Météo-France
Centre National de Recherches Météorologiques

ARPEGE-Climate Version 6.2

Algorithmic Documentation

JULY 2017

Contents

6 Horizontal diffusion 143

1

Basic hypotheses and related constants

Arpege-IFS is a complex code designed not only for weather forecast or climate simulation, but also for data assimilation, forecast pre- and post-processing. It has been extended, diversified and complexified since 1986 jointly by Météo-France and ECMWF. The present documentation restricts to the description of the French climate version of Arpege-IFS. Some features are not compatible with the version used by ECMWF. In this case, we will use the term Arpege. Some features are specific to the French climate version, and we will use the term Arpege-climat. The core of the model is cycle 37T1 of Arpege-IFS.

Arpege-climat is now an atmosphere-only model. So the calculations concerning the surface boundary layer, vegetation, snow cover and soil are done in Surfex (SURFace EXternalisée), which is another model. Surfex simulates the exchanges of momentum, heat, water, carbon dioxide concentration or chemical species between the surface and the atmosphere. It uses the concept of 'tile' to describe the surface (nature, town, sea, water) and can perform different parametrizations. Each surface grid box is made of the four adjacent tiles. The coverage of each of these surfaces is known through the global ECOCLIMAP database. Surfex receives atmospheric forcing terms, runs the surface schemes, computes the average surface fluxes over the nature, town, sea and water weighted by their respective fraction and sends them back to the atmosphere in addition to radiatives terms. All this information is then used as lower boundary conditions for the atmospheric radiation and turbulent schemes. The complete documentation on Surfex (Surfex main scientific documentation v2 and Surfex V8_0 user's guide) is available on the web site `http://www.umr-cnrm.fr/surfex//spip.php?rubrique141`.

The model relies upon a geometrical assumption: the thin layer approximation, and a certain number of phenomenological assumptions such as the law of perfect gases or the hydrostatic approximation. With these assumptions a set of basic constants is presented here.

1 Astronomical constants

This section follows the last recommendations of the *International Astronomical Union*. It should be noted that the formulas are not valid for dates too far away from the 1st January 2000 (more than one century).

1.1 Calendar

The calendar used is the *Gregorian* calendar. The dates are given in the form:

$$AAAAMMJJ,\ sssss$$

with:

$$\begin{cases} AAAA & \text{year,} \\ MM & \text{month,} \\ JJ & \text{day,} \\ sssss & \text{seconds in the day.} \end{cases}$$

1.2 Time

The length of the day is:

$$d = 86400\ s$$

Time t is expressed in seconds, the date of reference being 20000101.43200 (2451545.0 in *Julian* calendar). It is negative before this date and positive afterward. t is deduced from the calendar date by:

$$t = (JD - 2451545)86400 + sssss$$

with JD, date in the *Julian* calendar (E indicating the integer part):

$$JD = 1720994.5 + K + \mathrm{E}(365.25a) + \mathrm{E}(30.601(m+1)) + JJ$$

and:

$$a = \begin{cases} AAAA, & \text{if } MM > 2 \\ AAAA - 1, & \text{if } MM \leq 2 \end{cases}$$

$$m = \begin{cases} MM, & \text{if } MM > 2 \\ MM + 12, & \text{if } MM \leq 2 \end{cases}$$

$$K = 2 - \mathrm{E}(a/100) + \mathrm{E}(\mathrm{E}(a/100)/4)$$

1.3 Astronomical elements

In the following we set:

$$\theta = t/(dy_j)$$

with:

$$y_j = 365.25 \text{ days}$$

The constants between square brackets are not used in the model; however, we provide them because they form a consistent set with those needed by the model.

	half great axis	e_a	=	$149597870000\, m \pm 5\, 10^{-5}$	
[	excentricity			0.016704 ± 10^{-4}	]
[	inclination			$0 \pm 2\, 10^{-4}$	]
	mean longitude	e_l	=	$1.7535 + 6.283076\, \theta$	
[	longitude of perihelion			$1.79661 + 0.0000563\, \theta$	]
[	longitude of ascending node			6.1937 if $t < 0$, 3.0521 if $t > 0$	]
	mean anomaly	e_M	=	$6.240075 + 6.283020\, \theta$	
	Sun-Earth distance	R_s	=	$e_a(1.0001 - 0.0163 \sin(e_l)$ $+0.0037 \cos(e_l))$	

1.4 Sun trajectory relative to Earth

	mean longitude	l_s	=	$4.8951 + 6.283076\, \theta$	
	true longitude	L_s	=	$4.8952 + 6.283320\, \theta$ $-0.0075 \sin(e_l) - 0.0326 \cos(e_l)$ $-0.0003 \sin(2e_l) + 0.0002 \cos(2e_l)$	
[	true latitude			0	]

	obliquity	ϵ_s	=	0.409093	
	declination	δ_s	=	$\mathrm{Asin}(\sin(\epsilon_s)\sin L_s)$	
[	right ascension			$0 \leq \alpha_s \leq 2\pi$	]
[				$\cos(\alpha_s)\cos(\delta_s) = \cos(L_s)$	]
[				$\sin(\alpha_s)\cos(\delta_s) = \sin(L_s)\cos(\epsilon_s)$	]
	equation of time	Et	=	$591.8\sin(2l_s) - 459.4\sin(e_M)$	
	(true solar time −			$+39.5\sin(e_M)\cos(2l_s)$	
	mean solar time)			$-12.7\sin(4l_s) - 4.8\sin(2e_M)$	

Over the period 1980–2020, the relative accuracy on R_s is of $5\,10^{-4}$, the accuracy on the various angles is of $5\,10^{-4}\ rd$, and that on equation of time is of 10 s. These constants implicitly define the length of the sidereal year:

$$y_s = \frac{2\pi\, d\, y_j}{6.283076}$$

and therefore the length of sidereal day:

$$d_s = \frac{d}{1 + d/y_s}$$

and earth rotation:

$$\Omega = \frac{2\pi}{d_s}$$

2 Geometry, geoid

2.1 Geometry

We mentioned in the introduction that the thin layer assumption is the base of Arpege-IFS equations. To make sense, it requires the choice of one surface on which the equations are written.

As we write the momentum equation in vorticity-divergence, the Laplacian operator must have his kernel reduced to constant functions; we suppose moreover than the surface is of revolution around the axis of rotation of the planet.

2.2 Coordinate system

On the horizontal, we use longitude λ varying from 0 to 2π to parametrize the circles of revolution. The East is directed towards increasing longitudes. In the orthogonal direction, we use μ variable from -1 at South pole to $+1$ at North pole (by definition) to parametrize the surface generator.

On the vertical, we use a coordinate η varying from 0 at the top of the fluid to 1 at the bottom. This vertical coordinate has no geometrical significance with the ordinary metrics. The 3d metrics is obtained as the product of horizontal metrics by vertical one.

2.3 Geoid

The preceding assumptions imply that the vertical coordinate does not have any geometrical significance and that gravity is not explicitly used. In place we need two infinitely close equipotential surfaces between which the equations are written. We make the additional assumption that for the description of the Earth, equipotential surfaces are spheres of radius a (average value of the reference ellipsoid):

$$a = 6371229\ m$$

To transform an elevation value into geopotential in $J\,kg^{-1}$, it should be multiplied by the conventional value:

$$g = 9.80665\ m\,s^{-2}$$

If we used an ellipsoid instead of a sphere, gravity would vary with the latitude according to a formula close to that of Clairault, but it would not appear explicitly in the equations and the preceding remarks would remain valid.

Because of the sphericity assumption, the notions of "North, East, longitude ..." used above should not be taken in their geographical meaning since, as we will see further, the pole of the coordinate system is not necessarily in the Arctic.

3 Fundamental constants

light speed	c	$=$	$299792458\ m\,s^{-1}$
Planck's constant	h	$=$	$6.6260755\,10^{-34}\ J\,s$
Boltzmann's constant	k	$=$	$1.380658\,10^{-23}\ J\,K^{-1}$
Avogadro's number	N	$=$	$6.0221367\,10^{23}\ mol^{-1}$

4 Radiation

Stefan-Boltzmann's constant	σ	$=$	$\dfrac{2\pi^5 k^4}{15c^2h^3}$
solar constant	I_0	$=$	$1370\, W\, m^{-2}$

5 Thermodynamics, gas phase

The fluid is a mixture of dry air, of water in gas, liquid and solid phases.

gas constant	$\mathcal{R}$	$=$	Nk
dry air molar mass	M_a	$=$	$28.9644\, 10^{-3}\ kg\, mol^{-1}$
water vapor molar mass	M_v	$=$	$18.0153\, 10^{-3}\ kg\, mol^{-1}$

$$R_a = \frac{\mathcal{R}}{M_a}\ J\, kg^{-1}K^{-1}$$

$$R_v = \frac{\mathcal{R}}{M_v}\ J\, kg^{-1}K^{-1}$$

It is supposed here that dry air and water vapor are perfect gases. The maximum error is 0.1 %.

$$c_{p_a} = \frac{7}{2}\, R_a$$

$$c_{v_a} = \frac{5}{2}\, R_a$$

These quantities are not constant in the atmosphere. But this assumption is coherent with the approximation of perfect gases, and the error introduced is less than 1 %.

$$c_{p_v} = 4\, R_v$$

$$c_{v_v} = 3\, R_v$$

In the case of water vapor, supposing them constant leads to an error less than 5 %.

If one activates the optional “prognostic physical parametrizations”, atmosphere contains also four species: cloud water, cloud ice, rain water and rain ice.

6 Thermodynamics, liquid phase

water molar mass	M_l	$=$	M_v
massic volume	v_l	$=$	0

$$c_{p_l} = c_{v_l} = c_l = 4.218\,10^3\ J\,kg^{-1}K^{-1}(\text{value at triple point } T_t)$$

The identity between c_{p_l} and c_{v_l} is very well satisfied and is coherent with the constancy of the massic volume. The fact that c_l is constant is satisfied with less than 1 % error in the temperature range $[0°C, 30°C]$, but the error grows for the negative temperatures and reaches 12.5 % at $-40°C$.

7 Thermodynamics, solid phase

$$M_g = M_l$$

$$v_g = v_l$$

$$c_{p_g} = c_{v_g} = c_g = 2.106\,10^3\ J\,kg^{-1}K^{-1}(\text{value at } T_t)$$

Actually, c_g decreasing linearly with the temperature, the error introduced is 13 % at $-40°C$.

8 Thermodynamics, phase transition

triple point	T_t	$=$	$273.16\ K$

8.1 Vaporization

$$L_v(T) = L_v(T_t) + (c_{p_v} - c_l)(T - T_t)$$

$$L_v(T_t) = 2.5008\,10^6\ J\,kg^{-1}$$

It is supposed that L_v is independent of the pressure, which is accurate at 0.5 %, and is coherent with $v_l = 0$.

8.2 Sublimation

$$L_s(T) = L_s(T_t) + (c_{p_v} - c_g)(T - T_t)$$

$$L_s(T_t) = 2.8345\,10^6\ J\,kg^{-1}$$

$c_{p_v} - c_g$ is an order of magnitude weaker than $c_{p_v} - c_l$; however, to neglect the variation of L_s with temperature, it would be necessary to write: $c_g = c_{p_v}$.

8.3 Melting

$$L_f = L_s - L_v$$

9 Consequences on saturation

With $v_l = 0$, Clapeyron's equation becomes:

$$\frac{d\ln(e_s)}{dT} = \frac{L_v}{R_v T^2}$$

Using the expression of L_v, and integrating from:

$$(T_t,\ e_s(T_t) = 611.14\ Pa\,)$$

yields:

$$\ln(e_s) = \alpha_l - \frac{\beta_l}{T} - \gamma_l \ln T$$

with:

$$\begin{aligned} \alpha_l &= \ln(e_s(T_t)) + \frac{\beta_l}{T_t} + \gamma_l \ln T_t \\ \beta_l &= L_v(T_t)/R_v + \gamma_l T_t \\ \gamma_l &= \frac{c_l - c_{pv}}{R_v} \end{aligned}$$

In the presence of ice, the formula remains valid if the $_l$ are replaced by $_g$.

10 Thermodynamic functions

The thermodynamic functions are gathered in the block of declarations FCTTRM. They are divided into two groups: the first corresponds to the absolute functions and to the second to the approximate functions whose approximations are consistent with the above statements. This module is inserted in all the subroutines which require thermodynamic calculations. Thus coherence between the various parts of the code is ensured.

Notations:

- T_t temperature of water triple point
- $\gamma_l = (c_l - c_{p_v})/R_v$
- $\gamma_i = (c_i - c_{p_v})/R_v$
- $\beta_l = L_v(T_t)/R_v + \gamma_l T_t$
- $\beta_i = L_s(T_t)/R_v + \gamma_i T_t$
- $\alpha_l = \ln e_s(T_t) + \beta_l/T_t + \gamma_l \ln T_t$
- $\alpha_i = \ln e_s(T_t) + \beta_i/T_t + \gamma_i \ln T_t$
- δ index for liquid/solid calculation: $\delta = \begin{cases} 1 & \text{if liquid whatever T} \\ 0 & \text{if solid whatever T} \end{cases}$
- δ_T index of temperature positivity: $\delta_T = \begin{cases} 1 & \text{if} \quad \mathrm{T} \geq \mathrm{T}_t \\ 0 & \text{otherwise} \end{cases}$

10.1 Absolute Functions

Latent heat of vaporization:

$$L_v(T) = \mathrm{RLV}(T) = L_v(T_t) + (c_{p_v} - c_l)(T - T_t)$$

Latent heat of sublimation:

$$L_s(T) = \mathrm{RLS}(T) = L_s(T_t) + (c_{p_v} - c_i)(T - T_t)$$

Latent heat of fusion:

$$L_f(T) = \mathrm{RLF}(T) = L_s(T) - L_v(T)$$

Saturation pressure above liquid water:

$$e_{sl}(T) = \mathrm{ESW}(T) = \exp\left[\alpha_l - \frac{\beta_l}{T} - \gamma_l \ln T\right]$$

Saturation pressure above solid water:

$$e_{si}(T) = \mathrm{ESS}(T) = \exp\left[\alpha_i - \frac{\beta_i}{T} - \gamma_i \ln T\right]$$

Saturation pressure above liquid/solid water:

$$\begin{aligned} e_s(T) &= \mathrm{ES}(T) \\ &= \exp\left[\alpha_l + (\alpha_i - \alpha_l)\delta_T - \frac{\beta_l + (\beta_i - \beta_l)\delta_T}{T} - (\gamma_l + (\gamma_i - \gamma_l)\delta_T)\ln T\right] \end{aligned}$$

10.2 Functions in the model parametrizations

Saturation pressure:

$$\begin{aligned} e_s(T,\delta) &= \mathrm{FOEW}(T,\delta) \\ &= \exp\left[\alpha_l + (\alpha_i - \alpha_l)\delta - \frac{\beta_l + (\beta_i - \beta_l)\delta}{T} - (\gamma_l + (\gamma_i - \gamma_l)\delta)\ln T\right] \end{aligned}$$

Derivative of the logarithm of the saturation pressure:

$$\frac{\partial \ln e_s}{\partial T}(T,\delta) = \mathrm{FODLEW}(T,\delta) = \frac{\beta_l + (\beta_i - \beta_l)\delta - (\gamma_l + (\gamma_i - \gamma_l)\delta)T}{T^2}$$

Saturation specific moisture:

$$q_s = \mathrm{FOQS}(\frac{e_s}{p}) = \frac{e_s/p}{1 + (R_v/R_a - 1)\max(0,\, 1 - e_s/p)}$$

This formulation makes it possible to have:

$$q_s = \begin{cases} \dfrac{e_s/R_v}{(p - e_s)/R_a + e_s/R_v} & \text{if } e_s(T) \leq p \\[2ex] \dfrac{e_s}{p} & \text{if } e_s(T) \geq p \end{cases}$$

Derivative saturation of specific moisture:

$$\frac{\partial q_s}{\partial T} = \mathrm{FODQS}(q_s,\, \frac{e_s}{p},\, \frac{\partial \ln e_s}{\partial T}) = \frac{q_s - q_s^2}{1 - e_s/p}\frac{\partial \ln e_s}{\partial T}$$

Latent heat:

$$\begin{aligned} L(T,\delta) &= \mathrm{FOLH}(T,\delta) \\ &= R_v\left[\beta_l + (\beta_i - \beta_l)\delta - (\gamma_l + (\gamma_i - \gamma_l)\delta)T\right] \\ &= L_v(T_t) + \left[L_s(T_t) - L_v(T_t)\right]\delta + \left[c_{p_v} - c_l + (c_l - c_i)\delta\right](T - T_t) \end{aligned}$$

2

Dynamics equations

1 Introduction

Each equation of the model can generally write as:

$$\frac{dX}{dt} = \mathcal{A} + \mathcal{F}$$

where X is a prognostic variable, the evolution of which one wants to know. $\mathcal{A}$ represents all the effects which can be explicitly represented for the current resolution (often named "adiabatic effects"). They are:

- Coriolis force (momentum equation)
- pressure-gradient force term (momentum equation)
- conversion term (temperature equation)
- divergence term (continuity equation)

$\mathcal{F}$ represents all the sub-scale effects (often named "diabatic effects") which are calculated by physical parametrization routines. They are:

- radiation
- clouds and turbulence
- large-scale precipitations
- vertical diffusion

- convection
- orographic gravity wave drag
- soil, snow and vegetation

The time derivative of X means the total temporal derivative, including advection, also known as Lagrangian derivative.

2 Primitive equations in Eulerian form

Making the hydrostatic assumption, we use for vertical coordinate a hybrid coordinate $\eta(p, p_s)$ derived from the pressure coordinate p and terrain following. It must satisfy:

$$\left\{ \begin{array}{rcl} \eta(0, p_s) & = & 0 \\ \eta(p_s, p_s) & = & 1 \\ \dfrac{\partial \eta}{\partial p}(p, p_s) & > & 0 \end{array} \right.$$

This vertical coordinate η is defined by two functions $A(\eta)$ and $B(\eta)$, in such a way that the pressure at a given point is:

$$p = A(\eta) + B(\eta) p_s$$

where p_s is surface pressure. We have:

$$\begin{array}{ll} A(0) = 0 & A(1) = 0 \\ B(0) = 0 & B(1) = 1 \end{array}$$

ensuring for η-surfaces to follow orography at the bottom and to be pressure surfaces at the top. The model does not need to explicitly know the functional form of A and B, only their values at the interface of the layers are necessary.

The hydrostatic assumption leads to the equation:

$$\frac{\partial \Phi}{\partial \eta} = -\frac{RT}{p} \frac{\partial p}{\partial \eta}$$

which is used as a diagnostic equation to calculate geopotential Φ on level p by an integral starting at the lower boundary condition $\Phi(p_s) = \Phi_s$.

The evolution of the parameters which define the state of the atmosphere, horizontal wind $\vec{v}$, temperature T and mass moisture ratio q_v is controlled by the following equations, where the total temporal derivative is written as:

$$\frac{d\,X}{d\,t} = \frac{\partial X}{\partial t} + \vec{v}\,\nabla X + \dot{\eta}\frac{\partial X}{\partial \eta} \tag{1}$$

Momentum equation

$$\frac{d\,\overrightarrow{v}}{d\,t} + \underbrace{2\Omega \times \vec{v}}_{\text{Coriolis}} + \underbrace{RT\nabla \ln p + \nabla\Phi}_{\text{pressure force}} = -g\frac{\partial \eta}{\partial p}\frac{\partial \overrightarrow{F_{\vec{v}}}}{\partial \eta} + \overrightarrow{S_{\vec{v}}} + \overrightarrow{K_{\vec{v}}} \tag{2}$$

To conserve momentum in the vertical discretization, the acceleration term due to the pressure force is transformed into:

$$\frac{\partial \eta}{\partial p}\left(\Phi\nabla\frac{\partial p}{\partial \eta} - \frac{\partial \Phi \nabla p}{\partial \eta}\right) + \nabla\Phi$$

Thermodynamics equation

$$\frac{d\,T}{d\,t} - \underbrace{\kappa\,T\,\frac{\omega}{p}}_{\text{conversion}} = -\frac{g}{c_p}\frac{\partial \eta}{\partial p}\frac{\partial F_h}{\partial \eta} + S_h + K_h \tag{3}$$

Moisture equation

$$\frac{d\,q_v}{d\,t} = -g\frac{\partial \eta}{\partial p}\frac{\partial F_{q_v}}{\partial \eta} + S_{q_v} + K_{q_v} \tag{4}$$

In the above equations one takes:

$$\begin{aligned} R &= q_a R_a + q_v R_v \\ c_p &= q_a c_{p_a} + q_v c_{p_v} \\ \kappa &= \frac{R}{c_p} \end{aligned}$$

The terms in the right-hand members of Equations (2), (3) and (4) respectively represent vertical fluxes (noted F), sources (noted S) and horizontal diffusion (noted K) of momentum, enthalpy, and specific moisture.

The continuity equation is written as:

$$\frac{\partial}{\partial \eta}\left(\frac{\partial p}{\partial t}\right) + \nabla \cdot \left(\vec{v}\frac{\partial p}{\partial \eta}\right) + \frac{\partial}{\partial \eta}\left(\dot{\eta}\frac{\partial p}{\partial \eta}\right) = -g\frac{\partial F_p}{\partial \eta} \tag{5}$$

F_p is the mass flux, no source term being considered.

By integrating, one obtains the evolution equation of surface pressure:

$$\frac{\partial p_s}{\partial t} = -\int_0^1 \nabla \cdot \left(\vec{v} \frac{\partial p}{\partial \eta} \right) d\eta - gF_p(1)$$

vertical velocity in pressure coordinate:

$$\omega = -\int_0^\eta \nabla \cdot \left(\vec{v} \frac{\partial p}{\partial \eta} \right) d\eta + \vec{v} \cdot \nabla p - gF_p(\eta)$$

and vertical velocity:

$$\dot{\eta} \frac{\partial p}{\partial \eta} = -\frac{\partial p}{\partial t} - \int_0^\eta \nabla \cdot \left(\vec{v} \frac{\partial p}{\partial \eta} \right) d\eta - gF_p(\eta)$$

The momentum equations are integrated divergence and rotational form:

$$\begin{aligned} \frac{\partial \zeta}{\partial t} &= \nabla \times \left(\overrightarrow{H_{\vec{v}}} - g \frac{\partial \eta}{\partial p} \frac{\partial \overrightarrow{F_{\vec{v}}}}{\partial \eta} + \overrightarrow{S_{\vec{v}}} \right) + K_\zeta \\ \frac{\partial D}{\partial t} &= \nabla \cdot \left(\overrightarrow{H_{\vec{v}}} - g \frac{\partial \eta}{\partial p} \frac{\partial \overrightarrow{F_{\vec{v}}}}{\partial \eta} + \overrightarrow{S_{\vec{v}}} \right) - \Delta(\Phi + E_c) + K_D \end{aligned}$$

with:

$$\begin{aligned} H_u &= (\zeta + f)v - \dot{\eta} \frac{\partial u}{\partial \eta} + \frac{\partial \eta}{\partial p} \frac{\partial \Phi}{\partial \eta} \frac{1}{a} \frac{\partial p}{\partial \lambda} \\ H_v &= -(\zeta + f)u - \dot{\eta} \frac{\partial v}{\partial \eta} + \frac{\partial \eta}{\partial p} \frac{\partial \Phi}{\partial \eta} \frac{(1-\mu^2)}{a} \frac{\partial p}{\partial \mu} \\ E_c &= \frac{1}{2} \left(u^2 + v^2 \right) \end{aligned}$$

The wind is calculated from velocity potential χ and stream function ψ by:

$$\vec{v} = \nabla \chi + \nabla \times \psi$$

Velocity potential and stream function are obtained from divergence and vorticity by solving Poisson equations:

$$\begin{aligned}\chi &= \Delta^{-1}D\\ \psi &= \Delta^{-1}\zeta\end{aligned}$$

It is in these last three relations, the kernel of the Laplacian Δ is implicitly supposed to reduce to constant functions. It is then equivalent to know the wind or divergence and vorticity pair. This property is true on the sphere as well as on the torus.

3 Variable mesh

3.1 Stretched and tilted grid

Arpege makes it possible to increase the horizontal resolution on part of the sphere, while preserving locally the isotropy (Courtier and Geleyn, 1988). For that one uses a new set of coordinates (λ', μ'). First, North Pole is shifted at the point of coordinates (λ_0, μ_0) which becomes the new pole (or tilted pole). One defines new coordinates (λ_b, μ_b) by:

$$\begin{aligned}\mu_b &= \mu_0\mu + \sqrt{1-\mu_0^2}\sqrt{1-\mu^2}\cos(\lambda-\lambda_0)\\ \cos\lambda_b &= (1-\mu_b^2)^{-\frac{1}{2}}(\mu\sqrt{1-\mu_0^2} - \mu_0\sqrt{1-\mu^2}\cos(\lambda-\lambda_0))\\ \sin\lambda_b &= (1-\mu_b^2)^{-\frac{1}{2}}\sqrt{1-\mu^2}\sin(\lambda_0-\lambda)\end{aligned}$$

The origin longitude is the one which contains the geographical North Pole. Reciprocally:

$$\begin{aligned}\mu &= \mu_0\mu_b + \sqrt{1-\mu_0^2}\sqrt{1-\mu_b^2}\cos\lambda_b\\ \cos(\lambda-\lambda_0) &= (1-\mu^2)^{-\frac{1}{2}}(\mu_b\sqrt{1-\mu_0^2} - \mu_0\sqrt{1-\mu_b^2}\cos\lambda_b)\\ \sin(\lambda_0-\lambda) &= (1-\mu^2)^{-\frac{1}{2}}\sqrt{1-\mu_b^2}\sin\lambda_b\end{aligned}$$

Then, one carries out a stretching of the latitudes (without modifying longitudes, $\lambda' = \lambda_b$) obtained by homothety of a factor c on stereographic projection at the pole of stretching. It comes:

$$\mu' = \frac{(1-c^2)+(1+c^2)\mu_b}{(1+c^2)+(1-c^2)\mu_b}$$

and reciprocally:

$$\mu_b = \frac{(c^2-1)+(c^2+1)\mu'}{(c^2+1)+(c^2-1)\mu'}$$

3.2 Impact on the equations

The conformal transform of η-surfaces described above changes the model equations. The fields are represented by a base of functions defined on transformed surfaces. The equations are integrated on original surfaces. However modifications are necessary in the calculation of the horizontal derivative. It consists of multiplying them by a scale factor m. In the case of the present transform, this factor is:

$$m = \frac{c^2+1}{2c} + \frac{c^2-1}{2c}\mu'$$

At the pole of dilation ($\mu' = 1$) this factor is c. At the pole of contraction ($\mu' = -1$) it is $1/c$. The horizontal wind thus becomes:

$$\vec{v} = m\,\vec{v}\,'$$

and the horizontal gradient:

$$\nabla = m\,\nabla'$$

In going from the real sphere to the transformed sphere, the velocity potential χ and the stream function ψ, which are scalars, are invariant. The main modification consists of solving Poisson equation in a more complicated form:

$$\begin{aligned} \chi &= \Delta'^{-1}\frac{D}{m^2} \\ \psi &= \Delta'^{-1}\frac{\zeta}{m^2} \end{aligned}$$

where Δ'^{-1} is the same formal operator as Δ^{-1} but on the transformed sphere. As a consequence, the state variable of the model is: $\zeta' = \zeta/m^2$ and $D' = D/m^2$ (or equivalently ψ and χ).

