos(\lambda + \Omega t) - F_u^{tur-conv} \sin\theta \sin(\lambda + \Omega t) \right]$	cumulated
TA3TURCONV	$\delta t\, a\, F_u^{tur-conv} \cos\theta$	cumulated
TA1ONDEGREL	$\delta t\, a\, \left[F_v^{rel} \sin(\lambda + \Omega t) - F_u^{rel} \sin\theta \cos(\lambda + \Omega t) \right]$	cumulated
TA2ONDEGREL	$\delta t\, a\, \left[F_v^{rel} \cos(\lambda + \Omega t) - F_u^{rel} \sin\theta \sin(\lambda + \Omega t) \right]$	cumulated
TA3ONDEGREL	$\delta t\, a\, F_u^{rel} \cos\theta$	cumulated

Listing output

When an output on listing is requested,what is printed is the intersection position of $\vec{M}$ with the earth surface, in latitude and in longitude values:

$$\begin{aligned}\cos\theta_0 &= \frac{M_3}{\sqrt{M_1^2+M_2^2+M_3^2}}\\ \cos(\lambda_0+\Omega t) &= \frac{M_1}{\sqrt{M_1^2+M_2^2}}\end{aligned}$$

4.7 Entropy budget

Budget equation

$$\begin{aligned}\frac{\partial}{\partial t}(r_\eta s) = &-\mathrm{div}_\eta\,(r_\eta s\vec{v}) - \frac{\partial}{\partial\eta}(r_\eta s\dot{\eta}) + \frac{\partial}{\partial\eta}\left[s_l\left(F_p^{conv-l}+F_p^{stra-l}\right)+s_g\left(F_p^{conv-n}+F_p^{stra-n}\right)\right]\\ &-\frac{1}{T}\vec{v}\cdot\frac{\partial}{\partial\eta}\left(\vec{F}_v^{tur}+\vec{F}_v^{tur-conv}+\vec{F}_v^{rel}\right)-(s_v-s_a+c_{pv}-c_{pa})\frac{\partial}{\partial\eta}(F_q^{tur}+F_q^{tur-conv})\\ &+\frac{1}{T}(F_h^{sol}+F_h^{ther}+F_h^{tur}+F_h^{tur-conv})-(1-\delta_m)\left[s_a\frac{\partial F_p}{\partial\eta}+\frac{1}{T}c_{pa}\frac{\partial T}{\partial\eta}F_p\right]+\delta_m\frac{1}{T}F_p\frac{\partial\Phi}{\partial\eta}\end{aligned}$$

where

- $r_\eta = -\frac{1}{g}\frac{\partial p}{\partial\eta}$.
- $s = s_a + (s_v - s_a)q_v$.
- $s_a = c_{pa}\ln(\frac{T}{T_0}) - R_a\ln(\frac{p_a}{p_0}) + s_a^0$.
- $s_v = c_{pv}\ln(\frac{T}{T_0}) - R_v\ln(\frac{p_v}{p_0}) + s_v^0$.
- $s_l = c_w\ln(\frac{T}{T_0}) + s_l^0$.
- $s_g = c_g\ln(\frac{T}{T_0}) + s_g^0$.
- $s_a^0 = 6775\,Jkg^{-1}K^{-1}$.
- $s_v^0 = 10320\,Jkg^{-1}K^{-1}$.
- $s_l^0 = 3517\,Jkg^{-1}K^{-1}$.
- $s_g^0 = 2296\,Jkg^{-1}K^{-1}$.

The horizontal divergence term is computed as

$$\mathrm{div}_\eta\,(r_\eta s\vec{v}) = s\,\mathrm{div}_\eta\,(r_\eta\vec{v}) + r_\eta\vec{v}\cdot\vec{\nabla}s$$

where

- $\vec{v}\cdot\vec{\nabla}s = (s_v - s_a)\vec{v}\cdot\vec{\nabla}q_v + \vec{v}\cdot\left[c_p\vec{\nabla}\ln T - R\vec{\nabla}\ln p\right]$.
- $\vec{v}\cdot\vec{\nabla}\ln p = \mathtt{RTGR}\,\vec{v}\cdot\vec{\nabla}\pi$.

File output

`VSS0`	$\frac{1}{g}s\delta p$ (t=0)	
`VSS1`	$\frac{1}{g}s\delta p$ (t=NSTEP Δt)	
`TSSDIVFLUHOR`	$-\frac{\delta t}{g}\{s\,\mathrm{div}_\eta\,(\vec{v}\delta p)$	
	$+\delta p[(s_v - s_a)\vec{v}\cdot\vec{\nabla}q_v$	
	$+\frac{1}{T}c_p\vec{v}\cdot\vec{\nabla}T$	
	$-R\,\mathtt{ZRTGR}\,\vec{v}\cdot\vec{\nabla}\pi]\}$	cumulated
`TSSPRECICT`	$\delta t(1-\delta_m)[s_a\delta F_p + c_{p_a}F_p\frac{\delta T}{T}]$	
	$-\delta_m\delta t\frac{1}{T}F_p\delta\Phi$	cumulated
`FSSFLUVERTDYN`	$\frac{\delta t}{g}s\dot{\eta}\delta p$	cumulated
`FSSPRECISS`	$\delta t[s_l(F_p^{conv-l}+F_p^{stra-l})+s_g(F_p^{conv-n}+F_p^{stra-n})]$	cumulated
`TSSDISSIPTUR`	$\frac{\delta t}{T}\vec{v}\cdot\delta\vec{F}_v^{tur}$	cumulated
`TSSDISSIPCONV`	$\frac{\delta t}{T}\vec{v}\cdot\delta\vec{F}_v^{tur-conv}$	cumulated
`TSSDISSIPGREL`	$\frac{\delta t}{T}\vec{v}\cdot\delta\vec{F}_v^{rel}$	cumulated
`TSSTURQVTOT`	$\delta t(s_v - s_a + c_{p_v} - c_{p_a})(\delta F_q^{tur}+\delta F_q^{tur-conv})$	cumulated
`TSSRAY1`	$-\frac{\delta t}{T}(\delta F_h^{sol}+F_h^{ther})$	cumulated
`TSSTURCTTOT`	$-\frac{\delta t}{T}(\delta F_h^{tur}+F_h^{tur-conv})$	cumulated

4.8 Potential energy budget

Budget equation

$$-\frac{1}{g}\frac{\partial p}{\partial\eta}\,\vec{v}\left(\vec{\nabla}\Phi + RT\,\vec{\nabla}\ln p\right) = -RT\left[\frac{\omega}{gp}+\delta_m\frac{F_p}{p}\right]\frac{\partial p}{\partial\eta} - \frac{1}{g}\,\mathrm{div}_\eta\left(\Phi\frac{\partial p}{\partial\eta}\,\vec{v}\right)$$
$$-\frac{1}{g}\frac{\partial}{\partial\eta}\left[\Phi\left(\frac{\partial p}{\partial t}+\dot{\eta}\frac{\partial p}{\partial\eta}+\delta_m g F_p\right)\right]$$

File output

Three of these budget terms have already been mentioned, see «kinetic energy budget» et «enthalpy budget». Is added

`VEP0`	$\frac{1}{g}\,\Phi\delta p$ (t=0)	
`VEP1`	$\frac{1}{g}\,\Phi\delta p$ (t=NSTEP δt)	
`TEPDIVFLUHOR`	$-\frac{\delta t}{g}\,\mathrm{div}_\eta\,(\Phi\delta p\vec{v})$	cumulated
`FEPCONVERSIFL`	$-\frac{\delta t}{g}\,\Phi_{\tilde{\ell}}\sum_{k=1}^{\ell}\mathrm{div}_\eta\,(\delta p\vec{v})$	cumulated

4.9 Surface budgets

The DDH tool does not produce a closed soil budget, rather some surface fluxes or variables multiplied by land/sea mask `PITM` (1 on land, 0 at sea). The surface occupied by these points in each domain will also be computed. This operation makes it possible, to bring back the variable mean and the mean of fluxes to the sole surface of the continents.

