Diagnostics in Horizontal Domains (DDH)

Variables and budget equations, in horizontal mean

ARPEGE, ALADIN and AROME models

Guide for users and developers

July 24, 2008
## Contents

1 Main principles 7
1.1 User-defined horizontal domains 7
1.2 User-defined outputs 8
1.3 Scientific content of diagnostics 8
1.4 Output frequency 8
1.5 Internal representation of domains 8
1.6 Horizontal mean operator 9
1.7 Computation time of DDH 9

2 Producing DDH files: geometry, user namelist 11
2.1 Different kinds of horizontal domains 11
   2.1.1 Global domain 11
   2.1.2 Zonal bands 11
   2.1.3 Limited areas 12
2.2 Declarations. The NAMDDH namelist 15
   2.2.1 Declarations 15
      2.2.1.1 Type of domains 16
      2.2.1.2 Variables to budgetise 16
      2.2.1.3 Output on file or listing 16
      2.2.1.4 Software maintenance, debugging mode 16
      2.2.1.5 Control results reproducibility 17
      2.2.1.6 Number of zonal bands 17
   2.2.2 Declaration of limited domains 17
   2.2.3 Default values 20
2.3 Output occurrence control 20
2.4 Identification of domains in the code and in the outputs 21
2.5 Logical structure of output files 22
   2.5.1 File names 22
   2.5.2 Articles giving information about dimensions and dates 22
   2.5.3 Articles giving information about the type of domains 23
   2.5.4 Articles giving information about scientific fields 24

3 Budget equations and horizontal mean 25
3.1 Generic budget equation 25
   3.1.1 Term 1. Divergence of horizontal fluxes at the boundaries of the domain 25
   3.1.2 Term 2. Divergence of the adiabatic vertical flux 26
   3.1.3 Term 3. Adiabatic source term 26
   3.1.4 Term 4. Physical fluxes divergence term 26
   3.1.5 Term 5. Tendency term due to physical parametrizations 27
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Budget and diagnostics</td>
<td>31</td>
</tr>
<tr>
<td>4.1</td>
<td>Dry air mass budget</td>
<td>31</td>
</tr>
<tr>
<td>4.2</td>
<td>Water mass budget</td>
<td>32</td>
</tr>
<tr>
<td>4.3</td>
<td>Momentum budget</td>
<td>34</td>
</tr>
<tr>
<td>4.4</td>
<td>Kinetic energy budget</td>
<td>35</td>
</tr>
<tr>
<td>4.5</td>
<td>Enthalpy budget</td>
<td>36</td>
</tr>
<tr>
<td>4.6</td>
<td>Angular momentum budget</td>
<td>38</td>
</tr>
<tr>
<td>4.7</td>
<td>Entropy budget</td>
<td>40</td>
</tr>
<tr>
<td>4.8</td>
<td>Potential energy budget</td>
<td>41</td>
</tr>
<tr>
<td>4.9</td>
<td>Surface budgets</td>
<td>42</td>
</tr>
<tr>
<td>4.10</td>
<td>Instantaneous diagnostics</td>
<td>43</td>
</tr>
<tr>
<td>4.10.1</td>
<td>Relative humidity</td>
<td>43</td>
</tr>
<tr>
<td>4.10.2</td>
<td>Cloudiness</td>
<td>43</td>
</tr>
<tr>
<td>4.10.3</td>
<td>Vertical velocity</td>
<td>44</td>
</tr>
<tr>
<td>4.10.4</td>
<td>Passive variables</td>
<td>44</td>
</tr>
<tr>
<td>4.11</td>
<td>Cumulated mass</td>
<td>44</td>
</tr>
<tr>
<td>5</td>
<td>Budgets, AROME model</td>
<td>45</td>
</tr>
<tr>
<td>5.1</td>
<td>Microphysical process</td>
<td>45</td>
</tr>
<tr>
<td>5.2</td>
<td>Balance equations</td>
<td>47</td>
</tr>
<tr>
<td>5.2.1</td>
<td>Momentum</td>
<td>47</td>
</tr>
<tr>
<td>5.2.2</td>
<td>Turbulence kinetic energy</td>
<td>47</td>
</tr>
<tr>
<td>5.2.3</td>
<td>Enthalpy</td>
<td>48</td>
</tr>
<tr>
<td>5.2.4</td>
<td>Cloud water</td>
<td>49</td>
</tr>
<tr>
<td>5.2.5</td>
<td>Cloud ice</td>
<td>50</td>
</tr>
<tr>
<td>5.2.6</td>
<td>Rain</td>
<td>50</td>
</tr>
<tr>
<td>5.2.7</td>
<td>Snow</td>
<td>51</td>
</tr>
<tr>
<td>5.2.8</td>
<td>Graupel</td>
<td>51</td>
</tr>
<tr>
<td>5.3</td>
<td>Common Dynamics-Physics Interface CDPI</td>
<td>52</td>
</tr>
<tr>
<td>6</td>
<td>Software maintenance</td>
<td>53</td>
</tr>
<tr>
<td>6.1</td>
<td>Main arrays and their organization</td>
<td>53</td>
</tr>
<tr>
<td>6.1.1</td>
<td>Arrays describing domains</td>
<td>53</td>
</tr>
<tr>
<td>6.1.2</td>
<td>Data arrays</td>
<td>54</td>
</tr>
<tr>
<td>6.1.3</td>
<td>Main local arrays</td>
<td>56</td>
</tr>
<tr>
<td>6.1.3.1</td>
<td>Arrays for computations at each grid point</td>
<td>56</td>
</tr>
<tr>
<td>6.1.3.2</td>
<td>Arrays for synthesis over a user domain</td>
<td>57</td>
</tr>
<tr>
<td>6.2</td>
<td>Organization of the main functions</td>
<td>57</td>
</tr>
<tr>
<td>6.3</td>
<td>How to add new fields to budgets</td>
<td>60</td>
</tr>
<tr>
<td>6.4</td>
<td>Maintenance specificities, AROME model</td>
<td>60</td>
</tr>
<tr>
<td>6.4.1</td>
<td>New module YOMPHFT</td>
<td>60</td>
</tr>
<tr>
<td>6.4.2</td>
<td>Data flow</td>
<td>62</td>
</tr>
<tr>
<td>6.4.3</td>
<td>Setup</td>
<td>62</td>
</tr>
<tr>
<td>6.4.3.1</td>
<td>Subroutine SUOYOMA</td>
<td>62</td>
</tr>
<tr>
<td>6.4.3.2</td>
<td>Subroutine SUOYOMB</td>
<td>62</td>
</tr>
<tr>
<td>6.4.4</td>
<td>Entering physical fluxes and tendencies in DDH</td>
<td>62</td>
</tr>
</tbody>
</table>
## 6.4.5 Modified subroutines

6.4.5.1 Subroutine APL_AROME ........................................ 62
6.4.5.2 Subroutine AROEND_BUDGET ................................. 63
6.4.5.3 Subroutine CPDYDDH ......................................... 63
6.4.5.4 Subroutine CPG_DIA ........................................... 64
6.4.5.5 Subroutine CPG ................................................. 64
6.4.5.6 SU0PHY ....................................................... 64
6.4.5.7 Subroutine SUPHMNH ........................................ 64
6.4.5.8 Subroutine SUNDDH ......................................... 64
6.4.5.9 Subroutine PPFIDH ......................................... 65

## 6.4.6 Modified modules

6.4.6.1 Module YOMARPHY ........................................... 66
6.4.6.2 Module YOMLDDH ........................................... 66
6.4.6.3 Module YOMMNH ........................................... 66
6.4.6.4 Module YOMPHY ........................................... 66

## 6.4.7 New subroutines

6.4.7.1 Subroutine ARO_CPPHDDH .................................. 67
6.4.7.2 Subroutine ADDFT ........................................... 67
6.4.7.3 Subroutine ARO_INIPFT .................................. 67

## 6.4.8 Adding a new term in DDH for AROME ......................... 68

## 7 Using DDH files: the ddhtoolbox ................................. 69

7.1 Purpose .......................................................... 69
7.2 Synoptic view of ddhtoolbox utilities ........................ 69
7.3 Install the software ............................................. 70
7.4 Environment variables ........................................... 71
7.5 ddhb: get budgets ............................................... 71

## 8 History .......................................................... 73
Chapter 1

Main principles

This paper presents the diagnostics on horizontal domains (DDH) developed initially 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 developers 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 software by its geographical position, a scale factor, the orientation of the geographical North vector, a mean value of each variable as well as some horizontal derivatives.

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

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

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

1.1 User-defined horizontal domains

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

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

1.2 User-defined outputs

• a print on standard output of budgets on any given domain, in vertical mean.

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

1.3 Scientific content of diagnostics

• Budgets of mass / energy: balance of air and water masses, potential and internal energies, kinetic energy, momentum budget, entropy budget, can be activated from independent logical indicators.

• For surface, the DDH tool provides surface fluxes, but not soil budgets, as these soil budgets are given by the SURFEX software itself.

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

1.4 Output frequency

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

1.5 Internal representation of domains

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

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

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

1.7 Computation time of DDH

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

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

Having said that, using the DDH code to produce diagnostics at one point only, requires a large computation time, as computation is done on all points anyway.
Chapter 2

Producing DDH files: geometry, user namelist

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

2.1 Different kinds of horizontal domains

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

Recognized types:

- The global domain is activated if \( \text{LHDGLB} \) is true,
- Zonal bands are activated if \( \text{LHDZON} \) is true,
- Limited area domains or isolated points are activated if \( \text{LHDDOP} \) is true.

2.1.1 Global domain

- \( \text{LHDPRG} \), true for printing.
- \( \text{LHDEFG} \), true for producing a file.

2.1.2 Zonal bands

The total number of zonal bands is \( \text{NDHKD} \). 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 (\( \text{NDHZRP} \)).

To summarize

- \( \text{LHDPRZ} \) true induces the printing of the latitude band number \( \text{NDHZPR} \),
- \( \text{LHDEFZ} \) true induces the writing of a file,
- \( \text{NDHKD} \) specifies the number of latitude bands.
The principle of it is to divide the real sphere into \( \text{NDHKD} \) bands of geographical latitudes of equal surfaces. Each band is identified by its index \( j\text{k}\text{d} \) (\( 1 \leq j\text{k}\text{d} \leq \text{NDHKD} \)).

Let the zonal band between geographical latitudes \( \theta_{g,j\text{k}\text{d}} \) and \( \theta_{g,j\text{k}\text{d}+1} \), and notating

\[
\mu_{j\text{k}\text{d}} = \sin \theta_{g,j\text{k}\text{d}},
\]

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

\[
\text{NDHKD} \times 2\pi a^2 (\mu_{j\text{k}\text{d}} - \mu_{j\text{k}\text{d}+1}) = 4\pi a^2,
\]

with \( \mu_1 = 1 \). Therefore

\[
\mu_{j\text{k}\text{d}} = 1 - (j\text{k}\text{d} - 1) \frac{2}{\text{NDHKD}}
\]

The latitude of the zonal band \( j\text{k}\text{d} \) is

\[
\bar{\mu}_{j\text{k}\text{d}} = 1 - \left( j\text{k}\text{d} - \frac{1}{2} \right) \frac{2}{\text{NDHKD}}
\]

### 2.1.3 Limited areas

Several type of limited domains may be defined

- Type 1: an isolated point defined by its indexes \((j\text{lon}, j\text{gl})\),
- Type 2: a domain defined by its four corners identified by \((\text{geographical longitude}, \text{geographical latitude})\),
- Type 3: a domain defined by two opposite corners, identified by \((\text{geographical longitude}, \text{geographical latitude})\),
- 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 \((\lambda, \mu)\).

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 \text{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.
2.1. DIFFERENT KINDS OF HORIZONTAL DOMAINS

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).