The equations become then:

$$\begin{aligned} \frac{\partial \zeta'}{\partial t} &= \nabla' \times \left[\frac{1}{m}\left(\overrightarrow{H_{\vec{v}}} - g\frac{\partial \eta}{\partial p}\frac{\partial \overrightarrow{F_{\vec{v}}}}{\partial \eta} + \overrightarrow{S_{\vec{v}}}\right)\right] + K_{\zeta'} \\ \frac{\partial D'}{\partial t} &= \nabla' \cdot \left[\frac{1}{m}\left(\overrightarrow{H_{\vec{v}}} - g\frac{\partial \eta}{\partial p}\frac{\partial \overrightarrow{F_{\vec{v}}}}{\partial \eta} + \overrightarrow{S_{\vec{v}}}\right)\right] - \Delta'(\Phi + E_c) + K_{D'} \end{aligned}$$

3.3 Impact on post-processing

The files produced by ARPEGE-CLIMAT for restarting as well as for post-processing contain the model prognostic variables, even though, in the case of post-processing, ψ and χ are transformed into $\vec{v}\,'$. This is also true for momentum fluxes. As a consequence, both components of wind velocity or surface stress have to be multiplied by m before any comparison with observations or other model outputs. This operation can be done, for example, at the same time as the conversion from ARPEGE format to another format.

4 Lagrangian form of the primitive equations

The Lagrangian form of the equations of momentum, thermodynamics and moisture are respectively (2), (3) and (4). The continuity equation (5) has as a Lagrangian form:

$$\frac{d}{dt}\left(\frac{\partial p}{\partial \eta}\right) = -\frac{\partial p}{\partial \eta}\left(D + \frac{\partial \dot{\eta}}{\partial \eta}\right) - g\frac{\partial F_p}{\partial \eta} \tag{6}$$

It can take another form, more adapted to the semi-Lagrangian advection method :

$$\frac{d}{dt}\left[\left(\frac{\partial p}{\partial \eta}\right) J\right] = -g\frac{\partial F_p}{\partial \eta} \tag{7}$$

where J indicates the Jacobian of the transform which associates the position of a point at time t with its position at a reference time t_o.

On the stretched and tilted sphere, the continuity equation (6) becomes:

$$\frac{d}{dt}\left(\frac{\partial p}{\partial \eta}\right) = -\frac{\partial p}{\partial \eta}\left(m^2 D' + \frac{\partial \dot{\eta}}{\partial \eta}\right) - g\frac{\partial F_p}{\partial \eta}$$

and its Lagrangian form (7):

$$\frac{d}{dt}\left[\left(\frac{\partial p}{\partial \eta}\right) \frac{J'}{m^2}\right] = -g\frac{\partial F_p}{\partial \eta}$$

The momentum equation (2) becomes:

$$\frac{d\,m\vec{v}\,'}{d\,t} + m\left[2\Omega \times \vec{v}\,' + RT\nabla' \ln p + \nabla'\Phi\right] = -g\frac{\partial \eta}{\partial p}\frac{\partial \overrightarrow{F_{\vec{v}}}}{\partial \eta} + \overrightarrow{S_{\vec{v}}} + \overrightarrow{K_{\vec{v}}}$$

The equations for thermodynamics (3) and moisture (4) are formally unchanged.

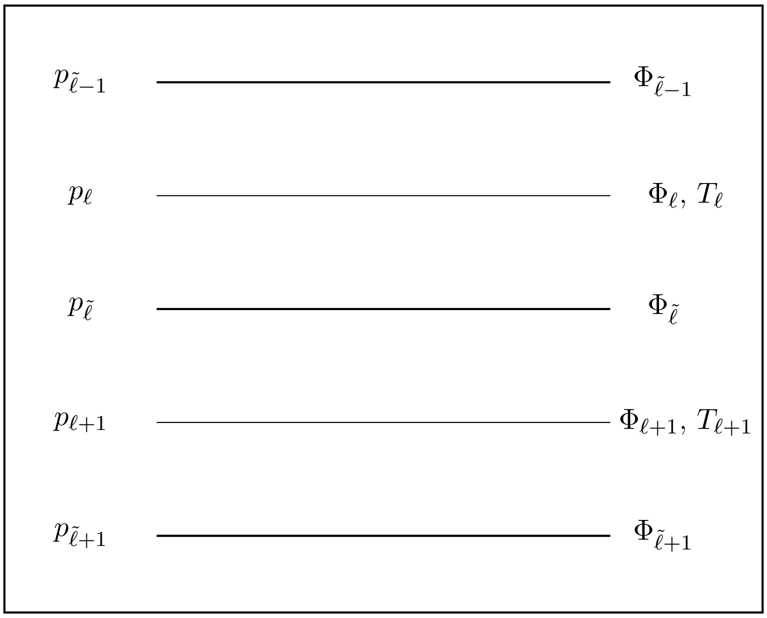


Figure 1: Position of variables on the vertical.

5 Vertical discretization

5.1 Model vertical levels

The atmosphere is vertically split into L layers, defined by the pressures at their interfaces, which are calculated by:

$$p_{\tilde{\ell}} = A_{\tilde{\ell}} + B_{\tilde{\ell}} p_s \qquad \tilde{\ell} = 0, \ldots, L \tag{8}$$

$A_{\tilde{\ell}}$ and $B_{\tilde{\ell}}$ are constants which define the vertical coordinate. The vertical distribution of the variables is presented in Figure 1, indices $_\ell$ relating to the mid-layers (also named full levels) and indices $_{\tilde{\ell}}$ to the inter-layers (also named half levels).

The values of $A_{\tilde{\ell}}$ and $B_{\tilde{\ell}}$ are imposed to the model. In earlier versions of Arpege-climat they were calculated from analytical functions. In recent cycles, one uses the same vertical discretization as in forecast models (Météo-France or ECMWF).

5.2 Vertical discretization of the equations

The vertical discretization scheme is defined according to Simmons and Burridge (1981). One introduces the operator δ which represents the variation of a variable between the two ends of a layer:

$$\delta p_\ell = p_{\tilde{\ell}} - p_{\tilde{\ell}-1}$$

Index $_\ell$ will be omitted when there is no ambiguity.

Continuity equation

The continuity equation is written as (F_m being the mass flux due to water cycle):

$$\frac{\partial}{\partial \eta}\left(\frac{\partial p}{\partial t}\right) + \nabla \cdot \left(\vec{v}\frac{\partial p}{\partial \eta}\right) + \frac{\partial}{\partial \eta}\left(\dot{\eta}\frac{\partial p}{\partial \eta}\right) = F_m$$

For a given layer, one writes it as:

$$\frac{\partial(\delta p)}{\partial t} = -\nabla \cdot (\vec{v}\,\delta p) - \delta\left(\dot{\eta}\frac{\partial p}{\partial \eta}\right) + F_m$$

One will note, in the following, the physical term of the discretized equation in the same way as the corresponding term of the continuous equation.

Summing on the vertical, one obtains the evolution equation of surface pressure:

$$\frac{\partial p_s}{\partial t} = -\sum_{\ell=1}^{L}\left[\delta p\, D + \delta B\, \vec{v}\cdot\nabla p_s\right] - g(P+E)$$

since:

$$\nabla \delta p = \delta B\, \nabla p_s$$

The vertical speed is obtained by summing the continuity equation from the top to the current level:

$$w_{\tilde{\ell}} = \left(\dot{\eta}\frac{\partial p}{\partial \eta}\right)_{\tilde{\ell}} = \sum_{k=1}^{\ell}\left[-\delta B_k\, \vec{v}_k\cdot\nabla p_s - \delta p_k D_k + F_{mk}\right] - B_{\tilde{\ell}}\frac{\partial p_s}{\partial t}$$

Eulerian advection

The vertical advections in momentum, thermodynamics and moisture equations are calculated using the scheme:

$$\left(w\frac{\partial X}{\partial p}\right)_\ell = \frac{1}{2\delta p}\left[w_{\tilde{\ell}}(X_{\ell+1}-X_\ell)+w_{\tilde{\ell}-1}(X_\ell - X_{\ell-1})\right]$$

This scheme ensures conservation of X and X^2. It results from the following form of the vertical advection:

$$w\frac{\partial X}{\partial p} = \frac{\partial wX}{\partial p} - X\frac{\partial w}{\partial p}$$

with the interpolation:

$$X_{\tilde{\ell}} = \frac{1}{2}(X_\ell + X_{\ell+1})$$

In the case of the semi-Lagrangian scheme, see Chapter **??**.

Hydrostatic equation

The equation of hydrostatic balance is integrated by using the centered scheme:

$$\Phi_{\tilde{\ell}-1} = \Phi_{\tilde{\ell}} - R_\ell T_\ell \ln\frac{p_{\tilde{\ell}-1}}{p_{\tilde{\ell}}}$$

Which gives, summing from the surface:

$$\Phi_{\tilde{\ell}} = \Phi_s + \sum_{k=L}^{\ell+1} R_k T_k \ln\frac{p_{\tilde{k}}}{p_{\tilde{k}-1}}$$

where Φ_s indicates surface geopotential.

To calculate the geopotential in the mid-layers one writes:

$$\Phi_\ell = \Phi_{\tilde{\ell}} + \alpha_\ell R_\ell T_\ell$$

where:

$$\begin{cases} \alpha_1 & = 1 \\ \alpha_\ell & = 1 - \dfrac{p_{\tilde{\ell}-1}}{\delta p_\ell}\ln\dfrac{p_{\tilde{\ell}}}{p_{\tilde{\ell}-1}} \end{cases} \tag{9}$$

The expression of Φ_ℓ is consistent with the discretization of the form:

$$\Phi = \frac{\partial p\Phi}{\partial p} + RT \tag{10}$$

Discretization of the pressure force

For conserving angular momentum, one writes the acceleration term due to the pressure force as:

$$\begin{aligned}\nabla\Phi + RT\,\nabla\ln p &= \left[\nabla\left(\Phi\frac{\partial p}{\partial\eta}\right) - \frac{\partial(\Phi\,\nabla p)}{\partial\eta}\right]\left(\frac{\partial p}{\partial\eta}\right)^{-1} \\ &= \nabla\Phi + \left[\Phi\nabla\left(\frac{\partial p}{\partial\eta}\right) - \frac{\partial(\Phi\,\nabla p)}{\partial\eta}\right]\left(\frac{\partial p}{\partial\eta}\right)^{-1}\end{aligned}$$

The discretization of this term yields:

$$(RT\nabla\ln p)_\ell = R_\ell T_\ell \underbrace{\frac{1}{\delta p_\ell}\left[\delta B_\ell + \frac{C_\ell}{\delta p_\ell}\ln\frac{p_{\tilde{\ell}}}{p_{\tilde{\ell}-1}}\right]}_{\text{ZRTGR}} \cdot\nabla p_s$$

with:

$$C_\ell = A_{\tilde{\ell}}B_{\tilde{\ell}-1} - A_{\tilde{\ell}-1}B_{\tilde{\ell}}$$

Term ZRTGR is also used in the calculation of the energy transformation term. It can be written in the simpler form:

$$\frac{1}{\delta p}\left[\alpha_\ell\delta B_\ell + B_{\tilde{\ell}-1}\ln\frac{p_{\tilde{\ell}}}{p_{\tilde{\ell}-1}}\right]$$

Energy transformation term

The energy transformation term is written as:

$$\frac{RT}{c_p}\frac{\omega}{p}$$

One writes:

$$\frac{\omega}{p} = \vec{v}\cdot\nabla\ln p - \frac{1}{p}\int_0^\eta\left[\vec{v}\cdot\nabla\left(\frac{\partial p}{\partial\eta}\right) + D\frac{\partial p}{\partial\eta} + F_m\right]d\eta$$

The first term of the right-hand member is evaluated in the same way as the corresponding term of the momentum equation:

$$(\vec{v}\cdot\nabla\ln p)_\ell = \underbrace{\frac{1}{\delta p_\ell}\left[\delta B_\ell + \frac{C_\ell}{\delta p_\ell}\ln\frac{p_{\tilde{\ell}}}{p_{\tilde{\ell}-1}}\right]}_{\text{ZRTGR}} \vec{v}_\ell\cdot\nabla p_s$$

and the second term:

$$-\frac{1}{\delta p_\ell}\left[\alpha_\ell\left(\nabla\cdot(\vec{v}_\ell\delta p_\ell)+F_{m\ell}\right)+\left(\ln\frac{p_{\tilde{\ell}}}{p_{\tilde{\ell}-1}}\right)\sum_{k=1}^{\ell-1}\left(\nabla\cdot(\vec{v}_k\delta p_k)+F_{mk}\right)\right]$$

where:

$$\nabla\cdot(\vec{v}_\ell\,\delta p_\ell) = \vec{v}_\ell\cdot\nabla\delta p_\ell + \delta p_\ell\, D_\ell$$

This discretization results from writing the last term in the form:

$$\frac{1}{p}\int X\,d\eta = \left(\frac{\partial p}{\partial\eta}\right)^{-1}\frac{\partial\ln p}{\partial\eta}\int X\,d\eta = \left[\frac{\partial}{\partial\eta}\left(\ln p\int X\,d\eta\right) - X\ln p\right]\left(\frac{\partial p}{\partial\eta}\right)^{-1}$$

in which one calculates the last logarithm of the pressure by:

$$\ln p = \frac{d}{d\,p}\,(p\ln p) - 1$$

3

Spectral transforms

1 Introduction

Arpege-IFS is a spectral global model. One part of computations is made in spectral space (semi-implicit scheme, horizontal diffusion scheme), the other part in grid-point space on a grid defined by a Gaussian quadrature. It is therefore necessary to perform spectral transforms from spectral space to grid-point space or vice-versa. The present chapter aims at giving a brief summary of the spectral method. For more algorithmic details on can report to Rochas and Courtier (1992) or to Temperton (1991). Computation aspects are described in details in Yessad (2007). Most subroutines are located in an independent library named TFL.

For a global spectral model, spectral transforms are a combination of a Legendre transform and a Fourier transform. A spectral limited-area model like Aladin uses a double Fourier representation for spectral fields.

2 Spectral representation

2.1 Spherical harmonics

The spherical Laplacian operator Δ on a sphere Σ of radius a admits as eigenvalues family $-n(n+1)/a^2$ with an order of $2n+1$. The eigenvectors are the surface spherical harmonics. An orthogonal base of an eigenspace is given by:

$$Y_n^m = P_n^m(\mu)e^{im\lambda}$$

where μ is the sine of the latitude and λ longitude. The $P_n^m(\mu)$ are the first-type Legendre polynomials. The standardization of the $P_n^m(\mu)$ is such as:

$$\iint_\sigma Y_n^m(\lambda,\mu) Y_{n'}^{m'}(\lambda,\mu) d\sigma = \delta_{n=n'}\delta_{m=m'} \iint_\sigma d\sigma$$

One has then, for $m \geq 0$:

$$P_n^m(\mu) = \sqrt{(2n+1)\frac{(n-m)!}{(n+m)!}}\ \frac{1}{2^n n!}(1-\mu^2)^{m/2}\frac{d^{n+m}}{d\mu^{n+m}}(\mu^2-1)^n \qquad (1)$$

And for $m \leq 0$:

$$P_n^{-m}(\mu) = P_n^m(\mu)$$

As mentioned above, the spherical harmonics satisfy:

$$\Delta Y_n^m = -\frac{n(n+1)}{a^2} Y_n^m$$

The Laplacian operator is invariant by rotation, his eigenspaces are thus also invariant by rotation. We deduce from it that under the effect of a rotation *i.e.* a change of pole, the coefficients of the decomposition of a field in spherical harmonics are exchanged at fixed n. For each n, there is a linear transformation (thus a matrix) which makes it possible to make the basic change. This property is preserved by the discretization if truncation is triangular. This is why triangular truncation is said to be isotropic.

2.2 Collocation grid

A collocation grid is selected for non-linear calculations which cannot be carried out directly on the coefficients of the spherical harmonics. At each time step, one passes from the spectral coefficients to the grid-point values and reciprocally. From the expression of the Y_n^m, the E-W transforms are Fourier transforms. To use fast Fourier transforms (FFT), one thus needs a regular grid in longitude. In the N-S direction, one uses a Gauss quadrature for the direct transform, therefore the latitudes are not equidistant.

At high latitudes, one takes less points on a latitude circle than Fourier modes, in order to maintain the grid almost isotropic. The collocation grid is said to be reduced (Hortal and Simmons, 1991). The grid is said to be Gaussian quadratic (or simply Gaussian) when the number of latitude circles is large enough (for a given truncation) so that the Gauss quadrature is exact for any product of two Legendre polynomials in the truncation. This is useful in the case of the Eulerian advection (product of velocity by gradient). When the approximate calculation of the integral is exact only for the Legendre polynomials of the truncation, the grid is said to be linear (Hortal, 1996).

2.3 Spectral transforms

To pass from the spectral coefficients to the grid points values, one uses the two formulas of direct evaluation:

$$A_m(\mu) = \sum_{n=|m|}^{\infty} A_n^m P_n^m(\mu)$$

and:

$$A(\lambda, \mu) = \sum_{m=-\infty}^{+\infty} A_m(\mu) e^{im\lambda}$$

where the A_m are called the Fourier coefficients.

The horizontal derivatives are calculated exactly by using the derivatives of the Legendre functions for the N-S direction and by multiplying by im for the E-W direction.

From the grid point fields, the Fourier coefficients are determined by:

$$A_m(\mu) = \frac{1}{2\pi} \int_0^{2\pi} A(\lambda, \mu)\, e^{-im\lambda}\, d\lambda$$

a formula which ensures that $A_0(\mu)$ is the average of field A along parallel μ. Integration is carried out numerically by using a fast Fourier transform (FFT). The integral in latitude is:

$$A_n^m = \frac{1}{2} \int_{-1}^{1} A_m(\mu) P_n^m(\mu)\, d\mu$$

As mentioned above, we use a Gauss quadrature, discrete version of the preceding integral:

$$A_n^m = \sum_{k=1}^{K} \omega(\mu_k) A_m(\mu_k) P_n^m(\mu_k)$$

where the μ_k are the K roots of the Legendre polynomial of degree K and the Gauss weights $\omega(\mu_k)$ are given by:

$$\omega(\mu_k) = \frac{1-\mu_k^2}{(NP_{N-1})^2}$$

Here, the Legendre polynomial is the one of the mathematicians, the squared norm of which is $1/(2n+1)$.

3 Horizontal discretization

3.1 Spectral truncation

In practical the expression of A is limited to a finite set of harmonics corresponding to $0 \leq n \leq N$ and $-n \leq m \leq n$. That defines a triangular truncation N. The truncated expansion of field A reads:

$$A(\lambda,\mu) = \sum_{m=-N}^{m=N} \sum_{n=|m|}^{n=N} A_n^m P_n^m(\mu) e^{im\lambda}$$

Due to the properties of $P_n^m(\mu)$, expression of A becomes for a real scalar field:

$$A(\lambda,\mu) = \sum_{m=0}^{m=N} \sum_{n=|m|}^{n=N} A_n^m P_n^m(\mu) e^{im\lambda}$$

3.2 Horizontal derivatives

Meridional derivative relative to latitude θ

For a variable A, meridional derivative is discretized in spectral space by the following formula:

$$\left(\cos\theta \frac{\partial A}{\partial \theta}\right)_n^m = -(n-1)e_n^m A_{n-1}^m + (n+2)e_{n+1}^m A_{n+1}^m$$

where $e_0^0 = 0$ and:

$$e_n^m = \sqrt{\frac{n^2-m^2}{4n^2-1}}$$

Zonal derivative relative to longitude λ

For a variable A, zonal derivative is discretized in spectral space by the following formula:

$$\left(\frac{\partial A}{\partial \lambda}\right)_n^m = i\,m A_n^m$$

Such a derivation can be made on Fourier coefficients by a multiplication by $i\,m$.

3.3 Spectral relationships for wind representation

The reduced components of the velocity are obtained by dividing the physical components by the mapping factor M. Divergence and vorticity are divided by M^2. As reduced divergence $D^{'}$ is obtained from velocity potential χ by a laplacian operator, and as reduced vorticity $\zeta^{'}$ is obtained similarly from stream function ψ, we have in spectral space:

$$D'^m_n = -\frac{n(n+1)}{a^2}\chi_n^m$$

$$\zeta'^m_n = -\frac{n(n+1)}{a^2}\psi_n^m$$

Relationship between $U^{'}$, ψ and χ:

$$(U^{'} a\cos\theta) = \frac{\partial \chi}{\partial \lambda} - \cos\theta \frac{\partial \psi}{\partial \theta}$$

the spectral discretization of which is:

$$(U^{'} a\cos\theta)_n^m = i\,m\chi_n^m + (n-1)e_n^m\psi_{n-1}^m - (n+2)e_{n+1}^m\psi_{n+1}^m$$

Relationship between $V^{'}$, ψ and χ:

$$(V^{'} a\cos\theta) = \frac{\partial \psi}{\partial \lambda} + \cos\theta \frac{\partial \chi}{\partial \theta}$$

the spectral discretization of which is:

$$(V^{'} a\cos\theta)_n^m = i\,m\psi_n^m - (n-1)e_n^m\chi_{n-1}^m + (n+2)e_{n+1}^m\chi_{n+1}^m$$

Relationship between $D^{'}$, $U^{'}$ and $V^{'}$:

$$D^{'} = \frac{1}{a\cos\theta}\left(\frac{\partial U^{'}}{\partial\lambda} + \frac{\partial(V^{'}\cos\theta)}{\partial\theta}\right)$$

which can be rewritten:

$$D^{'} = \frac{1}{a^2\cos^2\theta}\left(\frac{\partial(U^{'}a\cos\theta)}{\partial\lambda} + \cos\theta\frac{\partial(V^{'}a\cos\theta)}{\partial\theta}\right)$$

Relationship between $\zeta^{'}$, $U^{'}$ and $V^{'}$:

$$\zeta^{'} = \frac{1}{a\cos\theta}\left(\frac{\partial V^{'}}{\partial\lambda} - \frac{\partial(U^{'}\cos\theta)}{\partial\theta}\right)$$

which can be rewritten:

$$\zeta^{'} = \frac{1}{a^2\cos^2\theta}\left(\frac{\partial(V^{'}a\cos\theta)}{\partial\lambda} - \cos\theta\frac{\partial(U^{'}a\cos\theta)}{\partial\theta}\right)$$

Spectral discretizations allow to retrieve easily spectral components of fields $D^{'}a^2\cos^2\theta$ and $\zeta^{'}a^2\cos^2\theta$, but not directly spectral components of $D^{'}$ and $\zeta^{'}$ (requiring inversion of a penta-diagonal matrix). In fact, the algorithm involved to retrieve spectral coefficients of $D^{'}$ and $\zeta^{'}$ once known values of wind components is slightly different (requiring a division by $a\cos\theta$ in Fourier space), and is described in detail in Temperton (1991).

3.4 Relationship between dimension in spectral space and in grid point space

Quadratic grid, linear grid

Spectral space is defined by a triangular truncation N. Grid point space has *ndgl* latitudes and a maximum number of longitudes equal to *ndlon*. *ndlon* and *ndgl* are always even integers: if *ndlon* is a multiple of 4, $ndgl = ndlon/2$; if *ndlon* is not a multiple of 4, $ndgl = ndlon/2 + 1$. For a quadratic Gaussian grid, there is a relationship between these parameters to avoid aliasing on quadratic terms.

- If the stretching coefficient c is equal to 1 (no stretching), N is the maximum integer verifying the relationship $3 * N \leq (ndlon - 1)$.

- If the stretching coefficient c is greater than 1 (stretching), N is the maximum integer verifying the relationship $3 * N \leq \min(2 * ndgl - 3, ndlon - 1)$.

In a semi-Lagrangian scheme the advective quadratic terms disappear, so it is possible to use a smaller grid-point space: a linear grid. It is characterized by:

- If the stretching coefficient c is equal to 1 (no stretching), N is the maximum integer verifying the relationship $2 * N \leq (ndlon - 1)$.
- If the stretching coefficient c is greater than 1 (stretching), N is the maximum integer verifying the relationship $2 * N \leq \min(2 * ndgl - 3, ndlon - 1)$.

In ARPEGE-CLIMAT with $c > 1$, some aliasing is allowed, and the same N is taken as in the case $c = 1$.

Admissible dimensions for longitude

The current algorithm for FFT allows integers *ndlon* which can factorize as $2^{1+p_2} * 3^{p_3} * 5^{p_5}$. That limits the possibility of choosing the dimensions in a discontinuous subset of truncations and dimensions for Gaussian grid. In the range compatible with climate multi-year integrations, the admissible sizes (with even number of latitudes) are:

64 72 80 90 96 100 108 120 128 144 150 160 162 180 192 200 216 240 250 256 270 288 300 320 324 360 384 400 432 450 480 486 500 512 540 576 600 640 648 720

Reduced grid

To save memory and computation time (in particular in the physical parametrizations), the number of longitudes per latitude circle is reduced outside the tropics, in order to maintain a quasi-isotropic grid (note that the spectral triangular truncation allows an isotropic representation of the fields, despite the accumulation of grid points near the poles). This optimization is done at the expense of an aliasing error (Williamson and Rosinski, 2000). An algorithm is proposed to compute for a given truncation, the number of longitudes per latitude circle which is the best compromise between accuracy in the spectral transform an isotropy in the physical parametrizations.