Budget equation

- Surface temperature:

  $$\frac{\partial c_{ms}T_s}{\partial t} = \delta_{terre}\left\{Fh_{\tilde{L}}^{sol} + Fh_{\tilde{L}}^{ther} + L_v(T_s)E_l + L_n(T_s)E_n + F_{csa} - F_{csp} - L_{fonte}F_{fonte}\right\}$$

  where c_{ms} is the surfacic calorific capacity of the surface layer. A constant value is used here: $1/$`HSOL`.

- Deep temperature:

  $$\frac{\partial c_{mp}T_p}{\partial t} = \delta_{terre}F_{csp}$$

  where c_{mp} is the surfacic calorific capacity of the deep layer. A constant value is used here: `RTINER`/`HSOL`.

- Surface water content:

  $$\frac{\partial w_s}{\partial t} = \delta_{terre}\left\{F_p^{conv-l} + F_p^{stra-l} + E_l - F_{perco} + F_{fonte} - F_{ruiss}\right\}$$

- Deep water content:

  $$\frac{\partial w_p}{\partial t} = \delta_{terre}\left\{F_{perco} - F_{ruisp}\right\}$$

- Surface snow content:

  $$\frac{\partial w_n}{\partial t} = \delta_{neige}\delta_{terre}\left\{F_p^{conv-n} + F_p^{stra-n} + E_n - F_{fonte}\right\}$$

File output

The fields are written on 3 file articles:

- Variables at $t = 0$: article `S␣V0`

$\frac{1}{S_{\mathcal{D}}}\sum_{(j,k)\in\mathcal{D}}$ `PITM` $\sigma_{j,k}$	Land fraction
$c_{ms}T_s$	Surface layer: energy
$c_{mp}T_p$	Deep layer: energy
w_s	Surface layer: water content (surfacic mass)
w_n	Surface layer: snow content (surfacic mass)
w_p	Deep layer: water content (surfacic mass)

- Variables at $t = NSTEP \times \delta t$: article `S␣V1`

  The same articles as above.

- Flux cumulated from 0 to NSTEP: article `S␣FVRAC`

${F_h}_{\tilde{L}}^{sol}$	Solar radiation
${F_h}_{\tilde{L}}^{ther}$	Infra-red radiation
$L_v(T_s)E_l$	Latent heat (water)
$L_n(T_s)E_n$	Latent heat (snow)
F_{csa}	Sensible heat at surface
F_{csp}	Sensible heat between surface and deep layer
$L_{fonte}F_{fonte}$	Surface flux due to melting Flux de chaleur en surface lié à la fonte de neige
F_p^{stra-l}	Precipitation: resolved, rain
F_p^{stra-n}	Precipitation: resolved, snow
F_p^{conv-l}	Precipitation: subgrid-scale, rain
F_p^{conv-n}	Precipitation: subgrid-scale, snow
E_l	Surface evaporation (water)
E_n	Surface evaporation (snow)
F_{perco}	Percolation from surface to deep
F_{fonte}	Snow melting
F_{ruiss}	Surface run-off
F_{ruisp}	Deep run-off

4.10 Instantaneous diagnostics

4.10.1 Relative humidity

`VHR0`	$\frac{1}{g}H_r\delta p$ (t=0)
`VHR1`	$\frac{1}{g}H_r\delta p$ (t=NSTEP δt)

4.10.2 Cloudiness

`VNT0`	$\frac{1}{g}n_t\delta p$ (t=0)
`VNT1`	$\frac{1}{g}n_t\delta p$ (t=NSTEP δt)

4.10.3 Vertical velocity

`VOM0`	$\frac{1}{g}\omega\delta p$ (t=0)
`VOM1`	$\frac{1}{g}\omega\delta p$ (t=NSTEP δt)

4.10.4 Passive variables

The passive variables are diagnosed if `LHDHKS` is true and if `NFPASS` (number of passive variables in the model) is higher or equal to 1.

`Vxx0`	$\frac{1}{g}v_{xx}\delta p$ (t=0)
`Vxx1`	$\frac{1}{g}v_{xx}\delta p$ (t=NSTEP δt)

where xx is between 1 and `NFPASS`.

4.11 Cumulated mass

To make possible future conversion of tendencies and fluxes (extensive) in intensive values, the following value is diagnosed:

`PPP`	$\frac{1}{g}\delta t\delta p$	cumulated

Chapter 5

Budgets, AROME model

5.1 Microphysical process

In AROME physical parameterizations from MesoNH are used. The greatest difference compared to parameterizations used in ALADIN appears in microphysics. Microphysical processes in AROME with corresponding tendencies/fluxes are given in table 5.1. Each column belongs to one microphysical species; tendencies/fluxes written in it change its species amount. The last column shows tendencies/fluxes that change enthalpy due to microphysical processes. On output from MesoNH subroutines, only tendencies are available, but they are converted into fluxes at a later stage if LBUFLUX is .TRUE..

Microphysical process	vapour	cloud water	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			$F_{q_r}^{sfrz}$			$F_{q_g}^{sfr}$	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				$F_{q_i}^{autoi}$	$F_{q_s}^{autoi}$		
deposition on graupel	$F_{q_v}^{depg}$					$F_{q_g}^{dep}$	F_h^{depg}
auto-conversion of cloud water		$F_{q_l}^{autor}$	$F_{q_r}^{autor}$				
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		$F_{q_l}^{rim}$			$F_{q_s}^{rim}$	$F_{q_g}^{rim}$	F_h^{rim}
collection of raindrops			$F_{q_r}^{accs}$		$F_{q_s}^{accs}$	$F_{q_g}^{accs}$	F_h^{accs}
melting of aggregates					$F_{q_s}^{cmel}$	$F_{q_g}^{cmel}$	
contact freezing			$F_{q_r}^{cfrz}$	$F_{q_i}^{cfrz}$		$F_{q_g}^{cfrz}$	F_h^{cfrz}
wet growth		$F_{q_l}^{wetg}$	$F_{q_r}^{wetg}$	$F_{q_i}^{wetg}$	$F_{q_s}^{wetg}$	$F_{q_g}^{wetg}$	F_h^{wetg}
dry growth		$F_{q_l}^{dryg}$	$F_{q_r}^{dryg}$	$F_{q_i}^{dryg}$	$F_{q_s}^{dryg}$	$F_{q_g}^{dryg}$	F_h^{dryg}
melting of graupel			$F_{q_r}^{mltg}$			$F_{q_g}^{mltg}$	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 5.1: Microphysical fluxes.