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

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

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

Thus, let us go back to case 4. Quadrilateral 1 is declared as a guess for domain 1. Every single points included in this quadrilateral is given index 1. Then, quadrilateral 2 is declared and its points are given index 2, notably, those which previously had value 1. The shape of points \( \langle 1 \rangle \) 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. \textit{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 \texttt{NAMDDH} namelist

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

### 2.2.1 Declarations

\texttt{NAMDDH} regroups the main parameters controlling the diagnostics. Can be found logical indicators, some numerical parameters and also a table to declare possible limited domains.
2.2.1.1 Type of domains

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

2.2.1.2 Variables to budgetise

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

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

2.2.1.3 Output on file or listing

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

2.2.1.4 Software maintenance, debugging mode

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

Furthermore, this point must be declared as a domain. For more details about this option, refer to pages ?? and following.
2.2. DECLARATIONS. THE NAMDDH NAMELIST

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

\[
\begin{align*}
\text{BDEDDH(1, domain number)} &= \text{type} \\
\text{BDEDDH(2, domain number)} &= \text{virtual plane}
\end{align*}
\]

For type 1, point given by its indexes

\[
\begin{align*}
\text{BDEDDH(3, domain number)} &= \text{rjlon} \\
\text{BDEDDH(4, domain number)} &= \text{rjgl}
\end{align*}
\]
For type 2, quadrilateral given by its four corners

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

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

- For a domain which does not intersect the Greenwich meridian

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

and

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

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

\[(\lambda_1 \leq 360^\circ \text{ ou } \lambda_4 \leq 360^\circ) \quad \text{and} \quad (\lambda_2 > 360^\circ \text{ ou } \lambda_3 > 360^\circ)\]
For type 3, rectangular domain given by two opposite corners

\[
\text{BDDEDDH}(3, \text{domain number}) = \text{Longitude of corner #1, in degrees, } \lambda_1
\]

\[
\text{BDDEDDH}(4, \text{domain number}) = \text{Latitude of corner #1, in degrees, } \theta_1
\]

\[
(\text{BDDEDDH}(5, -), \text{BDDEDDH}(6, -)) = (\lambda_3, \theta_3)
\]

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

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

For type 4, points given by their geographical position

\[
\text{BDDEDDH}(3, \text{domain number}) = \text{longitude } \lambda_g \text{ in degrees}
\]

\[
\text{BDDEDDH}(4, \text{domain number}) = \text{latitude } \theta_g \text{ in degrees}
\]

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

**Allocation of points to the domains**

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

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

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

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

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

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

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., LHDDNT = .FALSE., LHDPGR = .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., 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) = pi (pi ≥ 0, i ∈ {1, ..., m}): output at time steps npi. In this case it will be preferable 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) = −pi (pi ≥ 0, i ∈ {1, ..., m}): output at hours npi.

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.

<table>
<thead>
<tr>
<th>Domain Identifier</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>Virtual plane</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
</tr>
<tr>
<td>number in this plane</td>
<td>λ₁</td>
<td>μ₁</td>
<td>λ₂</td>
<td>μ₂</td>
<td>λ₃</td>
<td>μ₃</td>
<td>λ₄</td>
<td>μ₄</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type-dependent information</td>
<td>r_jlon</td>
<td>r_jgl</td>
<td>λ_g</td>
<td>μ_g</td>
<td>user value</td>
<td>user value</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Type</td>
<td>Type 1. Point</td>
<td>Type 4. Point</td>
<td>Type 2. Quadrilateral</td>
<td>Type 3. Rectangle</td>
<td>Type 5. Globe</td>
<td>Type 6. Zonal band</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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 the 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 \( \text{DHGLeeee+nnnn} \)
Zonal bands \( \text{DHFOeeeee+nnnn} \)
Limited area domains \( \text{DHFDLeeee+nnn} \)

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 LHDKS is false, 1 if true,
3. 0 if LHMCI is false, 1 if true,
4. 0 if LHDET 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. NDHV, 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. NDHSV, number of instantaneous soil variables,
13. NDHFS, number of soil (dynamical) fluxes,
14. NDHFSP, number of soil (physical) fluxes,
15. number of domains in the file:
   - 1 for the globe,
   - NDH 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 ‘DOCDnnn’, where nnn is the name of the domain. This article is made of 11 words whose content has been described page 21.
2.5.4 Articles giving information about scientific fields

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

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

- $nnn$: number of the domain in the file. $nnn$ varies from 1 to $DOCFICHIER(17)$.
- $t$: type of field contained in the article:
  - $V$: variable profile, length $NFLEV$,  
  - $T$: tendency profile, length $NFLEV$,  
  - $F$: flux profile, length $NFLEV+1$,  
  - $S$: soil data, length: cf. page 42.
- $vv$: physical variable written in this file article:
  - $PP$: pressure,  
  - $QV$: specific water vapour content,  
  - $UU$: zonal momentum,  
  - $VV$: merional momentum,  
  - $KK$: kinetic energy,  
  - $CT$: enthalpy,  
  - $EN$: entropy,  
  - $M1$: angular momentum,  
  - $EP$: potential energy ($\Phi = g z$).

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

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

Some suffixes crop up quite frequently

- $ssssssssss = DIVFLUHOR$ for terms of the kind $\text{div}_\eta \left( \chi \tilde{v} \frac{\partial p}{\partial \eta} \right)$  
- $ssssssssss = FLUVERTDYN$ for terms $\chi \eta \frac{\partial p}{\partial \eta}$,  
- $ssssssssss = FLUDUAPLUI$ 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 $\chi$ be a variable of the model whose budget is required. The generic form of the $\chi$ budget may be written

$$
\frac{\partial}{\partial t} \left( \chi \frac{\partial p}{\partial \eta} \right) = - \text{div} \eta \left( \chi \bar{v} \frac{\partial p}{\partial \eta} \right) - \frac{\partial}{\partial \eta} \left( \chi \dot{\eta} \frac{\partial p}{\partial \eta} \right) + S_d \frac{\partial p}{\partial \eta} - g \frac{\partial F_{\phi}}{\partial \eta} - g S_{\phi} \frac{\partial G_{\phi}}{\partial \eta} \quad (1)
$$

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, $\chi \delta p_{\ell}$ for the discrete value in the layer $\ell$. In practice, the application of the vertical discretization leads this equation to

$$
\frac{\partial}{\partial t} \left( \chi \delta p \right) = - \text{div} \eta \left( \chi \bar{v} \delta p \right) - \delta \left( \chi \dot{\eta} \frac{\partial p}{\partial \eta} \right) + S_d \delta p - g \delta F_{\phi} - g S_{\phi} \delta G_{\phi} \quad (2)
$$

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

$$
\delta \xi_{\ell} = \xi_{\ell} - \xi_{\ell-1} \quad (3)
$$

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

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 $\chi$ gradient. For the initial conditions this will not always be the case. This term will not be complete, especially every time when $\chi$ depends on
the momentum (momentum itself, kinetic energy, angular momentum, etc). Whenever possible,

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

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 $\chi$.

### 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,

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

will be computed.

The vertical speed $\dot{\eta} \partial p/\partial \eta$ is computed by $\mathcal{GFC\mathcal{T}Y}$ 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

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

Some similar terms are to be found in the kinetic energy budget. They, as term 1, are in the "tendencies" category, expressed at $\ell$ levels.

### 3.1.4 Term 4. Physical fluxes divergence term

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

$$F_{\varphi_{\ell}T} = L (\eta, T) F_{\varphi_{\ell}q}^{\text{precip}}(\eta)$$

Some assumptions need to be introduced, like

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

where $L$ is an effective specific enthalpy of phase change.
3.2. HORIZONTAL MEAN

3.1.5 Term 5. Tendency term due to physical parametrizations

Such terms occur in the energy budgets, e.g., the dissipation term
\[ \vec{v} \frac{\partial F^{\text{tur+conv}}}{\partial \eta} \]
or in the entropy budget
\[ \frac{1}{T} \frac{\partial F^{\text{ray}}}{\partial \eta} \].

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 \( \text{div} (\chi \delta p \vec{v}) \)),
- **fluxes**, at the inter-layers (\( \delta F_{\chi} \)).

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

\[
\frac{\partial}{\partial t} \left( \frac{1}{g} \chi \delta p \right) = \left( \frac{1}{g} \chi \delta p \right) \text{tend} - \delta F_{\chi}
\]

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

\[
\frac{1}{S_D} \frac{\partial}{\partial t} \int_D \frac{1}{g} \chi \delta p \, d\sigma = \frac{1}{S_D} \int_D \left( \frac{1}{g} \chi \delta p \right) \text{tend} \, d\sigma - \frac{1}{S_D} \int_D \delta F_{\chi} \, d\sigma
\]

Some terms, such as the effect of the horizontal diffusion, cannot be diagnosed by \( \text{DDH} \): 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

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

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, \( \varpi_k \) is Gauss weight and \( J(k) \) the number of points on the circle of latitude \( k \).
Thus, $k$ refers to the latitudes of the gaussian grid and $j$ to the longitudes. On the stretched sphere the scale factor is a function of the spherical harmonics:

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

now with

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

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

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

Surface of domain $D$:

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

The horizontal mean of parameter $\chi$ on $D$ is written

$$\left[ \frac{1}{S_D} \int\int_D \chi \, d\sigma \right] = [\chi]_D = \frac{1}{S_D} \sum_{(j,k) \in D} \chi_{j,k} \sigma_{j,k}$$

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

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

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

$$[\chi]_{D_i} \, S_{D_i} = \sum_{p=1}^{P} \left( \sum_{(j,k) \in D_i(p)} \chi_{j,k} \sigma_{j,k} \right)$$

where $D(p)$ are the points belonging to $D$ 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 $D$ called for the user is

$$[\chi]_D = \frac{1}{S_D} \sum_{D_i \in D} \left[ \sum_{p=1}^{P} \left( \sum_{(j,k) \in D_i(p)} \chi_{j,k} \sigma_{j,k} \right) \right]$$

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.

• $\langle \langle$ condensation $\rangle \rangle$ 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

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

is integrated by the diagnostics $DDH$ as follow

$$\left( \frac{1}{g} \chi \delta p \right)^{n_{step}} - \left( \frac{1}{g} \chi \delta p \right)^0 = \Delta t \sum_{jstep=0}^{n_{step}-1} \left[ \left( \frac{1}{g} \chi \delta p \right) \right]_{jstep}$$

Where $\Delta t$ stands for $TSTEP$, the nominal time step, and $n_{step}$ 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 $\ell$ and a discretized domain $\mathcal{D}$

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

with

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

$$\left[ \frac{1}{g} \chi \delta p \right]_{\mathcal{D}}^{\ell} (t = NSTEP \times \Delta t) - \left[ \frac{1}{g} \chi \delta p \right]_{\mathcal{D}}^{\ell} (t = 0) = \delta t \sum_{n=0}^{n_{step}-1} \left[ \left( \frac{1}{g} \chi \delta p \right) \right]_{D}^{\ell} (n) + \delta t \left( \sum_{n=0}^{n_{step}-1} \left[ F_{\chi} \right]_{D}^{\ell} (n) \right)$$
and the vertical mean budget

\[
\sum_{\ell=1}^{NFLEV} \left[ \frac{1}{g} \chi \delta p \right]_D^{\ell} (t = NSTEP \times \delta t) - \sum_{\ell=1}^{NFLEV} \left[ \frac{1}{g} \chi \delta p \right]_D^{\ell} (t = 0) =
\]

\[
\delta t \sum_{\ell=1}^{NFLEV} \sum_{n=0}^{nstep-1} \left[ \frac{1}{g} \chi \delta p \right]_D^{\ell} (n) + \delta t \sum_{n=0}^{nstep-1} [F_{\chi}]^0_D (n) - \sum_{n=0}^{nstep-1} [F_{\chi}]^{NFLEV} (n)
\]
Chapter 4

