HORIZONTAL DIFFUSION COMPUTATIONS IN THE CYCLE 44 OF ARPEGE/IFS.

YESSAD K. (METEO-FRANCE/CNRM/GMAP/ALGO)

March 6, 2017

Abstract:

This documentation recalls some basics about horizontal diffusion schemes, and describes how they are implemented in ARPEGE/IFS and its LAM (limited area model) counterpart. The ECMWF way of implementing them (which is slightly different from the METEO-FRANCE one) is not described in detail. There is a main horizontal diffusion scheme, a sponge and a Rayleigh friction. An organigramme is provided.

Résumé:

Cette documentation rappelle quelques notions sur les schémas de diffusion horizontale, et décrit comment ces schémas sont utilisés et codés dans ARPEGE/IFS et sa version à domaine limité. La version de ces codes utilisée au CEP n'est pas décrite en détail, dans la mesure ou elle diffère un peu de ce qui est utilisé à METEO-FRANCE. Il y a un schéma principal de diffusion horizontale, une éponge et une friction de Rayleigh.

Contents

1 Introduction.

∗ **General considerations for ARPEGE/IFS (global model):** Horizontal diffusion computations are generally spectral computations. There is a main horizontal diffusion scheme in spectral space. In case of constant resolution, the main horizontal diffusion scheme is discretised by a purely diagonal operator in spectral space. The main horizontal diffusion scheme is implicit in order to remain stable even with high diffusion coefficients and are called after the semi-implicit scheme. Horizontal diffusion is called after inversion of the semi-implicit scheme and before nudging.

∗ **Horizontal diffusion in LAM model:** Horizontal diffusion computations are still spectral computations. There is a main horizontal diffusion scheme only. For the main horizontal diffusion scheme, the formulation looks like the unstretched global model one. There are specific routines for spectral calculations.

∗ **Sponge:** An additional sponge may be done near the top of the model. This sponge is done in spectral space for spectral variables and in grid-point space for purely grid-point variables.

∗ **Rayleigh friction:** There is a Rayleigh friction in the grid-point space computations which acts only in upper atmosphere for the U-component (and optionally for the V-component) of the horizontal wind.

∗ **Pressure gradient term filtering:** A specific filter has been implemented to filter pressure gradient term in spectral space. This filter is currently available for unstretched untilted global models and LAM models.

∗ **Two-time level semi-Lagrangian scheme:** equations are written for a leap-frog scheme. The algorithm remains the same for a two-time level semi-Lagrangian scheme.

∗ **Deep layer equations:** For convenience, equations are written with the geographical horizontal gradient operator ∇, which is used in the horizontal diffusion scheme for thin layer equations. For deep layer equations, the operator actually used is $\left[\frac{r}{a}\nabla\right]$, where r is the actual radius, to simulate a "pseudo-geographical" diffusion, instead of ∇: this approximation is reasonable and allows to avoid any tricky modification in the horizontal diffusion spectral computations; use of the true geographical gradient ∇ sticks to the grid-point calculations.

2 Formulation of the horizontal diffusion schemes.

2.1 Main horizontal diffusion scheme.

∗ **General formulation:** The main horizontal diffusion formulation is close to:

$$\left(\frac{\partial X}{\partial t}\right)_{\mathrm{hd}} = -\mathcal{K}_{\mathrm{X}} M \nabla'^{r} X \tag{1}$$

where $\mathcal{K}_{\mathrm{X}}$ is a vertically dependent and horizontally constant coefficient. $\mathcal{K}_{\mathrm{X}}$ is generally complex: $\exp(0.5\pi ir)$ multiplied by a real positive coefficient. r is the power of the horizontal diffusion scheme. M is the mapping factor.

In global model:

$$\mathcal{K}_{\mathrm{X}} = \exp(-0.5\pi ir) \left[\sqrt{\frac{N_{\mathrm{s}}(N_{\mathrm{s}}+1)}{a^2}}\right]^{-r} \Omega \, h_{\mathrm{X}} \, g \tag{2}$$

(a is the mean Earth radius, N_{s} is the truncation). g is horizontally constant, vertically dependent. Ω is the angular velocity of the Earth rotation (0.00007292115 s^{-1}). h_{X} is a constant coefficient for each prognostic variable. There are seven constants, one for vorticity (h_{ζ}), one for divergence (h_{D}), one for temperature (h_{T}), one for humidity (h_{q}), one for ozone ($h_{\mathcal{O}3}$), one for the extra GFL variables (h_{EXT}), one for the surface hydrostatic pressure (h_{SP}) which is also used for the equivalent height in the 2D model. There are additional constants for non-hydrostatic variables in the non-hydrostatic model. For divergence expression of h_{D} matches:

$$\frac{1}{\Omega h_{\mathrm{D}}} = \frac{2\pi a}{N_{\mathrm{dlon}}} \frac{(1+0.5r_{\mathrm{nlginc}})^{2.5}}{r_{\mathrm{dxtau}}} r_{\mathrm{dampdiv}} \tag{3}$$

r_{nlginc} is the increment from linear to quadratic/cubic grid ($r_{\mathrm{nlginc}} = 0$ for a linear grid, 1 for a quadratic grid, 2 for a cubic grid); r_{dxtau} is a tuning parameter; N_{dlon} is the number of grid-points on a latitude located near the equator of the computational sphere.

In LAM model:

$$\mathcal{K}_{\mathrm{X}} = \exp(-0.5\pi ir) \left[(2\pi)\sqrt{\frac{1}{\frac{L_{\mathrm{x}}^2}{N_{\mathrm{ms}}^2} + \frac{L_{\mathrm{y}}^2}{N_{\mathrm{s}}^2}}}\right]^{-r} \Omega \, h_{\mathrm{X}} \, g \tag{4}$$

(a is the mean Earth radius, L_{x} and L_{y} are respectively the zonal and the meridian lengths of the domain, these lengths being taken on a surface located at a distance a of the Earth centre, N_{ms} is the zonal truncation, N_{s} is the meridian truncation).

$$\frac{1}{\Omega h_{\mathrm{D}}} = [\Delta X]_{\mathrm{gp}} \frac{(1+0.5r_{\mathrm{nlginc}})^{2.5}}{r_{\mathrm{dxtau}}} r_{\mathrm{dampdiv}} \tag{5}$$

where:

$$[\Delta X]_{\mathrm{gp}} = \sqrt{0.5\left([\Delta X]_{\mathrm{gp,zonal}}^2 + [\Delta X]_{\mathrm{gp,merid}}^2\right)}$$

$[\Delta X]_{\mathrm{gp,zonal}}$ and $[\Delta X]_{\mathrm{gp,merid}}$ are respectively the zonal and meridian grid-point mesh-sizes taken at a location where the mapping factor is equal to 1.

