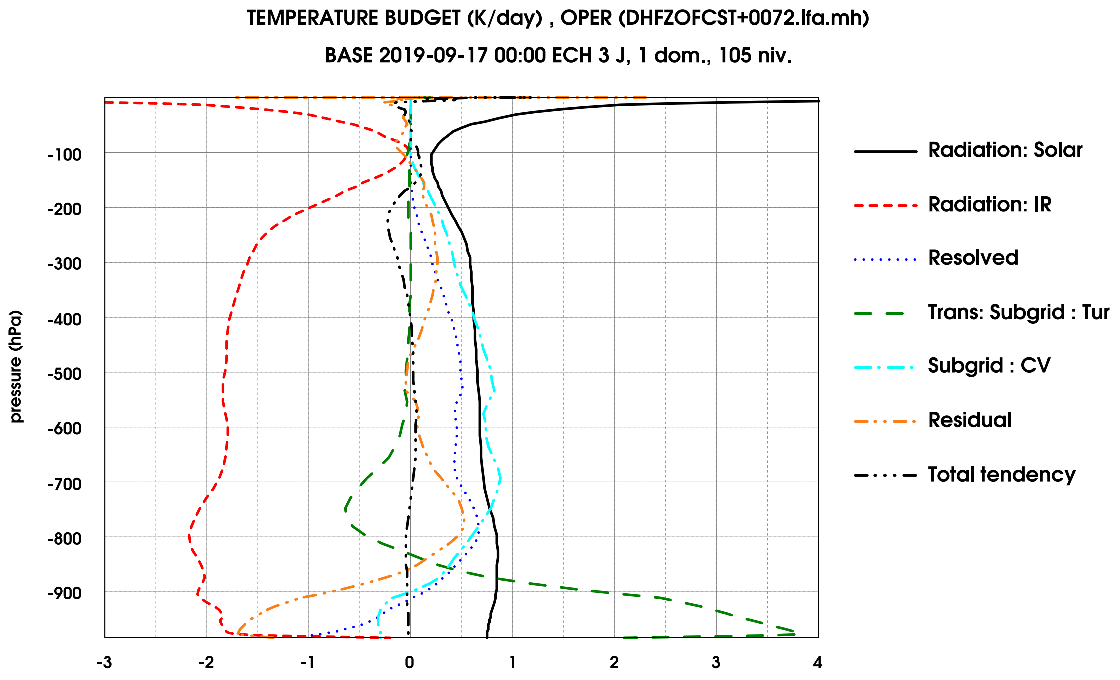
Diagnostics in Horizontal Domains (DDH)

Variables and budget equations, in horizontal mean

ARPEGE, ALADIN and AROME models

Guide for users and developpers

October 1, 2019



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.

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).

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` ($1 \leq \texttt{jkd} \leq \texttt{NDHKD}$).

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:

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

with $\mu_1 = 1$. Therefore

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

The latitude of the zonal band jkd is

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

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.

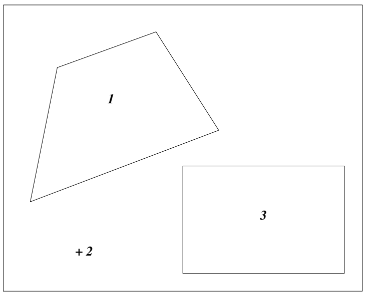


Figure 2.1: Case 1

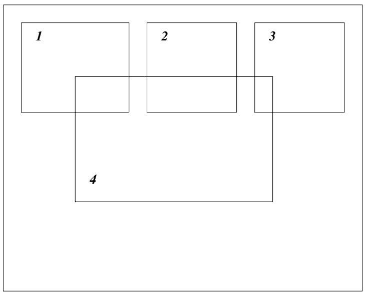


Figure 2.2: Case 2

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).

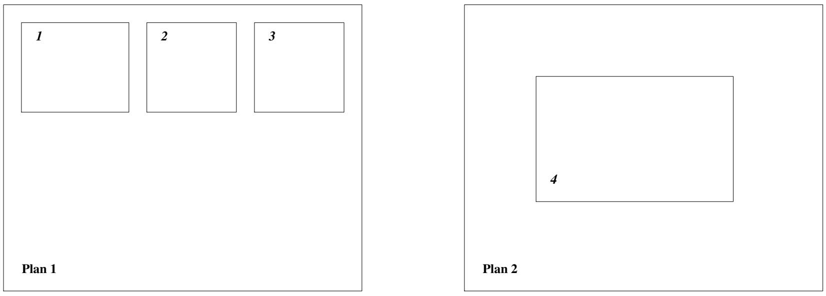


Figure 2.3: Case 2

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.

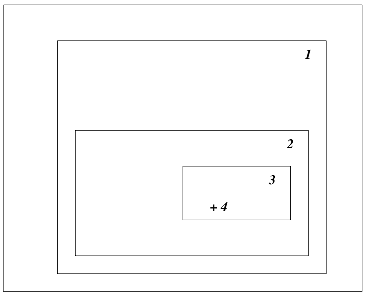


Figure 2.4: Case 3

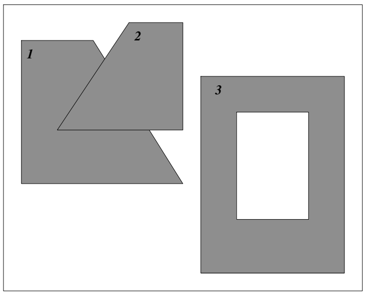


Figure 2.5: Case 4

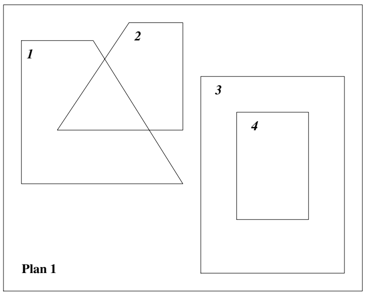


Figure 2.6: Case 4

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, … and not 1, 3, 5, … 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 global diagnostics on file,
- `LHDEFZ:` write zonal bands diagnostics on file,
- `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.