Budget and diagnostics

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

4.1 Dry air mass budget

Budget equation

\[
\frac{\partial r_p}{\partial t} = - \text{div}_\eta (r_\eta \vec{v}) - \frac{\partial}{\partial \eta} (r_\eta \dot{\eta}) + \delta_m \frac{\partial F_p}{\partial \eta}
\]

where

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

File output

| VPP0 | 1 \( \frac{\delta p}{g} \) (t=0) |
| VPP1 | 1 \( \frac{\delta p}{g} \) (t=NSSTE \( \delta t \)) |
| TPPDIVFLUHOR | \( - \frac{\delta t}{g} \) div\(_\eta(\vec{v}\delta p) \) cumulated |
| FPPFLUVERTDYN | \( \frac{\delta t}{g} \frac{\partial p}{\partial \eta} \) cumulated |
| FPPSUMFPL | \( \delta_m \delta t F_p \) cumulated |
If $\chi^*$ stands for a quantity modified by the mass exchange and if by $\chi$ the initial quantity, then

$$
\left( \dot{\eta} \frac{\partial p}{\partial \eta} \right)^* = \left( \dot{\eta} \frac{\partial p}{\partial \eta} \right) + \delta m g \left[ B \ell \left( F_p L + E \right) - F_{p\ell} \right]
$$

One should have at the lower limit

$$
F_{qL}^{\text{tur}} = E \left( 1 - \delta m q_v \right)
$$

Furthermore

$$
\left( \frac{\omega}{p} \right)^* \ell = \left( \frac{\omega}{p} \right) \ell - \delta m g \left[ \frac{1}{\partial p \partial \eta} \left( \alpha \ell \delta F_p + \ln \left( \frac{p_{\ell}}{p_{\ell-1}} \right) F_{p\ell-1} \right) \right]
$$

$$
\left( \frac{\partial \pi}{\partial t} \right)^* = \frac{\partial \pi}{\partial t} - \delta m g \left( E + F_{pL} \right)
$$

### 4.2 Water mass budget

#### Lagrangian equation

$$
\eta \frac{\partial q_{\psi}}{\partial t} = T_{q\psi}^{\text{diff-hor}} + \frac{\partial F_{q\psi}}{\partial \eta} - \delta m q_{\psi} \frac{\partial F_p}{\partial \eta}
$$

#### Budget equation

$$
\frac{\partial (\eta \eta q_{\psi})}{\partial t} = - \text{div} \eta \left( \eta \eta q_{\psi} \nabla \right) - \frac{\partial}{\partial \eta} (\eta \eta q_{\psi} \dot{\eta}) + \frac{\partial F_{q\psi}}{\partial \eta}
$$

where

- $q_{\psi} = q_v, q_l$ ou $q_n$.
- $\eta = \frac{1}{g} \frac{\partial p}{\partial \eta}$.
- $F_{qv} = F_{c\psi}^{\text{conv-l}} + F_{c\psi}^{\text{conv-n}} + F_{c\psi}^{\text{stra-l}} + F_{c\psi}^{\text{stra-n}} + F_{q\psi}^{\text{tur}} + F_{q\psi}^{\text{tur-conv}}$.
- $F_{ql} = F_{p\psi}^{\text{conv-l}} + F_{p\psi}^{\text{stra-l}} - F_{c\psi}^{\text{conv-l}} - F_{c\psi}^{\text{stra-l}} + F_{q\psi}^{\text{tur}} + F_{q\psi}^{\text{tur-conv}}$.
- $F_{qn} = F_{p\psi}^{\text{conv-n}} + F_{p\psi}^{\text{stra-n}} - F_{c\psi}^{\text{conv-n}} - F_{c\psi}^{\text{stra-n}} + F_{q\psi}^{\text{tur}} + F_{q\psi}^{\text{tur-conv}}$.
- $F_{q\psi}^{\text{tur}}$ contains the correction $F_{q\psi}^{\text{tur}<0}$ 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.
### 4.2. WATER MASS BUDGET

#### File output

<table>
<thead>
<tr>
<th>Variable</th>
<th>Equation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FQTPRECICOL</td>
<td>$\delta t F_{\text{conv}}^{p-l}$</td>
<td>cumulated</td>
</tr>
<tr>
<td>FQTPRECICON</td>
<td>$\delta t F_{\text{conv}}^{p-n}$</td>
<td>cumulated</td>
</tr>
<tr>
<td>FQTPRECISTL</td>
<td>$\delta t F_{\text{str}}^{c-l}$</td>
<td>cumulated</td>
</tr>
<tr>
<td>FQTPRECISTN</td>
<td>$\delta t F_{\text{str}}^{c-n}$</td>
<td>cumulated</td>
</tr>
<tr>
<td>FQTCONDECOL</td>
<td>$\delta t F_{\text{conv}}^{c-l}$</td>
<td>cumulated if LHDQLN=.TRUE.</td>
</tr>
<tr>
<td>FQTCONDECON</td>
<td>$\delta t F_{\text{conv}}^{c-n}$</td>
<td>cumulated if LHDQLN=.TRUE.</td>
</tr>
<tr>
<td>FQTCONDESTL</td>
<td>$\delta t F_{\text{str}}^{c-l}$</td>
<td>cumulated if LHDQLN=.TRUE.</td>
</tr>
<tr>
<td>FQTCONDESTN</td>
<td>$\delta t F_{\text{str}}^{c-n}$</td>
<td>cumulated if LHDQLN=.TRUE.</td>
</tr>
<tr>
<td>VQVO</td>
<td>$\frac{1}{g} q_v \delta p (t=0)$</td>
<td></td>
</tr>
<tr>
<td>VQV1</td>
<td>$\frac{1}{g} q_v \delta p (t=\text{NSTEP})$</td>
<td></td>
</tr>
<tr>
<td>TQVDIVFLUHOR</td>
<td>$-\frac{\delta t}{g} \text{div}_\eta (q_v \delta p \vec{v})$</td>
<td>cumulated</td>
</tr>
<tr>
<td>FQVFVLUVERTDYN</td>
<td>$\frac{\delta t}{g} q_{\text{v,} \eta} \frac{\partial p}{\partial \eta}$</td>
<td>cumulated</td>
</tr>
<tr>
<td>FQVTUR</td>
<td>$\delta t F_{\text{tur}}^{q_v}$</td>
<td>cumulated</td>
</tr>
<tr>
<td>FQVTURCONV</td>
<td>$\delta t F_{\text{tur-conv}}^{q_v}$</td>
<td>cumulated</td>
</tr>
<tr>
<td>FQVTURQNEGAT</td>
<td>$\delta t F_{\text{q&lt;0}}^{q_v}$</td>
<td>cumulated</td>
</tr>
<tr>
<td>VQL0</td>
<td>$\frac{1}{g} q_l \delta p (t=0)$</td>
<td>if LHDQLN=.TRUE.</td>
</tr>
<tr>
<td>VQL1</td>
<td>$\frac{1}{g} q_l \delta p (t=\text{NSTEP})$</td>
<td>if LHDQLN=.TRUE.</td>
</tr>
<tr>
<td>TQLDIVFLUHOR</td>
<td>$-\frac{\delta t}{g} \text{div}_\eta (q_l \delta p \vec{v})$</td>
<td>cumulated if LHDQLN=.TRUE.</td>
</tr>
<tr>
<td>FQLFLUVERTDYN</td>
<td>$\frac{\delta t}{g} q_{\text{l,} \eta} \frac{\partial p}{\partial \eta}$</td>
<td>cumulated if LHDQLN=.TRUE.</td>
</tr>
<tr>
<td>FQLTUR</td>
<td>$\delta t F_{\text{tur}}^{q_l}$</td>
<td>cumulated if LHDQLN=.TRUE.</td>
</tr>
<tr>
<td>FQLTURCONV</td>
<td>$\delta t F_{\text{tur-conv}}^{q_l}$</td>
<td>cumulated if LHDQLN=.TRUE.</td>
</tr>
<tr>
<td>FQLTURQNEGAT</td>
<td>$\delta t F_{\text{q&lt;0}}^{q_l}$</td>
<td>cumulated if LHDQLN=.TRUE.</td>
</tr>
<tr>
<td>VQN0</td>
<td>$\frac{1}{g} q_n \delta p (t=0)$</td>
<td>if LHDQLN=.TRUE.</td>
</tr>
<tr>
<td>VQN1</td>
<td>$\frac{1}{g} q_n \delta p (t=\text{NSTEP})$</td>
<td>if LHDQLN=.TRUE.</td>
</tr>
<tr>
<td>TQNDIVFLUHOR</td>
<td>$-\frac{\delta t}{g} \text{div}_\eta (q_n \delta p \vec{v})$</td>
<td>cumulated if LHDQLN=.TRUE.</td>
</tr>
<tr>
<td>FQNFVLUVERTDYN</td>
<td>$\frac{\delta t}{g} q_{\text{n,} \eta} \frac{\partial p}{\partial \eta}$</td>
<td>cumulated if LHDQLN=.TRUE.</td>
</tr>
<tr>
<td>FQNTUR</td>
<td>$\delta t F_{\text{tur}}^{q_n}$</td>
<td>cumulated if LHDQLN=.TRUE.</td>
</tr>
<tr>
<td>FQNTURCONV</td>
<td>$\delta t F_{\text{tur-conv}}^{q_n}$</td>
<td>cumulated if LHDQLN=.TRUE.</td>
</tr>
<tr>
<td>FQNTURQNEGAT</td>
<td>$\delta t F_{\text{q&lt;0}}^{q_n}$</td>
<td>cumulated if LHDQLN=.TRUE.</td>
</tr>
</tbody>
</table>
4.3 Momentum budget

Budget equation

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

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

- \( \vec{F}_{\text{tur}} \) is the turbulent flux,
- \( \vec{F}_{\text{tur-conv}} \) is the convective transport,
- \( \vec{F}_{\text{rel}} \) is the momentum flux due to gravity wave drag.

In the model \((u', v')\) 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 \( \cos \alpha \ G \) NORDM \( \sin \alpha \ GNORDL \). 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.

\[
\begin{align*}
\text{VUU0} & \quad \frac{1}{g} u \delta p \ (t=0) \\
\text{VVV0} & \quad \frac{1}{g} v \delta p \ (t=0) \\
\text{VUU1} & \quad \frac{1}{g} u \delta p \ (t=\text{NSTEP } \delta t) \\
\text{VVV1} & \quad \frac{1}{g} v \delta p \ (t=\text{NSTEP } \delta t) \\
\text{TUUDIVFLUHOR} & \quad \frac{\delta t}{g} u \ \text{div}_\eta (\delta \vec{v}) \quad \text{cumulated} \\
\text{TVVDIVFLUHOR} & \quad \frac{\delta t}{g} v \ \text{div}_\eta (\delta \vec{v}) \quad \text{cumulated}
\end{align*}
\]

The wind tendency due to advection of wind by itself is missing!
4.4 Kinetic energy budget

**Budget equation**

\[
\frac{\partial}{\partial t} \left( \frac{1}{g} k \frac{\partial p}{\partial \eta} \right) = -\frac{1}{g} k \text{div}_{\eta} \left( \vec{v} \frac{\partial p}{\partial \eta} \right) - \frac{1}{g} \frac{\partial}{\partial \eta} \left( k \frac{\partial p}{\partial \eta} \right) - \frac{1}{g} \frac{\partial p}{\partial \eta} \vec{v} \cdot \vec{\nabla} k
\]

\[-\frac{1}{g} \frac{\partial p}{\partial \eta} \vec{v} \left( \vec{\nabla} \Phi + RT \vec{\nabla} \ln p \right) - \delta_m \frac{\partial k F_p}{\partial \eta} - \vec{v} \frac{\partial}{\partial \eta} \left( F_{u}^{\text{tur}} + F_{v}^{\text{tur-conv}} + F_{v}^{\text{rel}} + F_{v}^{\text{meso}} \right)\]

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 \( \delta t \))

TKKDIVFLUHOR \( -\frac{\delta t}{g} k \text{div}_{\eta} (\delta p \vec{v}) \) cumulated (the advection term is missing!)