In both global and LAM models, for the other 3D upper air fields:

$$\frac{1}{\Omega h_{\mathrm{X}}} = \frac{r_{\mathrm{dampX}}}{r_{\mathrm{dampdiv}}} \frac{1}{\Omega h_{\mathrm{D}}} \tag{6}$$

which can be rewritten:

$$(\Omega h_{\mathrm{X}}) = \frac{r_{\mathrm{dampdiv}}}{r_{\mathrm{dampX}}} (\Omega h_{\mathrm{D}}) \tag{7}$$

∗ **Modifications brought to this scheme if the SLHD (semi-Lagrangian horizontal diffusion) interpolations are done in the semi-Lagrangian scheme:** When the advection scheme is a semi-Lagrangian one, it is possible to activate more diffusive interpolations in the semi-Lagrangian scheme ("SLHD" diffusion), and in this case the horizontal diffusion scheme must be modified:

- Less diffusion in the above formulation (generally by reducing the value of function $g(l)$).

- Adding of a second horizontal diffusion scheme for divergence and vorticity (and also vertical divergence in the NH model): this scheme has the same formulation as the previous one, but the following variables are changed:

  − h_{X} becomes $h_{\mathrm{hds}\,X}$ (currently $\frac{1}{\Omega h_{\mathrm{hds}\,X}} = \frac{1}{r_{\mathrm{damphds}}} \frac{1}{\Omega h_{\mathrm{X}}}$, i.e. $(\Omega h_{\mathrm{hds}\,X}) = r_{\mathrm{damphds}}(\Omega h_{\mathrm{X}})$).

4

- $\mathcal{K}_{\mathrm{X}}$ becomes $\mathcal{K}_{\mathrm{hds}\,X}$.
- the power r becomes s (s is generally above r).
- vertical profile g becomes g_{hds}.

The "SLHD" diffusion can now be activated on a subset of prognostic variables and the above modifications are applied only to this subset of variables.

∗ **Additional remarks:** In the unstretched version of global model, $M = 1$. In LAM model, the horizontal variations of M are generally neglected and M is replaced by a constant hidden in the coefficients $\mathcal{K}_{\mathrm{X}}$ and $\mathcal{K}_{\mathrm{hds}\,X}$. So diffusion looks like the unstretched global model one.

2.2 Sponge.

The sponge is a sort of horizontal diffusion applied near the top of the model; formulation is close to:

$$\left(\frac{\partial X}{\partial t}\right)_{\mathrm{hd}} = -K_{\mathrm{spng}\,\mathrm{X}}(X - X_{\mathrm{moy}}) \tag{8}$$

where $\mathcal{K}_{\mathrm{X}}$ is a vertically dependent and horizontally constant coefficient. X_{moy} is the horizontal average of X on the forecast domain.

2.3 Rayleigh friction.

The following scheme is applied on the U-component of the momentum equation.

$$\left(\frac{\partial U}{\partial t}\right)_{\mathrm{fric}} = -K_{\mathrm{fric}}U \tag{9}$$

$-K_{\mathrm{fric}}U$ is a Rayleigh friction which is activated if **LRFRIC**=.T. in **NAMDYNA**. K_{fric} is a vertical dependent coefficient which is non-zero only for a standard pressure lower than a threshold stored in variable **YRDYN%RRFPLM**.

This Rayleigh friction can be optionally applied on the V-component of the momentum equation.

$$\left(\frac{\partial V}{\partial t}\right)_{\mathrm{fric}} = -K_{\mathrm{fric}}V \tag{10}$$

2.4 Filtering pressure gradient term.

This filtering is activated in spectral space when **LGRADSP**=.T. . It is applied to vorticity component of pressure gradient term. Divergent component is kept unfiltered. In NH LAM models pressure departure is filtered too.

2.5 Nudging.

This scheme called after horizontal diffusion is a linear relaxation of prognostic variables towards pre-defined fields and is not properly saying a diffusion scheme, so it will not be described in detail in this documentation. Pre-defined fields (for example climatological fields) can be read on a file. Nudging coefficients can be different for each variable. Nudging coefficients are not vertically dependent, except for the upper atmosphere where they can be set to zero. Nudging is applied to all variables present in spectral space. There is also nudging for grid-point variables in grid-point space, such surface temperature, surface moisture, deep moisture and snow depth. Nudging is controlled by **LNUDG** in namelist **NAMNUD**.

3 Discretisation of the horizontal diffusion schemes.

3.1 Main horizontal diffusion scheme.

3.1.1 Main horizontal diffusion scheme in unstretched ARPEGE/IFS.

∗ **Discretisation:** Calculations are done in spectral space. The discretised equation which is coded is:

$$X^+_{(m,n)} - X^-_{(m,n)} = -\Omega \ h_X \ g(l) f(n, N, n_0(X), x_0, r) Dt X^+_{(m,n)} \tag{11}$$

where the superscripts + and − indicate respectively variables after horizontal diffusion and variables before horizontal diffusion, Dt is the time step in the first integration step and twice the time step later (leap frog scheme), n is the total wave number (between 0 and the truncation N_s), m is the zonal wave number (between $-n$ and n in a triangular truncation), r is the order of the horizontal diffusion operator, N is a reference wave number, $n_0(X)$ is a threshold depending on variable X (generally zero except for vorticity where it is 2 at METEO-FRANCE and 0 at ECMWF), x_0 is a threshold between 0 and 1. ∇' is the first order horizontal reduced derivative operator (i.e in the grid point space, $\nabla = M\nabla'$, where M is the mapping factor). This horizontal diffusion scheme is an implicit horizontal diffusion scheme (in the right-hand side member of the equation there is $X^+_{(m,n)}$ and not $X^-_{(m,n)}$). Equation (11) becomes (12):

$$X^+_{(m,n)} = \frac{1}{1 + K_X(n)Dt} X^-_{(m,n)} \tag{12}$$

where:

$$K_X(n) = \Omega \ h_X \ g(l) f(n, N, n_0(X), x_0, r) \tag{13}$$

In 2D models (shallow-water model for example), $g(l)$ is replaced by 1 and N is always equal to N_s.

∗ **Horizontal dependency for applications at METEO-FRANCE:** Expression of $f(n, N, n_0(X), x_0, r)$ is currently:

$$f(n, N, n_0(X), x_0, r) = \max\left(0., \left(\frac{\left(\frac{\max(0, n(n+1)-n_0)}{\max(0, N(N+1)-n_0)}\right)^{\frac{1}{2}} - x_0}{1 - x_0}\right)^r\right) \tag{14}$$

An exact discretisation of equation (1) would give:

$$f(n, N, n_0(X), x_0, r) = f(n, N_s, 0, 0, r) \tag{15}$$

One can notice that this discretisation is equivalent to apply a purely diagonal operator in spectral space.

N can be vertically constant (taken equal to the truncation N_s) or vertically variable (varies between N_s near the ground and N_2 near the top, with $N_2 < N_s$).

∗ **Horizontal dependency for applications at ECMWF:** Function f has a more tricky shape which is not detailed there.

∗ **Vertical dependency of g:** $g(l)$ only depends on the altitude, and several profiles have been coded, according to the value of namelist variable **YRDYN%NPROFILEHD**.