5.2 Balance equations

5.2.1 Momentum

FUUTUR	F_u^{tur}	vertical turbulence flux of u velocity component
FVVTUR	F_v^{tur}	vertical turbulence flux of v velocity component
FVWTUR	F_w^{tur}	vertical turbulence flux of u velocity component

5.2.2 Turbulence kinetic energy

FTETURB	F_{tke}^{tur}	turbulent flux of turbulent kinetic energy
FTEDYPRO	$F_{tke}^{tur-prod-dyn}$	dynamic production of turbulent kinetic energy
FTETERMPRO	$F_{tke}^{tur-prod-term}$	thermic production of turbulent kinetic energy
FTEDISS	$F_{tke}^{tur-diss}$	dissipation of turbulent kinetic energy

5.2.3 Enthalpy

FCTNEGC1	F_h^{negc1}	correction of negative specific ratios after advection
FCTCDEPI	F_h^{cdepi}	adjustment of water vapour, cloud water and cloud ice
FCTVCONV	F_h^{conv}	convection flux of enthalpy
FCTVTURB	F_h^{tur}	vertical turbulent flux of enthalpy
FCTDISSTUR	$F_h^{tur-diss}$	dissipation of turbulent kinetic energy
FCTNEGC	F_h^{negc}	correction of negative specific ratios after turbulence
FCTHENUI	F_h^{henuv}	heterogeneous nucleation of ice
FCTHON	F_h^{honl}	homogeneous nucleation of ice
FCTSFR	F_h^{sfr}	spontaneous freezing
FCTDEPS	F_h^{deps}	deposition on snow
FCTDEPG	F_h^{depg}	deposition on graupel
FCTREVA	F_h^{reva}	rain evaporation
FCTRIM	F_h^{rim}	riming by cloud droplets
FCTACCS	F_h^{accs}	collection of raindrops and snow on graupel

FCTCFRZ	F_h^{cfrz}	contact freezing of rain
FCTWETG	F_h^{wetg}	wet growth of graupel
FCTDRYG	F_h^{dryg}	dry growth of graupel
FCTMLTG	F_h^{mltg}	melting of graupel
FCTMLTI	F_h^{mlti}	melting of cloud ice
FCTBERFI	F_h^{berfi}	Bergeron-Findeisen effect
FCTRAYSOL1	$F_h^{raysol1}$	solar radiation
FCTRAYTER1	$F_h^{rayter1}$	earth radiation

5.2.4 Water vapour

FQVNEGC1	$F_{q_v}^{negc1}$	correction of negative specific ratios after advection
FQVDEPI	$F_{q_v}^{depi}$	adjustment of water vapour, cloud water and cloud ice
FQVVCONV	$F_{q_v}^{conv}$	convection flux of water vapour
FQVVTURB	$F_{q_v}^{tur}$	vertical turbulent flux of water vapour
FQVNEGC	$F_{q_v}^{negc}$	correction of negative specific ratios after turbulence
FQVHENUI	$F_{q_v}^{henu}$	heterogeneous nucleation of ice
FQVDEPS	$F_{q_v}^{deps}$	deposition on snow
FQVDEPG	$F_{q_v}^{depg}$	deposition on groupel
FQVREVA	$F_{q_v}^{reva}$	rain evaporation

5.2.5 Cloud water

FQLNEGC1	$F_{q_l}^{negc1}$	correction of negative specific ratios after advection
FQLCDEPI	$F_{q_l}^{cdepi}$	adjustment of water vapour, cloud water and cloud ice
FQLVCONV	$F_{q_l}^{conv}$	convection flux of cloud water
FQLVTURB	$F_{q_l}^{tur}$	vertical turbulent flux of cloud water
FQLNEGC	$F_{q_l}^{negc}$	correction of negative specific ratios after turbulence
FQLHON	$F_{q_l}^{hon}$	homogeneous nucleation of ice
FQLAUTO	$F_{q_l}^{autor}$	auto-conversion of cloud water
FQLACCR	$F_{q_l}^{accr}$	accretion of cloud wter on rain
FQLRIMS	$F_{q_l}^{rim}$	riming by cloud droplets
FQLWETG	$F_{q_l}^{wetg}$	wet growth of graupel
FQLDRYG	$F_{q_l}^{dryg}$	dry growth of graupel
FQLMLTI	$F_{q_l}^{mlti}$	melting of cloud ice
FQLBERFI	$F_{q_l}^{berfi}$	Bergeron-Findeisen effect

5.2.6 Rain

FQRNEGC	$F_{q_r}^{negc}$	correction of negative specific ratios after advection
FQRSEDI	F_{rp}	sedimentation
FQRSFR	$F_{q_r}^{sfrz}$	spontaneous freezing of rain
FQRAUTO	$F_{q_r}^{autor}$	auto-conversion of cloud water
FQRACCL	$F_{q_r}^{accr}$	accretion of cloud wter on rain
FQRREVA	$F_{q_r}^{reva}$	rain evaporation
FQRACCS	$F_{q_r}^{accs}$	collection of raindrops on graupel
FQRCFRZ	$F_{q_r}^{cfrz}$	contact freezing of rain
FQRWETG	$F_{q_r}^{wetg}$	wet growth of graupel
FQRDRYG	$F_{q_r}^{dryg}$	dry growth of graupel
FQRMLTG	$F_{q_r}^{mltg}$	melting of graupel

5.2.7 Cloud ice

FQINEGC1	$F_{q_i}^{negc1}$	correction of negative specific ratios after advection
FQICDEPI	$F_{q_i}^{cdepi}$	adjustment of water vapour, cloud water and cloud ice
FQICONV	$F_{q_i}^{conv}$	convection flux of cloud ice
FQITURB	$F_{q_i}^{tur}$	vertical turbulent flux of cloud ice
FQINEGC	$F_{q_i}^{negc}$	correction of negative specific ratios after turbulence
FQISEDI	F_{ip}	sedimentation
FQIHENU	$F_{q_i}^{henu}$	heterogeneous nucleation of ice
FQIHON	$F_{q_i}^{hon}$	homogeneous nucleation of ice
FQIAGGS	$F_{q_i}^{agg}$	collection of ice on snow
FQIAUTS	$F_{q_i}^{autoi}$	auto-conversion of ice to snow
FQICFRZ	$F_{q_i}^{cfrz}$	contact freezing of rain
FQIWETG	$F_{q_i}^{wetg}$	wet growth of graupel
FQIDRYG	$F_{q_i}^{dryg}$	dry growth of graupel
FQIMLT	$F_{q_i}^{mlti}$	melting of cloud ice
FQIBERFI	$F_{q_i}^{berfi}$	Bergeron-Findeisen effect