TKKCONVERSI1 \( -\frac{\delta t}{g} \delta p \vec{v} \left( \vec{\nabla} \Phi + RT \vec{\nabla} \ln p \right) \) cumulated

FKKFLUVERTDYN \( \frac{\delta t}{g} \frac{1}{\eta} \frac{\partial p}{\partial \eta} \) cumulated

FKKFLUDUAPLUI \( \delta t \delta_m k F_p \) cumulated

TKKDISSITUR \( -\delta t \vec{v} \delta F_{u}^{\text{tur}} \) cumulated

TKKDISSIPCONV \( -\delta t \vec{v} \delta F_{u}^{\text{tur-conv}} \) cumulated

TKKDISSIPGREL \( -\delta t \vec{v} \delta F_{v}^{\text{rel}} \) cumulated
4.5 Enthalpy budget

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

**Lagrangian equation in $T$**

$$
\frac{r_{\eta}c_p}{\partial t} \frac{dT}{\partial \eta} = r_{\eta}RT \frac{\omega}{p} + \frac{\partial F_{h}}{\partial \eta} - L_{v<1}(T) \frac{\partial F_{l}}{\partial \eta} - L_{v>n}(T) \frac{\partial F_{n}}{\partial \eta}
$$

$$+F_{p}^{\delta} \frac{\partial}{\partial \eta} \left\{ T \left[ c_{l} - c_{p_{a}}(1 - \delta_{m}) \right] \right\} + F_{n}^{\delta} \frac{\partial}{\partial \eta} \left\{ T \left[ c_{n} - c_{p_{a}}(1 - \delta_{m}) \right] \right\}
$$

$$-T \left[ (c_{p_{v}} - c_{p_{a}}) \frac{\partial F_{\text{tur}}^{\delta}}{\partial \eta} + (c_{l} - c_{p_{a}}) \frac{\partial F_{\text{tur}}^{\delta}}{\partial \eta} + (c_{n} - c_{p_{a}}) \frac{\partial F_{\text{tur}}^{\delta}}{\partial \eta} \right]
$$

$$+ \delta_{m} F_{p} \frac{\partial h}{\partial \eta}
$$

(4.1)

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

$$
\frac{r_{\eta} \partial h}{\partial t} = -r_{\eta} \left( \vec{\nabla} \cdot \vec{v} \right) h - \frac{r_{\eta} \partial h}{\partial \eta} + \frac{r_{\eta}RT \omega}{p}
$$

$$+ \frac{\partial}{\partial \eta} \left\{ F_{h} + F_{hp} + F_{p}^{l}T \left[ c_{l} - c_{p_{a}}(1 - \delta_{m}) \right] + F_{n}^{l}T \left[ c_{n} - c_{p_{a}}(1 - \delta_{m}) \right] - \delta_{m} c_{p} T F_{p} \right\}
$$

$$+ \delta_{m} F_{p} \frac{\partial h}{\partial \eta}
$$

(4.2)

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

**Budget equation**

$$
\frac{\partial}{\partial t} \left( r_{\eta}c_{p}T \right) = - \text{div}_{\eta} \left( r_{\eta}c_{p}T \vec{v} \right) - \frac{\partial}{\partial \eta} \left( r_{\eta}c_{p}T \vec{v} \right) + \frac{r_{\eta}RT \omega}{p}
$$

$$+ \frac{\partial}{\partial \eta} \left\{ F_{h} + F_{hp} + F_{p}^{l}T \left[ c_{l} - c_{p_{a}}(1 - \delta_{m}) \right] + F_{n}^{l}T \left[ c_{n} - c_{p_{a}}(1 - \delta_{m}) \right] \right\}
$$

$$+ \delta_{m} F_{p} \frac{\partial (\Phi + \frac{u^2 + v^2}{2})}{\partial \eta} - \vec{v} \cdot \frac{\partial F_{v}^{\text{phys}}}{\partial \eta}
$$

(4.3)

where

- $r_{\eta} = - \left( \frac{1}{g} \frac{\partial p}{\partial \eta} \right)$.
- $F_{h} = F_{h}^{\text{sol}} + F_{h}^{\text{ther}} + F_{h}^{\text{meso}} + F_{s}^{\text{tur}} + F_{s}^{\text{tur-conv}}$. 

The symbol $\vec{v}$ represents the velocity vector, $\partial h/\partial \eta$ represents the Eulerian equation in $h$, and $\partial T/\partial \eta$ represents the Lagrangian equation in $T$. The budget equation is used to calculate the change in enthalpy over time considering various terms related to heat, work, and dissipation.
• $F^{\text{tur-conv}}$ is the subgrid-scale transport of dry static energy $s = c_p T + \Phi$, due to deep convection.

  - $F_{hp} = F^l_{hp} + F^n_{hp}$.
  - $F^l_{hp} = F^{\text{conv-l}}_{hp} + F^{\text{stra-l}}_{hp}$.
  - $F^n_{hp} = F^{\text{conv-n}}_{hp} + F^{\text{stra-n}}_{hp}$.
  - $F^{\text{conv-l}}_{hp} = - \left[ L_v > l(T) + (c_l - c_{p_v}) T \right] F^c_{\text{conv-l}}$.
  - $F^{\text{conv-n}}_{hp} = - \left[ L_v > n(T) + (c_n - c_{p_v}) T \right] F^c_{\text{conv-n}}$.
  - $F^{\text{stra-l}}_{hp} = - \left[ L_v > l(T) + (c_l - c_{p_v}) T \right] F^c_{\text{stra-l}}$.
  - $F^{\text{stra-n}}_{hp} = - \left[ L_v > n(T) + (c_n - c_{p_v}) T \right] F^c_{\text{stra-n}}$.
  - $F^s_{\text{sol}}$ et $F^s_{\text{ther}}$ are the solar and infra-red fluxes.

  - $\vec{F}^s_{\text{phys}} = \vec{F}^s_{\text{tur}} + \vec{F}^s_{\text{tur-conv}} + \vec{F}^s_{\text{rel}} + \vec{F}^s_{\text{meso}}$.

  - $c_p = c_{p_a} q_a + c_{p_v} q_v + c_l q_l + c_n q_n$.

**File output**

| VCT0 | $\frac{1}{g} c_p T \delta p$ (t=0) |
| VCT1 | $\frac{1}{g} c_p T \delta p$ (t= NSTEP $\delta t$) |
| TCTDIVFLUHOR | $- \frac{\delta t}{g} \text{div}_\eta (c_p T \delta p \vec{v})$ cumulated |
| TCTCONVERSI2 | $\frac{\delta t}{g} RT \delta p (\omega/p)$ cumulated |
| TCTCONVERSI3 | $- \delta m \delta t F_p \delta \Phi$ cumulated |
| FCTFLUVERTDYN | $\frac{\delta t}{g} c_p T \frac{\partial p}{\partial \eta}$ cumulated |
| FCTTUR | $\delta t F^s_{\text{tur}}$ cumulated |
| FCTTURCONV | $\delta t F^s_{\text{tur-conv}}$ cumulated |
| FCTRAYSOL1 | $\delta t F^s_{\text{sol}}$ cumulated |
| FCTRAYER1 | $\delta t F^s_{\text{ther}}$ cumulated |
| FCTMESO | $\delta t F^s_{\text{meso}}$ cumulated |
| FCTPRECISTL | $- \delta t F^p_{\text{stra-l}}$ cumulated |
| FCTPRECISTN | $- \delta t F^p_{\text{stra-n}}$ cumulated |
| FCTPRECICOL | $- \delta t F^p_{\text{conv-l}}$ cumulated |
| FCTPRECCICOL | $- \delta t F^p_{\text{conv-n}}$ cumulated |
| FCTPRECCSCOL | $- \delta t F^p_{\text{conv-l}} T \left[ c_l - c_{p_a} (1 - \delta_m) \right]$ cumulated |
| FCTPRECCSCON | $- \delta t F^p_{\text{conv-n}} T \left[ c_n - c_{p_a} (1 - \delta_m) \right]$ cumulated |
| FCTPRECCSSTL | $- \delta t F^p_{\text{stra-l}} T \left[ c_l - c_{p_a} (1 - \delta_m) \right]$ cumulated |
| FCTPRECCSSTN | $- \delta t F^p_{\text{stra-n}} T \left[ c_n - c_{p_a} (1 - \delta_m) \right]$ cumulated |
4.6 Angular momentum budget

The angular momentum is defined by

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

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

\[ \vec{M} = M_1 \vec{i} + M_2 \vec{j} + M_3 \vec{k} \]

\[ = (av \sin \lambda - a(u + a\Omega \cos \theta) \sin \theta \cos \lambda) \vec{i} \]

\[ + (-av \cos \lambda - a(u + a\Omega \cos \theta) \sin \theta \sin \lambda) \vec{j} \]

\[ + a(u + a\Omega \cos \theta) \cos \lambda \vec{k} \]

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.
- \(\theta\) and \(\lambda\) latitude and longitude.

Budget equation

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

avec

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

File output

VA10 \( \frac{\delta p}{\eta} \delta M_1 (t=0) \)
VA20 \( \frac{\delta p}{\eta} \delta M_2 (t=0) \)
VA30 \( \frac{\delta p}{\eta} \delta M_3 (t=0) \)
VA11 \( \frac{\delta p}{\eta} \delta M_1 (t=NSTEP \ \delta t) \)
VA21 \( \frac{\delta p}{\eta} \delta M_2 (t=NSTEP \ \delta t) \)
VA31 \( \frac{\delta p}{\eta} \delta M_3 (t=NSTEP \ \delta t) \)
4.6. ANGULAR MOMENTUM BUDGET