The operational version of ARPEGE (case **YRDYN%NPROFILEHD**=3) uses the following profile: for the layer number l, expression of $g(l)$ is:

$$g(l) = \min\left(y_0 \frac{\Pi_{ref}}{\Pi_{st}(l)}, \frac{1}{y_3}\right) \tag{16}$$

where $\Pi_{st}(l)$ is the standard atmosphere pressure for the layer number l, Π_{ref} is a reference pressure (sea level pressure for the standard atmosphere: 1013.25 hPa). y_0 is between 0 and 1: $g(l) = 1$ if $\Pi_{st}(l)/\Pi_{ref}$ is above y_0 and $g(l)$ is above 1 if $\Pi_{st}(l)/\Pi_{ref}$ is below y_0. y_3 is between 0 and y_0, in practical it is significantly lower than y_0 and it is used to avoid too much diffusion in the high stratosphere and in the mesosphere.

A more flexible profile has been implemented (**YRDYN%NPROFILEHD**=2) in order to increase g in the stratosphere. Two additional reference pressures Π_{tp} and Π_{bt} are defined ($\Pi_{tp} < \Pi_{bt}$).

- For $\Pi_{st}(l) > \Pi_{bt}$, $g(l)$ is defined by formula (16).
- For $\Pi_{st}(l) < \Pi_{tp}$, $g(l)$ is set to a maximum value g_{mx} provided by namelist.

- For $\Pi_{\mathrm{st}}(l)$ between Π_{bt} and Π_{tp}, $g(l)$ writes:

$$g(l) = \frac{g_a(\Pi_{\mathrm{st}}(l)/\Pi_{\mathrm{bt}})^q + g_b}{1 + (\Pi_{\mathrm{st}}(l)/\Pi_{\mathrm{bt}})^q}$$

g_a and g_b are two constants computed in order that function $g(l)$ is continue for $\Pi_{\mathrm{st}} = \Pi_{\mathrm{bt}}$ and $\Pi_{\mathrm{st}} = \Pi_{\mathrm{tp}}$. q is an exponent ($q > 0$) allowing an additional freedom degree for vertical variations of g.

An alternate (ECMWF-type profile) is also coded (**YRDYN%NPROFILEHD**=1). Several formulae are coded according to the top of the model, leading to $g(l)$ functions varying between 1 at the surface and 16 at the top.

∗ **Vertical dependency of** N**:** The ARPEGE operational suite generally uses the constant profile $N = N_{\mathrm{s}}$. Climatic applications may require a stronger diffusion in the stratosphere; this is possible via a variable profile $N(l)$ depending on the standard pressure. A reference truncation N_2 is defined for the high stratosphere, and two pressures thresholds Π_1 and Π_2 are defined via variables y_1 and y_2 ($y_1 > y_2$ and are both between 0 and 1).

$$\Pi_1 = \Pi_{\mathrm{ref}} * y_1$$
$$\Pi_2 = \Pi_{\mathrm{ref}} * y_2$$

Definition of $N(l)$ is:
- For $\Pi_{\mathrm{st}} > \Pi_1$, $N(l)$ is equal to N_{s}.
- For $\Pi_{\mathrm{st}} < \Pi_2$, $N(l)$ is equal to N_2.
- For Π_{st} between Π_1 and Π_2, $N(l)$ writes:

$$N(l) = \mathrm{nint}\left[\left(\frac{\Pi_1}{\Pi_{\mathrm{st}}(l)}\frac{\Pi_{\mathrm{st}}(l) - \Pi_2}{\Pi_1 - \Pi_2}\right)N_{\mathrm{s}}{}^r + \left(\frac{\Pi_2}{\Pi_{\mathrm{st}}(l)}\frac{\Pi_{\mathrm{st}}(l) - \Pi_1}{\Pi_2 - \Pi_1}\right)N_2{}^r\right]^{1/r}$$

This vertical profile is available for all values of **YRDYN%NPROFILEHD** = 1 to 3. An old code (put under **YRDYN%NPROFILEHD**=4) is still present to take account of Π_2 and N_2 but this piece of code difficult to read and maintain is expected to disappear in the future. Users must switch on **YRDYN%NPROFILEHD**=3.

3.1.2 Main horizontal diffusion scheme in LAM model.

∗ **Discretisation:** Compared to global model, the main differences are:
- K_{X} also depends on the zonal wavenumber m.
- Function f also depends on m and on the zonal truncation N_{ms}, and has a different expression.
- n_0 is replaced by zero in the code (for convenience one will let n_0 in the following formulae).
- Equation (11) becomes:

$$X^+_{(m,n)} - X^-_{(m,n)} = -\Omega\ h_{\mathrm{X}}\ g(l)f(m,n,N_{\mathrm{m}},N,n_0(X),x_0,r)DtX^+_{(m,n)} \tag{17}$$

where N_{m} is a reference zonal wavenumber.
- Equation (12) becomes:

$$X^+_{(m,n)} = \frac{1}{1 + K_{\mathrm{X}}(m,n)Dt}X^-_{(m,n)} \tag{18}$$

- Equation (13) becomes:

$$K_{\mathrm{X}}(m,n) = \Omega\ h_{\mathrm{X}}\ g(l)f(m,n,N_{\mathrm{m}},N,n_0(X),x_0,r) \tag{19}$$

∗ **Horizontal dependency for applications at METEO-FRANCE:** Expression of $f(m,n,N_{\mathrm{m}},N,n_0(X),x_0,r)$ is currently:

$$f(m,n,N_{\mathrm{m}},N,n_0(X),x_0,r) = \max\left(0., \left(\frac{\left(\frac{n^2}{N^2} + \frac{m^2}{N_{\mathrm{m}}^2}\right)^{\frac{1}{2}} - x_0}{1 - x_0}\right)^r\right) \tag{20}$$

An exact discretisation of equation (1) would give:

$$f(m,n,N_{\mathrm{m}},N,n_0(X),x_0,r) = f(m,n,N_{\mathrm{ms}},N_{\mathrm{s}},0,0,r) \tag{21}$$

∗ **Vertical dependency of** g**:** Identical to the global model one.

∗ **Vertical dependency of** N**:** The algorithm available in the global model has been extended in the LAM model, but additional issues appear due to the fact that there are two truncations N_s and N_{ms}. We still define one reference truncation N_2 for the high stratosphere. We define two vertically varying truncations N and N_m and we consider the following cases:

- 1D model ($N_s = 1$, $N_{ms} = 0$):
  - $N(l)$ is equal to 1 at all levels.
  - $N_m(l)$ is equal to 0 at all levels.

- Meridian 2D model ($N_s \geq 1$, $N_{ms} = 0$):
  - $N(l)$ is computed as in global models from N_s to N_2.
  - $N_m(l)$ is equal to 0 at all levels.

- Zonal 2D model ($N_s = 0$, $N_{ms} \geq 1$):
  - $N(l)$ is equal to 0 at all levels.
  - $N_m(l)$ is computed as in global models from N_{ms} to N_2.

- Rectangular 3D model with $N_s > N_{ms}$:
  - $N(l)$ is computed as in global models from N_s to N_2.
  - $N_m(l)$ is equal to $\text{nint}((N_{ms}/N_s) * N(l))$.