4

Semi-lagrangian computations

1 Introduction.

1.1 General purpose of this documentation.

This chapter describes the set of equations used, and also the way to integrate the dynamics of the model with the semi-Lagrangian method currently implemented in ARPEGE/IFS. Equations will be written without horizontal diffusion scheme (which is treated in spectral computations), and without Rayleigh friction (which is done in grid-point space) in order to give a clearer presentation of the discretised equations. For additional information about horizontal diffusion scheme, report to documentation (IDDH). Extensions to ALADIN (cycle AL37T1) are not described in detail, but differences are briefly mentioned.

The following sets of equations will be described in this documentation:

- The 2D shallow-water equations model (configuration 201).
- The primitive equations hydrostatic model (configuration 1): thin layer and deep layer (according to White and Bromley, 1995) formulations.

In the current version of this documentation, some points are partly described:

- The option with finite element vertical discretisations **LVERTFE**=.T. . When **LVERTFE**=.T., the main modifications in the semi-Lagrangian part of the code are the following ones:

  - $\dot{\eta}\frac{\partial\Pi}{\partial\eta}$ is directly computed at full levels so the way of computing $\dot{\eta}$ to find the vertical displacement is modified.
  - All the vertical integrals use a matricial multiplication with special coefficients computed in the setup routine **SUVERTFE1** or **SUVERTFE3**; the vertical integration is done by routine **VERINT**.
  - All the vertical derivatives use a matricial multiplication with special coefficients computed in the setup routine **SUVERTFE3D**; the vertical derivation is done by routine **VERDER**.

- The spline cubic vertical interpolations used when **LVSPLIP**=.T. (in practical only for ozone, when **YO3_NL%LVSPLIP**=.T.). The detail of calculation of the interpolation weights is not currently given.

- The “semi-Lagrangian horizontal diffusion" interpolations (SLHD) used when **LSLHD**=.T. .

- Modified interpolations when **L3DTURB**=.T. .

1.2 Distributed memory code.

Some distributed code has been introduced for the semi-Lagrangian scheme, for some convenience expressions such “DM-local" or “DM-global" will be used to describe some distributed memory features.

- Expression “DM-local" for a quantity means “local to the couple of processors (*proca*,*procb*)": each processor has its own value for the quantity. Expression “DM-local computations" means that the computations are made independently in each processor on “DM-local" quantities, leading to results internal to each processor, which can be different from a processor to another one.

- Expression “DM-global" for a quantity means that it has a unic value available in all the processors. Expression “DM-global computations" means that the computations are either made in one processor, then the results are dispatched in all the processors, or the same computations are made in all the processors, leading to the same results in all the processors.

- In a routine description the mention “For distributed memory computations are DM-local" means that all calculations made by this routine are DM-local; the mention “For distributed memory computations are DM-global" means that all calculations made by this routine are DM-global; when no information is provided it means that a part of calculations is DM-local and the other part is DM-global.

1.3 Mass corrector.

Two different mass correctors are coded (but do not work in all configurations), one used at ECMWF, the other one for METEO-FRANCE climatic simulations. They are useful especially for climatic simulations, to correct the lack of conservativity of the semi-Lagrangian scheme. They are not described in this documentation.

1.4 Deep layer equations (according to White and Bromley, 1995).

They have been introduced in the hydrostatic model for the Eulerian scheme and most options of the semi-Lagrangian scheme.

2 Definition of Eulerian and semi-Lagrangian schemes.

2.1 Eulerian scheme.

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

$$\frac{\partial X}{\partial t} = -\mathbf{U}.\nabla_3 X + \dot{X} \tag{1}$$

where $\mathbf{U}$ is the 3D wind, ∇_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:

  $$M \mid \mathbf{V} \mid \frac{Dt}{2}\sqrt{\frac{N(N+1)}{r^2}} < 1 \tag{2}$$

  which can be rewritten:

  $$M \frac{\mid \mathbf{V} \mid}{r} \frac{Dt}{2}\sqrt{N(N+1)} < 1 \tag{3}$$

  where M is the mapping factor, Dt is the time-step at the first integration step and twice the time-step otherwise (leap-frog scheme), $\mid \mathbf{V} \mid$ is the horizontal wind modulus, N is the truncation, r is the distance between the point and the centre of the Earth.

- For the spectral limited area model ALADIN, the horizontal CFL condition writes as:

  $$M \frac{\mid \mathbf{V} \mid}{r} \frac{Dt}{2}(2\pi)\sqrt{\frac{1}{\frac{L_{\mathrm{x}}^2}{a^2 N_{\mathrm{m}}^2} + \frac{L_{\mathrm{y}}^2}{a^2 N_{\mathrm{n}}^2}}} < 1 \tag{4}$$

  where M is the mapping factor, Dt is the time-step at the first integration step and twice the time-step otherwise (leap-frog scheme), $\mid \mathbf{V} \mid$ is the horizontal wind modulus, N_{m} is the zonal truncation, N_{n} is the meridian truncation, a is the mean Earth radius, r is the distance between the point and the centre of the Earth, L_{x} (resp. L_{y}) is the zonal (resp. meridian) length of the ALADIN domain taken on a surface iso $r = a$.

The vertical CFL condition writes as:

$$\mid \dot{\eta} \mid \frac{Dt}{2}\Delta\eta < 1 \tag{5}$$

For a TL358L46 (triangular truncation 358, linear Gaussian grid, 46 levels) model with stretching coefficient c=2.4, that gives $\Delta t \simeq 2$ mn.

2.2 Semi-Lagrangian scheme.

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

$$\frac{dX}{dt} = \dot{X} \tag{6}$$

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 (roughly 1.5 times the cost of the Eulerian scheme in the TL358L46c2.4 model with full French 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 η-coordinates, $\dot{\eta} = \frac{d\eta}{dt}$. Lipschitz criterion writes for a three-time level semi-Lagrangian scheme:

$$\mid D + \frac{\partial \dot{\eta}}{\partial \eta} \mid \frac{Dt}{2} < 1 \tag{7}$$

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

$$\mid D + \frac{\partial \dot{\eta}}{\partial \eta} \mid \frac{\Delta t}{2} < 1 \tag{8}$$

Expressions “semi-Lagrangian scheme", “three-time level semi-Lagrangian scheme" and “two-time level semi-Lagrangian scheme" will be from now on abbreviated into “SL scheme", “3TL SL scheme" and “2TL SL scheme".

3 The 2D equations.

3.1 Notations for the 2D equations.

- $\mathbf{V}$ is the horizontal wind. Its zonal component (on the Gaussian grid) is denoted by U. Its meridian component (on the Gaussian grid) is denoted by V.
- D is the horizontal wind divergence.
- ζ is the horizontal wind vorticity.
- Φ is the equivalent height. Φ_{s} is the surface geopotential height (i.e. the orography). Φ^* is a reference equivalent height which is only used in the semi-implicit scheme and the linear model.
- $\mathbf{\Omega}$ is the Earth rotation angular velocity.
- ∇ is the first order horizontal gradient on η-surfaces.
- a is the Earth radius.
- $(\lambda_{\mathrm{bne}}, \theta_{\mathrm{bne}})$ are the longitude-latitude coordinates on a tilted and not stretched geometry, the tilting being the same as the one of the computational sphere.
- $\mathbf{k}$ is the unit vertical vector. One can write:

$$\mathbf{k} = \frac{\mathbf{r}}{\mid r \mid} = \frac{\mathbf{r}}{a}$$

3.2 The 2D shallow-water system of equations in spherical geometry.

Momentum equation.

Coriolis force can be treated explicitly ($\delta_{\vec{V}}$=0) or implicitly ($\delta_{\vec{V}}$=1) in the Lagrangian equation.

$$\frac{d\left(\vec{V} + \delta_{\vec{V}}(2\vec{\Omega} \wedge \vec{r})\right)}{dt} = [-2(1-\delta_{\vec{V}})(\vec{\Omega} \wedge \vec{V})] - \nabla\Phi \tag{9}$$

Continuity equation.

- **Conventional formulation.**

$$\frac{d(\Phi - (1-\delta_{TR})\Phi_{\mathrm{s}})}{dt} = -(\Phi - \Phi_s)D + \delta_{TR}\vec{V}\nabla(\Phi_s) \tag{10}$$

- **Lagrangian formulation.**

$$\frac{d((\Phi - \Phi_{\mathrm{s}})J)}{dt} = 0 \tag{11}$$

  J is a “Jacobian" quantity which satisfies to:

$$\frac{dJ}{dt} = -JD \tag{12}$$

4 The 3D equations in spherical geometry (ARPEGE/IFS).

4.1 Notations for the 3D equations.

- $\mathbf{V}$ is the horizontal wind. Its zonal component (on the Gaussian grid) is denoted by U. Its meridian component (on the Gaussian grid) is denoted by V.
- D is the horizontal wind divergence.
- ζ is the horizontal wind vorticity.
- T is the temperature.
- q is the humidity.
- $q_{\rm r}$: is the rain.
- Π is the hydrostatic pressure.
- $\Pi_{\rm s}$ is the hydrostatic surface pressure.
- $\boldsymbol{\Omega}$ is the Earth rotation angular velocity.
- $(\lambda_{\rm bne}, \theta_{\rm bne})$ are the longitude-latitude coordinates on a tilted and not stretched geometry, the tilting being the same as the one of the computational sphere.
- (λ, θ) are the geographical longitude-latitude coordinates.
- (Λ, Θ) are the computational sphere longitude-latitude coordinates.
- w is the z-coordinate vertical velocity: $w = \frac{dz}{dt}$.
- $\omega = \frac{d\Pi}{dt}$ is the total temporal derivative of the hydrostatic pressure.
- p is the pressure, $p_{\rm s}$ is the surface pressure.
- gz is the geopotential height.
- Φ is the total geopotential. $\Phi = gz$ in the thin layer equations, but not in the deep layer equations formulation of White and Bromley.
- $\Phi_{\rm s} = gz_{\rm s}$ is the surface geopotential (i.e. the orography).
- $\mathbf{r}$ is the vector directed upwards, the length of which is the Earth radius. The length of this vector is r. In the deep layer equations according to (White and Bromley, 1995), one uses an approximation of this radius, only depending on the hydrostatic pressure (“pseudo-radius").
- a is the average Earth radius near the surface.
- $W = \frac{dr_s}{dt}$ is the pseudo-vertical velocity used in some Coriolis and curvature terms in the deep layer equations according to (White and Bromley, 1995). $W = 0$ in the thin layer equations.
- $\mathbf{i}$ (resp. $\mathbf{j}$) is the unit zonal (resp. meridian) vector on the Gaussian grid.
- $\mathbf{k}$ is the unit vertical vector. One can write:

$$\mathbf{k} = \frac{\mathbf{r}}{r_s} = \frac{\mathbf{r}}{r}$$

- g is the gravity acceleration constant.
- In the case where vertical variations of g are taken into account, we denote by G the reference value of g at $r_s = a$.
- R is the gas constant for air and $R_{\rm d}$ the gas constant for dry air.
- $c_{\rm p}$ is the specific heat at constant pressure for air and $c_{\rm p_d}$ is the specific heat at constant pressure for dry air.
- $c_{\rm v}$ is the specific heat at constant volume for air and c_{vd} is the specific heat at constant volume for dry air.
- ∇ is the first order horizontal gradient on η-surfaces.

- α_{T} is a vertical-dependent coefficient used to define a thermodynamic variable $T+\delta_{TR}\frac{\alpha_{\mathrm{T}}\Phi_{\mathrm{s}}}{R_{\mathrm{d}}T_{ST}}$ less sensitive to orography than temperature T. Expression of α_{T} is:

$$\alpha_{\mathrm{T}} = B\left(-\frac{R_d}{g}\left[\frac{dT}{dz}\right]_{ST}\right)T_{ST}\left(\frac{\Pi_{ST}}{{\Pi_{\mathrm{s}}}^{st}}\right)^{\left(-\frac{R_d}{g}\left[\frac{dT}{dz}\right]_{ST}-1\right)} \tag{13}$$

where B is a vertically dependent and horizontally constant quantity which defines the vertical hybrid coordinate (see later paragraph “Definition of the vertical coordinate η", subsection (6.1)): B varies from 1 to 0 from bottom to top. Subscript “st" stands for “standard atmosphere".

- ρ is the mass per volume unit of air.

- M is the mapping factor.

- $\overline{M}$ is a reference mapping factor for the semi-implicit scheme.

- D_3 is the 3D divergence used in the NH model. Its expression is given by equation (2).

- τ, γ, ν, $\mathbf{L}^*$, ∂^* are linear operators used in the semi-implicit scheme (for more details, see documentation (IDSI) about semi-implicit scheme).

- T^* is a vertically-constant reference temperature which is used in the semi-implicit scheme and in some non-hydrostatic equations. IF **LSPRT**=.T. (use of virtual temperature in spectral transforms instead of real temperature), T^* is used as a reference virtual temperature (same default value).

- T_{a}^* is a cold vertically-constant reference temperature which is used in the semi-implicit scheme in the NH vertical divergence equation; it is recommended to have T_{a}^* lower than the current temperature.

- T_{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).

- Π^* is a reference hydrostatic pressure and ${\Pi_{\mathrm{s}}}^*$ is a reference hydrostatic surface pressure, which are used in the semi-implicit scheme and in some non-hydrostatic equations. These reference quantities are vertically dependent and “horizontally" (i.e. on η surfaces) constant. $\Delta\Pi^*$ are layer depths corresponding to a surface hydrostatic pressure equal to ${\Pi_{\mathrm{s}}}^*$.

- ${\Pi_{\mathrm{s}}}^{st}$ is a reference hydrostatic pressure equal to the surface pressure of the standard atmosphere (101325 Pa, variable **VP00**). Π_{ST} is a reference hydrostatic pressure defined at full levels and half levels corresponding to the surface reference hydrostatic pressure ${\Pi_{\mathrm{s}}}^{st}$ (stored in array **STPRE**).

- $\Delta\Phi^*$ is a reference geopotential depth computed on model layers, used in the non-hydrostatic model (more exactly in the non-linear part of the true 3D divergence). $\Delta\Phi^*$ is vertically dependent and “horizontally" (i.e. on η surfaces) constant.

- In the Wood and Staniforth deep-layer NH equations, a mass vertically integrated quantity $\tilde{\Pi}$ is introduced, in order to hide some metric terms, especially in the continuity equation: $\tilde{\Pi}$ replaces Π in the adiabatic equations (Π becomes, if needed, a diagnostic quantity). The definition of the hybrid vertical coordinate applies for $\tilde{\Pi}$, not for Π. Quantities δ and α (depths of logarithm of hydrostatic pressure) are replaced by $\tilde{\delta}$ and $\tilde{\alpha}$ (depths of logarithm of $\tilde{\Pi}$).

$$\frac{\partial\tilde{\Pi}}{\partial\Pi} = \frac{{r_s}^2}{a^2}\frac{G}{g}$$

4.2 The thin layer 3D primitive equation model: Lagrangian formulation.

Momentum equation.

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

$$\frac{d\left(\vec{V}+\delta_{\vec{V}}(2\vec{\Omega}\wedge\vec{r})\right)}{dt} = -2(1-\delta_{\vec{V}})(\mathbf{\Omega}\wedge\mathbf{V}) - \nabla\Phi - RT\nabla(\log\Pi) + \mathbf{F_V} \tag{14}$$

$\mathbf{F_V}$ is the physical contribution on horizontal wind.

Thermodynamic equation.

$$\frac{d\left(T+\delta_{TR}\frac{\alpha_{\mathrm{T}}\Phi_{\mathrm{s}}}{R_{\mathrm{d}}T_{ST}}\right)}{dt} = \frac{d\left(\delta_{TR}\frac{\alpha_{\mathrm{T}}\Phi_{\mathrm{s}}}{R_{\mathrm{d}}T_{ST}}\right)}{dt} + \frac{RT}{c_{\mathrm{p}}}\frac{\omega}{\Pi} + F_{\mathrm{T}} \tag{15}$$

F_{T} is the physical contribution on temperature. When $\delta_{TR}=1$ the Eulerian treatment of orography is applied and the prognostic variable is replaced by one variable less sensitive to the surface orography. This modification has been proposed by Ritchie and Tanguay (1996). See equation (13) for definition of α_{T}. Term $\frac{d\left(\delta_{TR}\frac{\alpha_{\mathrm{T}}\Phi_{\mathrm{s}}}{R_{\mathrm{d}}T_{ST}}\right)}{dt}$ only contains advection terms linked to horizontal variations of orography and vertical variations of the coefficient α_{T}.

Continuity equation.

The equation which is discretised is the vertically integrated Lagrangian formulation of continuity equation.

$$\int_{\eta=0}^{\eta=1}\frac{\partial B}{\partial\eta}\frac{d\left[\log\Pi_{\mathrm{s}}+\delta_{TR}\frac{\Phi_s}{R_dT_{st}}\right]}{dt}d\eta =$$
$$\int_{\eta=0}^{\eta=1}\frac{\partial B}{\partial\eta}\left[-\frac{1}{\Pi_{\mathrm{s}}}\int_{\eta=0}^{\eta=1}\nabla\left(\vec{V}\frac{\partial\Pi}{\partial\eta}\right)d\eta + \vec{V}\nabla\left[\log\Pi_{\mathrm{s}}+\delta_{TR}\frac{\Phi_s}{R_dT_{st}}\right] - \frac{1}{\Pi_{\mathrm{s}}}g\left[F_{\mathrm{m}}\right]_{\eta=1}\right]d\eta \tag{16}$$

Details leading to this formulation is given in part 6.2.3 of the documentation (IDEUL) about model equations and Eulerian dynamics.

Variable δ_{TR} is 0 or 1; when $\delta_{TR}=1$ the new variable is less sensitive to the orography (new variable proposed by Ritchie and Tanguay (1996) to reduce orographic resonance).

If one assumes that a volume of air occupied by rainfall drops is not replaced by dry air when drops are falling (case $\delta m=0$, variable **NDPSFI** is 0 in **NAMPHY**), F_{m} is replaced by zero and $\left[\dot{\eta}\frac{\partial\Pi}{\partial\eta}\right]_{\eta=1}$ is equal to zero.

If one assumes that a volume of air occupied by rainfall drops is replaced by dry air when drops are falling (case $\delta m=1$, variable **NDPSFI** is 1 in **NAMPHY**), F_{m} (diabatic flux) has to be taken in account and $\left[\dot{\eta}\frac{\partial\Pi}{\partial\eta}\right]_{\eta=1}$ is non-zero (more details are given in documentation (IDEUL)).

$\left[\dot{\eta}\frac{\partial\Pi}{\partial\eta}\right]_{\eta=0}$ is non-zero only when there is an upper radiative boundary condition (**LRUBC**=.TRUE.).

Advectable GFL variables: moisture, but also for example ozone, liquid water, ice, cloud fraction, TKE, aerosols and extra GFL variables equations.

Equation is written for moisture q, and is the same for the other advectable GFL variables.

$$\frac{dq}{dt} = F_{\rm q} \tag{17}$$

$F_{\rm q}$ is the physical contribution on moisture.

Non advectable pseudo-historic GFL variables:

Equation is written for rain $q_{\rm r}$, and is the same for the other non-advectable GFL variables. Since there is no advection, the SL equation is identical to the Eulerian equation.

$$\frac{\partial q_{\rm r}}{\partial t} = F_{q_{\rm r}} \tag{18}$$

4.3 The deep layer 3D primitive equation model according to White and Bromley, 1995: Lagrangian formulation.

Deep layer equations: basics and new features.

The following modifications are done, according to (White and Bromley, 1995):

- One takes account to the fact, that the distance to the Earth center is no longer a but a radius varying with the vertical. For conveniency (with the η vertical coordinate), one approximates the radius by a pseudo-radius r_s which depends only on the hydrostatic pressure Π. Two vertical lines are no longer parallel, so the section of a vertical column varies with the hydrostatic pressure.
- The vertical velocity is now taken in account in the Coriolis term through a pseudo-vertical velocity W defined by $W = \frac{dr_s}{dt}$. W also appears in some new curvature terms.
- The total geopotential Φ, which appears in the RHS of the wind equation, is no longer equal to the geopotential height gz.

Details about deep layer equations (definition and expression of r_s, W, geopotential relationships) is given in the documentation (IDEUL) about model equations and Eulerian dynamics and is not detailed again here. All calculations giving the below formulation of primitive equations is also given in the documentation (IDEUL).

Momentum equation.

Coriolis force can be treated explicitly ($\delta_{\vec{V}}$=0) or implicitly ($\delta_{\vec{V}}$=1) in the Lagrangian equation.

$$\frac{d\left(\vec{V} + \delta_{\vec{V}}(2\vec{\Omega} \wedge \vec{r})\right)}{dt} = (1-\delta_{\vec{V}})(-2\mathbf{\Omega}\wedge\mathbf{V}-2\mathbf{\Omega}\wedge W\mathbf{k}) - \frac{W}{r_s}\mathbf{V} - \nabla\Phi - (RT+\mu_{\rm s}R_{\rm d}T_{\rm r})\nabla(\log\Pi) + \mathbf{F_V} \tag{19}$$

$\mathbf{F_V}$ is the physical contribution on horizontal wind. See documentation (IDEUL) for definition of Φ and $\mu_{\rm s}$.

Thermodynamic equation.

Lagrangian tendency:

$$\frac{d\left(T+\delta_{TR}\frac{\alpha_{\mathrm{T}}\Phi_{\mathrm{s}}}{R_{\mathrm{d}}T_{ST}}\right)}{dt}=\frac{d\left(\delta_{TR}\frac{\alpha_{\mathrm{T}}\Phi_{\mathrm{s}}}{R_{\mathrm{d}}T_{ST}}\right)}{dt}+\frac{RT}{c_{\mathrm{p}}}\frac{\omega}{\Pi}+F_{\mathrm{T}} \tag{20}$$

F_{T} is the physical contribution on temperature. When $\delta_{TR}=1$ the Eulerian treatment of orography is applied and the prognostic variable is replaced by one variable less sensitive to the surface orography. This modification has been proposed by Ritchie and Tanguay (1996).

This equation is unchanged compared to its expression in the thin layer equations. The only change is the diagnostic expression of ω (see documentation (IDEUL) for details about expression and discretisation of ω).

Continuity equation.

The equation which is discretised is the vertically integrated Lagrangian formulation of continuity equation.

$$\begin{gathered}\int_{\eta=0}^{\eta=1}\frac{{r_s}^2}{a^2}\frac{\partial B}{\partial\eta}\frac{d\left[\log\Pi_{\mathrm{s}}+\delta_{TR}\frac{\Phi_s}{R_dT_{st}}\right]}{dt}d\eta= \\ \int_{\eta=0}^{\eta=1}\frac{{r_s}^2}{a^2}\frac{\partial B}{\partial\eta}\frac{\partial\log(\Pi_{\mathrm{s}})}{\partial t}d\eta+\int_{\eta=0}^{\eta=1}\frac{{r_s}^2}{a^2}\frac{\partial B}{\partial\eta}\frac{a}{r_s}\mathbf{V}\left[\frac{r_s}{a}\nabla\right]\left[\log\Pi_{\mathrm{s}}+\delta_{TR}\frac{\Phi_s}{R_dT_{st}}\right]d\eta\end{gathered} \tag{21}$$

where $\frac{\partial\log(\Pi_{\mathrm{s}})}{\partial t}$ is given by equation:

$$\begin{gathered}\frac{\partial\log(\Pi_{\mathrm{s}})}{\partial t}=-\left[\frac{a^2}{{r_s}^2}\right]_{\eta=1}\frac{1}{\Pi_{\mathrm{s}}}\int_{\eta=0}^{\eta=1}\left[\frac{r_s}{a}\nabla\right]\left(\frac{r_s}{a}\mathbf{V}\frac{\partial\Pi}{\partial\eta}\right)d\eta \\ -\frac{1}{\Pi_{\mathrm{s}}}\left[\dot{\eta}\frac{\partial\Pi}{\partial\eta}\right]_{\eta=1}+\frac{1}{\Pi_{\mathrm{s}}}\left[\frac{a^2}{{r_s}^2}\right]_{\eta=1}\left[\frac{{r_s}^2}{a^2}\right]_{\eta=0}\left[\dot{\eta}\frac{\partial\Pi}{\partial\eta}\right]_{\eta=0}-\frac{1}{\Pi_{\mathrm{s}}}g\left[F_{\mathrm{m}}\right]_{\eta=1}\left[\frac{a^2}{{r_s}^2}\right]_{\eta=1}\end{gathered} \tag{22}$$

Advectable GFL variables.

These equations are identical to the thin layer model ones.

Non advectable pseudo-historic GFL variables.

These equations are identical to the thin layer model ones.

5 Discretisation of the equations: general aspects.

This section does not describe in detail iterative centred-implicit schemes and describes only non-iterative schemes in detail. For more details about iterative centred-implicit schemes, see documentation (IDSI).

5.1 Notations.

$*$ Upper index:

- First integration step: $+$ (resp. m, o, $-$) for $t+\Delta t$ (resp. $t+0.5\Delta t$, t, t) quantity.
- Following integration steps: $+$ (resp. m, o, $-$) for $t+\Delta t$ (resp. $t+0.5\Delta t$, t, $t-\Delta t$) quantity.

$*$ Lower index:

F (resp. M and O) for final (resp. medium and origin) point.

$*$ Particular case of the first timestep in a SL3TL scheme:

Written discretisations are valid from the second integration step. Δt has to be replaced by $\frac{\Delta t}{2}$ for the first integration step (in this case the $t-\Delta t$ quantities are equal to the t quantities).