\[
\begin{align*}
\text{TA1DIVFLUHOR} & - \frac{\delta t}{g} M_1 \left[ \text{div}_\eta (\vec{v} \delta \rho) + a \Omega \cos \theta \delta B \frac{\partial \pi}{\partial x} \right] \quad \text{cumulated} \\
\text{TA2DIVFLUHOR} & - \frac{\delta t}{g} M_2 \left[ \text{div}_\eta (\vec{v} \delta \rho) + a \Omega \cos \theta \delta B \frac{\partial \pi}{\partial x} \right] \quad \text{cumulated} \\
\text{TA3DIVFLUHOR} & - \frac{\delta t}{g} M_3 \left[ \text{div}_\eta (\vec{v} \delta \rho) + a \Omega \cos \theta \delta B \frac{\partial \pi}{\partial x} \right] \quad \text{cumulated} \\
\text{FA1FLUVERTDY} & \frac{\delta t}{g} \frac{\delta \Phi}{\delta t} M_1 \dot{\gamma} \quad \text{cumulated} \\
\text{FA2FLUVERTDY} & \frac{\delta t}{g} \frac{\delta \Phi}{\delta t} M_2 \dot{\gamma} \quad \text{cumulated} \\
\text{FA3FLUVERTDY} & \frac{\delta t}{g} \frac{\delta \Phi}{\delta t} M_3 \dot{\gamma} \quad \text{cumulated} \\
\text{TA1ADJUST} & \frac{\delta t}{g} a \left[ - \left( \delta p \frac{\partial RT}{\partial y} + RT \delta B \frac{\partial \Phi}{\partial y} \right) \sin(\lambda + \Theta t) \\
& \quad + \left( \delta p \frac{\partial RT}{\partial x} + RT \delta B \frac{\partial \Phi}{\partial x} \right) \cos(\lambda + \Theta t) \sin \theta \right] \quad \text{cumulated} \\
\text{TA2ADJUST} & \frac{\delta t}{g} a \left[ - \left( \delta p \frac{\partial RT}{\partial y} + RT \delta B \frac{\partial \Phi}{\partial y} \right) \cos(\lambda + \Theta t) \\
& \quad + \left( \delta p \frac{\partial RT}{\partial x} + RT \delta B \frac{\partial \Phi}{\partial x} \right) \sin(\lambda + \Theta t) \sin \theta \right] \quad \text{cumulated} \\
\text{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 \quad \text{cumulated} \\
\text{TA1NONAX} & \frac{\delta t}{g} \frac{\delta \Phi}{\delta t} a^2 \Omega^2 \sin \theta \cos \sin(\lambda + \Theta t) \quad \text{cumulated} \\
\text{TA2NONAX} & \frac{\delta t}{g} \frac{\delta \Phi}{\delta t} a^2 \Omega^2 \sin \theta \cos \cos(\lambda + \Theta t) \quad \text{cumulated} \\
\text{FA1GRAV} & - \frac{\delta p}{g} a \left[ \sin \theta \cos(\lambda + \Theta t) \frac{\partial \Phi}{\partial x} + \sin(\lambda + \Theta t) \frac{\partial \Phi}{\partial y} \right] \quad \text{cumulated} \\
\text{FA2GRAV} & - \frac{\delta p}{g} a \left[ \sin \theta \cos(\lambda + \Theta t) \frac{\partial \Phi}{\partial x} + \cos(\lambda + \Theta t) \frac{\partial \Phi}{\partial y} \right] \quad \text{cumulated} \\
\text{FA3GRAV} & \frac{\delta p}{g} a \cos \theta \frac{\partial \Phi}{\partial x} \quad \text{cumulated} \\
\text{FA1FLUDUAPLUI} & \delta t \delta_m F_p M_1 \quad \text{cumulated} \\
\text{FA2FLUDUAPLUI} & \delta t \delta_m F_p M_2 \quad \text{cumulated} \\
\text{FA3FLUDUAPLUI} & \delta t \delta_m F_p M_3 \quad \text{cumulated} \\
\text{TA1TUR} & \delta t a \left[ F_{\text{v conv}} \sin(\lambda + \Theta t) - F_{\text{u conv}} \sin \theta \cos(\lambda + \Theta t) \right] \quad \text{cumulated} \\
\text{TA2TUR} & \delta t a \left[ F_{\text{v conv}} \cos(\lambda + \Theta t) - F_{\text{u conv}} \sin \theta \sin(\lambda + \Theta t) \right] \quad \text{cumulated} \\
\text{TA3TUR} & \delta t a F_{\text{u conv}} \cos \theta \quad \text{cumulated} \\
\text{TA1TURCONV} & \delta t a \left[ F_{\text{v conv}} \sin(\lambda + \Theta t) - F_{\text{u conv}} \sin \theta \cos(\lambda + \Theta t) \right] \quad \text{cumulated} \\
\text{TA2TURCONV} & \delta t a \left[ F_{\text{v conv}} \cos(\lambda + \Theta t) - F_{\text{u conv}} \sin \theta \sin(\lambda + \Theta t) \right] \quad \text{cumulated} \\
\text{TA3TURCONV} & \delta t a F_{\text{u conv}} \cos \theta \quad \text{cumulated} \\
\text{TA1ONDEGREE} & \delta t a \left[ F_{\text{v conv}} \sin(\lambda + \Theta t) - F_{\text{u conv}} \sin \theta \cos(\lambda + \Theta t) \right] \quad \text{cumulated} \\
\text{TA2ONDEGREE} & \delta t a \left[ F_{\text{v conv}} \cos(\lambda + \Theta t) - F_{\text{u conv}} \sin \theta \sin(\lambda + \Theta t) \right] \quad \text{cumulated} \\
\text{TA3ONDEGREE} & \delta t a F_{\text{u conv}} \cos \theta \quad \text{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:

\[
\cos \theta_0 = \frac{M_3}{\sqrt{M_1^2 + M_2^2 + M_3^2}}
\]

\[
\cos(\lambda_0 + \Omega t) = \frac{M_1}{\sqrt{M_1^2 + M_2^2}}
\]

4.7 Entropy budget

**Budget equation**

\[
\frac{\partial}{\partial t} (r_\eta s) = -\text{div}_\eta (r_\eta s \vec{v}) - \frac{\partial}{\partial \eta} (r_\eta s \vec{v}) + \frac{\partial}{\partial \eta} \left[ s_l \left( F_p^{\text{conv-l}} + F_p^{\text{stra-l}} \right) + s_g \left( F_p^{\text{conv-n}} + F_p^{\text{stra-n}} \right) \right] \\
- \frac{1}{T} \vec{v} \cdot \frac{\partial}{\partial \eta} \left( F_v^{\text{tur}} + F_v^{\text{tur-conv}} + F_v^{\text{rel}} \right) - (s_v - s_a + c_{pv} - c_{pa}) \frac{\partial}{\partial \eta} \left( F_q^{\text{tur}} + F_q^{\text{tur-conv}} \right) \\
+ \frac{1}{T} (F_h^{\text{sol}} + F_h^{\text{ther}} + F_h^{\text{turn}} + F_h^{\text{turn-conv}}) - (1 - \delta_m) \left[ s_a \frac{\partial F_p}{\partial \eta} + \frac{1}{T} c_{pa} \frac{\partial T}{\partial \eta} F_p \right] + \delta_m \frac{1}{T} F_p \frac{\partial \Phi}{\partial \eta}
\]

where

- \( r_\eta = -\frac{1}{g} \frac{\partial p}{\partial \eta} \)
- \( s = s_a + (s_v - s_a) q_v \)
- \( s_a = c_{pa} \ln \left( \frac{T}{T_0} \right) - R_a \ln \left( \frac{\rho_a}{\rho_0} \right) + s_0^a \)
- \( s_v = c_{pv} \ln \left( \frac{T}{T_0} \right) - R_v \ln \left( \frac{\rho_v}{\rho_0} \right) + s_0^v \)
- \( s_l = c_l \ln \left( \frac{T}{T_0} \right) + s_0^l \)
- \( s_g = c_g \ln \left( \frac{T}{T_0} \right) + s_0^g \)
- \( s_0^a = 6775 \text{ J kg}^{-1} \text{ K}^{-1} \)
- \( s_0^v = 10320 \text{ J kg}^{-1} \text{ K}^{-1} \)
- \( s_0^l = 3517 \text{ J kg}^{-1} \text{ K}^{-1} \)
- \( s_0^g = 2296 \text{ J kg}^{-1} \text{ K}^{-1} \)

The horizontal divergence term is computed as

\[
\text{div}_\eta (r_\eta s \vec{v}) = s \text{div}_\eta (r_\eta \vec{v}) + r_\eta \vec{v} \cdot \vec{\nabla} s
\]

where

- \( \vec{v} \cdot \vec{\nabla} s = (s_v - s_a) \vec{v} \cdot \vec{\nabla} q_v + \vec{v} \cdot \left[ c_p \vec{\nabla} \ln T - R \vec{\nabla} \ln p \right] \)
- \( \vec{v} \cdot \vec{\nabla} \ln p = RT \vec{v} \cdot \vec{\nabla} \pi \).
4.8 Potential energy budget

**Budget equation**

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

**File output**

Three of these budget terms have already been mentioned, see \( \langle \text{kinetic energy budget} \rangle \) et \( \langle \text{enthalpy budget} \rangle \). Is added

- VEP0
  \[ \frac{1}{g} \Phi \delta p \ (t=0) \]
- VEP1
  \[ \frac{1}{g} \Phi \delta p \ (t=\text{NSTEP } \Delta t) \]
- TEPDIVFLUHOR
  \[ - \frac{\delta t}{g} \text{div}_{\eta} (\Phi \delta \tilde{p} \tilde{v}) \]
- FEPCONVERSIFL
  \[ - \frac{\delta t}{g} \Phi \ell \sum_{k=1}^{\ell} \text{div}_{\eta} (\delta \tilde{p} \tilde{v}) \]
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 \( \text{PITM} \) (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:**
  \[
  \frac{\partial c_{ms} T_s}{\partial t} = \delta_{\text{terre}} \left\{ F_{h_s}^{\text{sol}} + F_{h_s}^{\text{ther}} + L_v(T_s) E_l + L_n(T_s) E_n + F_{csa} - F_{esp} - L_{\text{fonce}} F_{\text{fonte}} \right\}
  \]
  where \( c_{ms} \) is the surfacic calorific capacity of the surface layer. A constant value is used here: \( 1/\text{HSOL} \).

- **Deep temperature:**
  \[
  \frac{\partial c_{mp} T_p}{\partial t} = \delta_{\text{terre}} F_{esp}
  \]
  where \( c_{mp} \) is the surfacic calorific capacity of the deep layer. A constant value is used here: \( \text{RTINER}/\text{HSOL} \).

- **Surface water content:**
  \[
  \frac{\partial w_s}{\partial t} = \delta_{\text{terre}} \left\{ F_{p}^{\text{conv}} - l_{p} + F_{\text{stra}} - l_{p} + E_{l} - F_{\text{perco}} + F_{\text{fonte}} - F_{\text{ruiss}} \right\}
  \]

- **Deep water content:**
  \[
  \frac{\partial w_p}{\partial t} = \delta_{\text{terre}} \left\{ F_{\text{perco}} - F_{\text{ruisp}} \right\}
  \]

- **Surface snow content:**
  \[
  \frac{\partial w_n}{\partial t} = \delta_{\text{neige}} \delta_{\text{terre}} \left\{ F_{p}^{\text{conv}} - n + F_{p}^{\text{stra}} - n + E_{n} - F_{\text{fonte}} \right\}
  \]

**File output**

The fields are written on 3 file articles:

- **Variables at \( t = 0 \): article \( \text{SV0} \)**
  \[
  \frac{1}{\mathcal{D}} \sum_{(j,k) \in \mathcal{D}} \text{PITM} \sigma_{j,k} \text{ Land fraction}
  \]
  
  \begin{align*}
  c_{ms} T_s & \quad \text{Surface layer: energy} \\
  c_{mp} T_p & \quad \text{Deep layer: energy} \\
  w_s & \quad \text{Surface layer: water content (surfacic mass)} \\
  w_n & \quad \text{Surface layer: snow content (surfacic mass)} \\
  w_p & \quad \text{Deep layer: water content (surfacic mass)}
  \end{align*}

- **Variables at \( t = N \text{STEP} \times \delta t \): article \( \text{SV1} \)**
  The same articles as above.
• Flux cumulated from 0 to NSTEP: \( S_{\text{FVRAC}} \)

\[
\begin{align*}
F_{h_{\text{sol}}} & \quad \text{Solar radiation} \\
F_{h_{\text{ther}}} & \quad \text{Infra-red radiation} \\
L_{v}(T_s)E_l & \quad \text{Latent heat (water)} \\
L_{n}(T_s)E_n & \quad \text{Latent heat (snow)} \\
F_{\text{csa}} & \quad \text{Sensible heat at surface} \\
F_{\text{csp}} & \quad \text{Sensible heat between surface and deep layer} \\
L_{\text{fonde}}F_{\text{fonde}} & \quad \text{Surface flux due to melting Flux de chaleur en surface lié à la fonte de neige} \\
F_{p}^{\text{stra}} \quad & \text{Precipitation: resolved, rain} \\
F_{p}^{\text{stra}} \quad & \text{Precipitation: resolved, snow} \\
F_{p}^{\text{conv}} \quad & \text{Precipitation: subgrid-scale, rain} \\
F_{p}^{\text{conv}} \quad & \text{Precipitation: subgrid-scale, snow} \\
E_{l} & \quad \text{Surface evaporation (water)} \\
E_{n} & \quad \text{Surface evaporation (snow)} \\
F_{\text{perco}} & \quad \text{Percolation from surface to deep} \\
F_{\text{fonte}} & \quad \text{Snow melting} \\
F_{\text{ruiss}} & \quad \text{Surface run-off} \\
F_{\text{ruisp}} & \quad \text{Deep run-off} \\
\end{align*}
\]

4.10 Instantaneous diagnostics

4.10.1 Relative humidity

\[
\begin{align*}
\text{VHR0} & \quad \frac{1}{g} \bar{H}_r \delta p \quad (t=0) \\
\text{VHR1} & \quad \frac{1}{g} \bar{H}_r \delta p \quad (t=\text{NSTEP} \delta t)
\end{align*}
\]

4.10.2 Cloudiness

\[
\begin{align*}
\text{VNT0} & \quad \frac{1}{g} n_t \delta p \quad (t=0) \\
\text{VNT1} & \quad \frac{1}{g} n_t \delta p \quad (t=\text{NSTEP} \delta t)
\end{align*}
\]
4.10.3 Vertical velocity

\[ \frac{1}{g} \omega \delta p \ (t=0) \]

\[ \frac{1}{g} \omega \delta p \ (t=NSTEP \delta t) \]

4.10.4 Passive variables

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

\[ \frac{1}{g} v_{xx} \delta p \ (t=0) \]

\[ \frac{1}{g} v_{xx} \delta p \ (t=NSTEP \delta t) \]

where \( xx \) is between 1 and \( \text{NFPASS} \).

