Semi-lagrangian discretization

ARPEGE-Climat Version 5.1

September 2008

Contents

1 Introduction 3
  1.1 Purpose ........................................ 3
  1.2 Eulerian scheme .................................. 3
  1.3 Semi-Lagrangian scheme ........................... 4

2 Equations 5
  2.1 Notations ........................................ 5
  2.2 3D primitive equations hydrostatic model .......... 6

3 Generic discretization of the equations 7
  3.1 Notations ........................................ 7
  3.2 Discretization for a 3D variable: general case where the RHS has non-zero linear and non-linear terms (GMV variables) . . . 8
  3.3 Discretization for a 3D variable: particular case where the RHS has zero linear and non-linear terms (advectable GFL variables) ........................................ 10
  3.4 Non advectable pseudo-historic GFL variables .......... 12
  3.5 Discretization for a 2D variable in a 3D model (GMVS variables, for example continuity equation) .......................... 12
  3.6 Additional vertical derivatives ...................... 14
  3.7 Remarks for spline cubic vertical interpolations ........ 14
4 Computation of medium and origin points
   4.1 Medium point $M$ ............................................. 15
   4.2 Origin point $O$ ............................................. 18
   4.3 Refined re-computation of point $O$ ......................... 19

5 The SL discretization of the 3D primitive equation model
   5.1 Momentum equation .......................................... 19
   5.2 Thermodynamic equation ................................... 20
   5.3 Continuity equation ......................................... 20
   5.4 GFL variables ................................................ 21
   5.5 Quantities to be interpolated ............................... 21

6 $\mathcal{R}$ operator
   6.1 No tilting .................................................... 22
   6.2 Tilting ........................................................ 22

7 Longitudes and latitudes on the computational sphere

8 Interpolations and weights computations
   8.1 Interpolation grid and weights ............................. 24
       8.1.1 Horizontal interpolation grid and weights for bi-linear
            interpolations ........................................ 24
       8.1.2 Vertical interpolation grid and weights for vertical linear
            interpolations ....................................... 25
       8.1.3 Horizontal interpolation grid and weights for 12-point
            cubic interpolations ................................ 25
       8.1.4 Vertical interpolation grid and weights for vertical cubic
            4-point interpolations .............................. 26
       8.1.5 Vertical interpolation grid and weights for vertical cubic
            Hermite interpolations ............................. 26
       8.1.6 Interpolation grid and weights for tri-linear interpolations
            ................................................... 28
       8.1.7 Interpolation grid and weights for 32-point interpolations
            .................................................. 28
       8.1.8 Other grids and weights .............................. 28
1 Introduction

1.1 Purpose

This chapter describes the semi-Lagrangian scheme used in ARPEGE-CLIMAT. The ARPEGE-IFS code contains many other options. See Yessad (2007b) for a comprehensive description of all available features. Equations will be written without horizontal diffusion scheme (which is treated in spectral computations, see Chapter ??), in order to give a clearer presentation of the discretized equations.

1.2 Eulerian scheme

In Eulerian form of equations, the time dependency equation of a variable $X$ writes as:

$$\frac{\partial X}{\partial t} = -\vec{U} \cdot \nabla \overrightarrow{3}X + \dot{X}$$

(1)
where $\vec{U}$ is the 3D wind, $\nabla^3$ is the 3D gradient operator, $\dot{X}$ is the sum of the dynamical and physical contributions. $X(t + \Delta t)$ is computed knowing $X(t - \Delta t)$ at the same grid-point. Eulerian technique obliges to use a time-step that satisfies to the CFL (Courant Friedrich Levy) condition everywhere. For the variable-mesh spectral global model Arpege, the horizontal CFL condition writes as:

$$m | U | \Delta t \sqrt{\frac{N(N + 1)}{a^2}} < 1$$  \hspace{1cm} (2)

where $m$ is the mapping factor, $| U |$ is the 3D wind module, $N$ is the truncation, $a$ is the Earth radius. The vertical CFL condition writes as:

$$| \dot{\eta} | \Delta t \Delta \eta < 1$$  \hspace{1cm} (3)

For a T359L31 (triangular truncation 359, 31 levels) model with stretching coefficient $c=2.5$ (maximum horizontal resolution about 20 km), that gives $\Delta t \approx 2$ min.

1.3 Semi-Lagrangian scheme

In semi-Lagrangian form of equations, the time dependency equation of a variable $X$ writes as:

$$\frac{dX}{dt} = \dot{X}$$  \hspace{1cm} (4)

In a three-time level semi-Lagrangian scheme $X(t + \Delta t)$ is computed at a grid-point $F$ knowing $X(t - \Delta t)$ at the point $O$ (not necessary a grid-point) where the same particle is at the instant $t - \Delta t$. In a two-time level semi-Lagrangian scheme $X(t + \Delta t)$ is computed at a grid-point $F$ knowing $X(t)$ at the point $O$ (not necessary a grid-point) where the same particle is at the instant $t$. The semi-Lagrangian technique is more expensive for one time-step than the Eulerian technique because it is necessary to compute the positions of the origin point $O$ and the medium point $M$ along the trajectory and to interpolate some quantities at these points (1.5 times the cost of the Eulerian scheme in the T199L31 model with physics). But it allows to use larger time-steps: the stability condition is now the Lipschitz criterion (trajectories do not cross each other) and is less severe than the CFL condition.

$D$ is the divergence of the horizontal wind on the $\eta$-coordinates, $\dot{\eta} = \frac{d\eta}{dt}$.

Lipschitz criterion writes for a two-time level semi-Lagrangian scheme:

$$| D + \frac{\partial \dot{\eta}}{\partial \eta} | \frac{\Delta t}{2} < 1$$  \hspace{1cm} (5)

Arpege-climat using a two-time level scheme (2TL), the three-time level scheme (3TL) is not described in this chapter.
2 Equations

2.1 Notations

- \( \vec{V} \) is the horizontal wind. Its zonal component (on the Gaussian grid) is denoted by \( U \)
- \( D \) is the horizontal wind divergence
- \( w \) is the \( z \)-coordinate vertical velocity: \( w = \frac{dz}{dt} \)
- \( T \) is the temperature. \( T^* \) is a vertically-constant reference temperature which is used in the semi-implicit scheme. Default value is 300 K or 350 K according to configuration (for more details see documentation about semi-implicit scheme). \( T_s^{ST} \) is the reference standard atmosphere surface temperature (288.15 K). \( \left\{ \frac{dT}{dz} \right\}_{ST} \) is the standard atmosphere tropospheric gradient of temperature (−0.0065 K m\(^{-1}\))
- \( q \) is the humidity
- \( p \) is pressure, \( p_s \) is surface pressure. \( p^* \) is a reference pressure and \( p_s^* \) is a reference surface pressure, which are used in the semi-implicit scheme. These reference quantities are vertically dependent and “horizontally” (i.e. on \( \eta \)-surfaces) constant. Default value of \( p_s^* \) is 1000 hPa for a 2TL SL scheme. \( \Delta p^* \) are layer depths corresponding to a surface pressure equal to \( p_s^* \). \( p_s^{ST} \) is a reference pressure equal to the surface pressure of the standard atmosphere (101325 Pa, variable VP00). \( p^{ST} \) is a reference pressure defined on layers and inter-layers corresponding to the surface reference pressure \( p_s^{ST} \) (stored in array STPRE).
- \( \omega = \frac{dp}{dt} \) is the total temporal derivative of pressure
- \( \Phi \) is the geopotential, \( \Phi_s \) is the surface geopotential (i.e. the orography)
- \( \vec{\Omega} \) is the Earth rotation angular velocity
- \( \vec{r} \) is the vector directed upwards, the length of which is the Earth radius
- \( \vec{r} \) is the unit zonal vector on the Gaussian grid
- \( g \) is the gravity acceleration constant
- \( R \) is the gas constant for air, \( R_d \) the gas constant for dry air and \( R_v \) the gas constant for water vapor
- \( c_p \) is the specific heat at constant pressure for air and \( c_p_a \) is the specific heat at constant pressure for dry air
• $c_v$ is the specific heat at constant volume for air and $c_{va}$ is the specific heat at constant volume for dry air.

• $\nabla$ is the first order horizontal gradient on $\eta$-surfaces.

• $B$ is the quantity which defines the vertical hybrid coordinate (see chapter ??): $B$ varies from 1 to 0 from bottom to top. Layer value of $B$ is defined as the average of the adjacent inter-layer value of $B$.

• $\alpha_T$ is a vertical-dependent coefficient used to define a thermodynamic variable $T + \delta_{TR} \frac{\alpha T \Phi_s}{R_d T^{ST}}$ less sensitive to orography than temperature $T$. Expression of $\alpha_T$ is:

$$
\alpha_T = B \left( - \frac{R_d}{g} \left[ \frac{dT}{dz} \right]_{ST} \right) T^{ST} \left( \frac{p^{ST}}{p_{ST}^2} \right) \left( - \frac{R_d}{g} \left[ \frac{dT}{dz} \right]_{ST} - 1 \right)
$$

$\delta_{TR}$ being the switch to come back to temperature.