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

$$\boxed{\begin{array}{rcl} \texttt{BDEDDH}(1, \text{domain number}) & = & \text{type} \\ \texttt{BDEDDH}(2, \text{domain number}) & = & \text{virtual plane} \end{array}}$$

For type 1, point given by its indexes

$$\boxed{\begin{array}{rcl} \texttt{BDEDDH}(3, \text{domain number}) & = & \text{rjlon} \\ \texttt{BDEDDH}(4, \text{domain number}) & = & \text{rjgl} \end{array}}$$

For type 2, quadrilateral given by its four corners

$$\boxed{\begin{array}{rcl} \texttt{BDEDDH}(3, \text{domain number}) & = & \text{Longitude of corner \#1, in degrees, } \lambda_1 \\ \texttt{BDEDDH}(4, \text{domain number}) & = & \text{Latitude of corner \#1, in degrees, } \theta_1 \\ (\texttt{BDEDDH}(5,\ \text{-}\ ), \texttt{BDEDDH}(6,\ \text{-}\ )) & = & (\lambda_2, \theta_2) \\ (\texttt{BDEDDH}(7,\ \text{-}\ ), \texttt{BDEDDH}(8,\ \text{-}\ )\ ) & = & (\lambda_3, \theta_3) \\ (\texttt{BDEDDH}(9,\ \text{-}\ ), \texttt{BDEDDH}(10,\ \text{-}\ )\ ) & = & (\lambda_4, \theta_4) \end{array}}$$

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

- For a domain which does not intersect the Greenwich meridian

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

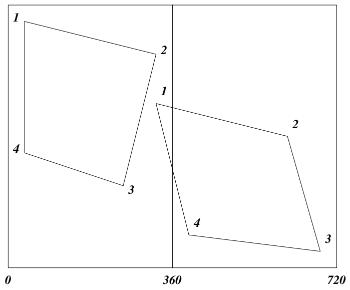


Figure 2.7: Constraints for quadrilateral declarations

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 $(\lambda_g + 2\pi,\, \mu_g)$ 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	$\mu_2 = \mu_1$	$\lambda_3 = \lambda_2$	μ_3	$\lambda_4 = \lambda_1$	$\mu_4 = \mu_3$	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 18.

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 40.

`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`: thermal energy,
`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 $\chi\,\frac{\partial p}{\partial \eta}$, 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

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

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 thermal energy 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 assumptions 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" latent heat, or difference in enthalpy due to 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

$$\vec{v}\,\frac{\partial F_{\varphi\vec{v}}^{tur+conv}}{\partial\eta}$$

or in the entropy budget

$$\frac{1}{T}\,\frac{\partial F_{\varphi T}^{ray}}{\partial\eta}$$

.

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:

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

now with

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

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

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

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

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

is integrated by the diagnostics `DDH` as follow

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

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

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

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

and the vertical mean budget

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

Chapter 4

Budget and diagnostics, ARPEGE model

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

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

File output

`VPP0`	$\frac{1}{g} \delta p$ (t=0)	
`VPP1`	$\frac{1}{g} \delta p$ (t=NSTEP δt)	
`TPPDIVFLUHOR`	$-\frac{\delta t}{g} \operatorname{div}_\eta (\vec{v}\, \delta p)$	cumulated
`FPPFLUVERTDYN`	$\frac{\delta t}{g} \dot{\eta} \frac{\partial p}{\partial \eta}$	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.
- $r_\eta = -\frac{1}{g}\frac{\partial p}{\partial \eta}$.
- $F_{q_v} = F_c^{conv-l} + F_c^{conv-n} + F_c^{stra-l} + F_c^{stra-n} + F^{tur}_{q_v} + F^{tur-conv}_{q_v}$.
- $F_{q_l} = F_p^{conv-l} + F_p^{stra-l} - F_c^{conv-l} - F_c^{stra-l} + F^{tur}_{q_l} + F^{tur-conv}_{q_l}$.
- $F_{q_n} = F_p^{conv-n} + F_p^{stra-n} - F_c^{conv-n} - F_c^{stra-n} + F^{tur}_{q_n} + F^{tur-conv}_{q_n}$.
- $F^{tur}_{q_\psi}$ 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`	$-\frac{\delta t}{g}u\,\mathrm{div}_\eta\left(\delta p\vec{v}\right)$	cumulated
`TVVDIVFLUHOR`	$-\frac{\delta t}{g}v\,\mathrm{div}_\eta\left(\delta p\vec{v}\right)$	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

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

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

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 miss
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 Thermal energy budget

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

Lagrangian equation in T

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

Eulerian equation in $s = c_pT + \Phi + \frac{u^2+v^2}{2}$

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

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

Budget equation

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

where

- $r_\eta = -\frac{1}{g}\frac{\partial p}{\partial \eta}$.
- $F_{c_pT} = F_{c_pT}^{sol} + F_{c_pT}^{ther} + F_{c_pT}^{meso} + F_s^{tur} + F_s^{tur-conv}$.
- $F_s^{tur-conv}$ is the subgrid-scale transport of dry static energy $s = c_pT + \Phi$, due to deep convection.

- $F_{c_pT_{prec}} = F^{l}_{c_pT_{prec}} + F^{n}_{c_pT_{prec}}$.
- $F^{l}_{c_pT_{prec}} = F^{conv-l}_{c_pT_{prec}} + F^{stra-l}_{c_pT_{prec}}$.
- $F^{n}_{c_pT_{prec}} = F^{conv-n}_{c_pT_{prec}} + F^{stra-n}_{c_pT_{prec}}$.
- $F^{conv-l}_{c_pT_{prec}} = -\left[L_{v>l}(T) + (c_l - c_{p_v})T\right] F^{conv-l}_{c}$.
- $F^{conv-n}_{c_pT_{prec}} = -\left[L_{v>n}(T) + (c_n - c_{p_v})T\right] F^{conv-n}_{c}$.
- $F^{stra-l}_{c_pT_{prec}} = -\left[L_{v>l}(T) + (c_l - c_{p_v})T\right] F^{stra-l}_{c}$.
- $F^{stra-n}_{c_pT_{prec}} = -\left[L_{v>n}(T) + (c_n - c_{p_v})T\right] F^{stra-n}_{c}$.
- $F^{sol}_{c_pT}$ et $F^{ther}_{c_pT}$ are the solar and infra-red fluxes.
- $\vec{F}^{phys}_{v} = \vec{F}^{tur}_{v} + \vec{F}^{tur-conv}_{v} + \vec{F}^{rel}_{v} + \vec{F}^{meso}_{v}$.
- $c_p = c_{p_a}q_a + c_{p_v}q_v + c_lq_l + c_nq_n$.