5.2.8 Snow

FQSNEGC	$F_{q_s}^{negc}$	correction of negative specific ratios after advection
FQSSEDI	F_{sp}	sedimentation
FQSDEPS	$F_{q_s}^{dep}$	deposition on snow
FQSAGGS	$F_{q_s}^{agg}$	collection of ice on snow
FQSAUTS	$F_{q_s}^{autoi}$	auto-conversion of ice to snow
FQSRIM	$F_{q_s}^{rim}$	riming by cloud droplets
FQSACC	$F_{q_s}^{accs}$	collection of raindrops and snow on graupel
FQSCMEL	$F_{q_s}^{cmel}$	melting of aggregates
FQSWETG	$F_{q_s}^{wetg}$	wet growth of graupel
FQSDRYG	$F_{q_s}^{dryg}$	dry growth of graupel

5.2.9 Graupel

FQGNEGC	$F_{q_g}^{negc}$	correction of negative specific ratios after advection
FQGSEDI	F_{gp}	sedimentation
FQGSFR	$F_{q_g}^{sfr}$	spontaneous freezing
FQGDEPG	$F_{q_g}^{dep}$	deposition on gropel
FQGRIM	$F_{q_g}^{rim}$	riming by cloud droplets
FQGACC	$F_{q_g}^{accs}$	collection of raindrops and snow on graupel

FQGCMEL	$F_{q_g}^{cmel}$	melting of aggregates
FQGCFRZ	$F_{q_g}^{cfrz}$	contact freezing of rain
FQGWETG	$F_{q_g}^{wetg}$	wet growth of graupel
FQGDRYG	$F_{q_g}^{dryg}$	dry growth of graupel
FQGMLT	$F_{q_g}^{mltg}$	melting of graupel

5.3 Common Dynamics-Physics Interface CDPI

FQVPL1	P_l'	Pseudo flux due to condensation
FQVPI1	P_i'	Pseudo flux due to sublimation
FQLPL2	P_l''	Pseudo flux due to evaporation of rain
FQIPI2	P_i''	Pseudo flux due to conversion of cloud ice to snow and graopel
FQRPL3	P_l'''	Pseudo flux due to evapotarion of rain
FQSPI3	P_i'''	Pseudo flux due to deposition on snow and groupel
FQGPG3	P_g'''	Pseudo flux due to deposition on snow and groupel
FQRPR0	P_r	Flux of falling rain drops
FQIPI0	P_i	Flux of falling cloud ice
FQSPS0	P_s	Flux of falling snow
FQGPG0	P_g	Flux of falling groupel

Chapter 6

Software maintenance

In this chapter, one will find a short description of DDH routines, and the organization of arrays. In a very practical manner, is described the necessary operations when adding a supplementary field.

6.1 Main arrays and their organization

The main arrays are in two modules:

- `YOMTDDH` for arrays receiving variables and cumulated tendencies and fluxes.
- `YOMMDDH` pour the other arrays, except logical variables which are in `YOMLDDH`.

6.1.1 Arrays describing domains

The distribution of grid-point points in the user's domains is inside `NDDHLA(NDLON,NDGL)` for zonal bands, `NDDHPU(NDLON,NDGL,NDHNPU)` for limited domains and isolated points.

- `NDHNPU:` number of planes used by the user.
- `NDDHI(NDLON,NDGL)` for the domain distribution,
- `HDSF (NDLON,NDGL)` for the weights $\sigma_{j,k}$ of each grid-point.

For each ($jlon$, jgl) point of the Gauss grid, within `NDDHI`, one will find the index of the external domain. It varies between 1 and `NDHIDH`.

- *Global domain:* `NDHIDH = 1` every points belong to domain 1.
- *Zonal bands:* `NDHIDH = NDHKD`
- *Limited areas:* `NDHIDH` depends very much on declaration details and is very unpredictable.

The following tables allow to reconstitute the user's domains using internal domains.

- `NLRDDH(NDHDDX,NDHKD):` integers contained in NLRDDH, from `NLRDDH(1,JKD)` to `NLRDDH(NLXDDH(JKD),JKD)`, are the internal domains whose reunion makes the latitude band `JKD`,

- HDSFLA(NDHKD): weight of each latitude band.

- NURDDH(NDHDDX,0:NDHBPX,NDHNPU): integers from NURDDH(1,JDOM,JMASK) to NURDDH(NUXDDH(JDOM,JMASK),JDOM,JMASK) are the domains whose reunion makes the user domain (JMASK,JDOM). JMASK is the index of the virtual plane, 1 $\leq$ JMASK $\leq$ NDHNPU, and JDOM is the index of the domain within the plane: 0 $\leq$ JDOM $\leq$ NDHBPU(JMASK). Points not assigned by users belong to the domain 0.

- HDSFDU(0:NDHBPX,NDHNPU): Weight of each user's domain.

The weight of the global domain is HDSFGL, from MODULE/YOMIDDH/.

6.1.2 Data arrays

Pointers contained within MODULE/DDHDIM/and /DDHPON/ from YOMMDDH identify the content of these tables. These tables are HDCV0(0:NFLEV,NDHCV,NDHIDH,NDHTSK) alias HDCVB0(NDHCV*(NFLEV+1),NDHIDH,NDHTSK): arrays at time 0 and tendencies/fluxes cumulated from 0 to NSTEP-1.
HDCV1(0:NFLEV,NDHCV,NDHIDH,NDHTSK) alias HDCVB1(NDHCV*(NFLEV+1),NDHIDH,NDHTSK): variables at time NSTEP and tendencies/fluxes cumulated from 0 to NSTEP.
HDCS0(NDHCS,NDHIDH,NDHTSK): soil variables at time 0 and cumulated fluxes from 0 to NSTEP-1.
HDCS1(NDHCS,NDHIDH,NDHTSK): soil variables at time NSTEP and cumulated fluxes from 0 to NSTEP.

- NDHIDH: number of internal domains.

- NDHTSK: number of tasks.

- NDHCV: total number of vertical profiles.

- NDHCS: total number of surface fields.

All vertical profiles are defined on NFLEV+1 words. Generally HDCV(0,field ,domain ,task)=0. Before describing these arrays, a few words on *logical pointers.* They are of two kinds

- Permanent pointers who, for each scientific options, show le number of fields of each categories. For the moment, the options are LHDHKS, LHDMCI and LHDENT. Categories are composed of variables, tendencies/dynamical fluxes, tendencies/physical fluxes (i.e. for the moment there is no distinction between tendency and flux).

- situation pointers which depend on chosen options for a given experiment.

Permanent pointers are initialized in SUNDDH. Every counted field corresponds to specific computation FORTRAN code lines inCPDYDDH and CPPHDDH, and to writing lines or other editions in PPEDDH and PPFIDH. A corresponding commentary is in the general nomenclature of Y0MTDDH. Situation pointers are also initialized in SUNDDH from permanent pointers and from logical options given by NAMDDH. They control global lengths in CPG, POSDDH and PPSYDH where fields are undifferentiated.