### 2.2 3D primitive equations hydrostatic model

**Momentum equation:**

Vector form of momentum equation is used. Coriolis force can be treated explicitly ($\delta \vec{V} = 0$) or implicitly ($\delta \vec{V} = 1$).

$$
d\left( \vec{V} + \delta \vec{V} (2 \vec{\Omega} \wedge \vec{r}) \right) = \left[ -2(1 - \delta \vec{V}) (\vec{\Omega} \wedge \vec{V}) \right] - \nabla \Phi - RT \nabla \ln p + F \vec{V}
$$

$F \vec{V}$ is the physical contribution on horizontal wind.

**Temperature equation:**

$$
\frac{dT}{dt} = \frac{RT \omega}{c_p} \frac{1}{p} + F_T
$$

$F_T$ is the physical contribution on temperature. This equation can be modified by replacing temperature $T$ by $T + \delta_{TR} \frac{\alpha T \Phi_s}{R_d T^{ST}}$ which is less sensitive to orography, as it is made for continuity equation (see next paragraph relative to continuity equation). This modification has been proposed by Ritchie and Tanguay (1996).

$$
\frac{d}{dt} \left( T + \delta_{TR} \frac{\alpha T \Phi_s}{R_d T^{ST}} \right) = \frac{d}{dt} \left( \delta_{TR} \frac{\alpha T \Phi_s}{R_d T^{ST}} \right) + \frac{RT \omega}{c_p} \frac{1}{p} + F_T
$$
See Equation (6) for definition of $\alpha_T$. Term $\frac{d}{dt} \left( \delta_{TR} \alpha_T \Phi_s \right)$ only contains advection terms linked to horizontal variations of orography and vertical variations of coefficient $\alpha_T$.

**Humidity equation:**

\[
\frac{dq}{dt} = F_q
\]  

(10)

$F_q$ is the physical contribution on specific humidity. This equation is also valid for other advective variables (GFL).

For non-advectable GFL variables (for example rain in the optional physics $q_r$), the equation is identical to the Eulerian equation:

\[
\frac{\partial q_r}{\partial t} = F_{q_r}
\]  

(11)

$F_{q_r}$ is the liquid precipitation flux

**Continuity equation:**

The impact of water phase changes on atmosphere mass balance is no longer taken into account in ARPEGE-CLIMAT because of the increasing complexity of the code and the unsafety of an option which is not supported by ECMWF.

The conservation of dry air mass, also known as continuity equation, is written as:

\[
\int_0^1 \frac{\partial B}{\partial \eta} \left[ \frac{\ln p_s + \delta_{TR} \Phi_s}{R_d T_{ST}} \right] d\eta = \int_0^1 \frac{\partial B}{\partial \eta} \left( -\frac{1}{p_s} \int_0^1 \nabla \left( \nabla \ln p + \delta_{TR} \frac{\Phi_s}{R_d T_{ST}} \right) d\eta + \frac{\nabla B}{\nabla \eta} \ln p + \frac{\delta_{TR} \Phi_s}{R_d T_{ST}} \right) d\eta
\]  

(12)

Variable $\delta_{TR}$ is 0 or 1; when $\delta_{TR} = 1$ the new variable is less sensitive to the orography (see temperature equation about orographic resonance).

3  **Generic discretization of the equations**

3.1  **Notations**

Upper index:

- $\delta t$ is half time-step
• First integration step: \((\text{resp. } m, o, -)\) for \(t + \Delta t\) (resp. \(t + \delta t, t, t\)) quantity.

• Following integration steps: \((\text{resp. } m, a, -)\) for \(t + \Delta t\) (resp. \(t + \delta t, t, t - \Delta t\)) quantity.

**Lower index:** \(F\) (resp. \(M\) and \(O\)) for final (resp. medium and origin) point.

**The different classes of prognostic variables:** Prognostic variables can be split into different classes:

• **3D variables**, the equation RHS of which has a non-zero adiabatic contribution and a non-zero semi-implicit correction contribution. They are called “GMV” in the code (“GMV” means “grid-point model variables”). This class of variables includes wind components and temperature.

• **3D advectable “conservative” variables**. The equation RHS of these variables has a zero adiabatic contribution, only the diabatic contribution (and the horizontal diffusion contribution) can be non-zero. They are called “GFL” in the code (“GFL” means “grid-point fields”). This class of variables may include liquid/ice cloud water, ozone and TKE.

• **3D non advectable pseudo-historic variables**. The equation RHS of these variables looks like the one of the 3D advectable “conservative” variables, but there is no advection. They are included in the GFL variables. This class of variables may include rain, snow, graupels, convective precipitation flux, stratiform precipitation flux, moisture convergence, total humidity variation or convective vertical velocity.

• **2D variables**, the equation RHS of which mixes 3D and 2D terms, has a non-zero adiabatic contribution and a non-zero semi-implicit correction contribution. They are called “GMVS” in the code (“GMVS” means “grid-point model variables for surface”). This class of variables includes the logarithm of surface pressure (continuity equation).

### 3.2 Discretization for a 3D variable: general case where the RHS has non-zero linear and non-linear terms (GMV variables).

**List of equations:**

• **Momentum equation**
• Temperature equation

**Generic notations:** Generic notation N(X)LAG stands for:

- NWLAG for momentum equation.
- NTLAG for temperature equation.

Generic notation P(X)L0, P(X)L9, P(X)T1 stands for:

- PUL0, PUL9, PUT1 for U-momentum equation.
- PVL0, PVL9, PVT1 for V-momentum equation.
- PTL0, PTL9, PTT1 for temperature equation.

Generic notation P(X)NLT9 stands for:

- PUNLT9 for U-momentum equation.
- PVNLT9 for V-momentum equation.
- PTNLT9 for temperature equation.

Generic notation for total term, linear term, non-linear term, physics:

- $A$ is the total term (sum of dynamical contributions)
- $B$ is the linear term (treated in the semi-implicit scheme)
- $A - \beta B$ is the non-linear term
- $\mathcal{F}$ is the sum of contributions computed in the physical parametrizations

**Other points:**

- High-order interpolations: in the following discretizations, “high-order interpolations” means 32-point interpolation for 3D terms (vertical interpolations are cubic), 12-points interpolations for 2D terms
- Uncentering: $\epsilon$ is a first-order “uncentering factor”; it allows to remove the noise due to gravity waves (orographic resonance)
- Vectors: for vectors like horizontal wind, a rotation operator $\mathcal{R}$ has to be applied from interpolation point to final point:
  - expression interpolated at $O$ has to be replaced by $\mathcal{R}^{OF}_{O}\{\text{this expression}\}$
  - expression interpolated at $M$ has to be replaced by $\mathcal{R}^{MF}_{M}\{\text{this expression}\}$
2TL vertical interpolating SL scheme: stable discretization (LSET-TLS=T) and first-order uncentering

The basic equation can be written as:

\[
\frac{dX}{dt} = A + F
\]  \hspace{1cm} (13)

When \( N(X)_{\text{LAG}}=3 \), Equation (13) is discretized by averaging the non-linear term along the trajectory in the RHS.

The \( t + \delta t \) non-linear term \( A^m - \beta B^m \) is calculated by a linear space and time extrapolation.

\[
\frac{1}{2}(1 + \epsilon)[A^o - \beta B^o]|_F + \frac{1}{2}(2 - \epsilon)[A^o - \beta B^o]|_O - \frac{1}{2}[A^- - \beta B^-]|_O
\]

This type of extrapolation is available only for \( N(X)_{\text{LAG}}=3 \). At the first time integration step, values at \( t + \delta t \) are set equal to initial values. Quantity \( A^o - \beta B^o \) has to be saved in a buffer \( P(X)_{\text{NLT9}} \) to be available as \( A^- - \beta B^- \) for the following time-step.

Equation (13) is discretized as follows:

\[
(X - (1 + \epsilon)\delta t \beta B)\frac{\xi}{F} = \{X^o + [(2 - \epsilon)\delta t A - (2 - \epsilon)\delta t \beta B]^o \\
- \delta t[A - \beta B]^o + [(1 - \epsilon)\delta t \beta B + \Delta t F]^o|_O \\
+ \{(1 + \epsilon)\delta t A - (1 + \epsilon)\delta t \beta B]^o\}_F
\]

Buffers content before interpolations for \( N(X)_{\text{LAG}}=3 \):

- \( P(X)_{\text{L0}}: [(2 - \epsilon)\delta t A - (2 - \epsilon)\delta t \beta B]^o - \delta t[A - \beta B]^o + [(1 - \epsilon)\delta t \beta B]^o \) for tri-linear interpolation at the origin point \( O \).
- \( P(X)_{\text{L9}}: X^o + [\Delta t F]^o \) for high-order interpolation at the origin point \( O \).
- \( P(X)_{\text{T1}}: [(1 + \epsilon)\delta t A - (1 + \epsilon)\delta t \beta B]^o \) then provisional add of quantity \( [(1 + \epsilon)\delta t \beta B]^o \) before \( t + \Delta t \) physics; evaluated at the final point \( F \).