$*$ 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, temperature (and the two additional non-hydrostatic variables in a non-hydrostatic model). The sub-class of thermodynamic variables includes T, and the two additional non-hydrostatic variables in a non-hydrostatic model. There are **NFTHER** thermodynamic variables.
- 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 includes for example humidity, liquid water, ice, cloud fraction, ozone, TKE, aerosols, and some extra fields. See documentation (IDEUL) for a comprehensive list of advectable GFL.
- 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 includes for example rain, snow, graupels, hail, convective precipitation flux, stratiform precipitation flux, SRC (second-order flux for AROME), forcings, easy diagnostics, greenhouse gases, reactive gases, moisture convergence, total humidity variation, standard deviation of the saturation depression, convective vertical velocity. See documentation (IDEUL) for a comprehensive list of non advectable pseudo-historic GFL.
- 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).

5.2 Discretisation for a 3D variable in a 3D model: general case where the RHS has non-zero linear and non-linear terms (GMV variables).

List of equations.

- Momentum equation.
- Temperature equation.
- Pressure departure variable (non-hydrostatic model only).
- Vertical divergence variable (non-hydrostatic model only).

Generic notations.

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

- **NWLAG** for momentum equation.
- **NTLAG** for temperature equation.
- **NSPDLAG** for pressure departure variable (non-hydrostatic model only).
- **NSVDLAG** for vertical divergence (non-hydrostatic model only).

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.
- **PSPDL0**, **PSPDL9**, **PSPDT1** for $\hat{Q}$ equation.
- **PSVDL0**, **PSVDL9**, **PSVDT1** for d equation.

Generic notation **P(X)NLT9** stands for:

- **PUNLT9** for U-momentum equation.
- **PVNLT9** for V-momentum equation.
- **PTNLT9** for temperature equation.
- **PSPDNLT9** for $\hat{Q}$ equation.
- **PSVDNLT9** for d equation.

Generic notation for total term, linear term, non linear term, physics: $\mathcal{A}$ is the total term (sum of dynamical contributions), $\mathcal{B}$ is the linear term (treated in the semi-implicit scheme), the difference $\mathcal{A} - \beta\mathcal{B}$ is the non-linear term. $\mathcal{F}$ is the sum of contributions computed in the physical parameterizations.

Description stands for the general case where linear and non-linear terms are gathered in the same buffer, and where no additional splitting is required to do diagnostics or to apply SLHD interpolations to a subset of the terms interpolated by a high-order interpolation. In some particular cases, additional splitting involving separate buffers may be required:

- In the NH model for options where linear terms must be separately interpolated (controlled by variables **LSLINL**, **LSLINLC1** and **LSLINLC2**). Linear terms are stored in arrays, the name of them has appendix **_SI**, or in parts of **PB1** with pointers, the name of them has appendix **_SI**.

- Separation between linear and non-linear terms may have to be done for some DDH diagnostics too (if **LRSIDDH**=T).
- When some diagnostics (for example DDH) impose that evaluation of the dynamics and of the physics at the origin point of the SL trajectory must be done separately in two different buffers (controlled by variable **NSPLTHOI** set to -1). Buffer **P(X)LF9** is used instead of **P(X)L9** to store quantities, the interpolation of which is a non-SLHD high-order one.
- When SLHD interpolations, if switched on, are applied only to a subset of the quantities interpolated by a high order interpolation (applied to $X(t)$ or $X(t-\Delta t)$ but not to the other terms); in particuliar physics do not use a SLHD interpolation in this case (controlled by variable **NSPLTHOI** set to 1). Buffer **P(X)LF9** is used instead of **P(X)L9** to store quantities, the interpolation of which is a non-SLHD high-order one.

Case **N(X)LAG**=4 is not described in detail: in this case diabatic terms are interpolated by trilinear interpolations and they enter the buffer containing terms interpolated by trilinear interpolations. Start from discretisations given for **N(X)LAG**=3 and move term $\Delta t\mathcal{F}$ from **P(X)L9** to **P(X)L0**.

Other points.

∗ **High-order interpolations:** In the following discretisations, “high-order interpolations" means 32 points interpolations for 3D terms (vertical interpolations are cubic), 12 points interpolations for 2D terms.

∗ **Uncentering:** ϵ is a first-order “uncentering factor". It allows to remove the noise due to gravity waves (orographic resonance).

∗ **Vectorial variables:** The following discretisations are written for scalar variables. For vectorial variables (for example the 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 $\mathcal{R}^{OF}\{\text{this expression}\}_O$.
- expression interpolated at M has to be replaced by $\mathcal{R}^{MF}\{\text{this expression}\}_M$.

3TL vertical interpolating SL scheme.

Equation

$$\frac{dX}{dt} = \mathcal{A} + \mathcal{F} \tag{23}$$

is discretised as follows:

$$(X - (1+\epsilon)\Delta t\beta\mathcal{B})^+_F = \{X^- + (1-\epsilon)\Delta t[\mathcal{A} - \beta\mathcal{B}]^o + [(1-\epsilon)\Delta t\beta\mathcal{B} + 2\Delta t\mathcal{F}]^-\}_O \\ +\{(1+\epsilon)\Delta t[\mathcal{A} - \beta\mathcal{B}]^o\}_F \tag{24}$$

Buffers content before interpolations for **N(X)LAG**=2:

- **P(X)L0** is not used.
- **P(X)L9**: $X^- + (1-\epsilon)\Delta t[\mathcal{A} - \beta\mathcal{B}]^o + [(1-\epsilon)\Delta t\beta\mathcal{B} + 2\Delta t\mathcal{F}]^-$ for high-order interpolation at the origin point O.
- **P(X)T1**: $(1+\epsilon)\Delta t[\mathcal{A} - \beta\mathcal{B}]^o$ then provisional add of quantity $[(1+\epsilon)\Delta t\beta\mathcal{B}]^o$ before t+dt physics; evaluated at the final point F.

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

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

2TL vertical interpolating SL scheme: conventional discretisation (LSETTLS=.F.) and first-order uncentering.

The $t + \frac{\Delta t}{2}$ non-linear term $\mathcal{A}^m - \beta\mathcal{B}^m$ used in the 2TL SL scheme is computed by a linear temporal extrapolation using the t and $t - \Delta t$ quantities at the same location. At the first time integration step, values at $t + \frac{\Delta t}{2}$ are set equal to initial values. This discretisation of the 2TL SL scheme follows (Mc Donald and Haugen, 1992). Quantity $\mathcal{A}^o - \beta\mathcal{B}^o$ has to be saved in a buffer **P(X)NLT9** to be available as $\mathcal{A}^- - \beta\mathcal{B}^-$ for the following timestep.

Equation (10) is discretised as follows:

$$\begin{aligned}(X - (1+\epsilon)\tfrac{\Delta t}{2}\beta\mathcal{B})^+_F = \{X^o + (1-\epsilon)\tfrac{\Delta t}{2}[\mathcal{A} - \beta\mathcal{B}]^m + [(1-\epsilon)\tfrac{\Delta t}{2}\beta\mathcal{B} + \Delta t\mathcal{F}]^o\}_O \\ +\{(1+\epsilon)\tfrac{\Delta t}{2}[\mathcal{A} - \beta\mathcal{B}]^m\}_F\end{aligned} \tag{25}$$

which can be rewritten, once expanded the extrapolation:

$$\begin{aligned}(X - (1+\epsilon)\tfrac{\Delta t}{2}\beta\mathcal{B})^+_F = [X^o + \tfrac{\Delta t}{2}(1-\epsilon)\mathcal{A}^o + \Delta t\mathcal{F}^o]_O \\ +0.5(1-\epsilon)\tfrac{\Delta t}{2}[(\mathcal{A} - \beta\mathcal{B})^o - (\mathcal{A} - \beta\mathcal{B})^-]_O + (1+\epsilon)\tfrac{\Delta t}{2}[1.5(\mathcal{A} - \beta\mathcal{B})^o - 0.5(\mathcal{A} - \beta\mathcal{B})^-]_F\end{aligned} \tag{26}$$

Buffers content before interpolations for **N(X)LAG**=2:

- **P(X)L0** is not used.
- **P(X)L9**: $X^o + \frac{\Delta t}{2}(1-\epsilon)\mathcal{A}^o + 0.5(1-\epsilon)\frac{\Delta t}{2}[(\mathcal{A} - \beta\mathcal{B})^o - (\mathcal{A} - \beta\mathcal{B})^-] + \Delta t\mathcal{F}^o$ for high-order interpolation at the origin point O.
- **P(X)T1**: $(1+\epsilon)\frac{\Delta t}{2}[1.5(\mathcal{A} - \beta\mathcal{B})^o - 0.5(\mathcal{A} - \beta\mathcal{B})^-]$ then provisional add of quantity $[(1+\epsilon)\frac{\Delta t}{2}\beta\mathcal{B}]^o$ before t+dt physics; evaluated at the final point F.

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

- **P(X)L0**: $\frac{\Delta t}{2}(1-\epsilon)\mathcal{A}^o + 0.5(1-\epsilon)\frac{\Delta t}{2}[(\mathcal{A} - \beta\mathcal{B})^o - (\mathcal{A} - \beta\mathcal{B})^-]$ for trilinear interpolation at the origin point O.
- **P(X)L9**: $X^o + [\Delta t\mathcal{F}]^o$ for high-order interpolation at the origin point O.
- **P(X)T1**: $(1+\epsilon)\frac{\Delta t}{2}[1.5(\mathcal{A} - \beta\mathcal{B})^o - 0.5(\mathcal{A} - \beta\mathcal{B})^-]$ then provisional add of quantity $[(1+\epsilon)\frac{\Delta t}{2}\beta\mathcal{B}]^o$ before t+dt physics; evaluated at the final point F.

2TL vertical interpolating SL scheme: conventional discretisation (LSETTLS=.F.) and pseudo-second order uncentering.

One starts to remove uncentering ϵ from the nonlinear terms and to apply a second-order uncentering $\epsilon_{\mathcal{X}}$ to linear terms, that yelds a term $\mathcal{B}^-$ in the discretisation. From property:

$$\mathcal{B}^- = 3\mathcal{B}^o - 2\mathcal{B}^m \qquad (27)$$

one can remove term $\mathcal{B}^-$ and show that discretisation is equivalent to replace β by $(1+\epsilon_{\mathcal{X}})\beta$. For more details, see part 5 (equations (37) and (38)) of (Simmons and Temperton, 1996). In equation (25), uncentering ϵ has to be replaced by zero, and β has to be replaced by $(1+\epsilon_{\mathcal{X}})\beta$. The $t+\frac{\Delta t}{2}$ non-linear term $\mathcal{A}^m - (1+\epsilon_{\mathcal{X}})\beta\mathcal{B}^m$ used in the 2TL SL scheme is computed by a linear temporal extrapolation using the t and $t-\Delta t$ quantities at the same location. At the first time integration step, values at $t+\frac{\Delta t}{2}$ are set equal to initial values and second-order uncentering is replaced by a first order uncentering. This discretisation of the 2TL SL scheme follows (Mc Donald and Haugen, 1992). Quantity $\mathcal{A}^o - (1+\epsilon_{\mathcal{X}})\beta\mathcal{B}^o$ has to be saved in a buffer **P(X)NLT9** to be available as $\mathcal{A}^- - (1+\epsilon_{\mathcal{X}})\beta\mathcal{B}^-$ for the following timestep.

Equation (10) is discretised as follows:

$$\begin{aligned}(X - (1+\epsilon_{\mathcal{X}})\tfrac{\Delta t}{2}\beta\mathcal{B})^+_F = {} & \{X^o + \tfrac{\Delta t}{2}[\mathcal{A} - (1+\epsilon_{\mathcal{X}})\beta\mathcal{B}]^m + [(1+\epsilon_{\mathcal{X}})\tfrac{\Delta t}{2}\beta\mathcal{B} + \Delta t\mathcal{F}]^o\}_O \\ & + \{\tfrac{\Delta t}{2}[\mathcal{A} - (1+\epsilon_{\mathcal{X}})\beta\mathcal{B}]^m\}_F \qquad (28)\end{aligned}$$

which can be rewritten, once expanded the extrapolation:

$$\begin{aligned}(X - (1+\epsilon_{\mathcal{X}})\tfrac{\Delta t}{2}\beta\mathcal{B})^+_F = {} & [X^o + \tfrac{\Delta t}{2}\mathcal{A}^o + \Delta t\mathcal{F}^o]_O + 0.5\tfrac{\Delta t}{2}[(\mathcal{A} - (1+\epsilon_{\mathcal{X}})\beta\mathcal{B})^o - (\mathcal{A} - (1+\epsilon_{\mathcal{X}})\beta\mathcal{B})^-]_O \\ & + \tfrac{\Delta t}{2}[1.5(\mathcal{A} - (1+\epsilon_{\mathcal{X}})\beta\mathcal{B})^o - 0.5(\mathcal{A} - (1+\epsilon_{\mathcal{X}})\beta\mathcal{B})^-]_F \qquad (29\end{aligned}$$

Buffers content before interpolations for **N(X)LAG**=2:

- **P(X)L0** is not used.
- **P(X)L9**: $X^o + \frac{\Delta t}{2}\mathcal{A}^o + 0.5\frac{\Delta t}{2}[(\mathcal{A} - (1+\epsilon_{\mathcal{X}})\beta\mathcal{B})^o - (\mathcal{A} - (1+\epsilon_{\mathcal{X}})\beta\mathcal{B})^-] + \Delta t\mathcal{F}^o$ for high-order interpolation at the origin point O.
- **P(X)T1**: $\frac{\Delta t}{2}[1.5(\mathcal{A} - (1+\epsilon_{\mathcal{X}})\beta\mathcal{B})^o - 0.5(\mathcal{A} - (1+\epsilon_{\mathcal{X}})\beta\mathcal{B})^-]$ then provisional add of quantity $[(1+\epsilon_{\mathcal{X}})\frac{\Delta t}{2}\beta\mathcal{B}]^o$ before t+dt physics; evaluated at the final point F.

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

- **P(X)L0**: $\frac{\Delta t}{2}\mathcal{A}^o + 0.5\frac{\Delta t}{2}[(\mathcal{A} - (1+\epsilon_{\mathcal{X}})\beta\mathcal{B})^o - (\mathcal{A} - (1+\epsilon_{\mathcal{X}})\beta\mathcal{B})^-]$ for trilinear interpolation at the origin point O.
- **P(X)L9**: $X^o + [\Delta t\mathcal{F}]^o$ for high-order interpolation at the origin point O.
- **P(X)T1**: $\frac{\Delta t}{2}[1.5(\mathcal{A} - (1+\epsilon_{\mathcal{X}})\beta\mathcal{B})^o - 0.5(\mathcal{A} - (1+\epsilon_{\mathcal{X}})\beta\mathcal{B})^-]$ then provisional add of quantity $[(1+\epsilon_{\mathcal{X}})\frac{\Delta t}{2}\beta\mathcal{B}]^o$ before t+dt physics; evaluated at the final point F.

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

The $t + \frac{\Delta t}{2}$ non-linear term $\mathcal{A}^m - \beta\mathcal{B}^m$ used in the 2TL SL scheme if **LSETTLS**=.F. is replaced in the case **LSETTLS**=.T. by a linear spatio-temporal extrapolation comparable to the one applied to the wind components for the research of trajectory (see formula (81)), except the fact that there is an additional uncentering. Term $\mathcal{A}^m - \beta\mathcal{B}^m$ is replaced by

$$0.5(1+\epsilon)[\mathcal{A}^o - \beta\mathcal{B}^o]_F + 0.5(2-\epsilon)[\mathcal{A}^o - \beta\mathcal{B}^o]_O - 0.5[\mathcal{A}^- - \beta\mathcal{B}^-]_O$$

This type of extrapolation is available only for **N(X)LAG**=3. At the first time integration step, values at $t + \frac{\Delta t}{2}$ are set equal to initial values. Quantity $\mathcal{A}^o - \beta\mathcal{B}^o$ has to be saved in a buffer **P(X)NLT9** to be available as $\mathcal{A}^- - \beta\mathcal{B}^-$ for the following timestep.

Equation (10) is discretised as follows:

$$(X - (1+\epsilon)\tfrac{\Delta t}{2}\beta\mathcal{B})^+_F = \{X^o + (2-\epsilon)\tfrac{\Delta t}{2}[\mathcal{A} - \beta\mathcal{B}]^o - \tfrac{\Delta t}{2}[\mathcal{A} - \beta\mathcal{B}]^- + [(1-\epsilon)\tfrac{\Delta t}{2}\beta\mathcal{B} + \Delta t\mathcal{F}]^o\}_O$$
$$+\{(1+\epsilon)\tfrac{\Delta t}{2}[\mathcal{A} - \beta\mathcal{B}]^o\}_F \quad (30)$$

which can be rewritten, once expanded the extrapolation:

$$(X - (1+\epsilon)\tfrac{\Delta t}{2}\beta\mathcal{B})^+_F = [X^o + \tfrac{\Delta t}{2}(1-\epsilon)\mathcal{A}^o + \Delta t\mathcal{F}^o]_O$$
$$+\tfrac{\Delta t}{2}[(\mathcal{A} - \beta\mathcal{B})^o - (\mathcal{A} - \beta\mathcal{B})^-]_O + [\tfrac{\Delta t}{2}(1+\epsilon)(\mathcal{A} - \beta\mathcal{B})^o]_F \quad (31)$$

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

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

2TL vertical interpolating SL scheme: stable discretisation (LSETTLS=.T.) and pseudo-second order uncentering.

In equation (30), uncentering ϵ has to be replaced by zero, and β has to be replaced by $(1+\epsilon_{\mathcal{X}})\beta$. At the first time integration step, values at $t + \frac{\Delta t}{2}$ are set equal to initial values and second-order uncentering is replaced by a first order uncentering. Quantity $\mathcal{A}^o - (1+\epsilon_{\mathcal{X}})\beta\mathcal{B}^o$ has to be saved in a buffer **P(X)NLT9** to be available as $\mathcal{A}^- - (1+\epsilon_{\mathcal{X}})\beta\mathcal{B}^-$ for the following timestep.

Equation (10) is discretised as follows:

$$(X - (1+\epsilon_{\mathcal{X}})\tfrac{\Delta t}{2}\beta\mathcal{B})^+_F$$
$$= \{X^o + \Delta t[\mathcal{A} - (1+\epsilon_{\mathcal{X}})\beta\mathcal{B}]^o - \tfrac{\Delta t}{2}[\mathcal{A} - (1+\epsilon_{\mathcal{X}})\beta\mathcal{B}]^- + [(1+\epsilon_{\mathcal{X}})\tfrac{\Delta t}{2}\beta\mathcal{B} + \Delta t\mathcal{F}]^o\}_O$$
$$+\{\tfrac{\Delta t}{2}[\mathcal{A} - (1+\epsilon_{\mathcal{X}})\beta\mathcal{B}]^o\}_F \quad (32)$$

which can be rewritten, once expanded the extrapolation:

$$(X - (1+\epsilon_{\mathcal{X}})\tfrac{\Delta t}{2}\beta\mathcal{B})^+_F = [X^o + \tfrac{\Delta t}{2}\mathcal{A}^o + \Delta t\mathcal{F}^o]_O$$
$$+\tfrac{\Delta t}{2}[(\mathcal{A} - (1+\epsilon_{\mathcal{X}})\beta\mathcal{B})^o - (\mathcal{A} - (1+\epsilon_{\mathcal{X}})\beta\mathcal{B})^-]_O + [\tfrac{\Delta t}{2}(\mathcal{A} - (1+\epsilon_{\mathcal{X}})\beta\mathcal{B})^o]_F \quad (33)$$

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

- **P(X)L0**: $\frac{\Delta t}{2}\mathcal{A}^o + \frac{\Delta t}{2}[(\mathcal{A} - (1+\epsilon_{\mathcal{X}})\beta\mathcal{B})^o - (\mathcal{A} - (1+\epsilon_{\mathcal{X}})\beta\mathcal{B})^-]$ for trilinear interpolation at the origin point O.
- **P(X)L9**: $X^o + [\Delta t\mathcal{F}]^o$ for high-order interpolation at the origin point O.
- **P(X)T1**: $\frac{\Delta t}{2}(\mathcal{A} - (1+\epsilon_{\mathcal{X}})\beta\mathcal{B})^o$ then provisional add of quantity $[(1+\epsilon_{\mathcal{X}})\frac{\Delta t}{2}\beta\mathcal{B}]^o$ before t+dt physics; evaluated at the final point F.

Specific treatment for some options in the vertical divergence equation.

$*$ **Option LGWADV=.T. .** When this option is activated, the SL scheme treats the Lagrangian equation of w instead of the one of d (or d_4). That implies the following steps in the code:

- Change of variable from d or d_4 to w.
- SL explicit treatment of the w equation. For finite difference vertical discretisation (**LVFE_GW**=F) w is given at half levels: that means that the SL trajectory must be computed also for trajectories ending at half levels. For finite element vertical discretisation (**LVFE_GW**=T) w is given at full levels. There is a specific treatment for $w_{\rm surf}$ (diagnostic condition at $t+\Delta t$).
- Calculation of the linear terms of the d equation for the semi-implicit scheme.
- Conversion from w^+ into d^+ or ${d_4}^+$. If **LVFE_GW**=T, this conversion is done by applying the vertical derivation to the temporal increment of w to obtain the temporal increment of d.
- Add the linear terms to d^+ or ${d_4}^+$.

It is necessary to keep the prognostic variable d or d_4 in the linear model: w in the linear model would lead to linear instabilities.

This option allows to remove spurious chimneys above slopes, and also spurious noise in some “bubble" tests.

$*$ **Option NVDVAR=4.** In this case we must discretize the term $\frac{d\mathtt{X}}{dt}$, and this is not done by a SL treatment of the equation of X (the RHS of this equation is not easy to compute).
If there is no predictor-corrector scheme activated, this term is diagnosed as follows:

- Three-time level semi-Lagrangian scheme:
$$\frac{d\mathtt{X}}{dt} = \frac{0.5\mathtt{X}_F^o + 0.5\mathtt{X}_O^o - \mathtt{X}_O^-}{\Delta t} \quad (34)$$
- Two-time level semi-Lagrangian scheme:
$$\frac{d\mathtt{X}}{dt} = \frac{\mathtt{X}_M^m - \mathtt{X}_O^o}{0.5\Delta t} \quad (35)$$
$\mathtt{X}_M^m$ is a symbolic denotation for the extrapolated $t+0.5\Delta t$ value of X. The way of extrapolating depends on **LSETTLS** and is the same as for the other terms evaluated at $t+0.5\Delta t$ (there is no interpolation at M).

If **LPC_FULL**=.T. and **LPC_NESC**=.F. these discretisations also apply to the predictor step; in this case there are two options:

- Option 1 (**ND4SYS**=1): include all the contributions of the evolution of X in the predictor-corrector scheme iterations.

- Option 2 (**ND4SYS**=2): include only the advective contributions evolution of `X` in the predictor-corrector scheme iterations; after the last step of the corrector step, include the non advective processes.

In some cases (ECMWF configurations with lagged physics) option 1 may generate instabilities above slopes.

∗ Assumptions at the surface (velocities). Calculation of $\frac{dd}{dt}$ for the layer $l = L$ requires the calculation of $\frac{dw_{\rm surf}}{dt}$. There are two options to compute $\frac{dw_{\rm surf}}{dt}$, controlled by the variable **LRDBBC** of **NAMDYN**.

- For **LRDBBC**=.F., one simply discretizes equation (**??**) with a SL treatment. The RHS of this equation contains $\frac{d\mathbf{V}_{\rm surf}}{dt}$ and $\mathbf{V}_{\rm surf}$, and the assumptions currently done about these quantities are:

  . $\mathbf{V}_{\rm surf} = \mathbf{V}_{l=L}$

  . $\frac{d\mathbf{V}_{\rm surf}}{dt} = \left[\frac{d\vec{V}}{dt}\right]_{l=L,{\rm adiab}}$, with an explicit treatment of the Coriolis term, even if **LADVF**=.T. or **LIMPF**=.T. .

- An alternate discretisation (option **LRDBBC**=.T.) is to evaluate $\frac{dw_{\rm surf}}{dt}$ by:

  SL3TL:

  $$\frac{dw_{\rm surf}}{dt} = \frac{w_F^+ - w_O^-}{2\Delta t} \tag{36}$$

  SL2TL:

  $$\frac{dw_{\rm surf}}{dt} = \frac{w_F^+ - w_O^o}{\Delta t} \tag{37}$$

  which requires additional interpolations to compute w_O.

  This option allows to remove spurious chimneys above slopes.

  w^+ is computed by:

  $$w^+ = \mathbf{V}^+_{l=L,{\rm prov}} \nabla\Phi_{\rm s}$$

  where $\mathbf{V}^+_{l=L,{\rm prov}}$ is the provisional $t + \Delta t$ value of $\mathbf{V}_{l=L}$ computed just after the interpolations (this is this provisional value which is used as input for the lagged physics).

  The treatment of $\frac{dd}{dt}$ for the layer $l = L$ is done according to the following steps:

  Calculation of $\left[\frac{dd}{dt}\right]_{l=L,{\rm lrdbbc=F}}$ (as if **LRDBBC** were .F.), $\left[\frac{dw_{\rm surf}}{dt}\right]_{\rm lrdbbc=F}$ and $\left[\frac{dw_{\rm surf}}{dt}\right]_{\rm lrdbbc=T}$.

  Calculation of $\left[\frac{dd}{dt}\right]_{l=L,{\rm lrdbbc=T}}$ by the following formula:

  $$\left[\frac{dd}{dt}\right]_{l=L,{\rm lrdbbc=T}} = \left[\frac{dd}{dt}\right]_{l=L,{\rm lrdbbc=F}} - \left(\left[\frac{dw_{\rm surf}}{dt}\right]_{\rm lrdbbc=T} - \left[\frac{dw_{\rm surf}}{dt}\right]_{\rm lrdbbc=F}\right)\left[\frac{gp}{R_{\rm d}T\Delta\Pi}\right]^o_{l=L,MM} \tag{38}$$

  where MM is the decentered middle of O and F (that requires a specific interpolation of $\left[\frac{gp}{R_{\rm d}T\Delta\Pi}\right]_{l=L}$ at O).

  For more details about these calculations and the additional interpolations required, see documentation (IDVNH3), especially the parts 1 to 3.

  The above formulae are valid for thin layer equations. For WS2003 deep-layer equations change Π into $\tilde{\Pi}$, g into G, w into $({r_s}^2/a^2)w$ in the above formulae.