4.11 Cumulated mass

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

\[ \frac{1}{g} \delta t \delta p \quad \text{cumulated} \]
Chapter 5

Budgets, AROME model

5.1 Microphysical process

In AROME physical parameterizations from MesoNH are used. The greatest difference compared to parameterizations used in ALADIN appears in microphysics. Microphysical processes in AROME with corresponding tendencies/fluxes are given in table 5.1. Each column belongs to one microphysical species; tendencies/fluxes written in it change its species amount. The last column shows tendencies/fluxes that change enthalpy due to microphysical processes. On output from MesoNH subroutines, only tendencies are available, but they are converted into fluxes at a later stage if LBUFLUX is .TRUE..
<table>
<thead>
<tr>
<th>Microphysical process</th>
<th>vapour</th>
<th>cloud water</th>
<th>rain</th>
<th>cloud ice</th>
<th>snow</th>
<th>graupel</th>
<th>enthalpy</th>
</tr>
</thead>
<tbody>
<tr>
<td>water vapour adjustment</td>
<td>$F^{cdepi}_{q_v}$</td>
<td>$F^{cdepi}_{q_l}$</td>
<td>$F^{cdepi}_{q_l}$</td>
<td>$F^{cdepi}_{h}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>heterogeneous nucleation</td>
<td>$F^{henu}_{q_v}$</td>
<td>$F^{henu}_{q_l}$</td>
<td>$F^{henu}_{h}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>homogeneous nucleation</td>
<td>$F^{hon}_{q_l}$</td>
<td>$F^{hon}_{q_l}$</td>
<td>$F^{hon}_{h}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>spontaneous freezing</td>
<td>$F^{deps}_{q_v}$</td>
<td>$F^{deps}_{q_s}$</td>
<td>$F^{deps}_{h}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>deposition on snow</td>
<td>$F^{deps}_{q_v}$</td>
<td>$F^{deps}_{q_s}$</td>
<td>$F^{deps}_{h}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>collection of ice on snow</td>
<td>$F^{agg}_{q_i}$</td>
<td>$F^{agg}_{q_s}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>auto-conversion of ice to snow</td>
<td>$F^{auto}_{q_i}$</td>
<td>$F^{auto}_{q_s}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>deposition on graupel</td>
<td>$F^{depg}_{q_v}$</td>
<td>$F^{depg}_{q_s}$</td>
<td>$F^{depg}_{h}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>auto-conversion of cloud water</td>
<td>$F^{autor}_{q_i}$</td>
<td>$F^{autor}_{q_s}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>accretion</td>
<td>$F^{accr}_{q_l}$</td>
<td>$F^{accr}_{q_s}$</td>
<td>$F^{accr}_{h}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>rain evaporation</td>
<td>$F^{reva}_{q_v}$</td>
<td>$F^{reva}_{q_s}$</td>
<td>$F^{reva}_{h}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>riming by cloud droplets</td>
<td>$F^{rim}_{q_l}$</td>
<td>$F^{rim}_{q_s}$</td>
<td>$F^{rim}_{h}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>collection of raindrops</td>
<td>$F^{accs}_{q_s}$</td>
<td>$F^{accs}_{q_s}$</td>
<td>$F^{accs}_{h}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>melting of aggregates</td>
<td>$F^{cmel}_{q_s}$</td>
<td>$F^{cmel}_{q_s}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>contact freezing</td>
<td>$F^{cfrz}_{q_i}$</td>
<td>$F^{cfrz}_{q_i}$</td>
<td>$F^{cfrz}_{q_g}$</td>
<td>$F^{cfrz}_{h}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>wet growth</td>
<td>$F^{wetg}_{q_i}$</td>
<td>$F^{wetg}_{q_i}$</td>
<td>$F^{wetg}_{q_g}$</td>
<td>$F^{wetg}_{h}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>dry growth</td>
<td>$F^{dryg}_{q_i}$</td>
<td>$F^{dryg}_{q_i}$</td>
<td>$F^{dryg}_{q_g}$</td>
<td>$F^{dryg}_{h}$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>melting of graupel</td>
<td>$F^{mltg}_{q_i}$</td>
<td>$F^{mltg}_{q_i}$</td>
<td>$F^{mltg}_{h}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>melting of cloud ice</td>
<td>$F^{mlti}_{q_i}$</td>
<td>$F^{mlti}_{q_i}$</td>
<td>$F^{mlti}_{h}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bergeron-Findeisen effect</td>
<td>$F^{berfi}_{q_i}$</td>
<td>$F^{berfi}_{q_i}$</td>
<td>$F^{berfi}_{h}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 5.1: Microphysical fluxes.
5.2 Balance equations

5.2.1 Momentum

- $F_{u}^{\text{tur}}$ vertical turbulence flux of $u$ velocity component
- $F_{v}^{\text{tur}}$ vertical turbulence flux of $v$ velocity component
- $F_{w}^{\text{tur}}$ vertical turbulence flux of $u$ velocity component

5.2.2 Turbulence kinetic energy

- $F_{\text{tke}}^{\text{tur}}$ turbulent flux of turbulent kinetic energy
- $F_{\text{tke}}^{\text{tur-prod-dyn}}$ dynamic production of turbulent kinetic energy
- $F_{\text{tke}}^{\text{tur-prod-term}}$ thermic production of turbulent kinetic energy
- $F_{\text{tke}}^{\text{tur-diss}}$ dissipation of turbulent kinetic energy
5.2.3 Enthalpy

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

- FCTCFRZ \( F_{h}^{cfroz} \): contact freezing of rain
- FCTWETG \( F_{h}^{wetg} \): wet growth of graupel
- FCTDRYG \( F_{h}^{dryg} \): dry growth of graupel
- FCTMLTG \( F_{h}^{mltg} \): melting of graupel
- FCTMLTI \( F_{h}^{mlti} \): melting of cloud ice
- FCTBERFI \( F_{h}^{berfi} \): Bergeron-Findeisen effect
- FCTRAYSOL1 \( F_{h}^{raysol1} \): solar radiation
- FCTRAYER1 \( F_{h}^{rayter1} \): earth radiation
5.2.4 Water vapour

\[ F_{q_v}^{\text{negc}} \] correction of negative specific ratios after advection
\[ F_{q_v}^{\text{depi}} \] adjustment of water vapour, cloud water and cloud ice
\[ F_{q_v}^{\text{conv}} \] convection flux of water vapour
\[ F_{q_v}^{\text{tur}} \] vertical turbulent flux of water vapour
\[ F_{q_v}^{\text{negc}} \] correction of negative specific ratios after turbulence
\[ F_{q_v}^{\text{henu}} \] heterogeneous nucleation of ice
\[ F_{q_v}^{\text{deps}} \] deposition on snow
\[ F_{q_v}^{\text{depg}} \] deposition on groupel
\[ F_{q_v}^{\text{reva}} \] rain evaporation

5.2.5 Cloud water

\[ F_{q_i}^{\text{negec}} \] correction of negative specific ratios after advection
\[ F_{q_i}^{\text{depi}} \] adjustment of water vapour, cloud water and cloud ice
\[ F_{q_i}^{\text{conv}} \] convection flux of cloud water
\[ F_{q_i}^{\text{tur}} \] vertical turbulent flux of cloud water
\[ F_{q_i}^{\text{negc}} \] correction of negative specific ratios after turbulence
\[ F_{q_i}^{\text{hon}} \] homogeneous nucleation of ice
\[ F_{q_i}^{\text{autor}} \] auto-conversion of cloud water
\[ F_{q_i}^{\text{accr}} \] accretion of cloud water on rain
\[ F_{q_i}^{\text{rim}} \] riming by cloud droplets
\[ F_{q_i}^{\text{wetg}} \] wet growth of graupel
\[ F_{q_i}^{\text{dryg}} \] dry growth of graupel
\[ F_{q_i}^{\text{mlti}} \] melting of cloud ice
\[ F_{q_i}^{\text{berfi}} \] Bergeron-Findeisen effect
5.2.6 Rain

- FQRNEG C $F_{\text{negc}}^r$ correction of negative specific ratios after advection
- FQRSEDI $F_{\text{se}}$ sedimentation
- FQRSFR $F_{\text{frz}}^r$ spontaneous freezing of rain
- FQRAUTO $F_{\text{auto}}^r$ auto-conversion of cloud water
- FQRACCL $F_{\text{accl}}^r$ accretion of cloud water on rain
- FQRREVA $F_{\text{reva}}^r$ rain evaporation
- FQRACCS $F_{\text{accs}}^r$ collection of raindrops on graupel
- FQRCFRZ $F_{\text{frz}}^r$ contact freezing of rain
- FQRWETG $F_{\text{wetg}}^r$ wet growth of graupel
- FQDRYG $F_{\text{dryg}}^r$ dry growth of graupel
- FQRMULT $F_{\text{mltg}}^r$ melting of graupel

5.2.7 Cloud ice

- FQINEG CI $F_{\text{negci}}^i$ correction of negative specific ratios after advection
- FQICDEPI $F_{\text{depi}}^i$ adjustment of water vapour, cloud water and cloud ice
- FQICONV $F_{\text{conv}}^i$ convection flux of cloud ice
- FQITURB $F_{\text{tur}}^i$ vertical turbulent flux of cloud ice
- FQINEG C $F_{\text{negci}}^i$ correction of negative specific ratios after turbulence
- FQISEDI $F_{\text{ip}}^i$ sedimentation
- FQIHENU $F_{\text{henu}}^i$ heterogeneous nucleation of ice
- FQIHON $F_{\text{hon}}^i$ homogeneous nucleation of ice
- FQIAGGS $F_{\text{agg}}^i$ collection of ice on snow
- FQIAUTS $F_{\text{autoi}}^i$ auto-conversion of ice to snow
- FQICFRZ $F_{\text{frzi}}^i$ contact freezing of rain
- FQIWETG $F_{\text{wetg}}^i$ wet growth of graupel
- FQIDRYG $F_{\text{dryg}}^i$ dry growth of graupel
- FQIMLT $F_{\text{mlti}}^i$ melting of cloud ice
- FQIBERFI $F_{\text{berfi}}^i$ Bergeron-Findeisen effect
### 5.2.8 Snow