File output

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

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

where

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

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`	$-\frac{\delta t}{g}\{s\,\mathrm{div}_\eta\,(\vec{v}\delta p)$	
	$+\delta p[(s_v - s_a)\vec{v}\cdot\vec{\nabla}q_v$	
	$+\frac{1}{T}c_p\vec{v}\cdot\vec{\nabla}T$	
	$-R\,\texttt{ZRTGR}\,\vec{v}\cdot\vec{\nabla}\pi]\}$	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`	$-\frac{\delta t}{T}(\delta F_{cpT}^{sol} + F_{cpT}^{ther})$	cumulated
`TSSTURCTTOT`	$-\frac{\delta t}{T}(\delta F_{cpT}^{tur} + F_{cpT}^{tur-conv})$	cumulated

4.8 Potential energy budget

Budget equation

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

$$-\frac{1}{g}\frac{\partial}{\partial\eta}\left[\Phi\left(\frac{\partial p}{\partial t} + \dot{\eta}\,\frac{\partial p}{\partial\eta} + \delta_m gF_p\right)\right]$$

File output

Three of these budget terms have already been mentioned, see «kinetic energy budget» et «thermal energy 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:

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

  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 `Sxx_0`

$\frac{1}{S_{\mathcal{D}}}\sum_{(j,k)\in\mathcal{D}} \mathtt{PITM}\,\sigma_{j,k}$	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 `Sxx_1`

  The same articles as above.

- Flux cumulated from 0 to NSTEP: articles `G01, ..., G17`

G01	$F_{cpT}{}^{sol}_{\tilde{L}}$	Solar radiation
G02	$F_{cpT}{}^{ther}_{\tilde{L}}$	Infra-red radiation
G03	$L_v(T_s)E_l$	Latent heat (water)
G04	$L_n(T_s)E_n$	Latent heat (snow)
G05	F_{csa}	Sensible heat at surface
G06	F_{csp}	Sensible heat between surface and deep layer
G07	$L_{fonte}F_{fonte}$	Surface flux due to melting Flux de chaleur en surface lié à la fonte de neige
G08	F_p^{stra-l}	Precipitation: resolved, rain
G09	F_p^{stra-n}	Precipitation: resolved, snow
G10	F_p^{conv-l}	Precipitation: subgrid-scale, rain
G11	F_p^{conv-n}	Precipitation: subgrid-scale, snow
G12	E_l	Surface evaporation (water)
G13	E_n	Surface evaporation (snow)
G14	F_{perco}	Percolation from surface to deep
G15	F_{fonte}	Snow melting
G16	F_{ruiss}	Surface run-off
G17	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`.

4.10.5 Free style variables

`SVGFS01`	$2\,m$ temperature
`SVGFS02`	specific water vapour at $2\,m$ temperature
`SVGFS03`	u at $2\,m$ level
`SVGFS04`	v at $2\,m$ level
`SVGFS05`	$g\,z$ orography
`SVGFS06`	$g\,z_0$, where z_0 is the dynamical roughness
`SVGFS07`	$g\,z_0$, where z_0 is the thermal roughness
`SVGFS08`	albedo
`SVGFS09`	boundary layer height

4.10.6 Free style fluxes

`SFGFS01`	downward solar radiation
`SFGFS02`	downward thermal radiation

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 and diagnostics, AROME model

5.1 Balance equations

5.1.1 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

5.1.2 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

5.1.3 Thermal energy

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

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

5.1.4 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

5.1.5 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

5.1.6 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

5.1.7 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

5.1.8 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

5.1.9 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

5.2 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

Using DDH files: the ddhtoolbox

6.1 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.

6.2 Install the software

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

Untar the ddhtoolbox.tar file.