5.3 Discretisation for a 3D variable in a 3D model: particular case where the RHS has zero linear and non-linear terms (advectable GFL variables).

List of equations.

Humidity equation, and for example:

- Ozone equation.
- Liquid water equation.
- Ice equation.
- Cloudiness equation.
- TKE.
- Aerosols equation.
- Extra GFL variables equations.

See documentation (IDEUL) for a comprehensive list of advectable GFL variables.

Generic notations.

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

- **PGFLL9**, **PGFLT1** for GFL variables.

Generic notation for total term, linear term, non linear term, physics: $\mathcal{A}$ is the total term (sum of dynamical contributions), $\mathcal{B}$ is the linear term (treated in the semi-implicit scheme), the difference $\mathcal{A} - \beta\mathcal{B}$ is the non-linear term. $\mathcal{F}$ is the sum of contributions computed in the physical parameterizations. In the present case $\mathcal{A}$ and $\mathcal{B}$ are equal to zero.

Description stands for the general case where diabatic and adiabatic terms are gathered in the same buffer, and where no additional splitting is required to do diagnostics or to apply SLHD interpolations to a subset of the terms interpolated by a high-order interpolation. In some particular cases, additional splitting involving separate buffers may be required:

- When some diagnostics (for example DDH) impose that evaluation of the dynamics and of the physics at the origin point of the SL trajectory must be done separately in two different buffers (controlled by variable **NSPLTHOI** set to -1). Buffer **P(X)LF9** is used instead of **P(X)L9** to store quantities, the interpolation of which is a non-SLHD high-order one.
- When diabatic terms use a different interpolation from the adiabatic one. That can be the case for non-zero values of **NSPLTHOI**, attribute **LPHYLIN** set to T. Buffer **P(X)LF9** is used instead of **P(X)L9** to store diabatic terms to be interpolated.

Other points.

$*$ **High-order interpolations:** In the following discretisations, “high-order interpolations" means: 32 points interpolations for 3D terms (vertical interpolations are cubic), 12 points 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%LVSPLIP** in **NAMGFL**).

$*$ **Uncentering:** ϵ is a first-order "uncentering factor". It allows to remove the noise due to gravity waves (orographic resonance).

3TL vertical interpolating SL scheme.

Equation (10) is discretised as follows:

$$X_F^+ = \{X^- + [2\Delta t\mathcal{F}]^-\}_O \tag{39}$$

Buffers content before interpolations:

- **P(X)L9**: $X^- + [2\Delta t\mathcal{F}]^-$ for high-order interpolation at the origin point O.
- **P(X)T1** contains zero; evaluated at the final point F.

2TL vertical interpolating SL scheme.

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

Equation (10) is discretised as follows:

$$X_F^+ = \{X^o + [\Delta t\mathcal{F}]^o\}_O \tag{40}$$

Buffers content before interpolations:

- **P(X)L9**: $X^o + [\Delta t\mathcal{F}]^o$ for high-order interpolation at the origin point O.
- **P(X)T1** contains zero; evaluated at the final point F.

$*$ **Remark for iterative centred-implicit schemes:** For options where this type of scheme involves the momentum equation (this is the case for the option **LPC_FULL**=.T.) X_F^+ has to be recomputed at all iterations of the iterative centred-implicit scheme since the origin point O is recomputed at each iteration.

Non advectable pseudo-historic GFL variables.

For these variables the discretization always writes (3TL SL and 2TL SL):

$$X_F^+ = \{X^o + [\Delta t\mathcal{F}]^o\}_F \tag{41}$$

and there are never interpolations.

5.4 Discretisation for a 2D variable in a 3D model (GMVS variables, for example continuity equation).

The equation which is now discretised is:

$$[\mathcal{R}_{inte}]_{(top,surf)} \left\langle \frac{\mathcal{W}_{\text{vei}}}{\Delta\eta} \frac{dX}{dt} \right\rangle = [\mathcal{R}_{inte}]_{(top,surf)} \left\langle \frac{\mathcal{W}_{\text{vei}}}{\Delta\eta} \mathcal{A} \right\rangle + [\mathcal{R}_{inte}]_{(top,surf)} \left\langle \frac{\mathcal{W}_{\text{vei}}}{\Delta\eta} \mathcal{F} \right\rangle \quad (42)$$

where:

$$[\mathcal{R}_{inte}]_{(top,surf)} \left\langle \frac{\mathcal{W}_{\text{vei}}}{\Delta\eta} \right\rangle = 1 \quad (43)$$

and $[\mathcal{R}_{inte}]_{(top,surf)}$ is the vertical integral matricial operator (the scalar product $[\mathcal{R}_{inte}]_{(top,surf)}\langle X\rangle$ is the discretisation of $\int_{\eta=0}^{\eta=1} X d\eta$, $\langle X\rangle$ is the vector containing the layer values of X: $(X_1; X_2; ...; X_l; ...; X_L)$).

In the thin layer equations or in the Wood and Staniforth formulation of deep-layer equations, expression of $\mathcal{W}_{\text{vei}}$ at full levels is:

$$[\mathcal{W}_{\text{vei}}]_l = \Delta B_l \quad (44)$$

In the deep layer equations (White and Bromley, 1995), expression of $\mathcal{W}_{\text{vei}}$ at full levels is:

$$[\mathcal{W}_{\text{vei}}]_l = \left[\frac{\Delta B_l \left[\frac{r_s{}^2}{a^2}\right]_l}{[\mathcal{R}_{inte}]_{(top,surf)} \left\langle \frac{\Delta B}{\Delta\eta} \frac{r_s{}^2}{a^2} \right\rangle} \right] \quad (45)$$

When the finite element vertical discretisation is activated (**LVERTFE**=.T.), $[\mathcal{R}_{inte}]_{(top,surf)}$ is a vector tricky to compute, it is precomputed in the setup routine **SUVERTFE1** or **SUVERTFE3** and stored in the array **RINTE** of **YOMCVER**. Vertical integrations are done in routine **VERINT**. For more details see the part of the appendix of documentation (IDEUL) which explains the computation of matrix $[\mathcal{R}_{inte}]$. When the finite difference vertical discretisation is activated (**LVERTFE**=.F.), $[\mathcal{R}_{inte}]_{(top,surf)}$ is simply the vector of coordinates $([\Delta\eta]_1; [\Delta\eta]_2; ...; [\Delta\eta]_l; ...; [\Delta\eta]_L)$ so:

- equation (42) rewrites:

$$\sum_{l=1}^{L} [\mathcal{W}_{\text{vei}}]_l \left(\frac{dX}{dt}\right)_l = \sum_{l=1}^{L} [\mathcal{W}_{\text{vei}}]_l \mathcal{A}_l + \sum_{l=1}^{L} [\mathcal{W}_{\text{vei}}]_l \mathcal{F}_l \quad (46)$$

- equation (45) rewrites:

$$[\mathcal{W}_{\text{vei}}]_l = \left[\frac{\Delta B_l \left[\frac{r_s{}^2}{a^2}\right]_l}{\sum_{k=1}^{k=L} [\Delta B_k] \left[\frac{r_s{}^2}{a^2}\right]_k} \right] \quad (47)$$

- routine **VERINT** is not used in this case.

List of equations.

- Continuity equation.

Generic notations.

Generic notation **N(X)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:

- $\mathcal{A}$ is the total term (sum of dynamical contributions): it is assumed to be a 3D term (sum of 3D and 2D contributions).
- $\mathcal{B}$ 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 $\mathcal{A} - \beta\mathcal{B}$ is the non-linear term, considered as a 3D term.
- $\mathcal{F}$ is the sum of contributions computed in the physical parameterizations; it is assumed to be a 2D term (vertical integral of a 3D term).

Description stands for the general case where linear and non-linear terms are gathered in the same buffer, and where no additional splitting is required to do diagnostics or to apply SLHD interpolations to a subset of the terms interpolated by a high-order interpolation. In some particular cases, additional splitting involving separate buffers may be required (see in part 5.2).

Other points.

$*$ **High-order interpolations:** In the following discretisations, “high-order interpolations" means: 32 points interpolations for 3D terms (vertical interpolations are cubic), 12 points interpolations for 2D terms.

$*$ **Uncentering:** ϵ is a first-order “uncentering factor". It allows to remove the noise due to gravity waves (orographic resonance).

$*$ **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 at the origin point, $[\mathcal{R}_{inte}]_{(top,surf)} \left\langle \left[\frac{\mathcal{W}_{\text{vei}}}{\Delta\eta}\right]_F [\text{surface quantity}]_O \right\rangle$ has no reason to be equal to $[\text{surface quantity}]_{O(\eta=1)}$, these quantities are generally different: this is $[\mathcal{R}_{inte}]_{(top,surf)} \left\langle \left[\frac{\mathcal{W}_{\text{vei}}}{\Delta\eta}\right]_F [\text{surface quantity}]_O \right\rangle$ which has to be computed. $\left\langle \left[\frac{\mathcal{W}_{\text{vei}}}{\Delta\eta}\right]_F [\text{surface quantity}]_O \right\rangle$ is the vector containing $\left[\frac{[\mathcal{W}_{\text{vei}}]_l}{[\Delta\eta]_l}\right]_F [\text{surface quantity}]_{O(l)}$, $l = 1$ to L. For **LVERTFE**=.F. this is $\sum_{l=1}^{L} [[\mathcal{W}_{\text{vei}}]_l]_F [\text{surface quantity}]_{O(l)}$.

3TL vertical interpolating SL scheme.

Equation (42) is discretised as follows:

$$
\begin{aligned}
&(X-(1+\epsilon)\Delta t\beta\mathcal{B})^{+}_{F} \\
&=[\mathcal{R}_{inte}]_{(top,surf)}\left\langle\left[\frac{\mathcal{W}_{\rm vei}}{\Delta\eta}\right]_{F}\{X^{-}+(1-\epsilon)\Delta t[\mathcal{A}-\beta\mathcal{B}]^{o}+[(1-\epsilon)\Delta t\beta\mathcal{B}+2\Delta t\mathcal{F}]^{-}\}_{O}\right\rangle \\
&+\{(1+\epsilon)\Delta t[\mathcal{A}-\beta\mathcal{B}]^{o}\}_{F}
\end{aligned}
\qquad (48)
$$

Buffers content before interpolations for **N(X)LAG**=2:

- **P(X3D)L9**: $(1-\epsilon)\Delta t[\mathcal{A}-\beta\mathcal{B}]^{o}+[(1-\epsilon)\Delta t\beta\mathcal{B}]^{-}$ for high-order interpolation at the origin point $O(l)$.
- **P(X2D)9**: $[X^{-}+2\Delta t\mathcal{F}]^{-}$ for horizontal high-order interpolation at the origin point $O(l)$.
- **P(X)T1**: $(1+\epsilon)\Delta t[\mathcal{A}-\beta\mathcal{B}]^{o}$ then provisional add of quantity $[(1+\epsilon)\Delta t\beta\mathcal{B}]^{o}$ before t+dt physics; evaluated at the final point F.

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

- **P(X3D)L9**: $(1-\epsilon)\Delta t[\mathcal{A}-\beta\mathcal{B}]^{o}+[(1-\epsilon)\Delta t\beta\mathcal{B}]^{-}$ for trilinear interpolation at the origin point $O(l)$.
- **P(X2D)9**: $[X^{-}+2\Delta t\mathcal{F}]^{-}$ for horizontal high-order interpolation at the origin point $O(l)$.
- **P(X)T1**: $(1+\epsilon)\Delta t[\mathcal{A}-\beta\mathcal{B}]^{o}$ then provisional add of quantity $[(1+\epsilon)\Delta t\beta\mathcal{B}]^{o}$ before t+dt physics; evaluated at the final point F.

2TL vertical interpolating SL scheme: conventional discretisation (LSETTLS=.F.) and first-order uncentering.

The $t+\frac{\Delta t}{2}$ non-linear term $\mathcal{A}^{m}-\beta\mathcal{B}^{m}$ used in the 2TL SL scheme is computed by a linear temporal extrapolation using the t and $t-\Delta t$ quantities at the same location. At the first time integration step, values at $t+\frac{\Delta t}{2}$ are set equal to initial values. This discretisation of the 2TL SL scheme follows (Mc Donald and Haugen, 1992). Quantity $\mathcal{A}^{o}-\beta\mathcal{B}^{o}$ has to be saved in a buffer **P(X)NLT9** to be available as $\mathcal{A}^{-}-\beta\mathcal{B}^{-}$ for the following timestep.

Equation (42) is discretised as follows:

$$
\begin{aligned}
&(X-(1+\epsilon)\tfrac{\Delta t}{2}\beta\mathcal{B})^{+}_{F} \\
&=[\mathcal{R}_{inte}]_{(top,surf)}\left\langle\left[\frac{\mathcal{W}_{\rm vei}}{\Delta\eta}\right]_{F}\{X^{o}+(1-\epsilon)\tfrac{\Delta t}{2}[\mathcal{A}-\beta\mathcal{B}]^{m}+[(1-\epsilon)\tfrac{\Delta t}{2}\beta\mathcal{B}+\Delta t\mathcal{F}]^{o}\}_{O}\right\rangle \\
&+\{(1+\epsilon)\tfrac{\Delta t}{2}[\mathcal{A}-\beta\mathcal{B}]^{m}\}_{F}
\end{aligned}
\qquad (49)
$$

which can be rewritten, once expanded the extrapolation:

$$
\begin{aligned}
(X-(1+\epsilon)\tfrac{\Delta t}{2}\beta\mathcal{B})^{+}_{F}&=[\mathcal{R}_{inte}]_{(top,surf)}\left\langle\left[\frac{\mathcal{W}_{\rm vei}}{\Delta\eta}\right]_{F}\{X^{o}+\tfrac{\Delta t}{2}(1-\epsilon)\mathcal{A}^{o}+\Delta t\mathcal{F}^{o}\}_{O}\right\rangle \\
&+0.5[\mathcal{R}_{inte}]_{(top,surf)}\left\langle\left[\frac{\mathcal{W}_{\rm vei}}{\Delta\eta}\right]_{F}(1-\epsilon)\tfrac{\Delta t}{2}\{(\mathcal{A}-\beta\mathcal{B})^{o}-(\mathcal{A}-\beta\mathcal{B})^{-}\}_{O}\right\rangle \\
&+\tfrac{\Delta t}{2}(1+\epsilon)[1.5(\mathcal{A}-\beta\mathcal{B})^{o}-0.5(\mathcal{A}-\beta\mathcal{B})^{-}]_{F}
\end{aligned}
\qquad (50)
$$

Buffers content before interpolations for **N(X)LAG**=2:

- **P(X3D)L9**: $(1-\epsilon)\frac{\Delta t}{2}[1.5(\mathcal{A}-\beta\mathcal{B})^o-0.5(\mathcal{A}-\beta\mathcal{B})^-]+[(1-\epsilon)\frac{\Delta t}{2}\beta\mathcal{B}]^o$ for high-order interpolation at the origin point $O(l)$.
- **P(X2D)9**: $X^o+[\Delta t\mathcal{F}]^o$ for horizontal high-order interpolation at the origin point $O(l)$.
- **P(X)T1**: $(1+\epsilon)\frac{\Delta t}{2}[1.5(\mathcal{A}-\beta\mathcal{B})^o-0.5(\mathcal{A}-\beta\mathcal{B})^-]$ then provisional add of quantity $[(1+\epsilon)\frac{\Delta t}{2}\beta\mathcal{B}]^o$ before t+dt physics; evaluated at the final point F.
- remark: another possibility is to add all the content of **P(X2D)9** to **P(X3D)L9** in order to remove one horizontal interpolation.

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

- **P(X3D)L9**: $(1-\epsilon)\frac{\Delta t}{2}[1.5(\mathcal{A}-\beta\mathcal{B})^o-0.5(\mathcal{A}-\beta\mathcal{B})^-]+[(1-\epsilon)\frac{\Delta t}{2}\beta\mathcal{B}]^o$ for trilinear interpolation at the origin point $O(l)$.
- **P(X2D)9**: $X^o+[\Delta t\mathcal{F}]^o$ for horizontal high-order interpolation at the origin point $O(l)$.
- **P(X)T1**: $(1+\epsilon)\frac{\Delta t}{2}[1.5(\mathcal{A}-\beta\mathcal{B})^o-0.5(\mathcal{A}-\beta\mathcal{B})^-]$ then provisional add of quantity $[(1+\epsilon)\frac{\Delta t}{2}\beta\mathcal{B}]^o$ before t+dt physics; evaluated at the final point F.

2TL vertical interpolating SL scheme: conventional discretisation (LSETTLS=.F.) and pseudo-second order uncentering.

One starts to remove uncentering ϵ from the nonlinear terms and to apply a second-order uncentering $\epsilon_{\mathcal{X}}$ to linear terms, that yelds a term $\mathcal{B}^-$ in the discretisation. From property given by formula (27), one can remove term $\mathcal{B}^-$ and show that discretisation is equivalent to replace β by $(1+\epsilon_{\mathcal{X}})\beta$. For more details, see part 5 (equations (37) and (38)) of (Simmons and Temperton, 1996). In equation (49), uncentering ϵ has to be replaced by zero, and β has to be replaced by $(1+\epsilon_{\mathcal{X}})\beta$. The $t+\frac{\Delta t}{2}$ non-linear term $\mathcal{A}^m-(1+\epsilon_{\mathcal{X}})\beta\mathcal{B}^m$ used in the 2TL SL scheme is computed by a linear temporal extrapolation using the t and $t-\Delta t$ quantities at the same location. At the first time integration step, values at $t+\frac{\Delta t}{2}$ are set equal to initial values and second-order uncentering is replaced by a first order uncentering. This discretisation of the 2TL SL scheme follows (Mc Donald and Haugen, 1992). Quantity $\mathcal{A}^o-(1+\epsilon_{\mathcal{X}})\beta\mathcal{B}^o$ has to be saved in a buffer **P(X)NLT9** to be available as $\mathcal{A}^- - (1+\epsilon_{\mathcal{X}})\beta\mathcal{B}^-$ for the following timestep.

Equation (42) is discretised as follows:

$$\begin{aligned}&(X-(1+\epsilon_{\mathcal{X}})\tfrac{\Delta t}{2}\beta\mathcal{B})^+_F\\ &=[\mathcal{R}_{inte}]_{(top,surf)}\left\langle\left[\frac{\mathcal{W}_{\text{vei}}}{\Delta\eta}\right]_F\{X^o+\tfrac{\Delta t}{2}[\mathcal{A}-(1+\epsilon_{\mathcal{X}})\beta\mathcal{B}]^m+[(1+\epsilon_{\mathcal{X}})\tfrac{\Delta t}{2}\beta\mathcal{B}+\Delta t\mathcal{F}]^o\}_O\right\rangle\\ &\quad+\{\tfrac{\Delta t}{2}[\mathcal{A}-(1+\epsilon_{\mathcal{X}})\beta\mathcal{B}]^m\}_F\end{aligned}\tag{51}$$

which can be rewritten, once expanded the extrapolation:

$$\begin{aligned}&(X-(1+\epsilon_{\mathcal{X}})\tfrac{\Delta t}{2}\beta\mathcal{B})^+_F=[\mathcal{R}_{inte}]_{(top,surf)}\left\langle\left[\frac{\mathcal{W}_{\text{vei}}}{\Delta\eta}\right]_F\{X^o+\tfrac{\Delta t}{2}\mathcal{A}^o+\Delta t\mathcal{F}^o\}_O\right\rangle\\ &+0.5[\mathcal{R}_{inte}]_{(top,surf)}\left\langle\left[\frac{\mathcal{W}_{\text{vei}}}{\Delta\eta}\right]_F\tfrac{\Delta t}{2}\{(\mathcal{A}-(1+\epsilon_{\mathcal{X}})\beta\mathcal{B})^o-(\mathcal{A}-(1+\epsilon_{\mathcal{X}})\beta\mathcal{B})^-\}_O\right\rangle\\ &\quad+\tfrac{\Delta t}{2}[1.5(\mathcal{A}-(1+\epsilon_{\mathcal{X}})\beta\mathcal{B})^o-0.5(\mathcal{A}-(1+\epsilon_{\mathcal{X}})\beta\mathcal{B})^-]_F\end{aligned}\tag{52}$$

Buffers content before interpolations for **N(X)LAG**=2:

- **P(X3D)L9**: $\frac{\Delta t}{2}[1.5(\mathcal{A}-(1+\epsilon_{\mathcal{X}})\beta\mathcal{B})^o - 0.5(\mathcal{A}-(1+\epsilon_{\mathcal{X}})\beta\mathcal{B})^-] + [(1+\epsilon_{\mathcal{X}})\frac{\Delta t}{2}\beta\mathcal{B}]^o$ for high-order interpolation at the origin point $O(l)$.
- **P(X2D)9**: $X^o + [\Delta t\mathcal{F}]^o$ for horizontal high-order interpolation at the origin point $O(l)$.
- **P(X)T1**: $\frac{\Delta t}{2}[1.5(\mathcal{A}-(1+\epsilon_{\mathcal{X}})\beta\mathcal{B})^o - 0.5(\mathcal{A}-(1+\epsilon_{\mathcal{X}})\beta\mathcal{B})^-]$ then provisional add of quantity $[(1+\epsilon_{\mathcal{X}})\frac{\Delta t}{2}\beta\mathcal{B}]^o$ before t+dt physics; evaluated at the final point F.
- remark: another possibility is to add all the content of **P(X2D)9** to **P(X3D)L9** in order to remove one horizontal interpolation.

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

- **P(X3D)L9**: $\frac{\Delta t}{2}[1.5(\mathcal{A}-(1+\epsilon_{\mathcal{X}})\beta\mathcal{B})^o - 0.5(\mathcal{A}-(1+\epsilon_{\mathcal{X}})\beta\mathcal{B})^-] + [(1+\epsilon_{\mathcal{X}})\frac{\Delta t}{2}\beta\mathcal{B}]^o$ for trilinear interpolation at the origin point $O(l)$.
- **P(X2D)9**: $X^o + [\Delta t\mathcal{F}]^o$ for horizontal high-order interpolation at the origin point $O(l)$.
- **P(X)T1**: $\frac{\Delta t}{2}[1.5(\mathcal{A}-(1+\epsilon_{\mathcal{X}})\beta\mathcal{B})^o - 0.5(\mathcal{A}-(1+\epsilon_{\mathcal{X}})\beta\mathcal{B})^-]$ then provisional add of quantity $[(1+\epsilon_{\mathcal{X}})\frac{\Delta t}{2}\beta\mathcal{B}]^o$ before t+dt physics; evaluated at the final point F.

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

The $t+\frac{\Delta t}{2}$ non-linear term $\mathcal{A}^m - \beta\mathcal{B}^m$ used in the 2TL SL scheme if **LSETTLS**=.F. is replaced in the case **LSETTLS**=.T. by a linear spatio-temporal extrapolation comparable to the one applied to the wind components for the research of trajectory (see formula (81)), except the fact that there is an additional uncentering. Term $\mathcal{A}^m - \beta\mathcal{B}^m$ is replaced by

$$0.5(1+\epsilon)[\mathcal{A}^o - \beta\mathcal{B}^o]_F + 0.5(2-\epsilon)[\mathcal{A}^o - \beta\mathcal{B}^o]_{O(l)} - 0.5[\mathcal{A}^- - \beta\mathcal{B}^-]_{O(l)}$$

This type of extrapolation is available only for **N(X)LAG**=3. At the first time integration step, values at $t+\frac{\Delta t}{2}$ are set equal to initial values. Quantity $\mathcal{A}^o - \beta\mathcal{B}^o$ has to be saved in a buffer **P(X)NLT9** to be available as $\mathcal{A}^- - \beta\mathcal{B}^-$ for the following timestep.

Equation (42) is discretised as follows:

$$\begin{aligned} & (X-(1+\epsilon)\tfrac{\Delta t}{2}\beta\mathcal{B})^+_F \\ = [\mathcal{R}_{inte}]_{(top,surf)} & \left\langle \left[\frac{\mathcal{W}\text{vei}}{\Delta\eta}\right]_F \{X^o + (2-\epsilon)\tfrac{\Delta t}{2}[\mathcal{A}-\beta\mathcal{B}]^o - \tfrac{\Delta t}{2}[\mathcal{A}-\beta\mathcal{B}]^- + [(1-\epsilon)\tfrac{\Delta t}{2}\beta\mathcal{B} + \Delta t\mathcal{F}]^o\}_O \right\rangle \\ & +\{(1+\epsilon)\tfrac{\Delta t}{2}[\mathcal{A}-\beta\mathcal{B}]^o\}_F \end{aligned} \tag{53}$$

which can be rewritten:

$$\begin{aligned} & (X-(1+\epsilon)\tfrac{\Delta t}{2}\beta\mathcal{B})^+_F \\ = [\mathcal{R}_{inte}]_{(top,surf)} & \left\langle \left[\frac{\mathcal{W}\text{vei}}{\Delta\eta}\right]_F \{X^o + (1-\epsilon)\tfrac{\Delta t}{2}\mathcal{A} + \Delta t\mathcal{F}]^o\}_O \right\rangle \\ +[\mathcal{R}_{inte}]_{(top,surf)} & \left\langle \left[\frac{\mathcal{W}\text{vei}}{\Delta\eta}\right]_F \tfrac{\Delta t}{2}\{[\mathcal{A}-\beta\mathcal{B}]^o - [\mathcal{A}-\beta\mathcal{B}]^-\}_O \right\rangle \\ & +\{(1+\epsilon)\tfrac{\Delta t}{2}[\mathcal{A}-\beta\mathcal{B}]^o\}_F \end{aligned} \tag{54}$$

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

- **P(X3D)L0** is not used.

- **P(X3D)L9**: $(2-\epsilon)\frac{\Delta t}{2}[\mathcal{A}-\beta\mathcal{B}]^o - \frac{\Delta t}{2}[\mathcal{A}-\beta\mathcal{B}]^- + [(1-\epsilon)\frac{\Delta t}{2}\beta\mathcal{B}]^o$ for trilinear interpolation at the origin point $O(l)$ (which can be rewritten: $(1-\epsilon)\frac{\Delta t}{2}\mathcal{A}+\frac{\Delta t}{2}\{[\mathcal{A}-\beta\mathcal{B}]^o-[\mathcal{A}-\beta\mathcal{B}]^-\}$).
- **P(X2D)0** is not used.
- **P(X2D)9**: $X^o+[\Delta t\mathcal{F}]^o$ for horizontal high-order interpolation at the origin point $O(l)$.
- **P(X)T1**: $(1+\epsilon)\frac{\Delta t}{2}[\mathcal{A}-\beta\mathcal{B}]^o$ then provisional add of quantity $[(1+\epsilon)\frac{\Delta t}{2}\beta\mathcal{B}]^o$ before t+dt physics; evaluated at the final point F.