- Rectangular 3D model with $N_s < N_{ms}$:
  - $N_m(l)$ is computed as in global models from N_{ms} to N_2.
  - $N(l)$ is equal to $\text{nint}((N_s/N_{ms}) * N_m(l))$.

- Square 3D model with $N_s = N_{ms}$:
  - $N(l)$ is computed as in global models from N_s to N_2.
  - $N_m(l)$ is equal to $N(l)$.

3.1.3 Main horizontal diffusion scheme in stretched ARPEGE/IFS.

∗ **Expression of mapping factor in spectral space.** Let us denote by:

- $a_c = 0.5 \left(c + \frac{1}{c} \right)$
- $b_c = 0.5 \left(c - \frac{1}{c} \right)$
- $e_{(0,0)} = 0$
- $e_{(m,n)} = \sqrt{\frac{n^2 - m^2}{4n^2 - 1}}$

Expression of M is:

$$M = a_c + b_c \mu \tag{22}$$

where μ is the sinus of computational sphere latitude. Expression of $[MX]_{(m,n)}$ is:

$$[MX]_{(m,n)} = b_c e_{(m,n)} X_{(m,n-1)} + a_c X_{(m,n)} + b_c e_{(m,n+1)} X_{(m,n+1)} \tag{23}$$

A multiplication by M in spectral space is equivalent to a multiplication by a symmetric tridiagonal matrix.

∗ **Discretisation:** Calculations are done in spectral space. The discretised equation which is coded is:

$$X_{(m,n)}^- =$$
$$+ b_c e_{(m,n)} \Omega \ h_X \ g(l) Dt f(n-1, N, n_0(X), x_0, r) X_{(m,n-1)}^+$$
$$+ (1 + a_c \Omega \ h_X \ g(l) Dt f(n, N, n_0(X), x_0, r)) X_{(m,n)}^+$$
$$+ b_c e_{(m,n+1)} \Omega \ h_X \ g(l) Dt f(n+1, N, n_0(X), x_0, r) X_{(m,n+1)}^+ \tag{24}$$

Equation (24) is equivalent to invert a tridiagonal matrix for each zonal wave number m. The computation of this matrix and a decomposition into a product of two triangular bidiagonal matrices are performed in the set-up routine **SUHDU**. At each time step these two triangular bidiagonal matrices are inverted in order to compute the $X_{(m,n)}^+$.

3.2 Modification of the horizontal diffusion scheme if "SLHD" interpolations are done in the semi-Lagrangian scheme.

∗ **Modified expression of** $g(l)$**:** We now need two vertical profiles $g(l)$ and $g_{\mathrm{hds}}(l)$.

- Expression of $g_{\mathrm{hds}}(l)$ has the same shape as $g(l)$ according to **YRDYN%NPROFILEHD**, but y_0 is replaced by $y_{\mathrm{hds}0}$ (we have the possibility to take $y_{\mathrm{hds}0}$ different from y_0).
- Constant s_{dred} is subtracted from $g(l)$: that provides a modified $g(l)$ (always ≥ 0).

∗ **Vertical profiles for** N**:** We now need two vertical profiles $N(l)$ and $N_{\mathrm{hds}}(l)$. Both refer to the same reference truncation N_2. In LAM models we also introduce the additional quantity $N_{\mathrm{mhds}}(l)$.

∗ **Unstretched global model:** A double diffusion is applied on vorticity and divergence (and also the vertical divergence variable in the NH model). It writes:

$$X^+_{(m,n)} = \left[\frac{1}{1 + K_{\mathrm{hds}X}(n)Dt}\right]\left[\frac{1}{1 + K_X(n)Dt}\right]X^-_{(m,n)} \tag{25}$$

where:

$$K_{\mathrm{hds}X}(n) = \Omega\ h_{\mathrm{hds}X}\ g_{\mathrm{hds}}(l)f(n, N_{\mathrm{hds}}, n_0(X), x_0, s) \tag{26}$$

∗ **LAM model:** A double diffusion is applied on vorticity and divergence (and also the vertical divergence variable in the NH model). It writes:

$$X^+_{(m,n)} = \left[\frac{1}{1 + K_{\mathrm{hds}X}(m,n)Dt}\right]\left[\frac{1}{1 + K_X(m,n)Dt}\right]X^-_{(m,n)} \tag{27}$$

where:

$$K_{\mathrm{hds}X}(m,n) = \Omega\ h_{\mathrm{hds}X}\ g_{\mathrm{hds}}(l)f(m, n, N_{\mathrm{mhds}}, N_{\mathrm{hds}}, n_0(X), x_0, s) \tag{28}$$

∗ **Stretched global model:** A double diffusion is applied on vorticity and divergence. It writes:

$$X^-_{(m,n)} =$$
$$+b_{\mathrm{c}}e_{(m,n)}\Omega\ h_X\ g(l)Dtf(n-1, N, n_0(X), x_0, r)X^{++}_{(m,n-1)}$$
$$+(1 + a_{\mathrm{c}}\Omega\ h_X\ g(l)Dtf(n, N, n_0(X), x_0, r))X^{++}_{(m,n)}$$
$$+b_{\mathrm{c}}e_{(m,n+1)}\Omega\ h_X\ g(l)Dtf(n+1, N, n_0(X), x_0, r)X^{++}_{(m,n+1)} \tag{29}$$

then:

$$X^{++}_{(m,n)} =$$
$$+b_{\mathrm{c}}e_{(m,n)}\Omega\ h_{\mathrm{hds}X}\ g_{\mathrm{hds}}(l)Dtf(n-1, N_{\mathrm{hds}}, n_0(X), x_0, s)X^+_{(m,n-1)}$$
$$+(1 + a_{\mathrm{c}}\Omega\ h_{\mathrm{hds}X}\ g_{\mathrm{hds}}(l)Dtf(n, N_{\mathrm{hds}}, n_0(X), x_0, s))X^+_{(m,n)}$$
$$+b_{\mathrm{c}}e_{(m,n+1)}\Omega\ h_{\mathrm{hds}X}\ g_{\mathrm{hds}}(l)Dtf(n+1, N_{\mathrm{hds}}, n_0(X), x_0, s)X^+_{(m,n+1)} \tag{30}$$

This discretisation requires to invert two tridiagonal symmetric matrices in spectral space.

3.3 Application of diffusion to each variable.

3.3.1 Application to GMV variables and to GFL variables in 3D models.

For GMV variables, horizontal diffusion is applied to:

- "reduced" vorticity ζ' with $n_0(\zeta) = 2$ at METEO-FRANCE, 0 at ECMWF.
- "reduced" divergence D' with $n_0(D) = 0$.
- for temperature, a variable which looks like $T - \alpha\log(\Pi_{\mathrm{s}})$ (see part 3.3.2), with $n_0(T) = 0$.
- in the NH model, the two additional NH variables, with $n_0 = 0$ (and also the auxiliary variable NHX if **NVDVAR**=4).
  About NH variables, some recommendations must be taken into account.
    - Vertical divergence and horizontal divergence (and NHX if **NVDVAR**=4) must be diffused with the same intensity .