cd ddhtoolbox/tools

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
```

6.3 Environment variables

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

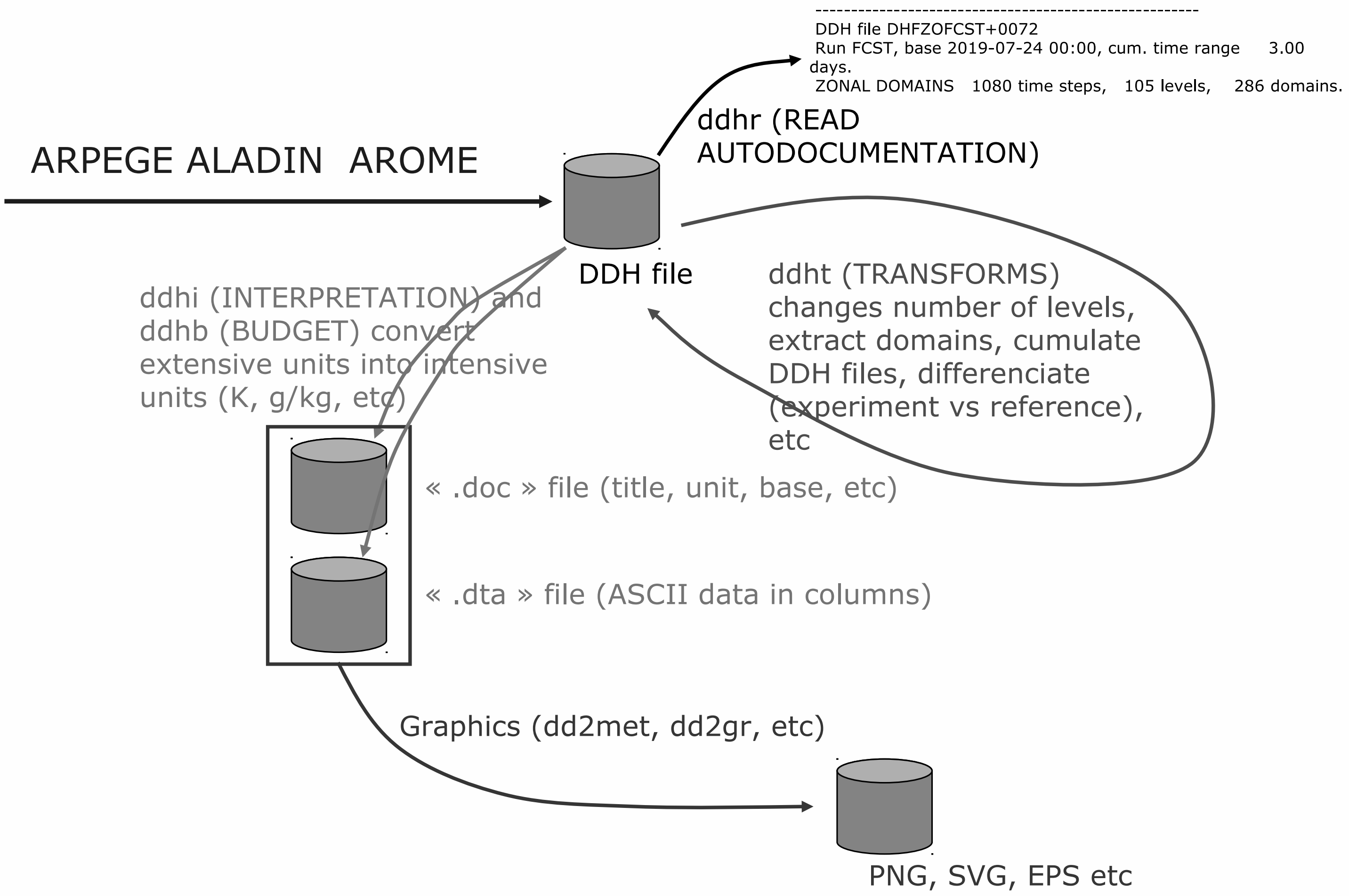
2. DDH tools (ddhi, ddhb) use the following environment variables, to be put in your ".profile" or ".bash_profile" files:

   ```
   export DDHTOOLBOX= the absolute PATH of the above ddhtoolbox directory

   export DDHI_LIST=$DDHTOOLBOX/ddh_budget_lists/conversion_list

   export DDHB_BPS=$DDHTOOLBOX/ddh_budget_lists
   ```

3. If the environment variable DDH_PLOT is set, some 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 the graphics can be done by dd2gr and visualisation by eog (or firefox), setting the environment variables in ".profile" or ".bash_profile" files: "export DDH_PLOT=dd2gr ; export VISU_G=eog". The graphics can also be done by dd2met (written by Yves Bouteloup, based on METVIEW) setting "export DDH_PLOT=dd2met".

6.4 Synoptic view of ddhtoolbox utilities



```
------------------------------------------------------
 DDH file DHFZOFCST+0072
 Run FCST, base 2019-07-24 00:00, cum. time range     3.00
days.
 ZONAL DOMAINS   1080 time steps,   105 levels,    286 domains.
```

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 described in the followinf sections.

6.4.1 ddhr: get autodocumentation

Get some DDH file autodocumentation on standard output: base, prediction range, etc.

Typing "ddhr" with no argument gives a documentation about the usage.

6.4.2 ddht: transform

Operates 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.

Typing "ddht" with no argument gives a documentation about the usage.

6.4.2.1 Difference experiment minus reference

```
=> ddht -cDIFFE_EXP_REFE -2FEXP -1FREF -sDIFF
```

makes the difference between DDH file FEXP and FREF, the result is DIFF file. The DIFF file is a DDH file. The prediction range of FEXP and FREF have to be equal. If they differ more than 0.001%, ddht aborts.

The script ddh- makes the same operation, with a shorter command line:

```
=> ddh- FEXP FREF DIFF
```

The ddh- script calls "ddht -cDIFFE_EXP_REFE".

It may be useful, in some situations, to make the difference between 2 files having different prediction ranges: for example, to compare the mean infra-red cooling from a 24h prediction with a 6h prediction, to study spin-up effects. The script ddh- makes such a difference: if the 2 prediction ranges are different, ddh- modifies the prediction range from one file, modifies all fluxes and tendencies accordingly (done through the ddhmech script), and then makes the "ddht -cDIFFE_EXP_REFE" difference.

6.4.3 ddhi: interpretation

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).

Typing "ddhi" with no argument gives a documentation about the usage.

Example: starting from a DDH file "DHFDLFCST+0024.domaine4", one needs to get an ASCII file containing the profile of water vapour q_v and temperature T. Create an ASCII file containing the list of articles:

```
lxgmap2:/home/piriou/ftn/ddh/ddhtoolbox/ddh_files/arome/cy35t1_arome_france_c744 => cat mylist
VQV1
VCT1
```

then type "ddhi DHFDLFCST+0024.domaine4 -lmylist":