3.3 Discretization for a 3D variable: particular case where the RHS has zero linear and non-linear terms (advectable GFL variables)

List of equations:

Humidity equation, and for example:
• Specific humidity equation
• Ozone equation
• Liquid water equation
• Ice equation
• TKE equation
• Extra GFL variables equations

Generic notations

Generic notation $P(X)L9\text{, }P(X)T1$ stands for:

• $PGFLL9\text{, }PGFL T1$ for GFL variables

In the present case $A$ and $B$ are equal to zero.

Other points:

• High-order interpolations: in the following discretizations, “high-order interpolations” means 32-point interpolations for 3D terms (vertical interpolations are cubic), 12-point interpolations for 2D terms. For ozone, vertical cubic interpolations can be replaced by vertical Hermite cubic interpolations (switch $YO3\_NL\%LHV$ in NAMGFL), or vertical spline cubic interpolations (switch $YO3\_NL\%LSPLIP$ in NAMGFL).

• Uncentering: $\epsilon$ is a first-order “uncentering factor”; It allows to remove the noise due to gravity waves (orographic resonance)

2TL vertical interpolating SL scheme

At the first time integration step, values at $t + \delta t$ are set equal to initial values. This discretization of the 2TL SL scheme follows (Mc Donald and Haugen, 1992).

Equation (13) is discretized as follows:

$$X^{\pm}_F = \{X^o + [\Delta t F]^o\}^O$$  \hspace{1cm} (14)

Buffers content before interpolations:

• $P(X)L9\text{: }X^o + [\Delta t F]^o$ for high-order interpolation at the origin point $O$

• $P(X)T1$ contains zero; evaluated at the final point $F$
3.4 Non advectable pseudo-historic GFL variables

For these variables the discretization always writes:

\[ X^+_F = \{ X^o + [\Delta t F]^o \}_F \]  

(15)

and there are no interpolations.

3.5 Discretization for a 2D variable in a 3D model (GMVS variables, for example continuity equation)

The equation which is now discretized is:

\[ [R_{\text{inte}}]_{\text{(top, surf)}} \left\langle \frac{W_{\text{v}}}{\Delta \eta} \frac{dX}{dt} \right\rangle = [R_{\text{inte}}]_{\text{(top, surf)}} \left\langle \frac{W_{\text{v}}}{\Delta \eta} A \right\rangle \]

\[ + [R_{\text{inte}}]_{\text{(top, surf)}} \left\langle \frac{W_{\text{v}}}{\Delta \eta} F \right\rangle \]

where:

\[ [R_{\text{inte}}]_{\text{(top, surf)}} \left\langle \frac{W_{\text{v}}}{\Delta \eta} \right\rangle = 1 \]  

(16)

and \([R_{\text{inte}}]_{\text{(top, surf)}}\) is the vertical integral matrix operator (the scalar product \([R_{\text{inte}}]_{\text{(top, surf)}}(X)\) is the discretization of \(\int_0^L X d\eta\), \(X\) is the vector containing the layer values of \(X\): \((X_1; X_2; \ldots; X_l; \ldots; X_L)\)).

In the thin layer equations, expression of \(W_{\text{v}}\) at full levels is:

\[ [W_{\text{v}}]_l = \Delta B_l \]  

(17)

List of equations:

- Continuity equation

Generic notations: Generic notation \(N(X)_{\text{LAG}}\) stands for

- NVLAG for continuity equation

Generic notations \(P(X2D)9\) (2D term), \(P(X)T1\) (2D term), \(P(X3D)L9\) (3D term), stand for
• PX9, PSPT1, PCL9 for continuity equation.

Generic notation P(X)NLT9 (3D term) stands for:

• PSPNLT9 for continuity equation.

Generic notation for total term, linear term, non-linear term, physics:

• $A$ is the total term (sum of dynamical contributions): it is assumed to be a 3D term (sum of 3D and 2D contributions).

• $B$ is the linear term (treated in the semi-implicit scheme): it is assumed to be a 2D term (vertical integral of a 3D term).

• the difference $A - \beta B$ is the non-linear term, considered as a 3D term.

Other points

Horizontal interpolation of 2D terms: since the horizontal position of the interpolation point is vertical dependent, horizontal interpolations of 2D quantities have to be done for each layer. For example, when interpolating a 2D surface variable $SV$ at the origin point, $[R_{inte}]_{(top, surf)} \left[ \left[ \frac{W_{v ei}}{\Delta \eta} \right]_F [SV]_O \right]$ has no reason to be equal to $[SV]_{O(\eta=1)}$, these quantities are generally different: this is $[R_{inte}]_{(top, surf)} \left[ \left[ \frac{W_{v ei}}{\Delta \eta} \right]_F [SV]_O \right]$ which has to be computed.

$\left[ \left[ \frac{W_{v ei}}{\Delta \eta} \right]_F [SV]_O \right]$ is the vector containing $\left[ \frac{W_{v ei}}{\Delta \eta} \right]_F [SV]_{O(l)}$, for $l = 1$ to $L$.

2TL vertical interpolating SL scheme: stable discretization (LSTLS=.T.) and first-order uncentering

The $t + \delta t$ non-linear term $A^m - \beta B^m$ is calculated by a linear space and time extrapolation:

$$\frac{1}{2} (1 + \epsilon) [A^0 - \beta B^0]_F + \frac{1}{2} (2 - \epsilon) [A^0 - \beta B^0]_{O(l)} - \frac{1}{2} [A^- - \beta B^-]_{O(l)}$$

This type of extrapolation is available only for N(X)LAG=3. At the first time integration step, values at $t + \delta t$ are set equal to initial values. Quantity $A^0 - \beta B^0$ has to be saved in a buffer P(X)NLT9 to be available as $A^- - \beta B^-$ for the following time-step.

Equation (13) is discretized as follows:
\[(X - (1 + \epsilon)\delta t\beta B)^+ = [\mathbf{R}_{\text{inte}}]_{\text{top, surf}} \left\{ \frac{\mathbf{W}_{\text{rej}}}{\Delta \eta} \right\} F \{X^o + \{(2 - \epsilon)\delta t \mathbf{A} - (2 - \epsilon)\delta t\beta \mathbf{B}\}^o - \delta t[\mathbf{A} - \beta \mathbf{B}]^- \right. \\
+ \left. \left. [(1 - \epsilon)\delta t\beta \mathbf{B} + \Delta t \mathbf{F}]^o \right\}_O + \{(1 + \epsilon)\delta t \mathbf{A} - (1 + \epsilon)\delta t\beta \mathbf{B}\}^o \} F\]

Buffers content before interpolations for N(X)LAG=3:

- P(X3D)L0 is not used.
- P(X3D)L9: \([(2 - \epsilon)\delta t \mathbf{A} - (2 - \epsilon)\delta t\beta \mathbf{B} - \delta t[\mathbf{A} - \beta \mathbf{B}]^- + \{(1 - \epsilon)\delta t\beta \mathbf{B}\}^o\] for tri-linear interpolation at the origin point O(l)
- P(X2D)0 is not used
- P(X2D)9: \(X^o + [\Delta t \mathbf{F}]^o\) for horizontal high-order interpolation at the origin point O(l)
- P(X)T1: \([(1 + \epsilon)\delta t \mathbf{A} - (1 + \epsilon)\delta t\beta \mathbf{B}]^o\) then provisional add of quantity \([(1 + \epsilon)\delta t\beta \mathbf{B}]^o\) before lagged physics; evaluated at the final point F

### 3.6 Additional vertical derivatives

If \(\delta_{TR}\) is non-zero, discretization of temperature equation needs to compute the vertical advection \(\left( \frac{\partial T}{\partial \eta} \right)\) (at full levels) of \(\alpha_T\). Layers values of \(\alpha_T\) (array RCORDIF) are used to define \(T + \delta_{TR} \frac{\alpha_T \Phi_s}{R_d T_{ST}}\), but inter-layer values of \(\alpha_T\) (array RCORDIH) are used to compute vertical advection.

### 3.7 Remarks for spline cubic vertical interpolations

In this case the vertical interpolation uses all model levels and can be written as the product of two vertical interpolations: the first one uses all model levels and can be done at \(F\) in the unlagged grid-point calculations (the intermediate quantity obtained is stored in the array P(X)SPL9), the second one is a 4 points interpolation, done in the lagged grid-point calculations in the interpolation routine. Interpolation routine uses both P(X)SPL9 (for interpolations) and P(X)L9 to apply a monotonicity constraint.
4 Computation of medium and origin points

4.1 Medium point $M$

The medium point is calculated in subroutines LARMES and LARMES2. Trajectories are great circles on the geographical sphere. The computation of the medium point $M$ location of the Lagrangian trajectory is performed by an iterative method described by Robert (1981) and adapted to the sphere by Rochas. In a 3TL SL scheme, the particle is at the point $M$ at time $t$ ($t + \delta t$ for the first integration step). In a 2TL SL scheme, the particle is at the point $M$ at $t + \delta t$. $M$ is at the middle position of the origin point $O$ and the final point $F$.