– It is recommended to diffuse pressure departure too.

- $\log(\Pi_s)$ should not be diffused (even if some diffusion code is implemented for this variable).

For GFL variables, horizontal diffusion is applied to:

- specific humidity q if spectral one, with $n_0(q) = 0$.
- ozone $\mathcal{O}3$ if spectral one, with $n_0(\mathcal{O}3) = 0$.
- extra GFL variables if spectral ones, with $n_0 = 0$.
- For the other advectable GFL variables and for the non-advectable GFL variables, no diffusion is currently coded.

3.3.2 Application to temperature.

Diffusion has to be done on a variable less sensitive to the orography than T, so one chooses a variable looking like $T - \alpha \log(\Pi_s)$ with a "good" formulation of α. $T - \alpha \log(\Pi_s)$ is diffused instead of T, where α is a coefficient depending on the altitude and using parameters related to standard atmosphere. Let us denote by $T_{s\ st}$ the surface temperature in a standard atmosphere (288.15 K), R the air constant, G the acceleration due to gravity, $\left[\frac{dT}{dz}\right]_{st}$ the tropospheric temperature vertical gradient in a standard atmosphere (-0.0065 K/m), $T_{t\ st}$ the tropopause temperature in a standard atmosphere (217.15 K), and Π_{ref} a reference temperature equal to 1013.25 hPa. If:

$$(T_{s\ st})(\Pi/\Pi_{ref})^{(R/G)\left[\frac{dT}{dz}\right]_{st}} > (T_{t\ st})$$

the expression of α_l for the layer number l is:

$$\alpha_l = -B_l(R/G)\left[\frac{dT}{dz}\right]_{st}(T_{s\ st})(\Pi/\Pi_{ref})^{-((R/G)\left[\frac{dT}{dz}\right]_{st}+1)} \tag{31}$$

where B_l is used in the definition of pressure on layers and interlayers (hybrid vertical coordinate). In the other cases:

$$\alpha_l = 0 \tag{32}$$

Temperature horizontal diffusion writes as:

$$(T^+_{(m,n)} - \alpha \log(\Pi_s)^-_{(m,n)}) = \frac{1}{1 + K_T(n)Dt}(T^-_{(m,n)} - \alpha \log(\Pi_s)^-_{(m,n)}) \tag{33}$$

Equation (33) yields (34):

$$T^+_{(m,n)} = (T^-_{(m,n)} + K_T(n)Dt\alpha \log(\Pi_s)^-_{(m,n)})/(1 + K_T(n)Dt) \tag{34}$$

For temperature $n_0(T) = 0$.

3.3.3 Application to GMV variables in 2D models.

Horizontal diffusion is performed for vorticity, divergence and equivalent height.

3.4 Other diffusive processes schemes.

3.4.1 Sponge.

We focus on the new sponge done in spectral space for GMV and spectral GFL, and in grid-point space for grid-point GFL variables.

Discretisation of equation (8) is equivalent to write:

$$X^+ - X_{moy} = (1 - K_{spng_X}Dt)(X^- - X_{moy})$$

Coefficient K_{spng_X} is only vertically-dependent.

In spectral space this is equivalent to write

$$X^+_{(m=0,n=0)} = X^-_{(m=0,n=0)}$$

$$X^+_{(m,n)} = (1 - K_{spng_X}Dt)X^-_{(m,n)}$$

We denote by R_{spng_X} the quantity $(1 - K_{spng_X}Dt)$. Coefficient R_{spng_X} depends on variable X, but with the following limitations: there is one coefficient for GMV, one coefficient for spectral GFL and one coefficient for grid-point GFL.

Two heights R_{sponbt_X} and R_{spontp_X} are defined ($R_{sponbt_X} < R_{spontp_X}$). The minimum value of R_{spng_X} at the top is R_{spngn_X}. There is an additional quantity $R_{spontau_X}$ introduced to control the shape of the function defining R_{spng_X}.

10

- For a standard atmosphere height z lower than $R_{\mathrm{sponbt\,X}}$, $R_{\mathrm{spng\,X}} = 1$.
- For a standard atmosphere height z bigger than $R_{\mathrm{spontp\,X}}$, $R_{\mathrm{spng\,X}} = R_{\mathrm{spngn\,X}}$.
- For a standard atmosphere height z between $R_{\mathrm{sponbt\,X}}$ and $R_{\mathrm{spontp\,X}}$:

$$R_{\mathrm{spng\,X}} = R_{\mathrm{spngn\,X}} + (1 - R_{\mathrm{spngn\,X}})R_{\mathrm{spontau\,X}}\frac{T}{Dt + R_{\mathrm{spontau\,X}}T}$$

where $T = \tan(0.5\pi(R_{\mathrm{spontp\,X}} - z)/(R_{\mathrm{spontp\,X}} - R_{\mathrm{sponbt\,X}}))$

For grid-point GFL variables, X_{moy} is always set to 0.

3.4.2 Rayleigh friction.

The following scheme is discretised in grid-point space for the U-component (and optionally for the V-component) of the momentum equation. **LRFRIC**=.T. (**NAMDYNA**) activates Rayleigh friction on U; **LRFRICISOTR**=.T. (**NAMDYNA**) activates additional Rayleigh friction on V. Rayleigh friction can be done in Fourier space instead of grid-point space if **LEGWWMS**=T (option used at ECMWF, available in unstretched untilted global model, and out of the scope of this documentation).

We now focus on calculation of coefficient K_{fric} which appears in equation (9). K_{fric} is a vertically dependent coefficient which is non-zero only for a standard pressure lower than Π_{rfric} (Π_{rfric} is stored in variable **YRDYN%RRFPLM** in **YOMDYN**). K_{fric} is computed in **SURAYFRIC** according the following formula:

- If $\Pi_{\mathrm{st}}(l) \leq \Pi_{\mathrm{rfric}}$:

$$K_{\mathrm{fric}}(l) = \frac{1 - \frac{1}{7.7}\tanh\left(R_{\mathrm{fz1}} - 7\log(100000/\Pi_{\mathrm{st}}(l))\right)}{R_{\mathrm{ftau}}} \tag{35}$$

- If $\Pi_{\mathrm{st}}(l) > \Pi_{\mathrm{rfric}}$: $K_{\mathrm{fric}}(l) = 0$

Formula (35) involves the following quantities:

- R_{fz1} is stored in variable **YRDYN%RRFZ1** in **YOMDYN** (it controls the inflexion point of the vertical profile).
- R_{ftau} is stored in variable **YRDYN%RRFTAU** in **YOMDYN** (it controls the intensity of diffusion).
- K_{fric} is stored in variable **YRDYN%RKRF** in **YOMDYN**.

3.4.3 Filtering pressure gradient term.

This filtering is activated when **LGRADSP**=.T. . Spectral filtering is done in routine **SPFILT** (global models) and **ESPFILT** (LAM models). It is applied to vorticity component of pressure gradient term. Divergent component is kept unfiltered. In NH LAM models pressure departure is filtered too.