Permanent pointers are

- `NDHVHK:` number of variables under `LHDHKS`
- `NDHFHKD:` number of fluxes/tendencies under `LHDHKS`
- `NDHFHKP:` number of fluxes/tendencies under `LHDHKS`
- `NDHTHK = NDHVHK+NDHFHKD+NDHFHKP`

In the same way, for the option `LHDMCI`

- `NDHVMC`
- `NDHFMCD`
- `NDHFMCP`
- `NDHTMC`

For the option `LHDENT`

- `NDHVEN`
- `NDHFEND`
- `NDHFENP`

Soil (under `LHDHKS`)

- `NDHVS`
- `NDHFSD`
- `NDHFSP`

(The total number is `NDHCS`).

Vertical profiles are splitted into categories

```
NDHFxxD = NDHAxxD + NDHBxxD
NDHFxxP = NDHAxxP + NDHBxxP
```

where (A) stands for tendencies and (B) for dynamical fluxes.
Situation pointers are

- `NDHVV:` number of variables in vertical profiles
- `NDHFVD:` number of fluxes/dynamical tendencies = `NDHAVD+NDHBVD`
- `NDHFVP:` number of fluxes/physical tendencies = `NDHAVP+NDHBVP`

The organisation is as follow

Champs	each field has `NFLEV+1` levels		
`1`	instantaneous values	variables	
`...`			
`...`			
`...`			
`NDHVV`			
`NDHVV+1`	cumulated values	`NDHAVD`	dynamical tendencies
`...`			
`...`			
`...`		`NDHBVD`	dynamical fluxes
`...`			
`NDHVV+NDHFVD`			
`NDHVV+NDHFVD+1`		`NDHAVP`	physical fluxes
`...`			
`...`			
`...`		`NDHBVP`	physical tendencies
`...`			
`NDHCV`			

For every categories (variables, dynamical tendency, dynamical flux, etc) one may find fields linked to `LHDHKS` (eventually) then those of `LHDMCI` (eventually) then those of `LHDENT` (eventually). Soil arrays work on the same principle. Fields from 1 to `NDHVV+NDHFVD` are computed in`CPDYDDH`. Fields from `NDHVV+NDHFVD+1` to `NDHCV` are computed in `CPPHDDH`.

6.1.3 Main local arrays

Arrays used as liaison between computations in each grid point and partial means in the one side and the overall table, the link between partial means and output on the other side, go into this category.

6.1.3.1 Arrays for computations at each grid point

`ZDHCV(KPROMA,0:NFLEV,NDHCVSU)` alias `PDHCV(KPROMA,NDHCVSU*(NFLEV+1))` fields in vertical profiles in `CPG`. Alias used in `CPCUDDH`
`ZDHCS(KPROMA,NDHCSSU)` soil fields.

- `KPROMA:` maximun number of points in the horizontal, by grid point task,
- `NDHCVSU = max(1,NDHCV)`
- `NDHCSSU = max(1,NDHCS)`

---SU lengths ensure that d'ARPEGE is properly working even if NDHCV=0 or NDHCS=0 (case(s) where DDH diagnostics are not activated).

6.1.3.2 Arrays for synthesis over a user domain

ZDHCV(0:NFLEV,NDHCVSU+NDHVV) alias PDHCV((NDHCVSU+NDHVV)*(NFLEV+1)) fields in vertical profiles. Alias used in PPSYDH.
ZDHCS(NDHCSSU+NDHVS) soil fields.

In this table, only one domain is present at a given time. It is organized as follow

Field	
1	variables at t
...	
NDHVV	
NDHVV+1	tendencies then fluxes
...	dynamical, cumulated
NDHVV+NDHFVD	from 0 to NSTEP-1
NDHVV+NDHFVD+1	fluxes, then tendencies
...	dynamical, cumulated
NDHCV	from 0 to NSTEP-1
NDHCV+1	variables at t=NSTEP*TSTEP
...	
NDHCV+NDHVV	

ZDHCV array structure.

6.2 Organization of the main functions

An inventory of the main DDH sub programmes and their calling tree is presented here. The following conventions are assumed:

- (—) name of the sub programme between parenthesis: sub programme whose main function is not to compute diagnostics; generally speaking, sub programmeARPEGE already existing.
- $\boxed{\text{m}}$ Multitask sub programme.
- $\boxed{\text{tci}}$ specific DDH sub programme in which all fields are undifferentiated
- $\boxed{\text{cci}}$ specific DDH sub programme in which every field is specified.

(SU0YOMA), 0 level initialisations calling

- (SULUN), initialisation of numbers of file logical units.

- `(SUCT0)`, initialisation of parameters of output frequencies.
- `SUNDDH`, initialisation of permanent logical as well as other default values. Read `NAMDDH`. Deduce dimensions (except `NDHIDH` and `NDHTSK`) and logicals.
- `(SUALLO)`, allocate global arrays.

`(SU0YOMB)`, initialisations from 0 level, calls

- `(SULEG)`, computation of Gauss weight $\omega(k)$.
- `(SUGEM1)`, geographical coordinates computation: (λ_q, μ_q) in each point.
- `SUMDDH`, verification and set up of domain declarations (from `BDEDDH` to `FNODDH`). Distribution of users domains in internal domains, computation of the number of internal domains `NDHIDH`. Computation of weights of interest for horizontal means (`HDSF`, `HDSFGL`, `HDSFLA`, etc...). Print of computed masks values, by calling `PRIMDDH`.
- `(SUOPH)`, generic name of DDH files (`CFNDDH` from MODULE `/YOMOPC/`).
- `(SUSC2)`, computation of the number of logical tasks: `NDHTSK = NSLBR - NDGSA + 1`.
- `SUALTDH`, allocate global arrays (MODULE `/YOMTDDH/`). Initialise these to 0.

`(CNT1)`, level 1 of the model, calls `SU1YOM`, initialization of output overcontrol: `N1DHP` and `N1DHF` (MODULE `/YOMCT1/`).
`(CNT4)`, management of the temporal loop, calls `(MONIO)`, determination of output time steps (`IDHFTS`, `IDHPTS`).
`(STEPO)`, control of the integration at the lowest level, calls

- `(SCAN2H)`, initialization of the input-output scheme, calls `ZERODDH`, transfer of fluxes/tendencies cumulated in time from `HDCVB1` to `HDCVB0` and zeroing of the part of `HDCS1` and `HDCVB1` tables which will receive the cumulated in the horizontal of variables at the current time.
- `(SCAN2M)` $\boxed{\text{m}}$, multi task interface of grid point computations, calls `CPG` $\boxed{\text{m}}$, grid point computation:
  - Declaration of local arrays `IDDHI` and `ZDHSF` (resp. for the domains distribution and for the points weight).
  - Declaration of `ZDHCV` et `ZDHCS` (fields resp. for 3D and 2D cumulated).
  - Interface from `NDDHI` and `HDSF` to `IDDHI` and `ZDHSF`.
  - Call to `CPDYDDH` $\boxed{\text{m}}$ $\boxed{\text{cci}}$, computation in every points of diagnosed atmospheric variables ($\delta p, q\delta p, C_pT\delta p$, etc...), of tendencies and of adiabatic fluxes, and possible call to `CPVRDH` (if the verification option is activated).
  - Call to `CPPHDDH` $\boxed{\text{m}}$ $\boxed{\text{cci}}$, computation in every points of fluxes and of tendencies du to physical parametrizations, soil computation, and possible call to `CPVRDH` (if the verification option is activated).
  - Call to `CPCUDDH` $\boxed{\text{m}}$ $\boxed{\text{tci}}$, partial horizontal mean of variables, stored in `HDCVB1` and `HDCVB0` if `NSTEP=0`, if `NSTEP` different from `NSTOP` temporal integration and partial horizontal mean of fluxes/tendencies in `HDCVB1`.