Notations:

- $R^{MF}$ is the rotation operator from medium point to final point (see section 6)
- $R^{OF}$ is the rotation operator from origin point to final point (see section 6)
- $r^F = C^F$ (C Earth center, $F$ final point)
- $r^M = C\bar{M}$ ($M$ medium point)
- $\phi^{MF}$: angle ($\hat{C}\bar{M}, \hat{C}F$)
- $\theta^F, \lambda^F$: latitude, longitude on the geographical sphere of $F$
- $\theta^M, \lambda^M$: latitude, longitude on the geographical sphere of $M$
- $\bar{V}^M$: interpolated horizontal wind at $M$ (at $t + \delta t$)
- $\bar{V}^O$: interpolated horizontal wind at $O$ (at $t + \delta t$)
- $a$ is the average Earth radius near the surface
- $\bar{r} = ak$
- $\Delta t$: time-step
- $\delta t$: half time-step
- $L$: number of layers of the model
Definition of the vertical coordinate $\eta$:

Research of medium point needs an exact definition of the vertical coordinate $\eta$. For the inter-layer number $l$ (between 0 and $L$), $\eta_l$ is defined by:

$$\eta_l = \frac{l}{L}$$  \hspace{1cm} (18)

For the layer number $l$ (between 1 and $L$), $\eta_l$ is defined by:

$$\eta_l = \frac{1}{2}(\eta_l + \eta_{l-1})$$  \hspace{1cm} (19)

Stable algorithm for 2TL SL scheme (LSETTLS=.T.)

Extrapolation of the wind:

This algorithm has been developed by Hortal (2002). The basic idea is to replace the purely temporal extrapolation by a space and time extrapolation:

$$R^{MF} \begin{bmatrix} \vec{V} \end{bmatrix}_M(t+\delta t) = \frac{3}{2} R^{NF} \begin{bmatrix} \vec{V} \end{bmatrix}_N(t) - \frac{1}{2} R^{OF} \begin{bmatrix} \vec{V} \end{bmatrix}_O(t - \Delta t)$$  \hspace{1cm} (20)

where $N$ is the position of the particle at time $t$ for a particle which goes from the origin point $O$ at time $t - \Delta t$ to $M$ at time $t + \delta t$. Assuming that the wind is constant along the trajectory one can write:

$$ON = 2NM = 0.5NF$$  \hspace{1cm} (21)

and evaluate the angular velocity $R^{NF} \begin{bmatrix} \vec{V} \end{bmatrix}_N(t)$ by

$$\frac{2}{3}R^{MF} \begin{bmatrix} \vec{V} \end{bmatrix}_M(t) + \frac{1}{3}R^{OF} \begin{bmatrix} \vec{V} \end{bmatrix}_O(t)$$ or

$$\frac{1}{3}R^{MF} \begin{bmatrix} \vec{V} \end{bmatrix}_M(t) + \frac{2}{3}R^{OF} \begin{bmatrix} \vec{V} \end{bmatrix}_O(t).$$

Expression of $\begin{bmatrix} \vec{V} \end{bmatrix}_M(t+\delta t)$ becomes:

$$R^{MF} \begin{bmatrix} \vec{V} \end{bmatrix}_M(t+\delta t) = \frac{1}{2} [\vec{V}]^F(t) + \frac{1}{2} R^{OF} (2[\vec{V}]^O(t) - [\vec{V}]^O(t - \Delta t))$$ \hspace{1cm} (22)

The same type of extrapolation is done for the $\eta$-coordinate vertical velocity. The algorithm of research of trajectory uses directly the RHS of this equation, and for all iterations the origin point $O$ is computed instead of the medium point $M$.

Algorithm

The origin point is defined by the following iterative scheme: for the iteration $k + 1$:

$$[\vec{r}]^O_{k+1} = [\vec{r}]^F \cos \phi_k - \frac{[\vec{V}]_k^F(t) + R^{OF} (2[\vec{V}]_k^O(t) - [\vec{V}]_k^O(t - \Delta t))}{| [\vec{V}]_k^F(t) + R^{OF} (2[\vec{V}]_k^O(t) - [\vec{V}]_k^O(t - \Delta t)) |} \sin \phi_k$$  \hspace{1cm} (23)
where:

$$\phi_k = \delta t \mid [\vec{V}]^F(t) + \mathcal{R}^{OF}(2[\vec{V}]_k^O(t) - [\vec{V}]_k^O(t - \Delta t)) \mid$$  \hspace{1cm} (24)

Some geometrical approximations (small angles on the sphere) yield:

$$[\bar{\vec{r}}]_{k+1}^O = [\vec{r}]^F \left(1 - \frac{\phi_k^2}{2}\right) - ([\vec{V}]^F(t) + \mathcal{R}^{OF}(2[\vec{V}]_k^O(t) - [\vec{V}]_k^O(t - \Delta t))) \delta t \left(1 - \frac{\phi_k^2}{6}\right)$$  \hspace{1cm} (25)

On the vertical:

$$\eta_{k+1}^O = \eta^F - 2\delta t(0.5\eta^F(t) + 0.5(2\dot{\eta}_k^O(t) - \dot{\eta}_k^O(t - \Delta t)))$$  \hspace{1cm} (26)

First iteration

One starts with $$M_0 = F, \ [\vec{V}]^F(t)$$ as a first guess for the space and time extrapolated horizontal angular velocity, $$\phi_0 = \Delta t \mid [\vec{V}]^F(t) \mid, \dot{\eta}^F(t)$$ as a first guess for the space and time extrapolated $$\eta$$-coordinate vertical wind. Quantities at $$t$$ are taken as a first guess and not quantities at $$(t + \delta t)$$, contrary to the case $$\text{LSETTLS}=F$$. The coordinates of $$O_1$$ are $$(2[\vec{V}](t) - [\vec{V}](t - \Delta t))$$. $$(2\dot{\eta}(t) - \dot{\eta}(t - \Delta t))$$ is interpolated at this point, which allows to compute the wind components which will be used for the next iteration.

Following iterations

$$[\vec{V}']$$ (of coordinates $$(\vec{u}', \vec{v}')$$ ) is a generic notation for 1/2([\vec{V}]^F(t) + \mathcal{R}^{OF}(2[\vec{V}]_k^O(t) - [\vec{V}]_k^O(t - \Delta t))).$

For horizontal displacement use equations:

$$\sin \theta_{k+1}^O = \sin \theta^F \cos \phi_k - \frac{[\vec{v}']}{|\vec{V}'|} \cos \theta^F \sin \phi_k$$  \hspace{1cm} (27)

$$\cos \theta_{k+1}^O \cos(\lambda_{k+1}^O - \lambda^F) = \cos \theta^F \cos \phi_k + \frac{[\vec{u}']}{|\vec{V}'|} \sin \theta^F \sin \phi_k$$  \hspace{1cm} (28)

$$\cos \theta_{k+1}^O \sin(\lambda_{k+1}^O - \lambda^F) = -\frac{[\vec{u}']}{|\vec{V}'|} \sin \phi_k$$  \hspace{1cm} (29)

For vertical displacement use equation:

$$\eta_{k+1}^O = \eta^F - \delta t(\eta^F(t) + (2\dot{\eta}_k^O(t) - \dot{\eta}_k^O(t - \Delta t)))$$  \hspace{1cm} (30)
4.2 Origin point $O$

The origin point is calculated in subroutines LARMES and LAINOR2.

In a 2TL SL scheme, the particle is at the point $O$ at time $t$.

$O$ is on the same great circle arc (on the geographical sphere) as $M$ and $F$ and the length of $OF$ is twice the length of $MF$. If angle $(\vec{r}_O^M, \vec{r}_F^F)$ is small (less than 10°, what is generally satisfied), one can write for horizontal displacement:

$$ [\vec{r}]^O - [\vec{r}]^F \simeq 2([\vec{r}]^M - [\vec{r}]^F) \quad (31) $$

For vertical displacement one can always write:

$$ \eta^O - \eta^F = 2(\eta^M - \eta^F) \quad (32) $$

One denotes by:

- $\phi = ([\vec{r}]^M, [\vec{r}]^F)$
- $[\vec{V}]'$ (of coordinates $[u', v']$) the last interpolated horizontal velocity.