2TL vertical interpolating SL scheme: stable discretisation (LSETTLS=.T.) and pseudo-second order uncentering.

In equation (53), uncentering ϵ has to be replaced by zero, and β has to be replaced by $(1+\epsilon_{\mathcal{X}})\beta$. At the first time integration step, values at $t+\frac{\Delta t}{2}$ are set equal to initial values and second-order uncentering is replaced by a first order uncentering. Quantity $\mathcal{A}^o-(1+\epsilon_{\mathcal{X}})\beta\mathcal{B}^o$ has to be saved in a buffer **P(X)NLT9** to be available as $\mathcal{A}^- -(1+\epsilon_{\mathcal{X}})\beta\mathcal{B}^-$ for the following timestep.

Equation (42) is discretised as follows:

$$\begin{aligned}
&(X-(1+\epsilon_{\mathcal{X}})\tfrac{\Delta t}{2}\beta\mathcal{B})^+_F \\
&= [\mathcal{R}_{inte}]_{(top,surf)}\left\langle\left[\frac{\mathcal{W}_{\mathrm{vei}}}{\Delta\eta}\right]_F\{X^o+\Delta t[\mathcal{A}-(1+\epsilon_{\mathcal{X}})\beta\mathcal{B}]^o-\tfrac{\Delta t}{2}[\mathcal{A}-(1+\epsilon_{\mathcal{X}})\beta\mathcal{B}]^-+[(1+\epsilon_{\mathcal{X}})\tfrac{\Delta t}{2}\beta\mathcal{B}+\Delta t\mathcal{F}]^o\}_O\right\rangle \\
&+\{\tfrac{\Delta t}{2}[\mathcal{A}-(1+\epsilon_{\mathcal{X}})\beta\mathcal{B}]^o\}_F
\end{aligned} \tag{55}$$

which can be rewritten:

$$\begin{aligned}
&(X-(1+\epsilon_{\mathcal{X}})\tfrac{\Delta t}{2}\beta\mathcal{B})^+_F \\
&= [\mathcal{R}_{inte}]_{(top,surf)}\left\langle\left[\frac{\mathcal{W}_{\mathrm{vei}}}{\Delta\eta}\right]_F\{X^o+\tfrac{\Delta t}{2}\mathcal{A}+\Delta t\mathcal{F}]^o\}_O\right\rangle \\
&+[\mathcal{R}_{inte}]_{(top,surf)}\left\langle\left[\frac{\mathcal{W}_{\mathrm{vei}}}{\Delta\eta}\right]_F\tfrac{\Delta t}{2}\{[\mathcal{A}-(1+\epsilon_{\mathcal{X}})\beta\mathcal{B}]^o-[\mathcal{A}-(1+\epsilon_{\mathcal{X}})\beta\mathcal{B}]^-\}_O\right\rangle \\
&+\{\tfrac{\Delta t}{2}[\mathcal{A}-(1+\epsilon_{\mathcal{X}})\beta\mathcal{B}]^o\}_F
\end{aligned} \tag{56}$$

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

- **P(X3D)L0** is not used.
- **P(X3D)L9**: $\Delta t[\mathcal{A}-(1+\epsilon_{\mathcal{X}})\beta\mathcal{B}]^o-\frac{\Delta t}{2}[\mathcal{A}-(1+\epsilon_{\mathcal{X}})\beta\mathcal{B}]^-+[(1+\epsilon_{\mathcal{X}})\frac{\Delta t}{2}\beta\mathcal{B}]^o$ for trilinear interpolation at the origin point $O(l)$ (which can be rewritten: $\frac{\Delta t}{2}\mathcal{A}+\frac{\Delta t}{2}\{[\mathcal{A}-(1+\epsilon_{\mathcal{X}})\beta\mathcal{B}]^o-[\mathcal{A}-(1+\epsilon_{\mathcal{X}})\beta\mathcal{B}]^-\}$).
- **P(X2D)0** is not used.
- **P(X2D)9**: $X^o+[\Delta t\mathcal{F}]^o$ for horizontal high-order interpolation at the origin point $O(l)$.
- **P(X)T1**: $\frac{\Delta t}{2}[\mathcal{A}-(1+\epsilon_{\mathcal{X}})\beta\mathcal{B}]^o$ then provisional add of quantity $[(1+\epsilon_{\mathcal{X}})\frac{\Delta t}{2}\beta\mathcal{B}]^o$ before t+dt physics; evaluated at the final point F.

5.5 Discretisation for a 2D variable in a 2D model.

The content of the part (5.2) is generally valid, but there are particular remarks.

List of equations.

- Momentum equation.
- Continuity equation.

Generic notations.

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

- **NWLAG** for momentum equation.
- **NVLAG** for continuity equation. Negative values of **NVLAG** are used for Lagrangian formulation of continuity equation (the following discretisations apply to the absolute value of **NVLAG**), only **NVLAG**=-2 is available in this case.

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.
- **PSPL0**, **PSPL9**, **PSPT1** for continuity equation.

Generic notation **P(X)NLT9** stands for:

- **PUNLT9** for U-momentum equation.
- **PVNLT9** for V-momentum equation.
- **PSPNLT9** for continuity equation.

Generic notation for total term, linear term, non linear term: $\mathcal{A}$ is the total term (sum of dynamical contributions), $\mathcal{B}$ is the linear term (treated in the semi-implicit scheme), the difference $\mathcal{A} - \beta\mathcal{B}$ is the non-linear term.

Other points.

$*$ **High-order interpolations:** In the following discretisations, “high-order interpolations" means 12 points interpolations.

$*$ **Uncentering:** ϵ is a first-order “uncentering factor". It allows to remove the noise due to gravity waves (orographic resonance).

$*$ **Vectorial variables:** The following discretisations are written for scalar variables. For vectorial variables (for example the 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 $\mathcal{R}^{OF}\{\text{this expression}\}_O$.
- expression interpolated at M has to be replaced by $\mathcal{R}^{MF}\{\text{this expression}\}_M$.

3TL SL scheme.

Equation:

$$\frac{dX}{dt} = \mathcal{A} \tag{57}$$

is discretised as follows:

$$\begin{aligned}(X - (1+\epsilon)\Delta t\beta\mathcal{B})^{+}_{F} = \{X^{-} + [(1-\epsilon)\Delta t\mathcal{A} - (1-\epsilon)\Delta t\beta\mathcal{B}]^{o} + [(1-\epsilon)\Delta t\beta\mathcal{B}]^{-}\}_{O} \\ +\{[(1+\epsilon)\Delta t\mathcal{A} - (1+\epsilon)\Delta t\beta\mathcal{B}]^{o}\}_{F}\end{aligned} \tag{58}$$

Buffers content before interpolations for **N(X)LAG**=2:

- **P(X)L0** is not used.
- **P(X)L9**: $X^{-} + [(1-\epsilon)\Delta t\mathcal{A} - (1-\epsilon)\Delta t\beta\mathcal{B}]^{o} + [(1-\epsilon)\Delta t\beta\mathcal{B}]^{-}$ for high-order interpolation at the origin point O.
- **P(X)T1**: $[(1+\epsilon)\Delta t\mathcal{A} - (1+\epsilon)\Delta t\beta\mathcal{B}]^{o}$; evaluated at the final point F.

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

- **P(X)L0**: $[(1-\epsilon)\Delta t\mathcal{A} - (1-\epsilon)\Delta t\beta\mathcal{B}]^{o} + [(1-\epsilon)\Delta t\beta\mathcal{B}]^{-}$ for bilinear interpolation at the origin point O.
- **P(X)L9**: X^{-} for high-order interpolation at the origin point O.
- **P(X)T1**: $[(1+\epsilon)\Delta t\mathcal{A} - (1+\epsilon)\Delta t\beta\mathcal{B}]^{o}$; evaluated at the final point F.

2TL SL scheme: conventional discretisation (LSETTLS=.F.) and first-order uncentering.

The $t + \frac{\Delta t}{2}$ non-linear term $\mathcal{A}^{m} - \beta\mathcal{B}^{m}$ used in the 2TL SL scheme is computed by a linear temporal extrapolation using the t and $t - \Delta t$ quantities at the same location. At the first time integration step, values at $t + \frac{\Delta t}{2}$ are set equal to initial values. This discretisation of the 2TL SL scheme follows (Mc Donald and Haugen, 1992). Quantity $\mathcal{A}^{o} - \beta\mathcal{B}^{o}$ has to be saved in a buffer **P(X)NLT9** to be available as $\mathcal{A}^{-} - \beta\mathcal{B}^{-}$ for the following timestep, when non-zero (i.e. only for continuity equation, if β=1).

Equation (57) is discretised as follows:

$$\begin{aligned}(X - (1+\epsilon)\tfrac{\Delta t}{2}\beta\mathcal{B})^{+}_{F} = \{X^{o} + [(1-\epsilon)\tfrac{\Delta t}{2}\mathcal{A} - (1-\epsilon)\tfrac{\Delta t}{2}\beta\mathcal{B}]^{m} + [(1-\epsilon)\tfrac{\Delta t}{2}\beta\mathcal{B}]^{o}\}_{O} \\ +\{[(1+\epsilon)\tfrac{\Delta t}{2}\mathcal{A} - (1+\epsilon)\tfrac{\Delta t}{2}\beta\mathcal{B}]^{m}\}_{F}\end{aligned} \tag{59}$$

Buffers content before interpolations for **N(X)LAG**=2:

- **P(X)L0** is not used.
- **P(X)L9**: $X^{o} + [(1-\epsilon)\frac{\Delta t}{2}\mathcal{A} - (1-\epsilon)\frac{\Delta t}{2}\beta\mathcal{B}]^{m} + [(1-\epsilon)\frac{\Delta t}{2}\beta\mathcal{B}]^{o}$ for high-order interpolation at the origin point O.
- **P(X)T1**: $[(1+\epsilon)\frac{\Delta t}{2}\mathcal{A} - (1+\epsilon)\frac{\Delta t}{2}\beta\mathcal{B}]^{m}$; evaluated at the final point F.

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

- **P(X)L0**: $[(1-\epsilon)\frac{\Delta t}{2}\mathcal{A} - (1-\epsilon)\frac{\Delta t}{2}\beta\mathcal{B}]^{m} + [(1-\epsilon)\frac{\Delta t}{2}\beta\mathcal{B}]^{o}$ for bilinear interpolation at the origin point O.
- **P(X)L9**: X^{o} for high-order interpolation at the origin point O.
- **P(X)T1**: $[(1+\epsilon)\frac{\Delta t}{2}\mathcal{A} - (1+\epsilon)\frac{\Delta t}{2}\beta\mathcal{B}]^{m}$; evaluated at the final point F.

$*$ **Remark for momentum equation:** β can only take the value 1 and in this case the non-linear term $\mathcal{A} - \mathcal{B}$ is zero; simplifications are made in the code; arrays **PUNLT9** and **PVNLT9** are useless and not allocated.

2TL SL scheme: stable discretisation (LSETTLS=.T.) and first-order uncentering.

The $t + \frac{\Delta t}{2}$ non-linear term $\mathcal{A}^m - \beta\mathcal{B}^m$ used in the 2TL SL scheme if **LSETTLS**=.F. is replaced in the case **LSETTLS**=.T. by a linear spatio-temporal extrapolation comparable to the one applied to the wind components for the research of trajectory (see formula (81)), except the fact that there is an additional uncentering. Term $\mathcal{A}^m - \beta\mathcal{B}^m$ is replaced by

$$0.5(1+\epsilon)[\mathcal{A}^o - \beta\mathcal{B}^o]_F + 0.5(2-\epsilon)[\mathcal{A}^o - \beta\mathcal{B}^o]_O - 0.5[\mathcal{A}^- - \beta\mathcal{B}^-]_O$$

This type of extrapolation is available only for **N(X)LAG**=3. At the first time integration step, values at $t + \frac{\Delta t}{2}$ are set equal to initial values. Quantity $\mathcal{A}^o - \beta\mathcal{B}^o$ has to be saved in a buffer **P(X)NLT9** to be available as $\mathcal{A}^- - \beta\mathcal{B}^-$ for the following timestep, when non-zero (i.e. only for continuity equation, if β=1).

Equation (57) is discretised as follows:

$$\begin{aligned}(X - (1+\epsilon)\tfrac{\Delta t}{2}\beta\mathcal{B})^+_F = \{X^o + [(2-\epsilon)\tfrac{\Delta t}{2}\mathcal{A} - (2-\epsilon)\tfrac{\Delta t}{2}\beta\mathcal{B}]^o - [\tfrac{\Delta t}{2}\mathcal{A} - \tfrac{\Delta t}{2}\beta\mathcal{B}]^- + [(1-\epsilon)\tfrac{\Delta t}{2}\beta\mathcal{B}]^o\}_O \\ +\{[(1+\epsilon)\tfrac{\Delta t}{2}\mathcal{A} - (1+\epsilon)\tfrac{\Delta t}{2}\beta\mathcal{B}]^o\}_F \qquad (60)\end{aligned}$$

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

- **P(X)L0**: $[(2-\epsilon)\frac{\Delta t}{2}\mathcal{A} - (2-\epsilon)\frac{\Delta t}{2}\beta\mathcal{B}]^o - [\frac{\Delta t}{2}\mathcal{A} - \frac{\Delta t}{2}\beta\mathcal{B}]^- + [(1-\epsilon)\frac{\Delta t}{2}\beta\mathcal{B}]^o$ for bilinear interpolation at the origin point O.
- **P(X)L9**: X^o for high-order interpolation at the origin point O.
- **P(X)T1**: $[(1+\epsilon)\frac{\Delta t}{2}\mathcal{A} - (1+\epsilon)\frac{\Delta t}{2}\beta\mathcal{B}]^o$; evaluated at the final point F.

$*$ **Remark for momentum equation:** β can only take the value 1 and in this case the non-linear term $\mathcal{A} - \mathcal{B}$ is zero; simplifications are made in the code; arrays **PUNLT9** and **PVNLT9** are useless and not allocated.

2TL SL scheme: pseudo-second order uncentering.

In cycle 37T1 of ARPEGE/IFS the pseudo-second order uncentering is not coded in the shallow-water model.

5.6 Additional vertical derivatives.

If δ_{TR} is non-zero, discretisation of temperature equation needs to compute the vertical advection $\left(\dot{\eta}\frac{d\alpha_{\mathrm{T}}}{d\eta}\right)$ (at full levels) of α_{T}. Layers values of α_{T} (array **RCORDIF**) are used to define $T + \delta_{TR}\frac{\alpha_{\mathrm{T}}\Phi_{\mathrm{s}}}{R_{\mathrm{d}}T_{ST}}$, but half level values of α_{T} (array **RCORDIH**) are used to compute vertical advection.

There is something similar in the $\hat{Q}$ equation if δ_{P} is non-zero (Wood and Staniforth deep-layer version of the NH-PDVD model, use of an array **RCORPDH**).

No longer existing option.

5.7 Case when some variables are evaluated at half levels.

The prognostic variables and the RHS of equations are generally evaluated at full levels in the discretisation. The previous general considerations valid to layer variables also apply to half level variables, but a “half level" trajectory has now to be computed (the origin O now matches a half level final point F).

- Horizontal displacement at half levels: the coordinates of the half level-trajectory interpolation point are computed as the average (with a vertical weight taking account of η) of the coordinates of the adjacent full level-trajectory interpolation points; this average is a lon-lat average on the computational sphere in spherical geometry, and a x-y average on the projection plane in plane geometry (ALADIN).
- Vertical displacement at half levels: the vertical coordinate of the half level-trajectory interpolation point is computed as the average (with a vertical weight taking account of η) of the vertical coordinates of the adjacent full level-trajectory interpolation points; no vertical displacement if there is no vertical displacement for the two adjacent full levels.
- No complete iterative recalculation of trajectory is done for half-level trajectories.

5.8 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.

6 Computation of medium and origin points.

Preliminary remark: the subsections (6.1), (6.2) and (6.3) are detailed for a spherical geometry and for the trajectory which is used in the advection of full level variables; the subsection (6.4) gives informations about the other cases (for example plane geometry and half level variables).

6.1 Medium point M (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 M. Rochas. In a 3TL SL scheme, the particle is at the point M at the instant t ($t+\Delta t/2$ for the first integration step). In a 2TL SL scheme, the particle is at the point M at the instant $t+\Delta t/2$. M is at the middle position of the origin point O and the final point F. Algorithm is described for deep layer equations; in the thin layer equations, replace simply r_s by a in formulae. For conveniency equations are written with the angular velocity $\mathbf{V}/r_s$ but actually this is rather $(a/r_s)\mathbf{V}$ which is used in the code. Parts (6.1), (6.1) and (6.1) are valid for non implicit iterative schemes. Part (6.1) is valid for a class of iterative centred-implicit schemes where the momentum equation is treated in an iterative centred-implicit manner (this is the case of the option **LPC_FULL** in ALADIN-NH).

* Notations:

- $\mathcal{R}^{MF}$ is the rotation operator from medium point to final point (see section 9).
- $\mathcal{R}^{OF}$ is the rotation operator from origin point to final point (see section 9).
- $\mathbf{r}^F = \mathbf{CF}$ (C Earth centre, F final point).
- $\mathbf{r}^M = \mathbf{CM}$ (M medium point).
- $\mathbf{r} = r_s\mathbf{k}$.
- ϕ^{MF}: angle $(\widehat{\mathbf{CM},\mathbf{CF}})$.
- θ^F,λ^F: latitude, longitude on the geographical sphere of F.
- θ^M,λ^M: latitude, longitude on the geographical sphere of M.
- $\mathbf{V}^M$: interpolated horizontal wind at M (wind at t in 3TL SL scheme, $t+0.5\Delta t$ in 2TL SL scheme).
- $\mathbf{V}^O$: interpolated horizontal wind at O (wind at t in 3TL SL scheme, $t+0.5\Delta t$ in 2TL SL scheme).
- a is the average Earth radius near the surface.
- Δt: time-step.
- δt:
  - In a 3TL SL scheme, $\delta t = 0.5\Delta t$ at the first integration step, $\delta t = \Delta t$ at the following integration steps (leap-frog scheme).
  - In a 2TL SL scheme, $\delta t = 0.5\Delta t$.
- L: number of layers of the model.
- A, B define hydrostatic pressure on the η levels ($\Pi = A + B\Pi_{\rm s}$, where $\Pi_{\rm s}$ is the hydrostatic surface pressure).
- ${\Pi_{\rm s}}^{st}$ is a reference hydrostatic pressure equal to the surface pressure of the standard atmosphere (variable **VP00**).
- Π_{ST} is a reference hydrostatic pressure defined at full levels and half levels corresponding to the surface reference hydrostatic pressure ${\Pi_{\rm s}}^{st}$ ($\Pi_{ST} = A + B{\Pi_{\rm s}}^{st}$): stored in array **STPRE**.

$*$ Definition of the vertical coordinate η: Research of medium point needs an exact definition of the vertical coordinate η. For the half level number $\bar{l}$ ($\bar{l}$ between 0 and L), $\eta_{\bar{l}}$ is defined by:

$$\eta_{\bar{l}} = \frac{A_{\bar{l}}}{{\Pi_{\mathrm{s}}}^{st}} + B_{\bar{l}} \tag{61}$$

if **LREGETA**=.F. (in namelist **NAMCT0**), and:

$$\eta_{\bar{l}} = \frac{\bar{l}}{L} \tag{62}$$

if **LREGETA**=.T. .
For the layer number l (l between 1 and L), η_l is defined by:

$$\eta_l = 0.5(\eta_{\bar{l}} + \eta_{\bar{l}-1}) \tag{63}$$

A specific definition of η may be required for the VFE operators if **LVERTFE**=.T.: it is controled by the key **LVFE_REGETA**.

Conventional algorithm (always used for 3TL SL scheme, case LSETTLST=LELTRA=.F. for 2TL SL scheme).

$*$ Extrapolation of the wind for 2TL SL scheme: The quantity $[\mathbf{V}/r_s]$ at $t + 0.5\Delta t$ used in the 2TL SL scheme is computed by a linear temporal extrapolation using the t and $t - \Delta t$ winds at the same location.

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

$$[\mathbf{r}/r_s]^M_{k+1} = [\mathbf{r}/r_s]^F \cos\phi_k - \frac{\mathcal{R}^{MF}([\mathbf{V}/r_s]^M_k)}{\mid [\mathbf{V}/r_s]^M_k \mid} \sin\phi_k \tag{64}$$

where:

$$\phi_k = \delta t \mid [\mathbf{V}/r_s]^M_k \mid \tag{65}$$

ϕ is a small angle:

$$\sin\phi \simeq \phi - \frac{\phi^3}{6} \tag{66}$$

and:

$$\cos\phi \simeq 1 - \frac{\phi^2}{2} \tag{67}$$

This approximation allows to simplify some calculations and avoids the occurrence of a division by 0 in the formula defining $\mathbf{r}^M_{k+1}$. Of course:

$$\sin\phi \simeq \phi\left(1 - \frac{\phi^2}{6}\right) \simeq \mid [\mathbf{V}/r_s] \mid \delta t \left(1 - \frac{\phi^2}{6}\right) \tag{68}$$

thus:

$$[\mathbf{r}/r_s]^M_{k+1} = [\mathbf{r}/r_s]^F \left(1 - \frac{\phi_k^2}{2}\right) - \mathcal{R}^{MF}([\mathbf{V}/r_s]^M_k)\delta t \left(1 - \frac{\phi_k^2}{6}\right) \tag{69}$$

On the vertical, for 3D model:

$$\eta^M_{k+1} = \eta^F - \delta t \dot{\eta}^M_k \tag{70}$$

$*$ **First iteration:** Let us start with $M_0 = F$, $[\mathbf{V}/r_s]^M_0 = [\mathbf{V}/r_s]^F$, $\phi_0 = \delta t \mid [\mathbf{V}/r_s]^F \mid$, $\dot{\eta}^M_0 = \dot{\eta}^F$. Horizontal wind $\mathbf{V}$ has components (u,v). Thus

$$\sin\theta^M_1 = \sin\theta^F \cos\phi_0 - \frac{[v/r_s]^F}{\mid [\mathbf{V}/r_s]^F \mid} \cos\theta^F \sin\phi_0 \tag{71}$$

$$\cos\theta^M_1 \cos(\lambda^M_1 - \lambda^F) = \cos\theta^F \cos\phi_0 + \frac{[v/r_s]^F}{\mid [\mathbf{V}/r_s]^F \mid} \sin\theta^F \sin\phi_0 \tag{72}$$

$$\cos\theta^M_1 \sin(\lambda^M_1 - \lambda^F) = -\frac{[u/r_s]^F}{\mid [\mathbf{V}/r_s]^F \mid} \sin\phi_0 \tag{73}$$

$$\eta^M_1 = \eta^F - \delta t \dot{\eta}^F_0 \tag{74}$$

This defines the coordinates of M_1. Then $[u/r_s]$, $[v/r_s]$, $\dot{\eta}$ are interpolated at this point, that gives $[\mathbf{V}/r_s]^M_1$ and $\dot{\eta}^M_1$. Tri-linear interpolations are used in the 3D primitive equation model, horizontal 12 points interpolations are used in the 2D shallow-water model (see section 11).

$*$ **Following iterations:** Let us denote by $\mathbf{V}^{'}(u^{'},v^{'}) = \mathcal{R}^{MF}(\mathbf{V}^M_k)$.

$$\sin\theta^M_{k+1} = \sin\theta^F \cos\phi_k - \frac{[v^{'}/r_s]}{\mid [\mathbf{V}'/r_s] \mid} \cos\theta^F \sin\phi_k \tag{75}$$

$$\cos\theta^M_{k+1} \cos(\lambda^M_{k+1} - \lambda^F) = \cos\theta^F \cos\phi_k + \frac{[v^{'}/r_s]}{\mid [\mathbf{V}'/r_s] \mid} \sin\theta^F \sin\phi_k \tag{76}$$

$$\cos\theta^M_{k+1} \sin(\lambda^M_{k+1} - \lambda^F) = -\frac{[u^{'}/r_s]}{\mid [\mathbf{V}'/r_s] \mid} \sin\phi_k \tag{77}$$

$$\eta^M_{k+1} = \eta^F - \delta t \dot{\eta}^M_k \tag{78}$$

This defines coordinates of M_{k+1}. Then, if it is not the last iteration, $[u/r_s]$, $[v/r_s]$, $\dot{\eta}$ are interpolated at this point, that gives $[\mathbf{V}/r_s]^M_{k+1}$ and $\dot{\eta}^M_{k+1}$. Tri-linear interpolations are used in the 3D primitive equation model, horizontal 12 points interpolations are used in the 2D shallow-water model (see section 11). This iterative algorithm quickly converges: 3 iterations are generally enough.

Stable algorithm for 2TL SL scheme (LSETTLST=.T., LELTRA=.F. in NAMDYN).