- `POSDDH`, output management, converts computation of internal domains into users domains, and gives the results on a file or listing.
  - `PPVFDH`, edition of verifications on a point.
  - `PPSYDH` $\boxed{\text{tci}}$, final horizontal means for a user domain, edition of arrays `ZDHCV` and `ZDHCS`, multiplication of variables by par $1/(gS_D)$ and of cumulated fields by $(\delta t/S_D)$.
  - `PPEDDH` $\boxed{\text{cci}}$, vertical mean budget edition.
  - `PPFIDH` $\boxed{\text{cci}}$, writing on file of results of diagnostics for each domain: articles of documentation and fields.
- `CPCUVDH`, cumulated in time either for a flux or a tendency in case of verification.

6.3 How to add new fields to budgets

This section describes the operations to implement in order to incorporate one or more new fields in the budgets. The sub programmes such as tci will work as long as the dimensions are updated.

Each new field enters into an option (LHDHKS, LHDMCI, LHDENT), and is a variable, a flux or a tendency. In the last two instances, the field may be either of diabatic origin or coming frm physical parametrization. Lastly, it can be a soil field. To identify these properties determines the permanent pointers which must be modified, followed by the sub programmes on which to intervene as well as the location of these sub programmes.

1. update of permanent pointer(s). In SUNDDH, increment the permanent pointer which corresponds to the option and to the category of the new field(s) (see page 54).

2. Instruction update. Add the field description in YOMTDDH.

3. Compute and store the field in PDHCV or PDHCS in the sub programme CPDYDDH or CPPHDDH.

4. Add the field(s) to the output file (PPFIDH).

5. Add the field(s) to the printed budget, in PPEDDH.

6.4 Maintenance specificities, AROME model

6.4.1 New module YOMPHFT

The procedure for adding a new term in DDH in ARPEGE/ALADIN has been described above. Namely, for each new term one line of program code must be added in CPDYDDH or in CPPHDDH and two lines in PPFIDH. In AROME many new fluxes/tendencies are introduced, and many old ones are not used. Adding new lines for each new term would make a quite unreadable code. For this reason and to avoid carrying 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 the corresponding flux/tendency in APFT.

In APFT, for each variable the values at initial time, current output time and terms appearing in conservation equation are written one after another. The order in which variables enter array APFT is given in 6.4.7.3. The number of variables is given by:

```
INTEGER(KIND=JPIM) :: NPROGVAR
```

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

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

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

```
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). The number of CDPI fluxes and their positions in APFT (third index) are defined with the 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)  :: NQI0      ! positions of Pi - flux of cloud ice
INTEGER(KIND=JPIM)  :: NQS0      ! positions of Ps - flux of snow
INTEGER(KIND=JPIM)  :: NQG0      ! positions of Pg - flux of graupel
```

Each physical parameterizations can be deactivated; this therefore changes the number of physical fluxes. The position of fluxes in APFT is then changed and for some fluxes the position in APFT array must be known for programming reasons. The position of these fluxes are given by the 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
  ```

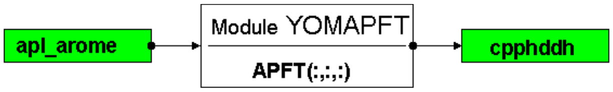


Figure 6.1: Data flow for DDH in ARPEGE/ALADIN.

6.4.2 Data flow

Due to the use of the 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_AROME and used in subroutine CPPHDDH. This is shown in Fig. 6.1.

6.4.3 Setup

To take into account the changes made in AROME to implement DDH, some changes in setting up the subroutines where needed.

6.4.3.1 Subroutine SU0YOMA

In SU0PHY new control variables LPHCDPI and NPHY are initialized.
In SUNDDH new control variables are initialized (see 6.4.5.8).

6.4.3.2 Subroutine SU0YOMB

In SUPHMNH, called in SUPHY, control variable LAROBU_ENABLE is initialized. Subroutines ARO_SUBUDGET and AROINI_BUDGET are called to initialize budget arrays and to save 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 6.4.7.3).

6.4.4 Entering physical fluxes and tendencies in DDH

Most of physical parameterizations in AROME are calculated in MesoNH subroutines. The general layout of implementing in AROME-MesoNH physical subroutines the extraction of tendencies, is shown in Fig. 6.2. Each MesoNH subroutine is called from an interface subroutine which itself is called in APL_AROME. Within the MesoNH subroutine a subroutine BUDGET is called to save physical tendencies in so called budget arrays. These arrays cannot be reached from APL_AROME, but can be reached using an interface subroutine, called AROEND_BUDGET, which is called at the end of APL_AROME. It puts physical fluxes or tendencies in the array APFT, which is passed to it as an actual argument.

6.4.5 Modified subroutines

6.4.5.1 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, the following two lines are added at the end of the radiation parameterizations: ZFRSO and ZFRTH

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

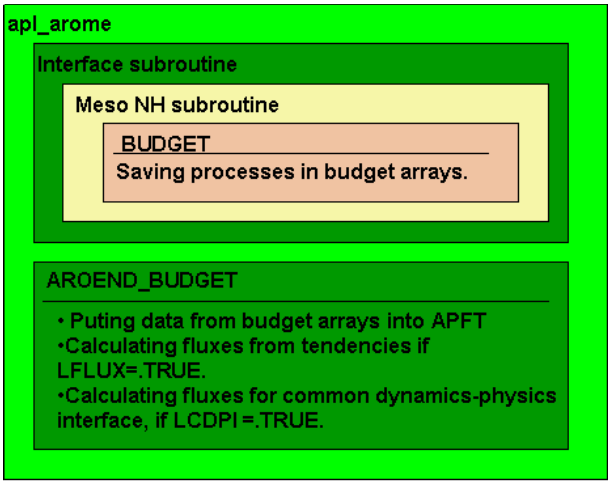

apl_arome
Interface subroutine
Meso NH subroutine
BUDGET
Saving processes in budget arrays.
AROEND_BUDGET
• Puting data from budget arrays into APFT
•Calculating fluxes from tendencies if LFLUX=.TRUE.
•Calculating fluxes for common dynamics-physics interface, if LCDPI =.TRUE.

Figure 6.2: Getting physical tendencies from MesoMN subroutine and saving in APFT.

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.

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

6.4.5.2 Subroutine AROEND_BUDGET

AROEND_BUDGET is an interface subroutine between the ARPEGE/ALADIN code and the MNH code. Its purpose is to save physical fluxes/tendencies in the 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 (LDAROCDPI.OR.LDHDCDPI) is .TRUE., CDPI fluxes and pseudo-fluxes are calculated and saved in APFT.

6.4.5.3 Subroutine CPDYDDH

Dynamical fluxes and tendencies were not changed for AROME only new variables are added. The difference in number and nature of variables in ARPEGE/ALADIN and AROME is treated by the SELECT CASE statement as follow:

```
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
```

6.4.5.4 Subroutine CPG_DIA

A call to subroutine ARO_CPPHDDH (see 6.4.7.1) is added. It is called, should the AROME physics be used. A new argument, PGPAR is added to get the surface temperature and humidity (see 6.4.5.5) for AROME.

6.4.5.5 Subroutine CPG

ZGPAR is added in call to CPG_DIA.

6.4.5.6 SU0PHY

Three new variables used for the control are initialized: LPHCDPI, NPHY and LBUFLUX.

6.4.5.7 Subroutine SUPHMNH

This subroutine is used in AROME for the setup of the 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 to be used, if LAROBU_ENABLE is .TRUE..

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

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

```
CALL ARO_INIAPFT(LAROBU_ENABLE)
```

6.4.5.8 Subroutine SUNDDH

This subroutine must be changed when new terms are entered in DDH. As the namelist NAMDDH is read in this subroutine, we setup the default values to three 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..

6.4.5.9 Subroutine PPFIDH

New variables are added as described in **??**, 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 on 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 does not have to be changed when new terms are introduced in the equations.

6.4.6 Modified modules

6.4.6.1 Module YOMARPHY

Outputs of physical parameterizations are supposed to be fluxes in ARPEGE/ALADIN, whereas in MNH they are always tendencies. To give a choice of either fluxes or tendencies a new switch is introduced. For details see **??**.

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

6.4.6.2 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.
- Two switches are connected to the possibility of having microphysical fluxes/tendencies in a form needed for CDPI (see **??**). It is possible to save microphysical fluxes/tendencies as they are calculated in the physical package, as in CDPI, or both (see **??**).

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

6.4.6.3 Module YOMMNH

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

```
LOGICAL :: LAROBU_ENABLE ! when it is .TRUE. tendencies are saved in budget arr
```

(see **??**)

```
! 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
```

6.4.6.4 Module YOMPHY

Two new control variables and three parameters are added in YOMPHY. LPHCDPI controls calculation of CDPI fluxes/tendencies. In various parts of the code there was a need to branch off because of possibility to use several physical packages. The best way to do it is by SELECT CASE statement. For this reason, a new variable NPHY is introduced. It is a rank of physical packages used in the model and it can take values defined with three new parameters each for one physical package of those used in ARPEGE/ALADIN/AROME. When a 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
```

6.4.7 New subroutines

6.4.7.1 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 to save physical fluxes and tendencies in PDHCV. For the time being, only fluxes are saved, and it 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
```

6.4.7.2 Subroutine ADDFT

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

6.4.7.3 Subroutine ARO_INIPFT

In this subroutine, memory is allocated for all allocatable arrays from module YOMAPFT (6.4.1) and values are assigned to all variables and arrays from YOMAPFT. Descriptions in YAPFT are sorted out by variable. The order of variables is as follow:

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 the execution of processes. The process in MesoNH means anything between: physical process, parameterisation, dynamical terms and all other terms in equations even some mathematical procedures like filtering.

The 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, and for each variable, first and last indexes in YAPFT 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 a MODULE interface is used, the positions of CPDI fluxes are determined.

6.4.8 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 (**??**). 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 still be done in SUNDDH. The procedure is as follow:

A. Adding a variable, dynamical flux or tendency or physical tendency: the same procedure as in **??**

B. Adding a physical flux

1. Do changes in subroutine SUNDDH (see 6.4.5.8),
2. Add a new term in subroutine ARO_INIPFT. First NAPHFT and NDDHFT must be changed. Then, the description of the new term must be put at the right place in YAPFT (see 6.4.7.3). Besides, changes in NBUPROC, NJBUDG1 and NJBUDG2 may be needed.

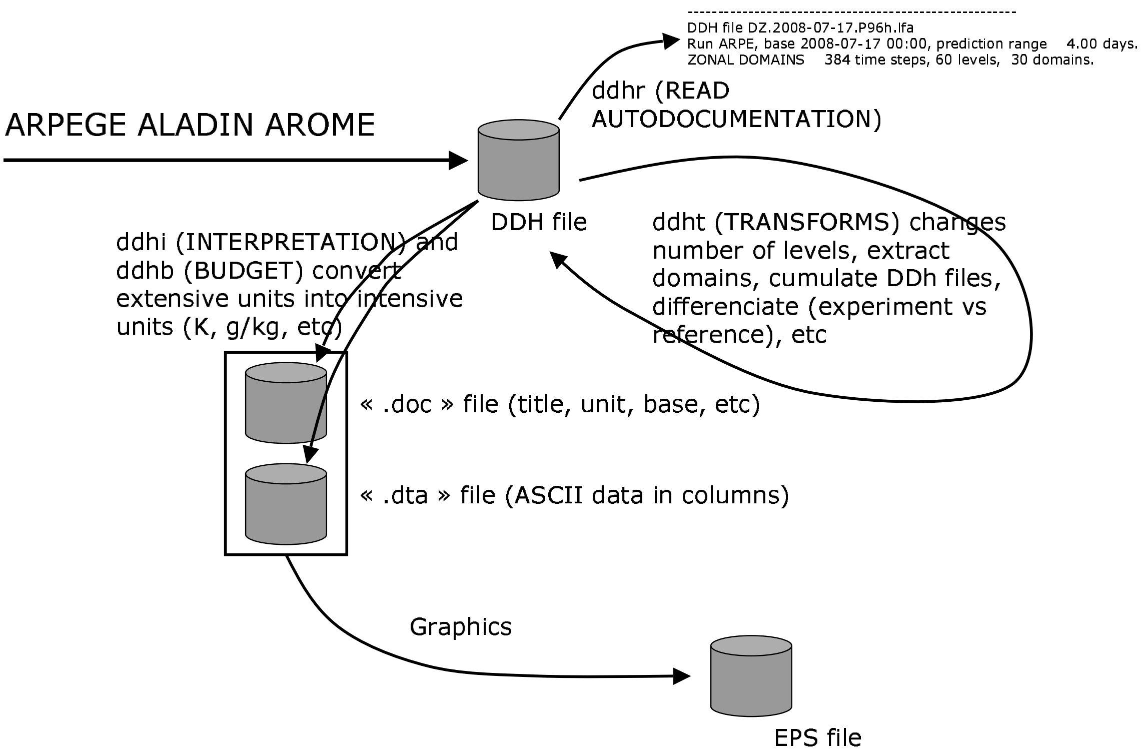
Chapter 7

Using DDH files: the ddhtoolbox

7.1 Purpose

ARPEGE, ALADIN and AROME models produce DDH files. The ddhtoolbox makes operations relevant to use these DDH files for scientifical development and research: produce ready-to-plot profiles of variables, tendencies and fluxes (ddhi), cumulate DDH files, differenciate DDH files, make horizontal and vertical means (ddht), get the budget of prognostic variables (ddhb), etc.

7.2 Synoptic view of ddhtoolbox utilities