Using the following identities:

$$ \cos 2\phi = 2 \cos^2 \phi - 1 \quad (33) $$

$$ \sin 2\phi = 2 \sin \phi \cos \phi \quad (34) $$

the origin point horizontal coordinates can be computed by:

$$ \sin \theta^O = \sin \theta^F \cos 2\phi - 2 \cos \phi \left[ \frac{[v']}{|\vec{V}'|} \cos \theta^F \sin \phi \right] \quad (35) $$

$$ \cos \theta^O \cos(\lambda^O - \lambda^F) = \cos \theta^F \cos 2\phi + 2 \cos \phi \left[ \frac{[u']}{|\vec{V}'|} \sin \theta^F \sin \phi \right] \quad (36) $$

$$ \cos \theta^O \sin(\lambda^O - \lambda^F) = -2 \cos \phi \left[ \frac{[u']}{|\vec{V}'|} \sin \phi \right] \quad (37) $$

Terms in brackets are already computed to determine $M$. 
4.3 Refined re-computation of point $O$

Option L2TLFF controls re-computation of the origin point using the average between the angular velocity at the origin point and the provisional $t + \Delta t$ angular velocity, according to the algorithm previously described. Only term $(2\vec{\Omega} \wedge a\vec{k})$ is computed (always analytically) at this improved position of $O$ (so L2TLFF is active only if LADVF=.T. or LADVFW=.T.). Refined re-computation of point $O$ is available only in a limited set of options. Equations system is integrated to find a first guess of $\vec{V}^F(t + \Delta t)$ and also a first guess of $p_s(t + \Delta t)$ which provides $1/2([\vec{V}]^F + R^O \vec{V}^O)$ is used to recompute $O$. A correction $(2\vec{\Omega} \wedge a\vec{k})(O$ improved) − $(2\vec{\Omega} \wedge \vec{k})(O$) is analytically computed and added to wind equation to find the “improved” value of $\vec{V}^F(t + \Delta t)$. Computations are currently made in routine LAPINEB and LADINE.

5 The SL discretization of the 3D primitive equation model

5.1 Momentum equation

Definition of $X$, $A$, $B$ and $F$, top and bottom values

\[
X = \vec{V} + \delta_\psi(2\vec{\Omega} \wedge \vec{r}) \tag{38}
\]

\[
A = [-2(1 - \delta_\psi)(\vec{\Omega} \wedge \vec{V})] - \vec{\nabla}\Phi - RT\vec{\nabla}(\log(p)) \tag{39}
\]

\[
B = -\vec{\nabla}\left[\frac{\gamma T}{ps} + \frac{RdT^{s}}{ps}p_s\right] + \beta C_o[-2(1 - \delta_\psi)(\vec{\Omega} \wedge \vec{V})] \tag{40}
\]

\[
F = \vec{F}\phi \tag{41}
\]

Top:

\[
\vec{V}_{\eta=0} = \vec{V}_{l=1} \tag{42}
\]

Bottom:

\[
\vec{V}_{\eta=1} = \vec{V}_{l=L} \tag{43}
\]

Remarks

- Coriolis term is treated implicitly ($\delta_\psi = 1$, LADVF=.T.) This means that $(2\vec{\Omega} \wedge \vec{r})$ is analytically computed
- With option L2TLFF, term $(2\vec{\Omega} \wedge \vec{r})$ is recomputed at an improved position of the origin point
5.2 Thermodynamic equation

**Definition of \( X, A, B \) and \( F \), top and bottom values**

\[
X = T + \delta_{TR} \frac{\alpha T \Phi_s}{R_d T_{ST}} \tag{44}
\]

\[
A = \frac{RT \omega}{c_p p} + \delta_{TR} \frac{\alpha T}{R_d T_{ST}} \vec{V} \nabla (\Phi_s) + \delta_{TR} \frac{\Phi_s}{R_d T_{ST}} \left( \eta \frac{d \alpha T}{d \eta} \right) \tag{45}
\]

\[
B = -\mathcal{M}^2 T \mathcal{D}' \tag{46}
\]

\[
F = F_T \tag{47}
\]

**Top:**

\[
T_{\eta=0} = T_{|z=1} \tag{48}
\]

**Bottom:**

\[
T_{\eta=1} = T_{|z=L} \tag{49}
\]

5.3 Continuity equation

**Definition of \( X, A, B, \) and \( F \)**

\[
X = \log p_s + \delta_{TR} \frac{\Phi_s}{R_d T_{st}} \tag{50}
\]

\[
A = -\frac{1}{p_s} \int_{\eta=0}^{\eta=1} \nabla \left( \vec{V} \frac{\partial p}{\partial \eta} \right) d\eta + \vec{V} \nabla \left[ \log p_s + \delta_{TR} \frac{\Phi_s}{R_d T_{st}} \right] \tag{51}
\]

\[
B = -\frac{\mathcal{M}}{\mathcal{M}^2} \nu D \tag{52}
\]

\[
B' = 0 \tag{53}
\]

\[
F = 0 \tag{54}
\]
5.4 GFL variables

We detail here the case of specific moisture.

**Definition of \( X \), \( A \), \( B \) and \( \mathcal{F} \), top and bottom values**

\[
X = q \quad (55)
\]

\[
A = 0 \quad (56)
\]

\[
B = 0 \quad (57)
\]

\[
\mathcal{F} = F_q \quad (58)
\]

Top:

\[
q_{\eta=0} = q_{l=1} \quad (59)
\]

Bottom:

\[
q_{\eta=1} = q_{l=L} \quad (60)
\]

The other GFL are treated similarly. Quantities are assumed constant above the middle of the upper layer and below the middle of the lower layer.

5.5 Quantities to be interpolated

The computation is performed in subroutine LACDYN.

**Research of trajectory**

When researching the medium point by an iterative algorithm, the interpolation at the origin point of \((U, V, \dot{\eta})\) is needed: a tri-linear interpolation is performed. For more details about interpolations, see section 8.

**RHS of equations**

The list of quantities to be interpolated has been described in subsections 3.2, 3.3 and 3.5 for each type of equation.

**Additional quantities to be interpolated at the origin point if L2TLFF**

The two components of the \( \vec{V} \) at time \( t \)
6 \ R \ \text{operator}

6.1 \ \text{No tilting}

To transport a vector along a trajectory (part of a great circle) from an origin point \(O\) to a final point \(F\) the following operator \(R^{OF}\) is defined:

\[
\vec{V}' = R^{OF}(\vec{V})
\]

where \(\vec{V}'\) has coordinates \((u', v')\), \(\vec{V}\) has coordinates \((u, v)\), and the relationship between \((u, v)\) and \((u', v')\) is:

\[
\begin{pmatrix}
  u' \\
  v'
\end{pmatrix} = \begin{pmatrix}
  p & q \\
  -q & p
\end{pmatrix} \begin{pmatrix}
  u \\
  v
\end{pmatrix}
\]

where:

\[
p = \frac{\vec{F} \vec{O} + \vec{F} \vec{O}}{1 + k^{F}k^{O}} = \frac{\cos \theta^{F} \cos \theta^{O} + (1 + \sin \theta^{F} \sin \theta^{O}) \cos(\lambda^{F} - \lambda^{O})}{1 + \cos \phi}
\]

\[
q = \frac{\vec{F} \vec{O} + \vec{F} \vec{O}}{1 + k^{F}k^{O}} = \frac{(\sin \theta^{F} + \sin \theta^{O}) \sin(\lambda^{F} - \lambda^{O})}{1 + \cos \phi}
\]

(Notations \(\theta^{O}, \theta^{F}, \lambda^{O}, \lambda^{F}, \phi; \text{ see section 4}).

\(p\) and \(q\) verify the following identity:

\[
p^2 + q^2 = 1
\]

Computation of \(p\) and \(q\) is made in subroutine LARCHE.

6.2 \ \text{Tilting}

The coordinates of \(\vec{V}'\) and \(\vec{V}\) are linked by the following relationship:

\[
\begin{pmatrix}
  u' \\
  v'
\end{pmatrix} = \begin{pmatrix}
  GNORDM & GNORDL \\
  -GNORDL & GNORDM
\end{pmatrix} \begin{pmatrix}
  p & q \\
  -q & p
\end{pmatrix} \times
\begin{pmatrix}
  \cos \alpha & -\sin \alpha \\
  \sin \alpha & \cos \alpha
\end{pmatrix} \begin{pmatrix}
  u \\
  v
\end{pmatrix}
\]

(66)
where:

\[
\cos \alpha = \frac{2c}{A \cos \Theta} \left[ \sin \theta_p \cos \theta^O - \sin \theta^O \cos \theta_p \cos (\lambda^O - \lambda_p) \right] \quad (67)
\]

\[
\sin \alpha = \frac{2c}{A \cos \Theta} \left[ \cos \theta_p \sin (\lambda^O - \lambda_p) \right] \quad (68)
\]

\[
A = (1 + c^2) + (1 - c^2)(\sin \theta_p \sin \theta^O + \cos \theta_p \cos \theta^O \cos (\lambda^O - \lambda_p)) \quad (69)
\]

and where:

- \(c\) is the stretching coefficient
- \(\Theta^O\) is the latitude on the computational sphere of the origin point \(O\)
- \((\theta_p, \lambda_p)\) are the latitude and longitude on the geographical sphere of the stretching pole
- \(p\) and \(q\) are computed like in the not tilted case (in subroutine LARCHE)
- \(\cos \alpha\) and \(\sin \alpha\) are also computed in subroutine LARCHE
- \((GNORDL, GNORDM)\) are the coordinates in the computational sphere of the unit vector directed towards the true north, computed in subroutine SUGEM2

7 Longitudes and latitudes on the computational sphere

For interpolations it is necessary to compute \((\Theta^O, \Lambda^O)\), latitude and longitude of the interpolation point \(O\) in the computational sphere. The iterative algorithm allowing to find \(O\) gives \((\theta^O, \lambda^O)\), latitude and longitude in the geographical sphere (more exactly \(\sin \theta^O, \cos \theta^O \cos \lambda^O - \lambda^F\) and \(\cos \theta^O \sin \lambda^O - \lambda^F\) where \((\theta^F, \lambda^F)\) are the coordinates of the final point on the geographical sphere). Transform formulas giving \((\Theta, \Lambda)\) on the computational sphere once knowing \((\theta, \lambda)\) on the geographical sphere are given by:

\[
\sin \Theta = \frac{(1 - c^2) + (1 + c^2)(\sin \theta_p \sin \theta + \cos \theta_p \cos \theta \cos (\lambda - \lambda_p))}{A} \quad (70)
\]

\[
\cos \Theta \cos \Lambda = \frac{2c(\cos \theta_p \sin \theta - \sin \theta_p \cos \theta \cos (\lambda - \lambda_p))}{A} \quad (71)
\]
\[ \cos \Theta \sin \Lambda = \frac{2c \cos \theta \sin(\lambda - \lambda_p)}{A} \]  

(72)

where:

- \( A = (1 + c^2) + (1 - c^2)(\sin \theta_p \sin \theta + \cos \theta_p \cos \theta \cos(\lambda - \lambda_p)) \)
- \( c \) is the stretching coefficient
- \((\theta_p, \lambda_p)\) are the latitude and longitude on the geographical sphere of the stretching pole
- Computation of \( \Theta, \Lambda \) is made in subroutine LARCHF

8 Interpolations and weights computations

8.1 Interpolation grid and weights

Computation is done in subroutine LASCAW.

8.1.1 Horizontal interpolation grid and weights for bi-linear interpolations

A 16 points horizontal grid is defined as it is shown in figure 1. The interpolation point \( O \) (medium or origin point) is between \( B_1, C_1, B_2 \) and \( C_2 \). \( \Lambda \) and \( \Theta \) are the longitudes and latitudes on the computational sphere. The weights are defined as follows:

- zonal weight number 1:

\[ ZDLO1 = \frac{\Lambda_O - \Lambda_{B_1}}{\Lambda_{C_1} - \Lambda_{B_1}} \]

- zonal weight number 2:

\[ ZDLO2 = \frac{\Lambda_O - \Lambda_{B_2}}{\Lambda_{C_2} - \Lambda_{B_2}} \]

- meridional weight:

\[ ZDLAT = \frac{\Theta_O - \Theta_{B_1}}{\Theta_{B_2} - \Theta_{B_1}} \]
8.1.2 Vertical interpolation grid and weights for vertical linear interpolations

A 4 points vertical grid is defined as it is shown in figure 2. The interpolation point $O$ (medium or origin point) is between $T_{i+1}$ and $T_{i+2}$. The vertical weight is defined by:

$$ZDVER = \frac{\eta_O - \eta_{T_{i+1}}}{\eta_{T_{i+2}} - \eta_{T_{i+1}}}$$

8.1.3 Horizontal interpolation grid and weights for 12-point cubic interpolations

A 16 points horizontal grid is defined as it is shown in figure 3. The interpolation point $O$ (medium or origin point) is between $B_1$, $C_1$, $B_2$ and $C_2$. The weights are defined as follows:

- zonal weight number 0:
  $$ZDLO0 = \frac{\Lambda_O - \Lambda_{B_0}}{\Lambda_{C_0} - \Lambda_{B_0}}$$

- zonal weight number 1:
  $$ZDLO1 = \frac{\Lambda_O - \Lambda_{B_1}}{\Lambda_{C_1} - \Lambda_{B_1}}$$

- zonal weight number 2:
  $$ZDLO2 = \frac{\Lambda_O - \Lambda_{B_2}}{\Lambda_{C_2} - \Lambda_{B_2}}$$

- zonal weight number 3:
  $$ZDLO3 = \frac{\Lambda_O - \Lambda_{B_3}}{\Lambda_{C_3} - \Lambda_{B_3}}$$

- meridional weights:
  $$ZCLA2 = \frac{(\Theta_O - \Theta_{B_0})(\Theta_O - \Theta_{B_2})(\Theta_O - \Theta_{B_3})}{(\Theta_{B_1} - \Theta_{B_0})(\Theta_{B_1} - \Theta_{B_2})(\Theta_{B_1} - \Theta_{B_3})}$$
  $$ZCLA3 = \frac{(\Theta_O - \Theta_{B_0})(\Theta_O - \Theta_{B_1})(\Theta_O - \Theta_{B_3})}{(\Theta_{B_2} - \Theta_{B_0})(\Theta_{B_2} - \Theta_{B_1})(\Theta_{B_2} - \Theta_{B_3})}$$
  $$ZCLA4 = \frac{(\Theta_O - \Theta_{B_0})(\Theta_O - \Theta_{B_1})(\Theta_O - \Theta_{B_2})}{(\Theta_{B_3} - \Theta_{B_0})(\Theta_{B_3} - \Theta_{B_1})(\Theta_{B_3} - \Theta_{B_2})}$$
8.1.4 Vertical interpolation grid and weights for vertical cubic 4-point interpolations

A 4-point vertical grid is defined as it is shown in figure 2. The interpolation point \( O \) (medium or origin point) is between \( T_{l+1} \) and \( T_{l+2} \). The vertical weights are defined by:

\[
ZCVE2 = \frac{(\eta_O - \eta_{T_l})(\eta_O - \eta_{T_{l+2}})(\eta_O - \eta_{T_{l+3}})}{(\eta_{T_{l+1}} - \eta_{T_l})(\eta_{T_{l+1}} - \eta_{T_{l+2}})(\eta_{T_{l+1}} - \eta_{T_{l+3}})}
\]

\[
ZCVE3 = \frac{(\eta_O - \eta_{T_l})(\eta_O - \eta_{T_{l+1}})(\eta_O - \eta_{T_{l+3}})}{(\eta_{T_{l+2}} - \eta_{T_l})(\eta_{T_{l+2}} - \eta_{T_{l+1}})(\eta_{T_{l+2}} - \eta_{T_{l+3}})}
\]

\[
ZCVE4 = \frac{(\eta_O - \eta_{T_l})(\eta_O - \eta_{T_{l+1}})(\eta_O - \eta_{T_{l+2}})}{(\eta_{T_{l+3}} - \eta_{T_l})(\eta_{T_{l+3}} - \eta_{T_{l+1}})(\eta_{T_{l+3}} - \eta_{T_{l+2}})}
\]

8.1.5 Vertical interpolation grid and weights for vertical cubic Hermite interpolations

A 4-point vertical grid is defined as it is shown in figure 2. The interpolation point \( O \) (medium or origin point) is between \( T_{l+1} \) and \( T_{l+2} \). First weights to compute vertical derivatives at layers \( l+1 \) and \( l+2 \) are computed. For a variable \( X \), \( \frac{\partial X}{\partial \eta} \) is computed as close as possible as \( \left( \frac{\dot{\eta} \frac{\partial X}{\partial \eta}}{\dot{\eta}} \right) \), but with additional approximations allowing to avoid horizontal interpolations for term \( \left( \frac{\dot{\eta} \frac{\partial p}{\partial \eta}}{\dot{\eta}} \right) \).