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Equation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FQSNegc</td>
<td>$F^\text{negc}_{qs}$</td>
<td>correction of negative specific ratios after advection</td>
</tr>
<tr>
<td>FQSedi</td>
<td>$F_{xp}$</td>
<td>sedimentation</td>
</tr>
<tr>
<td>FQSDEPS</td>
<td>$F^\text{dep}_{qs}$</td>
<td>deposition on snow</td>
</tr>
<tr>
<td>FQSAGGS</td>
<td>$F^\text{agg}_{qs}$</td>
<td>collection of ice on snow</td>
</tr>
<tr>
<td>FQSAUTS</td>
<td>$F^\text{autoi}_{qs}$</td>
<td>auto-conversion of ice to snow</td>
</tr>
<tr>
<td>FQSrim</td>
<td>$F^\text{rim}_{qs}$</td>
<td>riming by cloud droplets</td>
</tr>
<tr>
<td>FQSACC</td>
<td>$F^\text{acc}_{qs}$</td>
<td>collection of raindrops and snow on graupel</td>
</tr>
<tr>
<td>FQSCMEL</td>
<td>$F^\text{cmel}_{qs}$</td>
<td>melting of aggregates</td>
</tr>
<tr>
<td>FQSWETG</td>
<td>$F^\text{wetg}_{qs}$</td>
<td>wet growth of graupel</td>
</tr>
<tr>
<td>FQSDRYG</td>
<td>$F^\text{dryg}_{qs}$</td>
<td>dry growth of graupel</td>
</tr>
</tbody>
</table>

### 5.2.9 Graupel

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Equation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FQGnegc</td>
<td>$F^\text{negc}_{gq}$</td>
<td>correction of negative specific ratios after advection</td>
</tr>
<tr>
<td>FQGSedi</td>
<td>$F_{gp}$</td>
<td>sedimentation</td>
</tr>
<tr>
<td>FQGSFR</td>
<td>$F^\text{sfr}_{gq}$</td>
<td>spontaneous freezing</td>
</tr>
<tr>
<td>FQGDEPG</td>
<td>$F^\text{dep}_{gq}$</td>
<td>deposition on gropel</td>
</tr>
<tr>
<td>FQGRIM</td>
<td>$F^\text{rim}_{gq}$</td>
<td>riming by cloud droplets</td>
</tr>
<tr>
<td>FQGACC</td>
<td>$F^\text{acc}_{gq}$</td>
<td>collection of raindrops and snow on graupel</td>
</tr>
<tr>
<td>FQGCHEL</td>
<td>$F^\text{cmel}_{gq}$</td>
<td>melting of aggregates</td>
</tr>
<tr>
<td>FQGCFRZ</td>
<td>$F^\text{cfrz}_{gq}$</td>
<td>contact freezing of rain</td>
</tr>
<tr>
<td>FQGWETG</td>
<td>$F^\text{wetg}_{gq}$</td>
<td>wet growth of graupel</td>
</tr>
<tr>
<td>FQGDRYG</td>
<td>$F^\text{dryg}_{gq}$</td>
<td>dry growth of graupel</td>
</tr>
<tr>
<td>FQGMLT</td>
<td>$F^\text{mltg}_{gq}$</td>
<td>melting of graupel</td>
</tr>
</tbody>
</table>
5.3 Common Dynamics-Physics Interface CDPI

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>FQVPL1 $P'_i$</td>
<td>Pseudo flux due to condensation</td>
</tr>
<tr>
<td>FQVPI1 $P'_i$</td>
<td>Pseudo flux due to sublimation</td>
</tr>
<tr>
<td>FQLPL2 $P''_i$</td>
<td>Pseudo flux due to evaporation of rain</td>
</tr>
<tr>
<td>FQIPI2 $P''_i$</td>
<td>Pseudo flux due to conversion of cloud ice to snow and graupel</td>
</tr>
<tr>
<td>FQRPL3 $P'''_i$</td>
<td>Pseudo flux due to evaporation of rain</td>
</tr>
<tr>
<td>FQSPI3 $P''''_i$</td>
<td>Pseudo flux due to deposition on snow and graupel</td>
</tr>
<tr>
<td>FQGPG3 $P''''_g$</td>
<td>Pseudo flux due to deposition on snow and graupel</td>
</tr>
<tr>
<td>FQRPR0 $P_r$</td>
<td>Flux of falling rain drops</td>
</tr>
<tr>
<td>FQIPI0 $P_i$</td>
<td>Flux of falling cloud ice</td>
</tr>
<tr>
<td>FQSPS0 $P_s$</td>
<td>Flux of falling snow</td>
</tr>
<tr>
<td>FQGPG0 $P_g$</td>
<td>Flux of falling graupel</td>
</tr>
</tbody>
</table>
Chapter 6

Software maintenance

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

6.1 Main arrays and their organization

The main arrays are in two modules:

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

6.1.1 Arrays describing domains

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

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

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

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

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

- NLRDDH (NDHDDX, NDHKD) : integers contained in NLRDDH, from NLRDDH(1, JKD) to NLRDDH(NLXDDH(JKD), JKD), are the internal domains whose reunion makes the latitude band JKD.
• HDSFLA (NDHKD) : weight of each latitude band.

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

• HDSFDU (0 : NDHBPX, NDHNPU) : Weight of each user’s domain.

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

### 6.1.2 Data arrays

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

HDCV1 (0 : NFLEV, NDHCV, NDHIDH, NDHTSK) alias HDCVB1 (NDHCV * (NFLEV+1), NDHIDH, NDHTSK): variables at time NSTEP and tendencies/fluxes cumulated from 0 to NSTEP.

HDCS0 (NDHCS, NDHIDH, NDHTSK): soil variables at time 0 and cumulated fluxes from 0 to NSTEP-1.

HDCS1 (NDHCS, NDHIDH, NDHTSK): soil variables at time NSTEP and cumulated fluxes from 0 to NSTEP.

• NDHIDH: number of internal domains.

• NDHTSK: number of tasks.

• NDHCV: total number of vertical profiles.

• NDHCS: total number of surface fields.

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

• Permanent pointers who, for each scientific options, show the number of fields of each categories. For the moment, the options are LHDHKS, LHDNCI 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 CPDYYDDH and CPPHDDH, and to writing lines or other editions in PPFDDH and PPFIDH. A corresponding commentary is in the general nomenclature of Y0MTDDH. Situation pointers are also initialized in SUNDDH from permanent pointers and from logical options given by NAMDDH. They control global lengths in CPG, POSDDH and PPSYDH where fields are undifferentiated.

Permanent pointers are
6.1. MAIN ARRAYS AND THEIR ORGANIZATION

- **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 LHDLCI

- **NDHVMC**
- **NDHFMCMD**
- **NDHFMCPP**
- **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

\[
\text{NDHFxxD} = \text{NDHAxxD} + \text{NDHBxxD}
\]
\[
\text{NDHFxxP} = \text{NDHAxxP} + \text{NDHBxxP}
\]

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
### 6.1.3 Main local arrays

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

#### 6.1.3.1 Arrays for computations at each grid point

<table>
<thead>
<tr>
<th>Champs</th>
<th>each field has NFLEV+1 levels</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>instantiate values</td>
</tr>
<tr>
<td>...</td>
<td>variables</td>
</tr>
<tr>
<td>...</td>
<td>dynamical tendencies</td>
</tr>
<tr>
<td>NDHVV</td>
<td>dynamical fluxes</td>
</tr>
<tr>
<td>NDHVV+1</td>
<td>physical fluxes</td>
</tr>
<tr>
<td>...</td>
<td>physical tendencies</td>
</tr>
</tbody>
</table>

For every categories (variables, dynamical tendency, dynamical flux, etc) one may find fields linked to LHDKH (eventually) then those of LHDMC (eventually) then those of LHDE (eventually). Soil arrays work on the same principle. Fields from 1 to NDHV+NDHF are computed in CPDYDDH. Fields from NDHV+NDHF to NDHCV are computed in CPPHDDH.

- **KPRROMA**: maximum number of points in the horizontal, by grid point task,
- **NDHCVSU** = max(1, NDHCV)
- **NDHCS** = max(1, NDHCS)
---SU lengths ensure that d’ARPEGE is properly working even if NDHCV=0 or NDHCS=0 (case(s) where DDH diagnostics are not activated).

### 6.1.3.2 Arrays for synthesis over a user domain

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

ZDHCS (NDHCSSU+NDHVS) soil fields.

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

<table>
<thead>
<tr>
<th>Field</th>
<th>1</th>
<th>...</th>
<th>variables at t</th>
</tr>
</thead>
<tbody>
<tr>
<td>NDHVV</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NDHVV+1</td>
<td></td>
<td></td>
<td>tendencies then fluxes</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td>dynamical, cumulated</td>
</tr>
<tr>
<td>NDHVV+NDHFVD</td>
<td></td>
<td></td>
<td>from 0 to NSTEP-1</td>
</tr>
<tr>
<td>NDHVV+NDHFVD+1</td>
<td></td>
<td></td>
<td>fluxes, then tendencies</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td>dynamical, cumulated</td>
</tr>
<tr>
<td>NDHCV</td>
<td></td>
<td></td>
<td>from 0 to NSTEP-1</td>
</tr>
<tr>
<td>NDHCV+1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td>variables at t=NSTEP*TSTEP</td>
</tr>
<tr>
<td>NDHCV+NDHVV</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

ZDHCV array structure.

### 6.2 Organization of the main functions

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

- (—) name of the sub programme between parenthesis: sub programme whose main function is not to compute diagnostics; generally speaking, sub programme ARPEGE already existing.

- multitask sub programme.

- tci specific DDH sub programme in which all fields are undifferentiated

- 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: $(\lambda_q, \mu_q)$ in each point.
- SUMDDH, verification and set up of domain declarations (from BDEDHH 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: $\text{NDHTSK} = \text{NSLBR} - \text{NDGSA} + 1$.
- 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 ZEROODDH, 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) $m$, multi task interface of grid point computations, calls CPG $m$. 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 $m$ $cci$, 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 $m$ $cci$, computation in every points of fluxes and of tendencies due to physical parametrizations, soil computation, and possible call to CPVRDH (if the verification option is activated).
  - Call to CPCUDDH $m$ $tci$, 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, final horizontal means for a user domain, edition of arrays ZDHCV and ZDHCS, multiplication of variables by par $1/(gSD)$ and of cumulated fields by $(\delta t/SD)$.
  - PPEDDH, vertical mean budget edition.
  - PPFIDH, 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.
CHAPTER 6. SOFTWARE MAINTENANCE

6.3 How to add new fields to budgets

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

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

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

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

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

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

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

6.4 Maintenance specificities, AROME model

6.4.1 New module YOMPHFT

The procedure for adding a new term in DDH in ARPEGE/ALADIN has been described above. Namely, for each new term one line of program code must be added in CPDYDDH or in CPPHDDH and two lines in PPFIDH. In AROME many new fluxes/tendencies are introduced, and many old ones are not used. Adding new lines for each new term would make a quite unreadable code. For this reason and to avoid carrying fluxes/tendencies through subroutines a new module YOMPHFT is introduced. In this module a data type, TYPE_APFT, is defined to carry a description of data.

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

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

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

Dimensions of APFT are:

- 1. dimension: horizontal extend,
- 2. dimension: vertical extend,
- 3. dimension: NAPHFT (described later).
A dimension of YAPFT is NAPHFT; description in YAPFT must be at the same index value as the third index of the corresponding flux/tendency in APFT.