In global models, filter only depends on total wavenumber n; no filter is applied for $n + i \leq [(2/3)N_{\mathrm{s}} - 1]$, where $i = 0.83N_{\mathrm{s}}$; for $n + i > [(2/3)N_{\mathrm{s}} - 1]$ vorticity component is multiplied by a factor $1/(1 + fac^2)$ where $fac = 32\exp[-0.5 * (n + i - N_{\mathrm{s}})^2/(n + i - [(2/3)N_{\mathrm{s}} - 1])^2]$.

A specific formulation is used in LAM models.

4 Organigramme of the horizontal diffusion computations.

4.1 Order of spectral computations.

∗ **Global model:** Spectral computations are done in the 3D model in the following order:
- Mass corrector computations (**SPCMASCOR**).
- Semi-implicit computations in the hydrostatic model (**SPCSI**).
- Semi-implicit computations in the NH model (**SPNHSI**).
- Filtering of pressure gradient term (**SPFILT**).
- Main horizontal diffusion scheme and sponge (**SPCHOR**).
- Nudging (i.e. linear relaxation towards a pre-defined state) (**SPCHOR**).

In the 2D model the semi-implicit computations and the main horizontal diffusion scheme are done in **SPC2**.

∗ **LAM model:** Spectral computations are done in the 3D model in the following order:
- Semi-implicit computations in the hydrostatic model (**ESPCSI**).
- Semi-implicit computations in the NH model (**ESPNHSI**).
- Spectral nudging (coupling) from a coarser model (**ESPSC2R** for temporal interpolation providing coupler at the current timestep, **ESPCPL** for spectral relaxation).
- Filtering of pressure gradient term (**ESPFILT**).
- Main horizontal diffusion scheme and sponge (**ESPCHOR**).

∗ **Case of iterative centred-implicit schemes:** For **LPC_FULL**, horizontal diffusion can be called at the last iteration or at all iterations of the iterative algorithm according to the value of variable **YRDYN%LRHDI_LASTITERPC**.

4.2 Set-up and control routines until STEPO.

```
CNT0 ->
 * SU0YOMA ->
   - SUCT0
   - SUDYNA
   - SUGEOMETRY
 * SU0YOMB ->
   - SUDYN ->
     * SUALDYN
     * SUALDYNB or SUELDYNB
     * SUHDIR
     * SUHDF or SUHDF_EC (global model),
       SUHDF2 (2D global model) or SUEHDF (LAM model) [Main diffusion scheme only]
     * SUHDU -> SUHER [Main diffusion scheme in stretched global model only]
     * SURCORDI or SURCORDI_TH
     * SURAYFRIC
   - SUSPNG
 * CNT1 -> CNT2 -> CNT3 -> CNT4 ->
   - SUHDU -> SUHER [Main diffusion scheme in stretched global model only]
   - STEPO -> (organigramme not detailed)
```

∗ **Adjoint code:** For adjoint code **STEPO** is replaced by **STEPOAD**, organigramme is slightly different between **CNT0** and **STEPOAD**. For example **CNT3** is replaced by **CNT3AD**, **CNT4** is replaced by **CNT4AD**.

∗ **Tangent linear code:** For tangent linear code **STEPO** is replaced by **STEPOTL**, organigramme is slightly different between **CNT0** and **STEPOTL**. For example **CNT3** is replaced by **CNT3TL**, **CNT4** is replaced by **CNT4TL**.

∗ **DFI initialisation:** The sequence **SUHDU** → **SUHER** is also called by **DFI3** (digital filtering initialisation).

4.3 Direct code under STEPO or tangent linear code under STEPOTL for spectral calculations.

STEPO or **STEPOTL** →
- **SPCM** (global model only) →
    - **TRMTOS** (transposition routine for distributed memory).
    - **SPCMASCOR** (mass corrector).
    - **SPCSI** (semi-implicit scheme in the hydrostatic model).
    - **SPNHSI** (semi-implicit scheme in the NH model).
    - **TRSTOM** (transposition routine for distributed memory).
    - **SPFILT** (pressure gradient term filtering).
    - **SPCHOR** → **MXTURHD** (horizontal diffusion, sponge and nudging).
- **ESPCM** (LAM model only) →
    - **TRMTOS** (transposition routine for distributed memory).
    - **ESPCSI** (semi-implicit scheme in the hydrostatic model).
    - **ESPNHSI** (semi-implicit scheme in the NH model).
    - **ESPSC2R** and **ESPCPL** (spectral nudging (coupling) from a coarser model).
    - **ESPFILT** (pressure gradient term filtering).
    - **ESPCHOR** → **MXTURHD** (horizontal diffusion, sponge).
    - **TRSTOM** (transposition routine for distributed memory).
- **SPC2M** (global model only) → **SPC2** →
    - **MXTURS**
    - **MXTURE**
    - **MXPTMA**
    - **BALADSM**
- **GPNSPNG** (sponge for grid-point GFL variables).

4.4 Adjoint code under STEPOAD for spectral calculations.

STEPOAD →
- **SPCMAD** (global model only) →
    - **BRPTOB** (communication routine for distributed memory) → **PE2SET**
    - **SPCHORAD** → **MXTURHD** and **GATHERT** (horizontal diffusion).
    - **TRMTOS** (transposition routine for distributed memory).
    - **SPCSIAD** (semi-implicit scheme for the hydrostatic model).
    - **SPNHSIAD** (semi-implicit scheme for the NH model).
    - **TRSTOM** (transposition routine for distributed memory)
- **ESPCMAD** (LAM model only) →
    - **TRMTOS** (transposition routine for distributed memory).
    - **ESPCHORAD** → **MXTURHD** and **GATHERT** (horizontal diffusion).
    - **ESPCSIAD** (semi-implicit scheme for the hydrostatic model).
    - **TRSTOM** (transposition routine for distributed memory).
- **SPC2MAD** (ARPEGE/IFS only) → **SPC2AD** →
    - **MXTURS**
    - **MXTURE**
    - **MXPTMA**

4.5 Rayleigh friction done in grid-point space.

Rayleigh friction is done in the grid-point calculations under **CPG_GP** − > **GP_TNDLAGADIAB_UV** if **LEGWWMS**=F. There is some Rayleigh diffusion code (done in Fourier space) under the sequence

```
LTINV -> ASRE1 -> ASRE1B -> FSPGL_INT -> adiab/FSPGLH
```

activated only if **LEGWWMS**=T.

4.6 Action done by set-up routines.

- **SUCT0**: computes 0-level control variables.
- **SUGEOMETRY**: computes geometry object variables.
- **SUALDYN**, **SUALDYNB**, **SUELDYNB**: allocates some arrays used in the dynamics, in particuliar those used in the horizontal diffusion.
- **SUDYNA**: computes some variables used in the dynamics; for example **LGRADSP**.
- **SUDYN**: caller for the following ones:

    - **SUHDIR**: computes the coefficients **HDIR...**, **HRDIR...**, **RDAMP...**.
    - **SUHDF**: computes the arrays **RDI...**, **RDS...**.
    - **SUHDF_EC**: version of **SUHDF** used at ECMWF.
    - **SUHDF2**: cf. **SUHDF** but for 2D models.
    - **SUEHDF**: for LAM model; computes the arrays **RDI...E**, **RDS...E**.
    - **SURCORDI**: computes the array **RCORDIT**.
    - **SURAYFRIC**: computes the setup of the Rayleigh friction.
    - **SUHDU**: fills **RDHI** of **YOMDYN**. Array **RDHI** contains the non-zero diagonals of the two triangular bidiagonal matrices, the product of which defines the inverse of horizontal diffusion operator (right-hand side member of equation (24)).