- For layers other than the first or the last layer, discretization follows:
  \[
  \left( \frac{\partial X}{\partial \eta} \right)_{l+1} = \frac{1}{2} \frac{X_{l+2} - X_l}{\eta_{l+1} - \eta_l}
  \]
  \[(73)\]

- For layer \( l = 1 \), discretization assumes that \( \left( \frac{\dot{\eta} \frac{\partial p}{\partial \eta}}{\dot{\eta}} \right)_{l=0} = 0 \); discretization follows:
  \[
  \left( \frac{\partial X}{\partial \eta} \right)_{l=1} = \frac{X_{l=2} - X_{l=1}}{\eta_{l=1} - \eta_{l=0}}
  \]
  \[(74)\]

- For layer \( l = L \), discretization assumes that \( \left( \frac{\dot{\eta} \frac{\partial p}{\partial \eta}}{\dot{\eta}} \right)_{l=L} = 0 \); discretization follows:
  \[
  \left( \frac{\partial X}{\partial \eta} \right)_{l=L} = \frac{X_{l=L} - X_{l=L-1}}{\eta_{l=L} - \eta_{l=L-1}}
  \]
  \[(75)\]
The following weights are computed:

- For an interpolation point included between layers 2 and $L - 1$ ($l \geq 1$ and $l \leq L - 3$):
  
  \[
  V_{DERW11} = \frac{1}{2} \frac{\eta_{l+2} - \eta_{l+1}}{\eta_{l+1} - \eta_l}
  \]
  
  \[
  V_{DERW21} = \frac{1}{2} \frac{\eta_{l+2} - \eta_{l+1}}{\eta_{l+1} - \eta_l}
  \]
  
  \[
  V_{DERW12} = \frac{1}{2} \frac{\eta_{l+2} - \eta_{l+1}}{\eta_{l+2} - \eta_{l+1}}
  \]
  
  \[
  V_{DERW22} = \frac{1}{2} \frac{\eta_{l+2} - \eta_{l+1}}{\eta_{l+2} - \eta_{l+1}}
  \]

- For an interpolation point included between layers 1 and 2:
  
  \[
  V_{DERW11} = 0
  \]
  
  \[
  V_{DERW21} = \frac{\eta_{l=2} - \eta_{l=1}}{\eta_{l=1} - \eta_{l=0}}
  \]
  
  \[
  V_{DERW12} = \frac{1}{2} \frac{\eta_{l=2} - \eta_{l=1}}{\eta_{l=2} - \eta_{l=1}}
  \]
  
  \[
  V_{DERW22} = \frac{1}{2} \frac{\eta_{l=2} - \eta_{l=1}}{\eta_{l=2} - \eta_{l=1}}
  \]

such a case is extended to the case where the interpolation point is between the top and the first layer; in this case the interpolation becomes an extrapolation.

- For an interpolation point included between layers $L - 1$ and $L$:
  
  \[
  V_{DERW11} = \frac{1}{2} \frac{\eta_{l=L} - \eta_{l=L-1}}{\eta_{l=L-1} - \eta_{l=L-2}}
  \]
  
  \[
  V_{DERW21} = \frac{1}{2} \frac{\eta_{l=L} - \eta_{l=L-1}}{\eta_{l=L-1} - \eta_{l=L-2}}
  \]
  
  \[
  V_{DERW12} = \frac{\eta_{l=L} - \eta_{l=L-1}}{\eta_{l=L} - \eta_{l=L-1}}
  \]
\[ VDERW22 = 0 \]

such a case is extended to the case where the interpolation point is between the last layer and the ground; in this case the interpolation becomes an extrapolation.

Functions \( f_{H1}(ZDVER) \) to \( f_{H4}(ZDVER) \) (involved in any Hermite cubic interpolation) are:

- \( f_{H1}(\alpha) = (1 - \alpha)^2(1 + 2\alpha) \)
- \( f_{H2}(\alpha) = \alpha^2(3 - 2\alpha) \)
- \( f_{H3}(\alpha) = \alpha(1 - \alpha)^2 \)
- \( f_{H4}(\alpha) = -\alpha^2(1 - \alpha) \)

### 8.1.6 Interpolation grid and weights for tri-linear interpolations

A 64-point grid is defined as it is shown in figure 4. The interpolation point \( O \) (medium or origin point) is between \( B_{1,l+1}, C_{1,l+1}, B_{2,l+1}, C_{2,l+1}, B_{1,l+2}, C_{1,l+2}, B_{2,l+2} \) and \( C_{2,l+2} \). For the two levels \( l + 1 \) and \( l + 2 \) see section 8.1.1 corresponding to bi-linear horizontal interpolations for weights computations. For weights needed for vertical interpolations \( (ZDVER) \) see section 8.1.2 corresponding to linear vertical interpolations.

### 8.1.7 Interpolation grid and weights for 32-point interpolations

A 64-point grid is defined as it is shown in figure 4. The interpolation point \( O \) (medium or origin point) is between \( B_{1,l+1}, C_{1,l+1}, B_{2,l+1}, C_{2,l+1}, B_{1,l+2}, C_{1,l+2}, B_{2,l+2} \) and \( C_{2,l+2} \). For the two levels \( l \) and \( l + 3 \) see section 8.1.1 corresponding to bi-linear horizontal interpolations for weights computations. For the two levels \( l + 1 \) and \( l + 2 \) see section 8.1.3 corresponding to 12-point horizontal interpolations for weights computations. For weights needed for vertical interpolations \( (ZDVER) \) see section 8.1.2 corresponding to linear vertical interpolations.

### 8.1.8 Other grids and weights

The following interpolation systems are available but not described here:

- vertical cubic splines
8.2 Interpolations

8.2.1 Bilinear interpolation

It is done in subroutine LAIDLI. See figure 1 and section 8.1.1 for definition of ZDLO1, ZDLO2, ZDLAT and points B1, C1, B2 and C2.

For a quantity X, are computed successively:

- a linear interpolation on the longitude number 1:
  \[
  X_1 = X_{B_1} + ZDLO1(X_{C_1} - X_{B_1})
  \]
- a linear interpolation on the longitude number 2:
  \[
  X_2 = X_{B_2} + ZDLO2(X_{C_2} - X_{B_2})
  \]
- a meridional linear interpolation:
  \[
  X_{interpo} = X_1 + ZDLAT(X_2 - X_1)
  \]

8.2.2 Tri-linear interpolation

It is done in subroutine LAITLI. For layers \(l + 1\) and \(l + 2\) (see figure 4) bilinear horizontal interpolations give two interpolated values \(X_{l+1}\) and \(X_{l+2}\) (see section 8.2.1). Then the final interpolated value is given by the following expression:

\[
X_{interpo} = X_{l+1} + ZDVER(X_{l+2} - X_{l+1})
\]

8.2.3 Horizontal 12-point interpolation

It is done in subroutine LAIDDI or its shape-preserving version LAIDQM. See figure 3 and section 8.1.3 for definition of ZDLO0, ZDLO1, ZDLO2, ZDLO3, ZCLA2, ZCLA3 and ZCLA4 and points B0, C0, A1, B1, C1, D1, A2, B2, C2, D2, B3 and C3. Let us define:

- \(f_2(\alpha) = (\alpha + 1)(\alpha - 2)(\alpha - 1)/2\)
- \(f_3(\alpha) = -(\alpha + 1)(\alpha - 2)\alpha/2\)
• \( f_4(\alpha) = \alpha(\alpha - 1)(\alpha + 1)/6 \)

For a quantity \( X \), are computed successively:

• a linear interpolation on the longitude number 0:
  \[ X_0 = X_{B_0} + ZDLO0(X_{C_0} - X_{B_0}) \]

• a cubic 4-point interpolation on the longitude number 1:
  \[ X_1 = X_{A_1} + f_2(ZDLO1)(X_{B_1} - X_{A_1}) + f_3(ZDLO1)(X_{C_1} - X_{A_1}) + f_4(ZDLO1)(X_{D_1} - X_{A_1}) \]

• a cubic 4-point interpolation on the longitude number 2:
  \[ X_2 = X_{A_2} + f_2(ZDLO2)(X_{B_2} - X_{A_2}) + f_3(ZDLO2)(X_{C_2} - X_{A_2}) + f_4(ZDLO2)(X_{D_2} - X_{A_2}) \]

• a linear interpolation on the longitude number 3:
  \[ X_3 = X_{B_3} + ZDLO3(X_{C_3} - X_{B_3}) \]

• a meridional cubic 4-point interpolation:
  \[ X_{\text{interpo}} = X_0 + ZCLA2(X_1 - X_0) + ZCLA3(X_2 - X_0) + ZCLA4(X_3 - X_0) \]

There is a shape-preserving version LAIDQM of routine LAIDDI: after cubic 4-point interpolations on longitudes number 1 and 2, \( X_1 \) is bounded between \( X_{B_1} \) and \( X_{C_1} \) and \( X_2 \) is bounded between \( X_{B_2} \) and \( X_{C_2} \); after meridian cubic 4-point interpolation \( X_{\text{interpo}} \) is bounded between \( X_1 \) and \( X_2 \). Use of switches LQMW (momentum equation), LQMT (temperature equation), LQM (humidity equation), LQMV (passive scalar equations), LQMP (continuity equation), allow to use shape-preserving interpolation routine LAIDQM instead of LAIDDI.

8.2.4 Cubic 4-point vertical interpolation

See figure 2 and section 8.1.4 for definition of \( ZCVE2 \), \( ZCVE3 \) and \( ZCVE4 \).

The cubic 4-point vertical interpolation gives the final interpolated value:

\[
X_{\text{interpo}} = X_l + ZCVE2(X_{l+1} - X_l) + ZCVE3(X_{l+2} - X_l) + ZCVE4(X_{l+3} - X_l)
\]
8.2.5 Cubic Hermite vertical interpolation

See figure 2 and section 8.1.5 for definition of $VDERW_{11}$, $VDERW_{21}$, $VDERW_{12}$ and $VDERW_{22}$. See section 8.1.2 for definition of $ZDVER$.