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

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

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

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

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

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

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

```fortran
INTEGER(KIND=JPIM) :: NCDPIPHFT ! number of fluxes/tendencies in CDPI,
INTEGER(KIND=JPIM) :: NQVL1 ! positions of Pl'
INTEGER(KIND=JPIM) :: NQVI1 ! positions of Pi'
INTEGER(KIND=JPIM) :: NQLR2 ! positions of Pl''
INTEGER(KIND=JPIM) :: NQIS2 ! positions of Pi''
INTEGER(KIND=JPIM) :: NQRV3 ! positions of Pl'''
INTEGER(KIND=JPIM) :: NQSV3 ! positions of Pi'''
INTEGER(KIND=JPIM) :: NQGV3 ! positions of Pg'''
INTEGER(KIND=JPIM) :: NR0 ! positions of Pr - flux of rain
INTEGER(KIND=JPIM) :: NQI0 ! positions of Pi - flux of cloud ice
INTEGER(KIND=JPIM) :: NSQ0 ! positions of Ps - flux of snow
INTEGER(KIND=JPIM) :: NQG0 ! positions of Pg - flux of graupel
```

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

- for positions of momentum turbulent fluxes

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

- for positions of convective fluxes

```fortran
INTEGER(KIND=JPIM) :: NQLCONV ! third index in APFT for convective rain
INTEGER(KIND=JPIM) :: NQNCONV ! third index in APFT for convective snow
```
CHAPTER 6. SOFTWARE MAINTENANCE

6.4.2 Data flow

Due to the use of the module YOMPHFT, data flow for DDH data in AROME is simplified compared to ARPEGE/ALADIN. Physical fluxes and tendencies are saved in array APFT within subroutine APL_AROME and used in subroutine CPPHDDH. This is shown in Fig. 6.1.

6.4.3 Setup

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

6.4.3.1 Subroutine SU0YOMA

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

6.4.3.2 Subroutine SU0YOMB

In SUPHMNH, called in SUPHY, control variable LAROBU_ENABLE is initialized. Subroutines ARO_SUBBUDGET and AROINI_BUDGET are called to initialize budget arrays and to save physical tendencies in budget arrays within NMH subroutines. Also, ARO_INIAPFT is called to do a part of initializations needed for DDH in AROME (see 6.4.7.3).

6.4.4 Entering physical fluxes and tendencies in DDH

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

6.4.5 Modified subroutines

6.4.5.1 Subroutine APL_AROME

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

To put two radiation fluxes in APFT, the following two lines are added at the end of the radiation parameterizations: ZFRSO and ZFRTH

\[
\begin{align*}
APFT(JLON,JLEV,\text{MJJ2}(4) -1) &= ZFRSO(JLON,JLEV,1) \\
APFT(JLON,JLEV,\text{MJJ2}(4)) &= ZFRTH(JLON,JLEV,1)
\end{align*}
\]
Convection parameterization fluxes are not saved in budget arrays within MNH subroutines, but this is done in subroutine ARO_CONVBU, added at the end of convection calculations.

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

### Subroutine AROEND_BUDGET

AROEND_BUDGET is an interface subroutine between the ARPEGE/ALADIN code and the MNH code. Its purpose is to save physical fluxes/tendencies in the array APFT. MNH subroutines save them in budget arrays as tendencies. If LDLFLUX is .TRUE., fluxes are calculated from tendencies and saved in APFT, else, tendencies are saved. If (LDAROCDP1 OR LDHDCDPI) is .TRUE., CDPI fluxes and pseudo-fluxes are calculated and saved in APFT.

### Subroutine CPDYDDH

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

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

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

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

6.4.5.5 Subroutine CPG

ZGPAR is added in call to CPG_DIA.

6.4.5.6 SU0PHY

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

6.4.5.7 Subroutine SUPHMNH

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

\[ \text{LAROBU\_ENABLE}=\text{LAROBU\_ENABLE}.\text{OR.}(\text{LSDDH}.\text{AND.}.\text{NOT.ONLYVAR}).\text{OR.LPHCDPI} \]

Then budget arrays are prepared to be used, if LAROBU_ENABLE is .TRUE..

\[
\begin{align*}
\text{CALL ARO\_SUBBUDGET(ILON,ILEV,ZTSTEP)} \\
\text{CALL AROINI\_BUDGET(LAROBU\_ENABLE,CLUOUT,KULOUT,ZTSTEP,ISV,IRR,} & \\
& \text{CLRAD,CLDCONV,CLTURB,CLTURBDIM, CLCLOUD)}
\end{align*}
\]

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

\[
\text{CALL ARO\_INIAPFT(LAROBU\_ENABLE)}
\]

6.4.5.8 Subroutine SUNDDH

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

\[
\begin{align*}
\text{LONLYVAR} & = \text{.TRUE.} \\
\text{LHDORIGP} & = \text{.TRUE.} \\
\text{LHDCDPI} & = \text{.FALSE.}
\end{align*}
\]

Other variables that must be changed are:

\[
\begin{align*}
\text{NHDQLNVA} & \quad \text{! total number of variables for water} \\
\text{NHDQLNTD} & \quad \text{! number of dynamical tendencies for water when LHDHKS is .TRUE.} \\
\text{NHDQLNFD} & \quad \text{! number of dynamical fluxes for water when LHDHKS is .TRUE.} \\
\text{NHDQLNFP} & \quad \text{! number of physical fluxes for water when LHDHKS is .TRUE.}
\end{align*}
\]
6.4. MAINTENANCE SPECIFICITIES, AROME MODEL

NHDQLNFP ! number of physical tendencies for water when LHDHKS is .TRUE.
NDHVK ! total number of variables when LHDHKS is .TRUE.
NDHAKD ! total number of dynamical tendencies when LHDHKS is .TRUE.
NDHBHKD ! total number of dynamical fluxes when LHDHKS is .TRUE.
NDHAKP ! total number of physical fluxes when LHDHKS is .TRUE.
NDHBHKP ! total number of physical tendencies when LHDHKS is .TRUE.

This is done in SELECT CASE sentences like this one:

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

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

6.4.5.9 Subroutine PPFIDH

New variables are added as described in ??, but within a SELECT CASE statement like this:

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

Physical fluxes are saved on file in the following way:

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

The advantage of DO loop is that for AROME this subroutine does not have to be changed when new terms are introduced in the equations.
6.4.6 Modified modules

6.4.6.1 Module YOMARPHY

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

![LOGICAL :: LBUFLUX](image)

\[
\text{LBUFLUX} : \begin{cases} \text{If TRUE fluxes are calculated in AROEND\_BUDGET,} \\
\text{if FALSE, tendencies remain} \end{cases}
\]

6.4.6.2 Module YOMLDDH

Three switches are added to control the content of output file:

- **LONLYVAR** gives a possibility to save either only variables or variables and all other terms.
- Two switches are connected to the possibility of having microphysical fluxes/tendencies in a form needed for CDPI (see ??). It is possible to save microphysical fluxes/tendencies as they are calculated in the physical package, as in CDPI, or both (see ??).

![LOGICAL :: LONLYVAR](image)

\[
\text{LONLYVAR} : \begin{cases} \text{if .TRUE. only variables are saved} \end{cases}
\]

![LOGICAL :: LHDORIGP](image)

\[
\text{LHDORIGP} : \begin{cases} \text{if .TRUE. original fluxes from microphysical parameterization are saved} \end{cases}
\]

![LOGICAL :: LHDCDPI](image)

\[
\text{LHDCDPI} : \begin{cases} \text{if .TRUE. CDPI fluxes are saved} \end{cases}
\]

6.4.6.3 Module YOMMNH

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

![LOGICAL :: LAROBU_ENABLE](image)

\[
\text{LAROBU\_ENABLE} : \begin{cases} \text{when it is .TRUE. tendencies are saved in budget arrays} \end{cases}
\]

(see ??)

![INTEGER(KIND=JPIM),ALLOCATABLE](image)

\[
\text{NBUPROC}() \\
\text{NJBUDG1()} \\
\text{NJBUDG2()} \\
\text{number of budgets used in AROME, without passive scalars} \\
\text{JPAROBUD = 12}
\]

6.4.6.4 Module YOMPHY

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

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

6.4.7 New subroutines

6.4.7.1 Subroutine ARO_CPPHDDH

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

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

6.4.7.2 Subroutine ADDFT

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

6.4.7.3 Subroutine ARO_INIPFT

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

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

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

The following main tasks are done in ARO_INIPFT:

1. Calculating a number of fluxes/tendencies and allocations

2. Descriptions of fluxes/tendencies are saved in array YAPFT. At the same time, and for each variable, first and last indexes in YAPFT are saved in arrays MJJ1 and MJJ2.

3. Matching APFT with budget arrays; the first and the last index in budget arrays are saved in NJBUDG1 and NJBUDG2.

If a MODULE interface is used, the positions of CPDI fluxes are determined.

### 6.4.8 Adding a new term in DDH for AROME

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

A. Adding a variable, dynamical flux or tendency or physical tendency: the same procedure as in ARPEGE/ALADIN.

B. Adding a physical flux

1. Do changes in subroutine SUNDDH (see 6.4.5.8),

2. Add a new term in subroutine ARO_INIPFT. First NAPHFT and NDDHFT must be changed. Then, the description of the new term must be put at the right place in YAPFT (see 6.4.7.3). Besides, changes in NBUPROC, NJBUDG1 and NJBUDG2 may be needed.
Chapter 7

Using DDH files: the ddhtoolbox

7.1 Purpose

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

7.2 Synoptic view of ddhtoolbox utilities

ARPEGE, ALADIN or AROME models produce DDH files. Each file contains n domains, m levels, and for each domain-level all variables, fluxes and tendencies relevant to get a budget. The actions that can be performed on DDH files are
• **ddhr**: get some DDH file autodocumentation on standard output: base, prediction range, etc.

• **ddht**: operate transforms on DDH files: make the difference between a reference and an experiment, cumulate several DDH files, extract one or more domain(s) from a DDH file, extract one or more level(s), make an horizontal mean on all domains, make a vertical mean on all levels, etc. ddht generates in output a DDH file.

• **ddhi**: makes an interpretation of the raw data from a DDH file, in order to get ready to plot data, with intensive units (K/day, g/kg, etc). ddhi produces in output ASCII files:
  - a ".doc" file per field, containing autodocumentation (title, unit, base, prediction range, etc)
  - a ".dta" file per field, containing the data in columns.

• **ddhb** is a script to make the budget of prognostic variables.

• Typing "ddhr" or "ddhi" or "ddht" or "ddhb" with no argument gives a documentation about the usage.

• Graphics: the user may then use his own graphic tool to plot the ".doc" and ".dta" files.

### 7.3 Install the software

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

Untar the ddhtoolbox.tar file.

cd ddhtoolbox

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

1. Put the local directory in your PATH: export PATH=.:$PATH

2. Run install process:
   ```
   install clean
   install
   ```

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

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

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

```
DDHC DDH.lfi DDH.lfa
```
7.4 Environment variables

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

2. DDH tools (ddhi, ddhb) use the following environment variables, to be put in your ".profile" or ".bash_profile" files:
   
   ```
   export DDHI_LIST=..../ddhtoolbox/ddh_budget_lists/conversion_list
   export DDHB_BPS=..../ddhtoolbox/ddh_budget_lists
   ```

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

3. If the environment variable DDH_PLOT is set, 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 one uses a script to visualize, called dd2met (written by Yves Bouteloup) based on METVIEW to make the plots. The ".profile" or ".bash_profile" thus contains: "export DDH_PLOT=dd2met"

7.5 ddhb: get budgets

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

Example of ddhb use:

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

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

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

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

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

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

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

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

to see the complete list of physics or budget packages available for use in ddhb, and which variables. The user may read these directive files to see what DDH file articles are used for each prognostic variable budget.
The user may also create his own directive files. For example, to change the legends of the budgets (and translate them to French, German, etc), or to customize the scientifical budgets (change the list of file articles to be used for a given physical process). In this case, copy an existing $DDHB_BPS/arpege/*.fbl file into your directory, and modify it. This file can then be used by ddhb, in typing

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

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

where your modified directive file is `home/piriou/my_ddh_modifs/arpege/QV.fbl`
Chapter 8

History

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


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


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

2006: Write AROME DDH documentation (Tomislav Kovacic).

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


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