```
-------------------------------------------------------
DDH file DZ.2008-07-17.P96h.lfa
Run ARPE, base 2008-07-17 00:00, prediction range    4.00 days.
ZONAL DOMAINS     384 time steps, 60 levels,  30 domains.
```

ddhr (READ AUTODOCUMENTATION)
ARPEGE ALADIN AROME
DDH file
ddht (TRANSFORMS) changes number of levels, extract domains, cumulate DDh files, differenciate (experiment vs reference), etc
ddhi (INTERPRETATION) and ddhb (BUDGET) convert extensive units into intensive units (K, g/kg, etc)
« .doc » file (title, unit, base, etc)
« .dta » file (ASCII data in columns)
Graphics
EPS file

ARPEGE, ALADIN or AROME models produce DDH files. Each file contains n domains, m levels, and for each domain-level all variables, fluxes and tendencies relevant to get a budget. The actions that can be performed on DDH files are

- **ddhr:** get some DDH file autodocumentation on standard output: base, prediction range, etc.
- **ddht:** operate transforms on DDH files: make the difference between a reference and an experiment, cumulate several DDH files, extract one or more domain(s) from a DDH file, extract one or more level(s), make an horizontal mean on all domains, make a vertical mean on all levels, etc. ddht generates in output a DDH file.
- **ddhi:** makes an interpretation of the raw data from a DDH file, in order to get ready to plot data, with intensive units (K/day, g/kg, etc). ddhi produces in output ASCII files:
  - a ".doc" file per field, containing autodocumentation (title, unit, base, prediction range, etc)
  - a ".dta" file per field, containing the data in columns.
- **ddhb** is a script to make the budget of prognostic variables.
- Typing "ddhr" or "ddhi" or "ddht" or "ddhb" with no argument gives a documentation about the usage.
- Graphics: the user may then use his own graphic tool to plot the ".doc" and ".dta" files.

7.3 Install the software

Questions: Mailto: Jean-Marcel.Piriou@meteo.fr

Untar the ddhtoolbox.tar file.

cd ddhtoolbox

The install process uses the "uname -a" command to recognize the architecture of the current machine. So first type "uname -a" on the command line. Then check whether this is an already known type in both scripts install and lfa/install. If yes, and if the compiler option fits your needs, no change needs to be done. Else case, you will have to add an item in the "if ["$os_name"] else fi" statement, to give your compiler options.

1. Put the local directory in your PATH: export PATH=.:$PATH
2. Run install process:

   install clean

   install
3. An additional information: if you are running an ARPEGE - ALADIN - AROME code version earlier than cycle 32, you will need to convert the DDH files produced by ARPEGE - ALADIN - AROME, before using it with the ddhtoolbox utilities.

   This converter, DDHC, is already available on your computer: it is an entry point from the XRD library (libxrd.a) that is generated in the standard ARPEGE - ALADIN - AROME compilation process.

   The DDH files are converted this way (lets call DDH.lfi the file produced by ALADIN, and DDH.lfa the converted one):

   DDHC DDH.lfi DDH.lfa

7.4 Environment variables

1. Put the ddhtoolbox directory and the ddhtoolbox/lfa directory in your PATH, in order to access the ddhtoolbox executables from any directory on your computer.

2. DDH tools (ddhi, ddhb) use the following environment variables, to be put in your ".profile" or ".bash_profile" files:

   export DDHI_LIST=..../ddhtoolbox/ddh_budget_lists/conversion_list

   export DDHB_BPS=..../ddhtoolbox/ddh_budget_lists

   where "...." is to be filled by the absolute PATH of the ddhtoolbox directory.

3. If the environment variable DDH_PLOT is set, somme ddhtoolbox utilities call a script of your own, which makes the plot, starting from the information given by the autodocumentation file (suffix ".doc").

   Example: in Toulouse one uses a script to visualize, called dd2met (written by Yves Bouteloup) based on METVIEW to make the plots. The ".profile" or ".bash_profile" thus contains: "export DDH_PLOT=dd2met"

7.5 ddhb: get budgets

ddhb is a tool to make the budget of pronostic variables, starting from a DDH file. This file should contain only one domain. If it is not the case, use ddht to extract a single domain from your multi-domain DDH file.

Example of ddhb use:

```
ddhb -v arpege/QV -i DHFDLALAD+0036
```

In this example, one asks for the budget of the file DHFDLALAD+0036, this file has been produced by the arpege (or aladin) physics, the prognostic variable is QV (water vapour).

Typing "ddhb" with no argument gives a documentation about the ddhb use.

The ddhb script generates ASCII ready to plot files, and writes out on standard output the location of these files. Two kinds of files are produced:

1. **Data** files (suffix: dta): in ASCII colums.

2. **Autodocumentation** files (suffix: doc): title, unit, date of the prediction run, etc.

What ddhb basically does, as one types -v arpege/QV, is to read the $DDHB_BPS/arpege/QV.fbl file to get the directives to establish the budget of this variable. All information is in this file, and this file can be edited and/or modified by the user, for his own use.

```
find $DDHB_BPS -name "*.fbl" -print
```

to see the complete list of physics or budget packages available for use in ddhb, and which variables. The user may read these directive files to see what DDH file articles are used for each prognostic variable budget.

The user may also create his own directive files. For example, to change the legends of the budgets (and translate them to French, German, etc), or to customize the scientifical budgets (change the list of file articles to be used for a given physical process). In this case, copy an existing $DDHB_BPS/arpege/*.fbl file into your directory, and modify it. This file can then be used by ddhb, in typing

ddhb -d <BPSdirectory> -v <variable> -i <inputfile>

Example: ddhb -d /home/piriou/my_ddh_modifs -v arpege/QV -i dhfzoarpe+0096.mh

where your modified directive file is home/piriou/my_ddh_modifs/arpege/QV.fbl

Chapter 8

History

1991: Initial analysis and coding of the DDH software (Alain Joly).

1992: Introduction of entropy and kinetic energy budgets (Martin Janousek, Jean-Marcel Piriou).

1993: Introduction of relative humidity, liquid and ice water (Jean-Marcel Piriou).

1997: New surface fields, IFS model (Pedro Viterbo).

2000: Diagnose surface "tiles", IFS model (Christian Jakob).

2006: Extract MÉSO-NH physical data flow, interface to DDH routines (Tomislav Kovacic).

2006: Write AROME DDH documentation (Tomislav Kovacic).

2007-11: Rewrite the budget tool "ddhb", still based on the "ddhi" and "ddht" existing ones (Alex Deckmyn, Jean-Marcel Piriou, Tomas Kral).

2008-07: Draft translation of the French DDH documentation into English, by Jean Maziejewski International Sekretarski, approved by Jean-Marcel Piriou.

2008-07: Create the ddhtoolbox, write its documentation (Jean-Marcel Piriou).