See section 8.1.5 for definition of functions $f_{H1}$ to $f_{H4}$. The cubic Hermite vertical interpolation gives the final interpolated value:

$$X_{interpo} = f_{H1}(ZDVER)X_{l+1} + f_{H2}(ZDVER)X_{l+2} +$$

$$f_{H3}(ZDVER)(VDERW_{11}(X_{l+1} - X_l) + VDERW_{21}(X_{l+2} - X_{l+1}))+$$

$$f_{H4}(ZDVER)(VDERW_{12}(X_{l+2} - X_{l+1}) + VDERW_{22}(X_{l+3} - X_{l+2}))$$

8.2.6 32-point 3D interpolation with vertical cubic 4-point interpolation

It is done in subroutine LAITRI or its shape-preserving version LAITQM. For layers $l$ and $l+3$ (see figure 4) bilinear horizontal interpolations give two interpolated values $X_l$ and $X_{l+3}$ (see section 8.2.1). For layers $l+1$ and $l+2$ (see figure 4) 12-point horizontal interpolations give two interpolated values $X_{l+1}$ and $X_{l+2}$ (see section 8.2.3). The final interpolated value $X_{interpo}$ is a cubic 4 points vertical interpolation of $X_l$, $X_{l+1}$, $X_{l+2}$ and $X_{l+3}$ (see section 8.2.4).

There is a shape-preserving version LAITQM of routine LAITRI. In LAITQM 12 points horizontal interpolations for layers $l+1$ and $l+2$ are shape-preserving interpolations (see description of routine LAIQDM in section 8.2.3) and vertical cubic 4-point interpolation is shape-preserving: $X_{interpo}$ is bounded by $X_{l+1}$ and $X_{l+2}$. Use of switches LQMW (momentum equation), LQMT (temperature equation), LQM (humidity equation), LQMV (passive scalar equations), LQMP (continuity equation), allows to use shape-preserving interpolation routine LAITQM instead of LAITRI.

8.2.7 32-point 3D interpolation with vertical cubic Hermite interpolation

It is done in subroutine LAIHVT or its shape-preserving version LAIHVTQM. For layers $l$ and $l+3$ (see figure 4) bilinear horizontal interpolations give two interpolated values $X_l$ and $X_{l+3}$ (see section 8.2.1). For layers $l+1$ and $l+2$ (see figure 4) 12-point horizontal interpolations give two interpolated values $X_{l+1}$ and $X_{l+2}$ (see section 8.2.3). The final interpolated value $X_{interpo}$ is a cubic Hermite vertical interpolation of $X_l$, $X_{l+1}$, $X_{l+2}$ and $X_{l+3}$ (see section 8.2.5).
There is a shape-preserving version LAIHVTQM of routine LAIHVT. In LAIHVTQM 12-point horizontal interpolations for layers \( l + 1 \) and \( l + 2 \) are shape-preserving interpolations (see description of routine LAIDQM in section 8.2.3) and vertical cubic Hermite interpolation is shape-preserving: \( X_{\text{interpo}} \) is bounded by \( X_{l+1} \) and \( X_{l+2} \). Use of switch LQMV (passive scalar equations), allows to use shape-preserving interpolation routine LAIHVTQM instead of LAIHVT.

8.2.8 Other algorithms

The following interpolation methods are also available but are not described here:

- diffusive bilinear interpolation (subroutine LAIDLIHD)
- diffusive tri-linear interpolation (subroutine LAITLIHD)
- 48-point 3D interpolation with vertical cubic spline interpolation (subroutine LAITVSPCQM)
- semi-lagrangian horizontal diffusion (subroutine LAITSLD)
- horizontal 16-point linear least-square fit interpolation
- 32-point with linear least-square fit horizontal and linear vertical interpolation (subroutine LAISMOO)
- horizontal 12-point spline interpolation (subroutine LAIDSP)
- 32-point 3D cubic spline interpolation
- semi-lagrangian horizontal diffusion with cubic splines (subroutine LAITSLDSP)
Figure 1: Interpolation horizontal grid for bilinear interpolations.
Figure 2: Interpolation vertical grid for linear and cubic vertical interpolations.
$x_0(B_0) = x_1(B_1) = x_2(B_2) = x_3(B_3) = 0; \ x_0(C_0) = x_1(C_1) = x_2(C_2) = x_3(C_3) = 1.$

Figure 3: Interpolation horizontal grid for 12-point interpolations.
- points used in 32-point interpolations.
- points not used in 32-point interpolations.

Figure 4: Interpolation grid for trilinear and 32-point interpolations.
9 Computation of $\dot{\eta}$ on layers

This quantity is needed to find the height of the medium and origin points. $\dot{\eta}$ (resp. $\dot{\eta}$) is the $\eta$-coordinate vertical velocity $\dot{\eta}$ at the inter-layer $l$ (resp. layer $l$), $\Delta p_l$ is the pressure depth of layer $l$.

$\dot{\eta}$ can be written:

$$\dot{\eta} = \left( \eta \frac{\partial p}{\partial \eta} \right) \frac{\partial \eta}{\partial p}$$  \hspace{1cm} (76)

where $\left( \eta \frac{\partial p}{\partial \eta} \right)$ is computed at inter-layers (and stored in the array PEVEL) using a vertical integration of continuity equation, and $\left( \frac{\partial \eta}{\partial p} \right)$ is computed at layers.

Discretization of Equation (76) is:

$$\dot{\eta}_l = \frac{1}{2} \left( \left( \eta \frac{\partial p}{\partial \eta} \right)_l + \left( \eta \frac{\partial p}{\partial \eta} \right)_{l-1} \right) \frac{\eta_l - \eta_{l-1}}{\Delta p_l}$$  \hspace{1cm} (77)

10 Lateral boundary conditions

10.1 Extra longitudes

Let us denote by $LX$ the number of longitudes (in the array NLOENG for each latitude in the code). For a quantity $X$, let us define:

- $X$(longitude number 0)$=X$(longitude number $LX$).
- $X$(longitude number $LX+1$)$=X$(longitude number 1).
- $X$(longitude number $LX+2$)$=X$(longitude number 2).

These extra computations are necessary for all interpolated fields. For distributed memory computations are done when making the halo (routine SLCOMM or SLCOMM1+SLCOMM2A which exchange data with other processors).

10.2 Extra latitudes

Let us denote by $lx$ the number of latitudes (NDGLG in the code): latitudes number -1,0,$lx$ + 1,$lx$ + 2 are respectively the symmetric of latitudes number 2,1,$lx$,$lx$ − 1. These extra computations are necessary for all interpolated fields. For distributed memory computations are done in SLEXTPOL or SLEXTPOL1A+SLEXTPOL2.
10.3 Vertical boundary conditions

Vertically interpolations for layer variables at the medium point: The medium point has a vertical coordinate always included between $\eta_{t=0}$ and $\eta_{t=L}$ in case of vertical interpolating scheme. Therefore no extrapolated values are needed.

Vertically cubic 4-point interpolations for layer variables at the origin point: When the origin point is above the layer number 2 (resp. below the layer number $L - 1$), the vertical cubic 4-point interpolations using data of the layers number 1, 2, 3 (resp. $L - 2$, $L - 1$, $L$) and the extra-layer number 0 (resp. $L + 1$) are degenerated into linear interpolations between the layers number 1 and 2 (resp. $L - 1$ and $L$). The extrapolated values at the extra-layer number 0 (resp. $L + 1$) are always multiplied by a weight equal to 0 and are set to 0 in subroutine LAVABO. This algorithm extends itself to the case where the origin point is between the top (resp. surface) and the layer number 1 (resp. $L$), but in this case the interpolation using data of the layers number 1 and 2 (resp. $L - 1$ and $L$) becomes an extrapolation.

Vertically cubic 4-point interpolations for inter-layer variables at the origin point: When the origin point is above the inter-layer number 1 (resp. $L - 1$), the vertical cubic 4-point interpolations using data of the inter-layers number -1, 0, 1, 2 (resp. $L - 2$, $L - 1$, $L$ and $L + 1$) are degenerated into linear interpolations between the inter-layers numbers 0 and 1 (resp. $L - 1$ and $L$).

Vertically cubic Hermite interpolations for layer variables at the origin point: When the origin point is above the layer number 2 (resp. $L - 1$), interpolation is still a vertical cubic Hermite one, computation of vertical derivatives is modified for layer number 1 (resp. $L$). This algorithm extends to the case where the origin point is between the top (resp. ground) and the layer number 1 (resp. $L$), but in this case the interpolation using data of the layers number 1 and 2 (resp. $L - 1$ and $L$) becomes an extrapolation.

Vertically cubic spline interpolations for layer variables at the origin point: Some top and bottom values are computed and the vertical interpolation always uses 4 points.
References