```
lxgmap2:/home/piriou/ftn/ddh/ddhtoolbox/ddh_files/arome/cy35t1_arome_france_c744 => ddhi DHFDLFCST+0024.domaine4 -lmylist
default list file:
/home/piriou/ftn/ddh/ddhtoolbox/ddh_budget_lists/conversion_list
----------------------
---DDHI-CHAMPS--------
----------------------
Fichier d'entree: DHFDLFCST+0024.domaine4
calling lisc
lisc:/home/piriou/ftn/ddh/ddhtoolbox/ddh_budget_lists/conversion_list
lisc: read           413  fields.
----------------------
---DDHI-COORDONNEES---
----------------------
DHFDLFCST+0024.domaine4.tmp.VCT1.dta
DHFDLFCST+0024.domaine4.tmp.VCT1.doc
DHFDLFCST+0024.domaine4.tmp.VQV1.dta
DHFDLFCST+0024.domaine4.tmp.VQV1.doc
```

ddhi reads the DDH file, reads the VCT1 and VQV1 articles, converts the units of these data: for example the VCT1 article is $c_p\, T\, \frac{\delta p}{g}$, ddhi converts it into T, and thus divides by c_p and by the $\frac{\delta p}{g}$ field. The conversion uses a conversion list file (whose name is given to ddhi by the DDHI_LIST environment variable), which tells ddhi how to convert each DDH article.

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.

To know the complete list of variables, tendencies and fluxes that can be tranformed into ASCII data by ddhi, type "lfaminm FDDH", where FDDH is the name of the DDH file. This gives the list of all articles. The variables are article names beginning with "V", the tendencies are article names beginning with "T", the fluxes are article names beginning with "F". These article names can be put the "mylist" file as described above. If one of these articles is not present in the conversion list file DDHI_LIST, ddhi will not know how to convert it. In this case, simply edit the DDHI_LIST file, and add a line describing how this article has to be converted.

6.4.4 ddhb: budgets of prognostic variables

ddhb is a tool to make the budget of pronostic variables, starting from a DDH file. Typing "ddhb" with no argument gives a documentation about the ddhb use.

This DDH 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.

6.4.4.1 Get a first budget plot

Example of ddhb use:

"ddhb -v QV -i DHFDLALAD+0036".

In this example, one asks for the budget of the prognostic variable is QV (water vapour), from the file DHFDLALAD+0036. What ddhb basically does, as one types -v QV, is to read in the DDH file the list of articles containing fluxes or tendencies of QV: all articles "FQV*" and "TQV*" of the DDH file are used to build the QV budget.

The ddhb script then writes ASCII ready to plot 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.

6.4.4.2 More advanced use to get budget plots

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, two methods:

1. **Create your own directive file, "from scratch":** "ddh2fbl FDDH DIR" will read the FDDH file, and produce the directive files on the $DDHB_BPS/DIR directory. Then, one simply needs to type

   "ddhb -v DIR/VAR -i FDDH" to get the budget of the variable VAR. If one wants to modify the legends of the curves, one edits the $DDHB_BPS/DIR/VAR.fbl file, and then reruns ddhb.

   How does ddh2fbl work? It reads inside the DDH file all article names, lists the articles of the type V??0 (examples: ??=CT, QV, etc). For each of these variables, lists all F??* and T??* articles. And writes the directive file containing this list. The resulting list is thus consistent with the DDH file. ddh2fbl makes the assumption that all budget items are articles beginning with F or T. This rule, presently true in ARPEGE - ALADIN - AROME, should thus be respected in the future to guarantee a proper work of ddh2fbl.

2. **Modify existing reference directive files:**

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

   to see the complete list of physics or budget packages available for use in ddhb, and which variables. Copy a directory containing directive files under your own name, and then modify it. Example:

```
lxgmap2: => find $DDHB_BPS -name "*.fbl" -print
/home/piriou/ftn/ddh/ddhtoolbox/ddh_budget_lists/arome_cy35t1/QG.fbl
/home/piriou/ftn/ddh/ddhtoolbox/ddh_budget_lists/arome_cy35t1/QR.fbl
/home/piriou/ftn/ddh/ddhtoolbox/ddh_budget_lists/arome_cy35t1/QL.fbl
/home/piriou/ftn/ddh/ddhtoolbox/ddh_budget_lists/arome_cy35t1/QI.fbl
/home/piriou/ftn/ddh/ddhtoolbox/ddh_budget_lists/arome_cy35t1/QS.fbl
/home/piriou/ftn/ddh/ddhtoolbox/ddh_budget_lists/arome_cy35t1/CT.fbl
/home/piriou/ftn/ddh/ddhtoolbox/ddh_budget_lists/arome_cy35t1/QV.fbl
/home/piriou/ftn/ddh/ddhtoolbox/ddh_budget_lists/arome_cy35t1/TE.fbl
/home/piriou/ftn/ddh/ddhtoolbox/ddh_budget_lists/arpege/KK.fbl
/home/piriou/ftn/ddh/ddhtoolbox/ddh_budget_lists/arpege/QT_old.fbl
/home/piriou/ftn/ddh/ddhtoolbox/ddh_budget_lists/arpege/QT.fbl
/home/piriou/ftn/ddh/ddhtoolbox/ddh_budget_lists/arpege/CT_simplified.fbl
/home/piriou/ftn/ddh/ddhtoolbox/ddh_budget_lists/arpege/QV_2006-06_and_previous.fbl
/home/piriou/ftn/ddh/ddhtoolbox/ddh_budget_lists/arpege/QV_simplified.fbl
/home/piriou/ftn/ddh/ddhtoolbox/ddh_budget_lists/arpege/CT.fbl
/home/piriou/ftn/ddh/ddhtoolbox/ddh_budget_lists/arpege/QV.fbl
lxgmap2: => cp -r $DDHB_BPS/arpege/ $DDHB_BPS/myown_arpege
lxgmap2: => vi $DDHB_BPS/myown_arpege/CT.fbl
lxgmap2: => ddhb -v myown_arpege/CT -i DHFDLALAD+0036
```

6.5 Graphics

All above mentioned tools (ddhi, ddhr, ddhb, etc) produce read-to-plot ASCII files. The user may then use his own graphic tool to plot these ".doc" and ".dta" files. As mentioned in page 52, if the environment variable DDH_PLOT is set, some ddhtoolbox utilities call a script of your own, which makes the plot, starting from the information given by the autodocumentation file (suffix ".doc").

Chapter 7

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.

7.1 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`.

7.1.1 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/`.

7.1.2 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 in`CPDYDDH` and `CPPHDDH`, and to writing lines or other editions in `PPEDDH` and `PPFIDH`. A corresponding commentary is in the general nomenclature of `YOMTDDH`. 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`.

7.1.3 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.

7.1.3.1 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).

7.1.3.2 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.

7.2 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.

7.3 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 $\boxed{\text{tci}}$ 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 58).
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.

7.4 New dataflow for DDH

7.4.1 Introduction

A new coding approach has been proposed in 2009 for extracting diagnostics from the Arome/MesoNH physical parametrisations. It can be used in other parts of the IFS/ARPEGE software. Physical quantities are recorded into a flexible data structure in the parametrisations, and readable by higher level routines. The data structure (a linked list of ad hoc Fortran 90 types) is automatically allocated and indexed as needed by low-level routines, so that physicists can freely choose which quantities they want to record, and how they want to process them. This technical approach greatly simplifies software clarity and maintenance.

Main applications are (1) to provide an easy access to various Arome/MesoNH physical quantities at the level of the physics calling interface and (2) to replace existing DDH in Arpege/Aladin/Alaro if satisfying results are obtained after intensive testing.

7.4.2 Achievements-Future developments

The software is developped progressively and is expected to replace the existing DDH dataflow in the different models after a period of testing. User's feedbacks will be very important to trace potential weaknesses of the present code. Here is the timetable of foreseen code evolutions:

- cy35t1: new dataflow available in Arome only for 3D fields. For Arpege/Aladin/Alaro, old DDH structures are kept.
- cy35t2: new dataflow can be used in all models (LFLEXDIA=.TRUE.) but by default old dataflow is used only in Arpege/Aladin/Alaro. 2D fields are available in the new dataflow.
- 2009: intensive testing period whith expected improvements in the code. Renewing of DDH operators for horizontal averaging may be necessary.
- 2010: complete switch to new dataflow ? (Would affect IFS code also...)

7.4.3 General basics of the new dataflow

This section describes the content of file xrd/module/ddh_mix.F90 which contains all the functionalities of the new dataflow. It can be thought as an externalized functionality of the code. New dataflow features are present in the code under the LFLEXDIA switch.

Description

The dataflow consists in self allocatable structures similar to GFL but more flexible. This section describes how they are defined, the possible architecture of the code being discussed in section 7.4.5. Each extracted quantity (variable, flux, tendencies...) will be characterized through a Fortran 90 structure type (named here DDHFLEX) which defines several attributes corresponding to this quantity.

The structure type named DDHFLEX is given here:

```
TYPE DDHFLEX
  CHARACTER(LEN=11)::CNAME !name of field
  CHARACTER(LEN=1)::CFLUX !'F' if flux 'V' if variable 'T' if tendency
  CHARACTER(LEN=3)::CMOD  ! 'ARP','ARO': name of model
  LOGICAL:: LKDDH !TRUE if to be stored into DDH
  ! rfield has to be a pointer because allocatable not allowed in structure ty
  REAL(KIND=JPRB),DIMENSION(:,:),POINTER:: RFIELD  ! value of retrieved field
  INTEGER(KIND=JPIM):: NFIELDIND! position of flux in ddh array
END TYPE DDHFLEX
```

Following attributes are used:

- CNAME is the name of the field as it will appear in the output file. CNAME has to respect the following conventions:
  - First letter has to be either 'F' for a flux, 'V' for a variable or 'T' for a tendency.
  - Second and third letter describes the conservation equation to which the budget applies (see DDH documentation for details): CT (temperature), QV (water vapour), ...
- CFLUX is a sting that informs about the nature of the quantity stored in the structure:
  - CFLUX='F' for a flux
  - CFLUX='T' for a tendency
  - CFLUX='V' for a variable
- CMOD gives information on the model's name
  - CMOD='ARO' for AROME
  - CMOD='ARP' for ARPEGE, ALADIN and ALARO (by default but if you wish other labels can be introduced)
- LKDDH is a flag set to .TRUE. if the field has to be processed by DDH operators and to .FALSE. otherwise.
- RFIELD is a pointer corresponding to the value of the field (it will be explained later why it has to be a pointer)

- NFIELDIND is an integer that gives the number of the processed field within the list of all fields.

These attributes are important because they document the structure content itself (important for debugging purposes) and they determine which operations the extracted field will undergo at the place where it is recorded, before being stored (for instance conversion from potential temperature to temperature...)

The various extracted fields are gathered into an allocatable array of structure of type DDH, called here RDDH_DESCR and whose last dimension corresponds to the total number of extracted fields:

```
TYPE(DDHFLEX),ALLOCATABLE,DIMENSION(:):: RDDH_DESCR
```

The attribute *allocatable* being forbidden inside a type structure, the field is not directly stored inside RDDH_DESCR but defined through a pointer to a large array called RDDH_FIELD:

```
REAL,DIMENSION(:,:,:),ALLOCATABLE,TARGET::RDDH_FIELD ! target of RFIELD
! first two dims are the same as PFIELD, the third being the number of stored
```

Extracting a field from the physics

For adding a field into the diagnostics, you only need to call subroutine ADD_FIELD_3D and that's all ! The first argument of ADD_FIELD_3D will be the field to store and the others will correspond to the associated attributes (for instance "call ADD_FIELD3D(field_to_store,'name_of_field','F','CT'....)")

Arguments of ADD_FIELD_3D(PMAT,CDNAME,CDFLUX,CDMOD,LDINST,LDDH) are the following:

- PMAT: the array to be stored. It has to be with levels in the same order than in Arpege part of the code. If you are in a .mnh subroutine just use subroutine INVERT_VLEV.MNH before calling ADD_FIELD_3D in order to have levels ordered as in Arpege.
- CDNAME: name of field. It is constructed the following way:
    - CDNAME(1): 'F' if flux ,'T' if tendency,'V' if variable
    - CDNAME(1:2): type of variable ('CT','QI','QV','QR',...)
    - CDNAME(3:): name of process
- CDFLUX: 'F' if flux ,'T' if tendency,'V' if variable
- CDMOD: 'ARO' if AROME, 'ARP' otherwise (but you may add some other label if you wish)
- LDINST:'TRUE' if instaneous field
- LDDH:'TRUE' if field is stored to be in DDH

When using add_field_3D it is extremely important to have the right attributes in the right order. So be careful ! Have a look at xrd/module/ddh_mix.F90 if any doubt.

Here are some examples:

```
CALL ADD_FIELD_3D(ZTMPAF,'VQI','V','ARP',.TRUE.,.TRUE.)

CALL ADD_FIELD_3D(ZTMPAF(:,:),CLNAME,'T','ARP',.TRUE.,.TRUE.)