∗ **Extrapolation of the wind:** The previous algorithm with **LSETTLST**=.F. can sometimes generate instability (especially when applied to the vertical displacement) so a stable algorithm has been developed by M. Hortal at ECMWF. For more details about theoretical aspects see (Hortal, 1998), (Hortal, 2002). The basic idea is to replace the purely temporal extrapolation by a spatio-temporal extrapolation:

$$\mathcal{R}^{MF}[\mathbf{V}/r_s]^M(t+0.5\Delta t) = 1.5\mathcal{R}^{NF}[\mathbf{V}/r_s]^N(t) - 0.5\mathcal{R}^{OF}[\mathbf{V}/r_s]^O(t-\Delta t) \qquad (79)$$

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 $t+0.5\Delta t$. Assuming that the wind is constant along the trajectory one can write:

$$ON = 2NM = 0.5NF \qquad (80)$$

and evaluate the angular velocity $\mathcal{R}^{NF}[\mathbf{V}/r_s]^N(t)$ by $2/3\mathcal{R}^{MF}[\mathbf{V}/r_s]^M(t)$+$1/3\mathcal{R}^{OF}[\mathbf{V}/r_s]^O(t)$ or $1/3[\mathbf{V}/r_s]^F(t) + 2/3\mathcal{R}^{OF}[\mathbf{V}/r_s]^O(t)$. Expression of $[\mathbf{V}/r_s]^M(t+0.5\Delta t)$ becomes:

$$\mathcal{R}^{MF}[\mathbf{V}/r_s]^M(t+0.5\Delta t) = 0.5[\mathbf{V}/r_s]^F(t) + 0.5\mathcal{R}^{OF}(2[\mathbf{V}/r_s]^O(t) - [\mathbf{V}/r_s]^O(t-\Delta t)) (81)$$

The same type of extrapolation is done for the η-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 $k+1$:

$$[\mathbf{r}/r_s]^O_{k+1} = [\mathbf{r}/r_s]^F \cos\phi_k - \frac{0.5[\mathbf{V}/r_s]^F(t) + 0.5\mathcal{R}^{OF}(2[\mathbf{V}/r_s]^O_k(t) - [\mathbf{V}/r_s]^O_k(t-\Delta t))}{\mid 0.5[\mathbf{V}/r_s]^F(t) + 0.5\mathcal{R}^{OF}(2[\mathbf{V}/r_s]^O_k(t) - [\mathbf{V}/r_s]^O_k(t-\Delta t)) \mid} \sin\phi_k (82)$$

where:

$$\phi_k = 2\delta t \mid 0.5[\mathbf{V}/r_s]^F(t) + 0.5\mathcal{R}^{OF}(2[\mathbf{V}/r_s]^O_k(t) - [\mathbf{V}/r_s]^O_k(t-\Delta t)) \mid \qquad (83)$$

Approximations given by equations (66), (67) and (68) are still valid (change δt by $2\delta t$ in (68)), thus:

$$[\mathbf{r}/r_s]^O_{k+1} =$$

$$[\mathbf{r}/r_s]^F \left(1 - \frac{\phi_k^2}{2}\right) - (0.5[\mathbf{V}/r_s]^F(t) + 0.5\mathcal{R}^{OF}(2[\mathbf{V}/r_s]^O_k(t) - [\mathbf{V}/r_s]^O_k(t-\Delta t)))(2\delta t)\left(1 - \frac{\phi_k^2}{6}\right) (84)$$

On the vertical, for 3D model:

$$\eta^O_{k+1} = \eta^F - 2\delta t(0.5\dot{\eta}^F(t) + 0.5(2\dot{\eta}^O_k(t) - \dot{\eta}^O_k(t-\Delta t))) \qquad (85)$$

∗ **First iteration:** One starts with $M_0 = F$, $[\mathbf{V}/r_s]^F(t)$ as a first guess for the spatio-temporally extrapolated horizontal angular velocity, $\phi_0 = 2\delta t \mid [\mathbf{V}/r_s]^F(t) \mid$, $\dot{\eta}^F(t)$ as a first guess for the spatio-temporally extrapolated η-coordinate vertical wind. Remark that quantities at t are taken as a first guess and not quantities at $(t+0.5\Delta t)$, contrary to the case **LSETTLST**=.F. . Use equations (71) to (74) replacing δt by $2\delta t$ and the superscript M by O. This defines the coordinates of O_1; $(2[\mathbf{V}/r_s](t) - [\mathbf{V}/r_s](t-\Delta t))$ and $(2\dot{\eta}(t) - \dot{\eta}(t-\Delta t))$ are interpolated at this point, that allows to compute the wind components which will be used for the next iteration.

$*$ **Following iterations:** $[\mathbf{V}'/r_s]$ (of coordinates $([u'/r_s],[v'/r_s])$) is a generic notation for $(0.5[\mathbf{V}/r_s]^F(t)+0.5\mathcal{R}^{OF}(2[\mathbf{V}/r_s]^O_k(t)-[\mathbf{V}/r_s]^O_k(t-\Delta t)))$. For horizontal displacement use equations (75) to (77) replacing δt by $2\delta t$ and the superscript M by O. For vertical displacement use equation

$$\eta^O_{k+1}=\eta^F-2\delta t(0.5\dot{\eta}^F(t)+0.5(2\dot{\eta}^O_k(t)-\dot{\eta}^O_k(t-\Delta t))) \quad (86)$$

Alternate stable algorithm for 2TL SL scheme (LELTRA=.T. in NAMDYN).

$*$ **Extrapolation of the horizontal angular velocity:** The quantity $[\mathbf{V}/r_s]$ at $t+0.5\Delta t$ used in the 2TL SL scheme is computed using the RHS of the $[\mathbf{V}/r_s]$ equation with explicit formulation of Coriolis term. We denote this RHS by $RHSAV$ in subsubsection (6.1). The extrapolated value of $[\mathbf{V}/r_s]$ is given by:

$$[\mathbf{V}/r_s]_{\text{ext}}=[\mathbf{V}/r_s](t)+0.5\Delta tRHSAV \quad (87)$$

$*$ **Extrapolation of the vertical velocity:** The RHS is assumed to be zero, so:

$$\dot{\eta}_{\text{ext}}=\dot{\eta}(t) \quad (88)$$

There is no extrapolation at all.

$*$ **Algorithm:** The medium point is defined by the following iterative scheme: for the iteration $k+1$, a provisional position of the origin point is computed using equations (64) to (70), replacing "M" by "O" and "δt" by "Δt".

$*$ **First iteration:** Let us start with $O_0=F$, $[\mathbf{V}/r_s]^O_0=[\mathbf{V}/r_s]^F_{\text{ext}}$, $\phi_0=\Delta t\mid[\mathbf{V}/r_s]^F_{\text{ext}}\mid$, $\dot{\eta}^O_0=\dot{\eta}^F_{\text{ext}}$. Horizontal angular velocity $[\mathbf{V}/r_s]_{\text{ext}}$ has components $(u/r_s,v/r_s)$. Use equations (71) to (74), replacing "M" by "O" and "δt" by "Δt" to define coordinates of O_1. Then u/r_s,v/r_s,$\dot{\eta}$ are interpolated at this point, that gives $[\mathbf{V}/r_s]^O_1$ and $\dot{\eta}^O_1$ (subscript "ext" is omitted). Tri-linear interpolations are used in the 3D primitive equation model, horizontal 12 points interpolations are used in the 2D shallow-water model (see section 11).

$*$ **Following iterations:** Use equations (75) to (78), replacing "M" by "O" and "δt" by "Δt" to define coordinates of O_{k+1}. Then u/r_s,v/r_s,$\dot{\eta}$ are interpolated at this point, that gives $[\mathbf{V}/r_s]^O_{k+1}$ and $\dot{\eta}^O_{k+1}$ (subscript "ext" is omitted). Tri-linear interpolations are used in the 3D primitive equation model, horizontal 12 points interpolations are used in the 2D shallow-water model (see section 11). This iterative algorithm quickly converges: 3 iterations are generally enough.

Algorithm used with the iterative centred-implicit schemes.

$*$ **Preliminar remarks:** Iterative centred-implicit schemes are used to improve stability and it has been shown that this type of scheme has to be used when non-hydrostatic equations are advected by a SL2TL scheme. There are several manners to do iterative centred-implicit schemes; these schemes are not described in detail here; for some of them which are not obsolescent, see documentation (IDSI). When the momentum equation is treated by an iterative centred-implicit scheme, the semi-Lagrangian trajectory has to be recomputed at each iteration of the iterative centred-implicit scheme and interpolations have to be done again. In this case the algorithm of research trajectory is modified. The case is currently encountered with a SL2TL scheme and **LPC_FULL**=.T. . The following algorithm will be described for a SL2TL scheme but it can be extended to a SL3TL scheme (replacing instant t by instant $t-\Delta t$).

$*$ **Extrapolation of the wind if non-extrapolating option (LPC_NESCT=.T.):** No extrapolation is done. The first iteration of the iterative centred-implicit scheme uses $[\mathbf{V}/r_s](t)$ and $\dot{\eta}(t)$. The following iterations of the iterative centred-implicit scheme use the $[\mathbf{V}/r_s](t+\Delta t)$ and $\dot{\eta}(t+\Delta t)$ of the previous iteration to start the research of trajectory. As for the "**LSETTLST**" option the algorithm computes the origin point O.

$*$ **Extrapolation of the wind if extrapolating option (LPC_NESCT=.F.):** The only difference with the previous case is for the first iteration of the iterative centred-implicit scheme: the wind which is used is now $1.5[\mathbf{V}/r_s](t) - 0.5[\mathbf{V}/r_s](t-\Delta t)$ and $1.5\dot{\eta}(t) - 0.5\dot{\eta}(t-\Delta t)$.

$*$ **Algorithm:** One denotes by index "k" the numbering of the SL-trajectory research algorithm iteration, and by index "(i)" the number of the iterative centred-implicit scheme iteration (in variable **NCURRENT_ITER**).

The origin point is defined by the following iterative scheme: for the iteration $k+1$ of the SL-trajectory research algorithm:

$$\left[[\mathbf{r}/r_s]^O_{k+1}\right]_{(i)} = [\mathbf{r}/r_s]^F \left[\cos\phi_k\right]_{(i)} - \frac{0.5\mathcal{R}^{OF}[\mathbf{V}/r_s]^O_k(t) + 0.5\left[[\mathbf{V}/r_s]^F(t+\Delta t)\right]_{(i-1)}}{\mid 0.5\mathcal{R}^{OF}[\mathbf{V}/r_s]^O_k(t) + 0.5\left[[\mathbf{V}/r_s]^F(t+\Delta t)\right]_{(i-1)} \mid} \left[\sin\phi_k\right]_{(i)} \quad (89)$$

where:

$$\left[\phi_k\right]_{(i)} = 2\delta t \mid 0.5\mathcal{R}^{OF}[\mathbf{V}/r_s]^O_k(t) + 0.5\left[[\mathbf{V}/r_s]^F(t+\Delta t)\right]_{(i-1)} \mid \quad (90)$$

Approximations given by equations (66), (67) and (68) are still valid (change δt by $2\delta t$ in (68)), thus:

$$\left[[\mathbf{r}/r_s]^O_{k+1}\right]_{(i)} = [\mathbf{r}/r_s]^F \left(1 - \frac{\left[\phi^2_k\right]_{(i)}}{2}\right) -$$

$$\left(0.5\mathcal{R}^{OF}[\mathbf{V}/r_s]^O_k(t) + 0.5\left[[\mathbf{V}/r_s]^F(t+\Delta t)\right]_{(i-1)}\right)(2\delta t)\left(1 - \frac{\left[\phi^2_k\right]_{(i)}}{6}\right) \quad (91)$$

On the vertical, for 3D model:

$$\left[\eta^O_{k+1}\right]_{(i)} = \eta^F - 2\delta t\left(0.5\dot{\eta}^O_k(t) + 0.5\left[\dot{\eta}^F(t+\Delta t)\right]_{(i-1)}\right) \quad (92)$$

For $i=0$: $[[\mathbf{V}/r_s](t+\Delta t)]_{(i=0)} = [\mathbf{V}/r_s](t)$ and $[\dot{\eta}(t+\Delta t)]_{(i=0)} = \dot{\eta}(t)$ if **LPC_NESCT**=.T. ; $[[\mathbf{V}/r_s](t+\Delta t)]_{(i=0)} = 2[\mathbf{V}/r_s](t) - [\mathbf{V}/r_s](t-\Delta t)$ and $[\dot{\eta}(t+\Delta t)]_{(i=0)} = 2\dot{\eta}(t) - \dot{\eta}(t-\Delta t)$ if **LPC_NESCT**=.F. .

$*$ **First iteration of the research of SL trajectory:** One starts with $M_0 = F$, $\left[[\mathbf{V}/r_s]^F(t+\Delta t)\right]_{(i-1)}$ as a first guess for the spatio-temporally extrapolated horizontal wind, $[\phi_0]_{(i)} = 2\delta t \mid \left[[\mathbf{V}/r_s]^F(t+\Delta t)\right]_{(i-1)} \mid$, $\left[\dot{\eta}^F(t+\Delta t)\right]_{(i-1)}$ as a first guess for the spatio-temporally extrapolated η-coordinate vertical wind. Use equations (71) to (74) replacing δt by $2\delta t$ and the superscript M by O in the SL-trajectory research. This defines the coordinates of $[O_1]_{(i)}$; $[\mathbf{V}/r_s](t)$ and $\dot{\eta}(t)$ are interpolated at this point, that allows to compute the wind components which will be used for the next iteration of the research of SL trajectory.

$*$ **Following iterations of the research of SL trajectory:** $[\mathbf{V}^{'}/r_s]$ (of coordinates $([u^{'}/r_s],[v^{'}/r_s])$ is a generic notation for $0.5\mathcal{R}^{OF}[\mathbf{V}/r_s]_k^O(t)$+0.5 $\left[[\mathbf{V}/r_s]^F(t+\Delta t)\right]_{(i-1)}$. For horizontal displacement use equations (75) to (77) replacing δt by $2\delta t$ and the superscript M by O otherwise. For vertical displacement use equation

$$\left[\eta_{k+1}^O\right]_{(i)} = \eta^F - 2\delta t\left(0.5\dot{\eta}_k^O(t) + 0.5\left[\dot{\eta}^F(t+\Delta t)\right]_{(i-1)}\right) \tag{93}$$

$*$ **Alternate stable algorithm for 2TL SL scheme (LELTRA=.T. in NAMDYN):** For each iteration of the iterative centred-implicit scheme the algorithm is the same as for explicit schemes; at each iteration the RHS of the momentum equation is updated with the “provisional" $t+\Delta t$ information computed at the previous iteration.

6.2 Origin point O (subroutines LARMES and LAINOR2).

In a 3TL SL scheme, the particle is at the point O at the instant $t-\Delta t$ (t for the first integration step). In a 2TL SL scheme, the particle is at the point O at the instant 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 $(\widehat{[\mathbf{r}/r_s]^O,[\mathbf{r}/r_s]^F})$ is small (less than 10^o, what is generally satisfied), on can write for horizontal displacement:

$$[\mathbf{r}/r_s]^O - [\mathbf{r}/r_s]^F \simeq 2([\mathbf{r}/r_s]^M - [\mathbf{r}/r_s]^F) \tag{94}$$

For vertical displacement on can always write:

$$\eta^O - \eta^F = 2(\eta^M - \eta^F) \tag{95}$$

One denotes by:

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

Using the following identities:

$$\cos 2\phi = 2\cos^2\phi - 1 \tag{96}$$

$$\sin 2\phi = 2\sin\phi\cos\phi \tag{97}$$

the origin point horizontal coordinates can be computed by:

$$\sin\theta^O = \sin\theta^F\cos 2\phi - 2\cos\phi\left[\frac{[v^{'}/r_s]}{\mid[\mathbf{V}'/r_s]\mid}\cos\theta^F\sin\phi\right] \tag{98}$$

$$\cos\theta^O\cos(\lambda^O-\lambda^F) = \cos\theta^F\cos 2\phi + 2\cos\phi\left[\frac{[v^{'}/r_s]}{\mid[\mathbf{V}'/r_s]\mid}\sin\theta^F\sin\phi\right] \tag{99}$$

$$\cos\theta^O\sin(\lambda^O-\lambda^F) = -2\cos\phi\left[\frac{[u^{'}/r_s]}{\mid[\mathbf{V}'/r_s]\mid}\sin\phi\right] \tag{100}$$

Terms in brackets are already computed to determine M.

6.3 Refined recomputation of point O.

$*$ **Option L2TLFF for RW2TLFF=1:** Option (switch **L2TLFF** in **YOMDYN**) controls recomputation 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 $(2\mathbf{\Omega}\wedge a\mathbf{k})$ is computed (always analytically) at this improved position of O (so **L2TLFF** is active only if **LADVF**=.T. or **LADVFW**=.T.). Refined recomputation of point O is available only in a limited set of options. In the following sections 7, 8 and **??** discretised equations are written with notation O for all quantities. Equations system is integrated to find a first guess of $\mathbf{V}^F(t+\Delta t)$ and also a first guess of $\Pi_s(t+\Delta t)$ which provides $r_s(t+\Delta t)$, then $0.5([\mathbf{V}/r_s]^F + \mathcal{R}^{OF}[\mathbf{V}/r_s]^O)$ is used to recompute O. A correction $(2\mathbf{\Omega}\wedge a\mathbf{k})(O_{\mathrm{improved}}) - (2\mathbf{\Omega}\wedge\mathbf{k})(O)$ is analytically computed and added to wind equation to find the "improved" value of $\mathbf{V}^F(t+\Delta t)$. In the deep layer equations and in the cases where a multiplicative factor r_s/a is required, this factor remains interpolated at O and is never re-interpolated at the refined origin point (this is too tricky to code and too expensive also). Computations are currently made in routine **LAPINEB** and **LADINE**.

$*$ **Options L2TLFF for RW2TLFF between 0 and 1:** In this case the improved position ON of the origin point O is computed as a linear interpolation between O and its position for **RW2TLFF**=1 on a great circle bow on geographical sphere. For **RW2TLFF**=0. ON=O. For an idealised straight displacement with a wind of constant acceleration and constant direction, one can show that the "exact" position of the origin point is given by **RW2TLFF**=1/3 (Yessad, internal paper in French).

6.4 Remarks.

$*$ **Shallow water model:** Computations remain valid for horizontal coordinates (there is no vertical movement, equations containing η have not to be considered). There are remaining some old features (compute M in **LARMES2** even for **LPC_FULL**=.T., **LELTRA**=.T. or **LSETTLST**=.T., then O in **LAINOR2**).

$*$ **Trajectory going out of atmosphere for trajectories ending at a layer final point:** If the origin point O is found above the top of the atmosphere (resp. under the ground) it is put on the top of the atmosphere (resp. on the ground). Then the position of the medium point is recomputed (when necessary), that gives necessary a point between the bottom and the top of the atmosphere. At last the origin point is bounded by a vertical position η_O between the top of the atmosphere and the layer $l=1$, according to the namelist variable **VETAON** in **NAMDYN** (resp. between the layer $l=L$ and the ground, according to the namelist variable **VETAOX** in **NAMDYN**). Upper bound of O is $\eta_O = \eta_{l=1} + (\mathbf{VETAON} - 1)(\eta_{l=1} - \eta_{\bar{l}=0})$. Lower bound of O is $\eta_O = \eta_{l=L} + (1 - \mathbf{VETAOX})(\eta_{\bar{l}=L} - \eta_{l=L})$.

$*$ **Case of interpolations applied to half level variables:** That produces for example in the non-hydrostatic scheme when the half level variable w is advected instead of the full level vertical divergence variable (option **LGWADV**=.T.). In this case one needs to define an origin $O(\bar{l})$ for a half level trajectory (which ends at a half level final point $F(\bar{l})$). The following rules are applied to compute such a kind of trajectory:

- The horizontal displacement from $F(\bar{l})$ is a weighted average of the horizontal displacements from $F(l-1)$ and $F(l)$ (see below for vertical displacement).
- At the top (resp. bottom) the horizontal displacement from $F(\bar{l}=0)$ (resp. $F(\bar{l}=L)$) is equal to the horizontal displacement from $F(l=1)$ (resp. $F(l=L)$).

- The rule applied to the horizontal displacement is also applied to the vertical displacement for $\bar{l}$ between 1 and $L-1$. For example, if $\bar{l}$ is between 1 and $L-1$:

$$\eta_{O(\bar{l})} = + \left(1 - \frac{\eta_{O(\bar{l})} - \eta_{O(l)}}{\eta_{O(l+1)} - \eta_{O(l)}}\right) \eta_{O(l)} + \frac{\eta_{O(\bar{l})} - \eta_{O(l)}}{\eta_{O(l+1)} - \eta_{O(l)}} \eta_{O(l+1)}$$

- A particle coming from the top or the bottom has no vertical displacement (that assumes that $\dot{\eta} = 0$ at the top and the bottom, and so that excludes the options **LRUBC**=.T., **NDPSFI**=1 and $\Pi_{\mathrm{top}} > 0$ which are not consistent with the constraint $(\dot{\eta}_{\mathrm{top}} = 0; \dot{\eta}_{\mathrm{surf}} = 0)$).
- If the trajectory goes above the top of the atmosphere, it is bounded at the top of the atmosphere.
- If the trajectory goes below the surface, it is bounded at the surface.
- Computation of the position of $O(\bar{l})$ is done in routine **LARCINHA**.

∗ **Plane geometry (ALADIN):** Computation of the SL trajectory is made on the projected plane geometry. **ELARMES** and **ELARMES2** are called instead of **LARMES** and **LARMES2**.

∗ **Treatment of $\dot{\eta}$ in the upper stratosphere:** For some options of the code the horizontal interpolations applied on $\dot{\eta}$ are replaced by “least-square" interpolations; that allows to remove some instabilities. See documentation (IDSVTSM) for more details.

7 The SL discretisation of the 2D shallow-water system of equations (spherical geometry).

7.1 Momentum equation.

* Definition of X, $\mathcal{A}$ and $\mathcal{B}$.

$$X = \vec{V} + \delta_{\vec{V}}(2\vec{\Omega} \wedge \vec{r}) \tag{101}$$

$$\mathcal{A} = [-2(1 - \delta_{\vec{V}})(\vec{\Omega} \wedge \vec{V})] - \nabla\Phi \tag{102}$$

$$\mathcal{B} = -\nabla\Phi \tag{103}$$

* Remarks.

- In the cycle 37T1 of ARPEGE/IFS Coriolis term can be treated explicitly ($\delta_{\vec{V}} = 0$) or implicitly ($\delta_{\vec{V}} = 1$). Use switch **LADVF** in namelist **NAMDYN**. **LADVF**=.F. corresponds to ($\delta_{\vec{V}} = 0$). **LADVF**=.T. corresponds to ($\delta_{\vec{V}} = 1$). If **LADVF**=.T., term ($2\mathbf{\Omega} \wedge \mathbf{r}$) is analytically computed.
- For a limited set of options, term ($2\mathbf{\Omega} \wedge \mathbf{r}$) can be recomputed at an improved position of the origin point (**RW2TLFF**>0 in **NAMDYN**).
- Coriolis term can also be put in the semi-implicit scheme by tuning β_{Co} (which has a sense only if **LADVF**=.F.). Values $\beta_{\mathrm{Co}} = 0$ (**LIMPF**=.F. in namelist **NAMDYN**) and $\beta_{\mathrm{Co}} = 1$ (**LIMPF**=.T.) are available in the cycle 37T1. Caution: do not use **LIMPF**=.T. in variable resolution in cycle 37T1 (formulation of spectral computations is not correct in this case for the semi-implicit scheme).
- If β=1, the non linear term $([-2(1 - \delta_{\vec{V}})(\vec{\Omega} \wedge \vec{V})] - \nabla\Phi) - \beta(-\nabla\Phi)$ is zero.

7.2 Continuity equation.

Conventional formulation (positive value of NVLAG in the namelist NAMDYN).

* Definition of X, $\mathcal{A}$ and $\mathcal{B}$.

$$X = (\Phi - \Phi_s) \tag{104}$$

$$\mathcal{A} = -(\Phi - \Phi_s)D + \delta_{TR}\vec{V}\nabla(\Phi_s) \tag{105}$$

$$\mathcal{B} = -\Phi^* \overline{M}^2 D^{'} \tag{106}$$

$\delta_{TR} = 1$ plays the same role as the “Tanguay-Ritchie" modification for 3D model.

Lagrangian formulation (negative value of NVLAG in the namelist NAMDYN).

$*$ Definition of X, $\mathcal{A}$ and $\mathcal{B}$.

$$X = (\Phi - \Phi_{\rm s})J \tag{107}$$

$$\mathcal{A} = 0 \tag{108}$$

$$\mathcal{B} = -\Phi^* \overline{M}^2 D^{'} J \tag{109}$$

where J is a “Jacobian" quantity defined by its Lagrangian derivative (see equation (12)).

$*$ Calculation of J if 3TL scheme. Jacobian quantities are computed by: $J^- = (1 - \Delta t D^o)/(1 + \Delta t D^o)$, $J^o = 1/(1 + \Delta t D^o)$ and $J^+ = 1$.

$*$ Discretisation if 2TL scheme. In cycle 37T1 of ARPEGE/IFS the option **NVLAG**=-2 is not coded for the 2TLSL scheme.

7.3 Quantities to be interpolated.

When researching the medium point by an iterative algorithm, the interpolation at the medium point (or the origin point **LSETTLST**=.T. in **NAMDYN**) of the two components of the horizontal wind is needed: a 12 points interpolation is used. For other quantities to be interpolated, see section 5. For more details about interpolations, see section 11.

8 The SL discretisation of the 3D primitive equation model.

Remark: the detailed discretisation of each part of the RHS is given in the documentation (IDEUL). Some notations used in the expression of linear term $\mathcal{B}$ (like τ, γ, μ, ν) are given in the documentation (IDSI).

8.1 Thin layer formulation of the momentum equation.

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

$$X = \vec{V} + \delta_{\vec{V}}(2\vec{\Omega} \wedge \vec{r}) \tag{110}$$

$$\mathcal{A} = -2(1 - \delta_{\vec{V}})(\mathbf{\Omega} \wedge \mathbf{V}) - \nabla\Phi - RT\nabla(\log \Pi) \tag{111}$$

$$\mathcal{B} = -\nabla \left[\gamma T + \frac{R_d T^*}{\Pi_{\mathrm{s}}{}^*} \Pi_{\mathrm{s}} \right] + \beta_{Co}[-2(1 - \delta_{\vec{V}})(\vec{\Omega} \wedge \vec{V})] \tag{112}$$

$$\mathcal{F} = \mathbf{F_V} \tag{113}$$

Top:

$$\mathbf{V}_{\eta=0} = \mathbf{V}_{l=1} \tag{114}$$

Bottom if $\delta m = 0$:

$$\mathbf{V}_{\eta=1} = \mathbf{V}_{l=L} \tag{115}$$

Bottom if $\delta m = 1$:

$$\mathbf{V}_{\eta=1} = 0 \tag{116}$$

* Remarks.