- **SUSPNG** (encapsulated in module **SPNG_MOD**): computes the setup of the sponge.

4.7 Control routines.

CNT0, **CNT1**, **CNT2**, **CNT3**, **CNT4**, **CNT3AD**, **CNT4AD**, **CNT3TL**, **CNT4TL**, **STEPO**, **STEPOAD**, **STEPOTL** are control routines. **STEPO** (resp **STEPOAD**, **STEPOTL**) manages one direct (resp. adjoint, tangent linear) integration timestep.

4.8 Action done by routines under STEPO, STEPOAD or STEPOTL.

∗ **Algorithmic routines:**

- **SPCHOR**: spectral horizontal diffusion schemes (except Rayleigh friction) and nudging in the 3D model.
- **SPFILT**: filtering of pressure gradient term in spectral space.
- **SPC2**: spectral space computations, including the semi-implicit scheme and all horizontal diffusion schemes in the 2D model.
- **SPCM**: interface routine for **SPCHOR**.
- **SPC2M**: interface routine for **SPC2**.
- **ESPCM**, **ESPC2M**, **ESPCHOR**, **ESPC2** are called instead of **SPCM**, **SPC2M**, **SPCHOR**, **SPC2** in LAM model.
- These routines have adjoints (same names + "AD"). Horizontal diffusion on post-processed fields and nudging are not yet coded in **SPCHORAD**.
- **BALADSM**: solve linear balance equation in spectral space to convert vorticity into geopotential (used for vorticity 2D equation only).

∗ **Linear algebra routines (project "algor"):**

- **MXTURE**: called by **SPC2** and inverts bidiagonal or tridiagonal triangular linear systems.
- **MXTURHD** is a version of **MXTURE** well suited for horizontal diffusion.
- **MXTURS**: combines two calls to **MXTURE** to invert a set of symmetric tridiagonal or pentadiagonal matrices, the decomposition LU of which is known. Called by **SPC2**.
- **MXPTMA**: products of a set of tridiagonal or pentadiagonal matrices by a set of matrices or vectors. Called by **SPC2**.
- **SUHER**: performs the LU factorisations of a set of non symmetric tridiagonal matrices.

∗ **Distributed memory routines (for ex. transposition routines) doing communications between processors:** the list of transposition and communication routines used in horizontal diffusion scheme is the following: **TRSTOM**, **TRMTOS**, **BRPTOB**, **MPL_ALLREDUCE**, **GATHERT**, **PE2SET** and **SET2PE**; see documentation (IDDM) about distributed memory features for the action of these routines.

5 Some distributed memory features in spectral space.

- The total number of processors involved in the A-level parallelisation is **NPRTRW**.

- The total number of processors involved in the B-level parallelisation is **NPRTRV**.

- The total number of processors **NPROC=NPRTRW∗NPRTRV**.

- One processor treats only a subset of zonal wave numbers.

- A call to **SPCHOR** can treat several wave numbers (currently all the zonal wave numbers treated by the current processor). For computations needing matricial operations (geographical component of unified diffusion scheme for example) calculations are split zonal wave number by zonal wave number inside **SPCHOR**.

- For 2D model a call to **SPC2** treats only one zonal wave number.

- For the 3D model there is a splitting into **NFLEVL** layers packets when **NPRTRV**>1. One B-set processor treats **NFLEVL** layers; an exception is done in the adjoint code **SPCHORAD** when diffusing variables using the temperature array, some calculations on all the **NFLEVG** layers are necessary to compute a vertical integral which does not exist in the direct code. That means that additional transpositions (**TRSTOM** in the direct code) are necessary between the semi-implicit calculations of **SPCSI** and the horizontal diffusion calculations of **SPCHOR** to convert the fields from the **NFLEVG** structure required in the semi-implicit calculations to the **NFLEVL** structure required in the horizontal diffusion calculations.

- LAM model does not completely follow global model for the second level of distribution: the **NFLEVL** structure is not yet implemented in **ESPCHOR**, and **ESPCHOR** works like **ESPCSI**. That explains why the order of calls is **SPCSI**, **TRSTOM**, **SPCHOR** in global model, but **ESPCSI**, **ESPCHOR**, **TRSTOM** in LAM model.

6 Module and namelist variables.

These modules are auto-documented so description of each variable is provided in the code source. We can recall here the most important variables to know for each module:

- Modules for geometry, in particuliar:

  - **SPGEOM_MOD** (spectral geometry).
  - **YEMGEO** (LAM model geometry): in particular ELX, ELY, EDELX, EDELY. Some of these variables are in namelist **NEMGEO**.
  - **YOMGEM** (computational space grid-point horizontal geometry). Some of these variables are in namelist **NAMGEM**.
  - **YOMLAP** (Laplace space).
  - **YOMVERT** (vertical geometry).

- **SPNG_MOD** (module for new sponge):

  - LNSPONGE: to switch on new sponge.
  - RSPONBT (R_{sponbt_X}), RSPONTP ($R_{\mathrm{spontp}X}$), RSPONTAU (R_{spontau_X}), RSPONGN (R_{spngn_X}), RSPONGF (R_{spng_X}).

  Some of these variables are in namelist **NAMSPNG**.

- **YOMARG** (0-level control, former command line) and **YOMCT0** (0-level control):

  - NCONF, LECMWF, LELAM (in **NAMARG**).
  - LR3D, LR2D, LRSHW, LRVEQ.
  - LRPLANE (in **NAMCT0**).

- **YOMDIM**, **YOMDIMV** and **YOMDIMH** (dimensioning): most of variables. Some of these variables are in namelist **NAMDIM**.

- **YOMDYNA** (adiabatic dynamics: first part):

  - LSLHD (internal model key for SLHD = semi-Lagrangian diffusion computation).
  - LSLHD_W, LSLHD_T, LSLHD_SPD, LSLHD_SVD, LSLHD_OLD, LSLHD_STATIC, LSLHDQUAD (keys for SLHD).
  - SLHDKMIN, SLHDKMAX, SLHDEPSH, SLHDEPSV (values used in SLHD).
  - LRFRIC and LRFRICISOTR (Rayleigh friction).

  Some of these variables are in namelist **NAMDYNA**.

- **YOMDYN** (adiabatic dynamics: second part). All these variables are attributes of YRDYN.