CALL ADD_FIELD_3D(PFRSO(:,:,1),'FCTRAYSO','F','ARP',.TRUE.,.TRUE.)
```

ADD_FIELD performs the following tasks:

- when in the code a specific field is supplied as argument for the first time in the execution, the last dimension of arrays DDH_FIELD and DDH_DESCR is incremented in order to add space for the new field to store. The code determines if a field is encountered for the first time by testing the field's name. This reallocation of arrays may slow the code and fragment memory during the first time step, but it avoids going through complicated setups. One could also preallocate the arrays according to a first guess of the dimensions, as chosen by the user.
- at every time step the field is stored in RDDH_FIELD through the pointer RFIELD
- at every time step, some transformations are done on the field according to its nature (and documented by its attributes), for instance conversion from θ to T... These operations also depend on the physics used (Meso-NH, Arpege...). Here it will be possible for users to add parts corresponding to specific needs, and to document them through attributes.

7.4.4 Activating and modifying the new dataflow

7.4.4.1 Using DDH products included in documentation

The DDH documentation helds as a reference for the formulation of the budget equations and for the list of terms present by default in the DDH files. If you just need these products, just set the DDH namelist according to your need and you just have to plot the ddh files using the ddhtoolbox. In Arome, new dataflow is activated by default. For Arpege, Aladin and Alaro you have to set LFLEXDIA=.TRUE. in namelist in order to use the new dataflow. Otherwise old dataflow is used. We recommand to use the new dataflow since old dataflow is kept for the moment only for compatibility with ECMWF and validation purposes.

7.4.4.2 Adding terms to the already existing DDH products

You just have to call ADD_FIELD_3D (Make sure that you have imported this function by adding in your file USE DDH_MIX,ONLY :ADD_FIELD_3D) If you want to add a term to an existing budget equation, just use the same name for the variable ('CT','QR'...) than in the rest of the code. Otherwise you are free to introduce a new name. If you are in a .mnh subroutine, you have to proceed in two steps:

- First you have to transform your array on NLEV+2 levels to an array on NLEV levels in reverse order (to go from the "MNH" word to "Arpege" word). There is a subroutine dedicated to this transformation INVERT_VLEV.MNH
- Then use ADD_FIELD_3D on the transformed array.

7.4.4.3 Using the dataflow for extracting terms from the physics but not for DDH

It is possible by just setting LDDH to .FALSE. when calling ADD_FIELD_3D to use the flexible dataflow for retrieving fields out from the physics and use them elsewhere. Once the field is stored using ADD_FIELD_3D, you just have to go through the flexible structure once to have the index MYINDEX of your field that you can use later on by accessing RDDH_FIELD(:,:,MYINDEX):

```
DO II=1,NTOTFIELD
  IF (RDDH_DESCR(II)\%CNAME=='MYNAME') THEN
   MYINDEX=RDDH_DESCR(II)\%NFIELDINF
  ENDIF
ENDDO

% your field is stored in RDDH\_FIELD(:,:,MYINDEX)
```

For the time being the previous lines of code are not in the common cycles, if you feel that there should be just send an email to the DDH team.

7.4.4.4 Miscellaneous

If the budget package in Méso-NH is maintained (BUDGET routine) there is nothing to do in the DDH part of the code, except in the following situations:

- **New species are added in Arome.**
  - In this case, a label for it first has to be introduced.
    If this is an hydrometeor you have to add an entry to CLVARNAME in APL_AROME (it corresponds to the names of hydrometeors ordered the same way than in PTENDR) and report it coherently in MODDB_INTBUDGET. Increase also by one dimension of TAB_VARMULT array. **Beware to use the same ordering of variable than in Méso-NH calls to budget !!!** If this is not an hydrometeor, it may not be present in the Méso-NH budgets and thus we recommend to use combination of INVERT_VLEV and ADD_FIELD_3D.
  - The transformation applied to this field has to be defined.
    In ARO_SUINTBUDGET, increase by one the last dimension of TAB_VARMULT and have it pointing on the TCON2 (equal to PQDM) since it is an hydrometeor.
  - In APL_AROME, check that loops on last dimension on PTENDR include this new hydrometer.
  - In CPDYDDH just use ADD_FIELD_3D to add the value of the variable corresponding to your new hydrometeor.

- **Order of subroutines is changed in APL_AROME.** In this case make sure that ARO_STARTBU and ARO_SUNINTBUDGET are called at the right place.

7.4.5 Architecture of the code

Subroutine ADD_FIELD_3D and associated modules are in xrd/module/ddh_mix.F90. This subroutine contains all elements for using the new dataflow.

However the use of the new dataflow in the part of the Arome code originating form Méso-NH required some interfacing described in the following subsection.

7.4.5.1 Calling tree

Example: correction of negative QL values by the AROME physics.

- The correction is done in aro_rain_ice.mnh (in mpa/micro/externals).
- To activate the DDH budget of the QL tendency, due to this correction, aro_rain_ice.mnh calls BUDGET:

```
IF (LBUDGET_RC) CALL BUDGET (PRS(:,:,:,2) * PRHODJ(:,:,:), 7,'NEGA_BU_RRC')
```

- The routine BUDGET (in mpa/micro/internals) used for AROME runs differs from that used for MNH runs. The AROME BUDGET routine converts the MNH name 'NEGA_BU_RRC' into the DDH name 'TQLNEGA', and then calls ADD_FIELD:

```
IF (CLPROC/='INIF') CALL ADD_FIELD_3D(ZVARS,CLDDH,'T','AROME',LINST,LDDH)
```

- ADD_FIELD_3D (in xrd/module/ddh_mix.F90) allocates the relevant arrays if not already allocated, and then writes the real data of the tendency into the RDDH_FIELD array from module ddh_mix.

7.4.5.2 Organization of the data flow

The DDH diagnostic facility performs some domain averaging and budget computation after the diagnostic extraction. These operations are performed at each timestep, after the physics computations, so that the raw recorded fields are accessible as NPROMA packets at the level of APLPAR/APL_AROME, where they may be used for other purposes.

For the DDH domain averaging, the Arpege subroutine cpcuddh.F90 (see DDH documentation for more details) is used and averaged fields are then written into file in ppfidh.F90 (which will be simplified since now with the self-documented structure, a loop on all elements in DDH_DESCR can generate the names of the fields to be written into the DDH file). The subroutine cpcuddh.F90 uses arrays (hdcvbx stored in module yomtddh) whose size is computed in setups (the total number of fluxes/tendencies depend on the options used for physics). Since these setups are no longer used with the new data flow, these arrays are allocated with an estimated size (larger than expected value) for the time being but we are thinking at a way to have them reallocated or initialized elsewhere in the code after a dummy call to the code that only computes the total size of DDH arrays (like the call to stepo from cnt4.F90 if CFU/XFU diagnostics are switched on).

Figure 7.1 summarizes the new data flow (which is the same for Arpege and Arome) within a time step.

7.4.5.3 Application to DDH in Arome

The new dataflow is used in Arome since cy35t1 for DDH diagnostics. Méso-NH code already uses its own diagnostics through the sophisticated budgets and advantage is taken form the work already performed there in order to avoid duplication of effort. MNH's budgets are called through the call of the subroutine budget after each process. This subroutine is able to perform operations on the stored quantity. In order to keep the maximum level of compatibility between MNH and Arome code, it was chosen to keep the calls to budget unchanged in the Méso-NH code and to write a new budget subroutine that would be called in Arome instead of the budget from MASDEV. This subroutine, located in /mpa/micro/externals, supresses first and last level of MNH fields and reverses the order of the vertical levels.

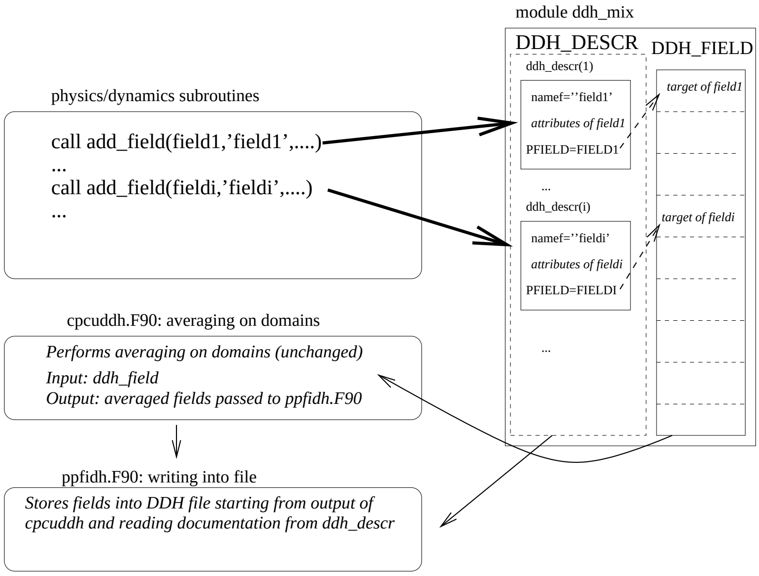


Figure 7.1: Organization of the data flow within a time step. Subroutine ADD_FIELD stores the field and the associated description into DDH_DESCR after possible transformations (bold arrows). Averaging on the domains is performed as in Arpege in cpcuddh.F90, the output being written into file in ppfidh.F90 using the description of the fields stored as attribute in DDH_DESCR.

In Arome there are two different ways to have terms in the DDH products. The first is to use ADD_FIELD_3D after a call INVERT_VLEV as shown previously. The second possibility is to use the budgets from Méso-NH as in the first version of the DDH code in Arome. We have used a combination of the two methodes in order to take advantage of the validation performed by the Méso-NH team of the budget packages.

- Variables are stored in cpdyddh.F90 using ADD_FIELD_3D since the part of the code is common with Arpege/Aladin/Alaro.
- Within APL_AROME, adjustment and radiation are retrieved using ADD_FIELD_3D and other processes through budgets from Méso-NH.

Interfacing with Méso-NH budgets works the following way:

- ARO_SUINTBUDGET stores quantities (Exner function...) necessary to transform tendencies from Méso-NH (in θ,r variables) to tendencies in (T,q) variables into the module MODDB_INTBUDGET
- ARO_STARTBU stores initial values of tendencies for each variable
- Within Méso-NH, subroutine “BUDGET” is called. The BUDGET subroutine from Méso-NH is replaced by a new subroutine (/mpa/micro/internals/budget.mnh) called the same way with the same arguments that transforms tendencies of Méso-NH variables (θ...) to tendencies on the desired variables (c_pT...) and skips the Méso-NH processing.

7.4.6 Remaining issues specific to the new dataflow

Some issues are still to be dealt with in the new dataflow:

- Performances. If faster in Arome than the old code, there is still room for improvement in terms of computational performances.

- OpenMP. The code has to be tested and optimized for OpenMP parallelization. For the time being, validation has only been done on the NEC platform from Météo-France.

- Elarging the flexibility of the code to the DDH operators for domain averaging. Some arrays like PDDHCV_TOT still have to be initalized at the beginning of the time step and thus we don't fully benefit from the flexibility of the new dataflow. Thinking about how to deal with that without affecting the part of he code used by ECMWF is ongoing.

7.4.7 Conclusion

This new version of the dataflow offers not only more facilities to add new quantities in the diagnostics but also more flexibility in terms of possible uses of these diagnostics. For developers, since the new code is considerably smaller and readable than the current one in Arome, it will be easier to debug and maintain when physics evolve in the future. We also expect an increase in the code's performance for Arome's DDH since the Meso-NH budgets part of the code (with a lot of unused (in Arome) options slowing the code) will be skipped.

An another important aspect is that this tool, after being successfully implemented in Arome can now be used in Arpege/Aladin/Aladin. Before going on with further work to upgrade this prototype version towards a beta version, discussion between the different possible users of this type of diagnostics is needed in order to raise possible new issues and needs regarding what different users would like these structures to offer.

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 AROME 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).

2009: Interface AROME physics with DDH, new flexible dataflow for AROME, ALADIN and ARPEGE (Olivier Rivière).

2018: Dynamical DDH: semi-lagrangian advection, horizontal diffusion, semi-implicit (Fabrice Voitus).

2019: Flexible DDH: adding a new field in calling NEW_ADD_FIELD_3D (Fabrice Voitus).