  - Main horizontal diffusion scheme:
    * HDIR[X] ($1/(\Omega h_X)$).
    * HRDIR[X] ((Ωh_X), i.e. 1/HDIR[X]).
    * FRANDH (x_0 in equation (14)).
    * REXPDH (r = order of the horizontal diffusion operator).
    * SLEVDH and SLEVDH3 (thresholds y_0 and y_3 in formula defining $g(l)$).
    * SLEVDH1 and SLEVDH2 (thresholds y_1 and y_2 in formula defining $N(l)$).
    * NSREFDH (N_2).
    * NPROFILEHD (type of vertical profile for function $g(l)$).
    * RPROFHDBT, RPROFHDTP, RPROFHDMX, RPROFHDEX (used if NPROFILEHD=2).
    * LRDISPE_EC (controls horizontal function used in the horizontal diffusion).
    * RDI[X] (contains K_X, see equation (13)).
    * RCORDIT (contains α, used in equations (31) and (33)).
    * RRDXTAU (r_{dxtau}), RDAMP[X] ($r_{\mathrm{damp}X}$).
    * LSTRHD.
    * RDHI.

  - Additional quantities used if SLHD scheme is activated:
    * HDSR[X] ($1/(\Omega h_{\mathrm{hds}X})$).
    * HRDSR[X] ($\Omega h_{\mathrm{hds}X}$).
    * REXPDHS (s = order of the additional horizontal diffusion operator).

16

* SLEVDHS (thresholds y_{hds_0} in formula defining $g_{\mathrm{hds}}(l)$).
* SLEVDHS1 and SLEVDHS2 (thresholds y_{hds_1} and y_{hds_2} in formula defining $N_{\mathrm{hds}}(l)$).
* RDS[X] (contains $K_{\mathrm{hds}\,X}$, see equation (26)).
* RDAMPHDS (coefficient r_{damphds}).
* RDAMP[X]S: cf. RDAMP[X] for additional diffusion.
* SDRED (coefficient s_{dred}).
* RDHS (cf. RDHI for additional diffusion).
   - Rayleigh friction:
      * RKRF (K_{fric}), NMAXLEVRF, RRFZ1 (R_{fz1}), RRFPLM (Π_{rfric}), RRFTAU (R_{ftau}).

Some of these variables are in namelist **NAMDYN**.

- **YEMDYN** (LAM model dynamics). All these variables are attributes of YREDYN.
   - RDIVORE, RDIDIVE, RDITE, RDIEGFL, RDISPE, RDIPDE, RDIVDE (LAM counterparts of RDIVOR to RDIVD).
   - RDSVORE, RDSDIVE, RDSVDE (LAM counterparts of RDSVOR to RDSVD).
   - REFILV and REFILD (for pressure gradient filtering).

- **YOMGFL** (GFL dataflow, in particular GFL%Y[X]%LSLHD and GFL%Y[X]_NL%LSLHD).

- **YOMMP0** and **YOMMP** (distributed memory environment, see documentation (IDDM) for more details).

- **YOMNUD** (nudging); some of variables are in namelist **NAMNUD**.

- **YOMSP** and **YOMSP5** (spectral arrays).

- **SPECTRAL_FIELDS_MOD** (spectral arrays, in particuliar type SPECTRAL_FIELD), in project "algor".

7 References.

7.1 Publications:

- Hoskins, B. J., 1980: Representation of the Earth topography using spherical harmonics. *Mon. Wea. Rev.*, **108**, 111-115.

- Jakimow, G., E. Yakimiw, and A. Robert, 1992: An implicit formulation for horizontal diffusion in grid-point models. *Mon. Wea. Rev.*, **120**, 124-130.

- Koshyk, J. N., and G. J. Boer, 1995: Parameterization of dynamical subgrid-scale processes in a spectral G.C.M. . *J. Atmos. Sci.*, **52**, 965-976.

- Sardeshmukh, P. D. and B. J. Hoskins, 1984: Spatial smoothing on the sphere. *Mon. Wea. Rev.*, **112**, 2524-2529.

- Stephenson, D. B., 1994: The impact of changing the horizontal diffusion scheme on the northern winter climatology of a general circulation model. *Quart. J. Roy. Meteor. Soc.*, **120**, 211-226.

- Váña F., P. Bénard, J.F. Geleyn, A. Simon, and Y. Seity, 2008: Semi-Lagrangian advection scheme with controlled damping: an alternative to nonlinear horizontal diffusion in a numerical weather prediction model. *Quart. J. Roy. Meteor. Soc.*, **134**, 523-537.

- Yessad, K. and P. Bénard, 1996: Introduction of a local mapping factor in the spectral part of the METEO-FRANCE global variable mesh numerical forecast model. *Quart. J. Roy. Meteor. Soc.*, **122**, 1701-1719.

7.2 Internal notes:

- (TDECDYN) 2016: IFS technical documentation (CY41R2). Part III: dynamics and numerical procedures. Available at "https://software.ecmwf.int/wiki/display/IFS/Official+IFS+Documentation".

- (TDECTEC) 2016: IFS technical documentation (CY41R2). Part VI: technical and computational procedures. Available at "https://software.ecmwf.int/wiki/display/IFS/Official+IFS+Documentation".

- (IDBAS) Yessad, K., 2017: Basics about ARPEGE/IFS, ALADIN and AROME in the cycle 44 of ARPEGE/IFS (internal note, available on the intranet server "http://www.umr-cnrm.fr/gmapdoc/").

- (IDEUL) Yessad, K., 2017: Integration of the model equations, and Eulerian dynamics, in the cycle 44 of ARPEGE/IFS (internal note, available on the intranet server "http://www.umr-cnrm.fr/gmapdoc/").

- (IDSI) Yessad, K., 2017: Semi-implicit spectral computations in the cycle 44 of ARPEGE/IFS (internal note, available on the intranet server "http://www.umr-cnrm.fr/gmapdoc/").

- (IDTS) Yessad, K., 2017: Spectral transforms in the cycle 44 of ARPEGE/IFS (internal note, available on the intranet server "http://www.umr-cnrm.fr/gmapdoc/").

- (IDSL) Yessad, K., 2017: Semi-Lagrangian computations in the cycle 44 of ARPEGE/IFS (internal note, available on the intranet server "http://www.umr-cnrm.fr/gmapdoc/").

- (IDFPOS) Yessad, K., 2017: FULL-POS in the cycle 44 of ARPEGE/IFS (internal note, available on the intranet server "http://www.umr-cnrm.fr/gmapdoc/").

- (IDRD) Yessad, K., 2017: Sphere to sphere transforms in spectral space in the cycle 44 of ARPEGE/IFS: configuration 911. (internal note, available on the intranet server "http://www.umr-cnrm.fr/gmapdoc/").

- (IDDM) Yessad, K., 2017: Distributed memory features in the cycle 44 of ARPEGE/IFS (internal note, available on the intranet server "http://www.umr-cnrm.fr/gmapdoc/").

- (IDLAM) Zagar, M., and C. Fischer, 2007: The ARPEGE/ALADIN Tech'Book: Implications of LAM aspects on the global model code for CY33/AL33. Internal note, 31pp, available from the authors. Available on the internet server "http://www.umr-cnrm.fr/gmapdoc/".