- In the cycle 37T1 of ARPEGE/IFS Coriolis term can be treated explicitly ($\delta_{\vec{V}} = 0$) or implicitly ($\delta_{\vec{V}} = 1$). Use switch **LADVF** in namelist **NAMDYN**. **LADVF**=.F. corresponds to ($\delta_{\vec{V}} = 0$). **LADVF**=.T. corresponds to ($\delta_{\vec{V}} = 1$). If **LADVF**=.T., term $(2\mathbf{\Omega} \wedge \mathbf{r})$ is analytically computed.
- For a limited set of options, term $(2\mathbf{\Omega} \wedge \mathbf{r})$ can be recomputed at an improved position of the origin point (**RW2TLFF**>0 in **NAMDYN**).
- Coriolis term can also be put in the semi-implicit scheme by tuning β_{Co} (which has a sense only if **LADVF**=.F.). Values $\beta_{\mathrm{Co}} = 0$ (**LIMPF**=.F. in namelist **NAMDYN**) and $\beta_{\mathrm{Co}} = 1$ (**LIMPF**=.T.) are available in the cycle 37T1. Caution: do not use **LIMPF**=.T. in variable resolution in cycle 37T1 (formulation of spectral computations is not correct in this case for the semi-implicit scheme).

8.2 White and Bromley deep layer formulation of the momentum equation.

∗ Definition of X, $\mathcal{A}$, $\mathcal{B}$ and $\mathcal{F}$, top and bottom values.

$$X = \vec{V} + \delta_{\vec{V}}(2\vec{\Omega} \wedge \vec{r}) \tag{117}$$

$$\mathcal{A} = (1 - \delta_{\vec{V}})(-2\mathbf{\Omega} \wedge \mathbf{V} - 2\mathbf{\Omega} \wedge W\mathbf{k}) - \frac{W}{r_s}\mathbf{V} - \nabla\Phi - (RT + \mu_{\mathrm{s}} R_{\mathrm{d}} T_{\mathrm{r}})\nabla(\log \Pi) \tag{118}$$

$$\mathcal{B} = -M\nabla^{'} [\gamma T + R_{\mathrm{d}} T^{*} \log(\Pi_{\mathrm{s}})] + \beta_{\mathrm{Co}}[-2(1 - \delta_{\vec{V}})(\mathbf{\Omega} \wedge \mathbf{V})] \tag{119}$$

$$\mathcal{F} = \mathbf{F}_{\mathbf{V}} \tag{120}$$

Top:

$$\mathbf{V}_{\eta=0} = \mathbf{V}_{l=1} \tag{121}$$

Bottom if $\delta m = 0$:

$$\mathbf{V}_{\eta=1} = \mathbf{V}_{l=L} \tag{122}$$

Bottom if $\delta m = 1$:

$$\mathbf{V}_{\eta=1} = 0 \tag{123}$$

∗ Remarks.

- In the cycle 37T1 of ARPEGE/IFS Coriolis term can be treated explicitly ($\delta_{\vec{V}} = 0$) or implicitly ($\delta_{\vec{V}} = 1$). Use switches **LADVF** and **LADVFW** in namelist **NAMDYN**. (**LADVF**;**LADVW**)=(.F.;.F.) corresponds to ($\delta_{\vec{V}} = 0$). (**LADVF**;**LADVW**)=(.T.;.T.) corresponds to ($\delta_{\vec{V}} = 1$). If (**LADVF**;**LADVW**)=(.T.;.F.) term ($2\mathbf{\Omega} \wedge a\mathbf{k}$) is treated implicitly and the remaining Coriolis term is treated explicitly. If (**LADVF**;**LADVW**)=(.F.;.T.) term ($2\mathbf{\Omega} \wedge (r_s - a)\mathbf{k}$) is treated implicitly and the remaining Coriolis term is treated explicitly. If at least **LADVF**=.T. or **LADVFW**=.T., term ($2\mathbf{\Omega} \wedge a\mathbf{k}$) is analytically computed. If **LADVFW**=.T., r_s/a (at $t - \Delta t$ if SL3TL or t if SL2TL) has to be interpolated at O, and r_s/a has also to be computed at $t + \Delta t$ using a provisional value of $\log \Pi_{\mathrm{s}}$ at $t + \Delta t$.
- For a limited set of options, term ($2\mathbf{\Omega} \wedge a\mathbf{k}$) can be recomputed at an improved position of the origin point (**RW2TLFF**>0 in **NAMDYN**).
- The horizontal part of the Coriolis term can also be put in the semi-implicit scheme by tuning β_{Co} (which has a sense only if **LADVF**=.F.). Values $\beta_{\mathrm{Co}} = 0$ (**LIMPF**=.F. in namelist **NAMDYN**) and $\beta_{\mathrm{Co}} = 1$ (**LIMPF**=.T.) are available in the cycle 37T1. Caution: do not use **LIMPF**=.T. in variable resolution in cycle 37T1 (formulation of spectral computations is not correct in this case for the semi-implicit scheme). The combination (**LIMPF**;**LADVW**)=(.T.;.T.) is possible but does not eliminate completely a residual explicit term linked to Coriolis force in the RHS of the momentum equation.

8.3 Thermodynamic equation.

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

$$X = T + \delta_{TR} \frac{\alpha_{\rm T} \Phi_{\rm s}}{R_{\rm d} T_{ST}} \tag{124}$$

$$\mathcal{A} = \frac{RT}{c_{\rm p}} \frac{\omega}{\Pi} + \delta_{TR} \frac{\alpha_T}{R_d T_{ST}} \vec{V} \nabla(\Phi_s) + \delta_{TR} \frac{\Phi_s}{R_d T_{ST}} \left(\dot{\eta} \frac{d\alpha_T}{d\eta} \right) \tag{125}$$

$$\mathcal{B} = -\tau(\overline{M}^2 D^{'}) \tag{126}$$

$$\mathcal{F} = F_{\rm T} \tag{127}$$

Top:

$$T_{\eta=0} = T_{l=1} \tag{128}$$

Bottom if $\delta m = 0$:

$$T_{\eta=1} = T_{l=L} \tag{129}$$

Bottom if $\delta m = 1$ (output of physics):

$$T_{\eta=1} = T_{\rm s} \tag{130}$$

8.4 Thin layer formulation of the continuity equation.

$*$ Definition of X, $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{F}$.

$$X = \log \Pi_{\rm s} + \delta_{TR} \frac{\Phi_s}{R_d T_{st}} \tag{131}$$

$$\mathcal{A} = -\frac{1}{\Pi_{\rm s}} \int_{\eta=0}^{\eta=1} \nabla \left(\vec{V} \frac{\partial \Pi}{\partial \eta} \right) d\eta + \vec{V} \nabla \left[\log \Pi_{\rm s} + \delta_{TR} \frac{\Phi_s}{R_d T_{st}} \right] \tag{132}$$

$$\mathcal{B} = -\frac{\overline{M}^2}{M^2} \nu D \tag{133}$$

$$\mathcal{F} = \left(\frac{F_{\rm m}}{\Pi_{\rm s}} \right) \tag{134}$$

$*$ Remarks:

- $\mathcal{A}$ is a sum of 3D terms (the advection term) and 2D terms (the other terms).
- $\mathcal{B}$ and $\mathcal{F}$ are 2D terms (vertical integrals).

8.5 White and Bromley deep layer formulation of the continuity equation.

$*$ Definition of X, $\mathcal{A}$, $\mathcal{B}$, and $\mathcal{F}$.

$$X = \log \Pi_{\mathrm{s}} + \delta_{TR} \frac{\Phi_s}{R_d T_{st}} \tag{135}$$

$$\mathcal{A} = - \left[\frac{a^2}{{r_s}^2} \right]_{\eta=1} \frac{1}{\Pi_{\mathrm{s}}} \int_{\eta=0}^{\eta=1} \left[\frac{r_s}{a} \nabla \right] \left(\frac{r_s}{a} \mathbf{V} \frac{\partial \Pi}{\partial \eta} \right) d\eta + \frac{a}{r_s} \mathbf{V} \left[\frac{r_s}{a} \nabla \right] \left[\log \Pi_{\mathrm{s}} + \delta_{TR} \frac{\Phi_s}{R_d T_{st}} \right] - \frac{1}{\Pi_{\mathrm{s}}} \left[\dot{\eta} \frac{\partial \Pi}{\partial \eta} \right]_{\eta=1} + \frac{1}{\Pi_{\mathrm{s}}} \left[\frac{a^2}{{r_s}^2} \right]_{\eta=1} \left[\frac{{r_s}^2}{a^2} \right.$$

$$\mathcal{B} = -\nu (\overline{M}^2 D^{'}) \tag{137}$$

$$\mathcal{F} = \left(\frac{F_{\mathrm{m}}}{\Pi_{\mathrm{s}}} \right) \tag{138}$$

$*$ Remarks:

- $\mathcal{A}$ is a sum of 3D terms (the advection term) and 2D terms (the other terms).
- $\mathcal{B}$ and $\mathcal{F}$ are 2D terms (vertical integrals).

8.6 Moisture equation.

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

$$X = q \tag{139}$$

$$\mathcal{A} = 0 \tag{140}$$

$$\mathcal{B} = 0 \tag{141}$$

$$\mathcal{F} = F_{\mathrm{q}} \tag{142}$$

Top:

$$q_{\eta=0} = q_{l=1} \tag{143}$$

Bottom if $\delta m = 0$:

$$q_{\eta=1} = q_{l=L} \tag{144}$$

Bottom if $\delta m = 1$ (see **CPQSOL**, relative humidity is the same for $\eta = \eta_L$ and $\eta = 1$):

$$q_{\eta=1} = q_{\mathrm{surf}} \tag{145}$$

8.7 Other advectable GFL variables.

Equations are discretised as for humidity equation. Vertical boundary conditions: quantities are assumed constant above the middle of the upper layer and below the middle of the lower layer in case $\delta m = 0$; quantities are assumed constant above the middle of the upper layer in case $\delta m = 1$; quantities other than q are assumed to be zero at the surface in case $\delta m = 1$.

8.8 Case of lagged physics.

All the previous discretisations have been written with not lagged physics (interpolated at the origin point O).

- For lagged physics (**LAGPHY**=.T., **LSLPHY**=.F.): the previous discretisations are done without physics, then the provisional

  $$(X^{+} - (1+\epsilon_{\mathcal{X}})\frac{\Delta t}{2}\beta\mathcal{B}^{+} + (1+\epsilon_{\mathcal{X}})\frac{\Delta t}{2}\beta\mathcal{B}^{o})_F$$

  or

  $$(X^{+} - (1+\epsilon)\frac{\Delta t}{2}\beta\mathcal{B}^{+} + (1+\epsilon)\frac{\Delta t}{2}\beta\mathcal{B}^{o})_F$$

  is used as input to the lagged physics.
- For split physics used at ECMWF (**LEPHYS**=.T., **LAGPHY**=.T., **LSLPHY**=.T.): one part of the physics is interpolated at the origin point (t or $t-\Delta t$ physics according to **LTWOTL**), the remainder is evaluated at the final point ($t+\Delta t$ physics). The physical contribution is put in a separate interpolation buffer (name **P(X)P9**) and tri-linearly interpolated. The way to compute the non-lagged contribution is different than the way used at METEO-FRANCE: compute it at the previous timestep as a lagged contribution, then saving it (by the routine **GPSAVTEND**) from one timestep to the following one (where it is restored by calling the routine **GPGETTEND** and added to the interpolation buffer by calling the routine **GPADDSLPHY**). Partition between the non-lagged and lagged contribution is done by a linear partition of coefficient **RSLWX**.

8.9 Quantities to be interpolated (computation under subroutine LACDYN).

Research of trajectory.

When researching the medium point by an iterative algorithm, the interpolation at the medium point (or the origin point **LSETTLST**=.T., **LELTRA**=.T. or **LPC_FULL**=.T.) of $([U/r_s], [V/r_s], \dot{\eta})$ is needed: a tri-linear interpolation is performed. For more details about interpolations, see section 11.

RHS of equations.

The list of quantities to be interpolated has been described in subsections 5.2, 5.3 and 5.4 for each type of equation.

Additional quantities to be interpolated at the origin point if RW2TLFF>0.

The two components of the $[\mathbf{V}/r_s]$ at time t (if 2TL SL scheme) or $t-\Delta t$ (if 3TL SL scheme) when not available after the other interpolations (for example if not lagged physics). These additional interpolations are useless if lagged physics (or adiabatic run) and **NWLAG**=3.

9 $\mathcal{R}$ operator.

9.1 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 $\mathcal{R}^{OF}$ is defined:

$$\mathbf{V}^{'} = \mathcal{R}^{OF}(\mathbf{V}) \tag{146}$$

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

$$\begin{pmatrix} u^{'} \\ v^{'} \end{pmatrix} = \begin{pmatrix} p & q \\ -q & p \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} \tag{147}$$

where:

$$p = \frac{\mathbf{i}^F\mathbf{i}^O + \mathbf{j}^F\mathbf{j}^O}{1 + \mathbf{k}^F\mathbf{k}^O} = \frac{\cos\theta^F\cos\theta^O + (1 + \sin\theta^F\sin\theta^O)\cos(\lambda^F - \lambda^O)}{1 + \cos\phi} \tag{148}$$

$$q = \frac{\mathbf{i}^F\mathbf{j}^O + \mathbf{j}^F\mathbf{i}^O}{1 + \mathbf{k}^F\mathbf{k}^O} = \frac{(\sin\theta^F + \sin\theta^O)\sin(\lambda^F - \lambda^O)}{1 + \cos\phi} \tag{149}$$

(Notations θ^O,θ^F,λ^O,λ^F,ϕ: see section 6.).

p and q verify the following identity:

$$p^2 + q^2 = 1 \tag{150}$$

Computation of p and q is made in subroutine **LARCHE**.

9.2 Tilting.

The coordinates of $\mathbf{V}^{'}$ and $\mathbf{V}$ are linked by the following relationship:

$$\begin{pmatrix} u^{'} \\ v^{'} \end{pmatrix} = \begin{pmatrix} GNORDM & GNORDL \\ -GNORDL & GNORDM \end{pmatrix} \begin{pmatrix} p & q \\ -q & p \end{pmatrix} \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} \tag{151}$$

where:

$$\cos\alpha = \frac{2c}{A\cos\Theta^O}[\sin\theta_{\rm p}\cos\theta^O - \sin\theta^O\cos\theta_{\rm p}\cos(\lambda^O - \lambda_{\rm p})] \tag{152}$$

$$\sin\alpha = \frac{2c}{A\cos\Theta^O}[\cos\theta_{\rm p}\sin(\lambda^O - \lambda_{\rm p})] \tag{153}$$

$$A = (1 + c^2) + (1 - c^2)(\sin\theta_{\rm p}\sin\theta^O + \cos\theta_{\rm p}\cos\theta^O\cos(\lambda^O - \lambda_{\rm p})) \tag{154}$$

and where:

- c is the stretching coefficient.
- Θ^O is the latitude on the computational sphere of the origin point O.
- $(\theta_{\rm p}, \lambda_{\rm p})$ 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**.

9.3 Plane geometry (ALADIN).

The curvature of the Earth is now taken into account in computing an operator $\mathcal{R}^{OF}$ in the routine **ELARCHE** instead of computing curvature terms. Expressions of p and q are different from the ones of ARPEGE and are not detailed here.

10 Computation of longitudes and latitudes on the computational sphere.

For interpolations it is necessary to compute (Θ^O, Λ^O), latitude and longitude of the interpolation point O in the computational sphere. The iterative algorithm allowing to find O gives (θ^O, λ^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 (θ^F, λ^F) are the coordinates of the final point on the geographical sphere). Transform formulae giving (Θ, Λ) on the computational sphere once knowing (θ, λ) on the geographical sphere are given by equations (155) to (157).

$$\sin\Theta = \frac{(1-c^2) + (1+c^2)(\sin\theta_{\mathrm{p}}\sin\theta + \cos\theta_{\mathrm{p}}\cos\theta\cos(\lambda-\lambda_{\mathrm{p}}))}{A} \tag{155}$$

$$\cos\Theta\cos\Lambda = \frac{2c(\cos\theta_{\mathrm{p}}\sin\theta - \sin\theta_{\mathrm{p}}\cos\theta\cos(\lambda-\lambda_{\mathrm{p}}))}{A} \tag{156}$$

$$\cos\Theta\sin\Lambda = \frac{2c\cos\theta\sin(\lambda-\lambda_{\mathrm{p}})}{A} \tag{157}$$

where:

- $A = (1+c^2) + (1-c^2)(\sin\theta_{\mathrm{p}}\sin\theta + \cos\theta_{\mathrm{p}}\cos\theta\cos(\lambda-\lambda_{\mathrm{p}}))$
- c is the stretching coefficient.
- $(\theta_{\mathrm{p}}, \lambda_{\mathrm{p}})$ are the latitude and longitude on the geographical sphere of the stretching pole.
- Computation of Θ, Λ is made in subroutine **LARCHE**.

$*$ **Plane geometry (ALADIN):** The SL trajectory is already computed on the computational grid, so equivalent transformation formulae from geographical space to computational space are useless.

11 Interpolations and weights computations.

11.1 Interpolation grid and weights (subroutine LASCAW).

Horizontal interpolation grid and weights for bi-linear interpolations.

$*$ **Definitions:** A 16 points horizontal grid is defined as it is shown in figure 11.1, but only 4 of these 16 points are used in the interpolations. The interpolation point O (medium or origin point) is between B_1, C_1, B_2 and C_2. Λ and Θ are the longitudes and latitudes on the computational sphere. The following 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:
$$ZDLO2 = \frac{\Lambda_O - \Lambda_{B_2}}{\Lambda_{C_2} - \Lambda_{B_2}}$$
- meridian weight:
$$ZDLAT = \frac{\Theta_O - \Theta_{B_1}}{\Theta_{B_2} - \Theta_{B_1}}$$

$*$ **Computations:**

- The weights $ZDLO1$ and $ZDLO2$ are computed then stored in the array **PDLO**.
- The weight $ZDLAT$ is computed then stored in the array **PDLAT**.
- The memory address (in SL arrays) of the data concerning the points A_1 and A_2 are computed then stored in the array **KL0** or **KLH0**. Memory address of the data concerning the points B_1, B_2, C_1 and C_2 can be easily computed in interpolations routines knowing these ones of A_1 and A_2.
- Interpolations use data of points B_1, B_2, C_1 and C_2.

Vertical interpolation grid and weights for vertical linear interpolations.

$*$ **Definitions:** A 4 points vertical grid is defined as it is shown in figure 11.2, but only 2 of these 4 points are used in the interpolations. The interpolation point O (medium or origin point) is between T_{l+1} and T_{l+2}. The vertical weight is defined by:

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

$*$ **Computations:**

- The weight $ZDVER$ is computed then stored in the array **PDVER**.
- The level number l of T_l is stored in the array **KLEV**.
- Interpolations use data of points T_{l+1} and T_{l+2}.

$*$ **Remark:** The same formulae and computations are valid for half level data, simply replace the layer index l by the half level index $\bar{l}$.

Horizontal interpolation grid and weights for 12 points cubic interpolations.

∗ **Definitions:** A 16 points horizontal grid is defined as it is shown in figure 11.3, but only 12 of these 16 points are used in the interpolations. The interpolation point O (medium or origin point) is between B_1, C_1, B_2 and C_2. The following weights are defined as follows:

- zonal linear weights for latitudes 0, 1, 2, 3:

$$ZDLO0 = \frac{\Lambda_O - \Lambda_{B_0}}{\Lambda_{C_0} - \Lambda_{B_0}}$$

$$ZDLO1 = \frac{\Lambda_O - \Lambda_{B_1}}{\Lambda_{C_1} - \Lambda_{B_1}}$$

$$ZDLO2 = \frac{\Lambda_O - \Lambda_{B_2}}{\Lambda_{C_2} - \Lambda_{B_2}}$$

$$ZDLO3 = \frac{\Lambda_O - \Lambda_{B_3}}{\Lambda_{C_3} - \Lambda_{B_3}}$$

- zonal cubic weights for latitude 1:

$$ZCLO11 = f_1(ZDLO1)$$
$$ZCLO12 = f_2(ZDLO1)$$
$$ZCLO13 = f_3(ZDLO1)$$

  where:
  - $f_1(\alpha) = (\alpha + 1)(\alpha - 2)(\alpha - 1)/2$
  - $f_2(\alpha) = -(\alpha + 1)(\alpha - 2)\alpha/2$
  - $f_3(\alpha) = \alpha(\alpha - 1)(\alpha + 1)/6$

- zonal cubic weights for latitude 2:

$$ZCLO21 = f_1(ZDLO2)$$
$$ZCLO22 = f_2(ZDLO2)$$
$$ZCLO23 = f_3(ZDLO2)$$

- meridian cubic weights:

$$ZCLA1 = \frac{(\Theta_O - \Theta_{B_0})(\Theta_O - \Theta_{B_2})(\Theta_O - \Theta_{B_3})}{(\Theta_{B_1} - \Theta_{B_0})(\Theta_{B_1} - \Theta_{B_2})(\Theta_{B_1} - \Theta_{B_3})}$$

$$ZCLA2 = \frac{(\Theta_O - \Theta_{B_0})(\Theta_O - \Theta_{B_1})(\Theta_O - \Theta_{B_3})}{(\Theta_{B_2} - \Theta_{B_0})(\Theta_{B_2} - \Theta_{B_1})(\Theta_{B_2} - \Theta_{B_3})}$$

$$ZCLA3 = \frac{(\Theta_O - \Theta_{B_0})(\Theta_O - \Theta_{B_1})(\Theta_O - \Theta_{B_2})}{(\Theta_{B_3} - \Theta_{B_0})(\Theta_{B_3} - \Theta_{B_1})(\Theta_{B_3} - \Theta_{B_2})}$$

∗ **Computations:**

- The zonal linear weights $ZDLO0$, $ZDLO1$, $ZDLO2$ and $ZDLO3$ are computed then stored in the array **PDLO**.
- The zonal cubic weights $ZCLO11$, $ZCLO12$, $ZCLO13$, $ZCLO21$, $ZCLO22$, $ZCLO23$ are computed then stored in the array **PCLO**.
- The meridian cubic weights $ZCLA1$, $ZCLA2$ and $ZCLA3$ are computed then stored in the array **PCLA**. One can notice that the denominators of $ZCLA1$, $ZCLA2$ and $ZCLA3$ do not depend on coordinates of O and can be pre-computed in the subroutine **SULEG**, in array **RIPI**.
- The memory address (in SL arrays) of the data concerning the points A_0, A_1, A_2 and A_3 are stored in the array **KL0** or **KLH0**. Once knowing these addresses one can easily retrieve the other points addresses in the interpolations routines.
- Interpolations use data of the following 12 points B_0, C_0, A_1, B_1, C_1, D_1, A_2, B_2, C_2, D_2, B_3 and C_3.

$*$ **Extension to diffusive interpolations (SLHD):** The way to take account of the diffusive properties of interpolations has been redesigned, and now it is completely contained in the calculation of cubic weights: we pass from conventional cubic interpolations to diffusive SLHD cubic interpolations simply by changing the way of computing the cubic weights. The comprehensive way of computing the SLHD cubic weights is not provided in this documentation (because it leads to rather tricky formulae), but details about the calculations can be found in documentation (IDSLIF2) and looking in routine **LASCAW** (**ELASCAW** in ALADIN). We just give a sum-up of the way to compute the cubic weights.

- Meridian diffusive cubic weights:

$$ZCLA1_{\mathrm{slhd}} = ZCLA1 + ZINCRM1$$

$$ZCLA2_{\mathrm{slhd}} = ZCLA2 + ZINCRM2$$

$$ZCLA3_{\mathrm{slhd}} = ZCLA3 + ZINCRM3$$

  where each of increments $ZINCRM1$, $ZINCRM2$ and $ZINCRM3$ is a linear combination of cubic Lagrangian weights $ZCLA1$, $ZCLA2$ and $ZCLA3$ and diffusive weights (linear for **LSLHD_OLD**=.T., quadratic otherwise).

  Coefficients of these linear combinations depend on:

  - Constants in time C_{sldw}, describing Laplacian smoother.
  - A pre-computed quantity κ (computed in **GP_KAPPA**) depending on horizontal flow deformation, ranging from 0 to 1; κ is equal to 1 if **LSLHD_STATIC**=T.
  - Lower and upper bounds ($\kappa_{\min}$ and $\kappa_{\max}$) respectively stored in variables **SLHDKMIN** and **SLHDKMAX**, used to construct limit interpolators (0 - cubic Lagrange, 1 - linear/quadratic; they are not restricted to range 0-1).

- Zonal diffusive cubic weights for latitude number 1:

$$ZCLO11_{\mathrm{slhd}} = ZCLO11 + ZINCRL11$$

$$ZCLO12_{\mathrm{slhd}} = ZCLO12 + ZINCRL12$$

$$ZCLO13_{\mathrm{slhd}} = ZCLO13 + ZINCRL13$$

  where each of increments $ZINCRL11$, $ZINCRL12$ and $ZINCRL13$ is a linear combination of cubic Lagrangian weights $ZCLO11$, $ZCLO12$ and $ZCLO13$ and diffusive weights (linear for **LSLHD_OLD**=.T., quadratic otherwise).

  Coefficients of these linear combinations depend on:

  - Constants like C_{slhdepsh}.
  - κ (see above).
  - Lower and upper bounds $\kappa_{\min}$ and $\kappa_{\max}$ (see above).

- Zonal diffusive cubic weights for latitude number 2:

$$ZCLO21_{\mathrm{slhd}} = ZCLO21 + ZINCRL21$$

$$ZCLO22_{\mathrm{slhd}} = ZCLO22 + ZINCRL22$$

$$ZCLO23_{\mathrm{slhd}} = ZCLO23 + ZINCRL23$$

  where each of increments $ZINCRL21$, $ZINCRL22$ and $ZINCRL23$ is a linear combination of cubic Lagrangian weights $ZCLO21$, $ZCLO22$ and $ZCLO23$ and diffusive weights (linear for **LSLHD_OLD**=.T., quadratic otherwise).

  Coefficients of these linear combinations depend on C_{slddepsh}, κ, $\kappa_{\min}$, $\kappa_{\max}$ like the zonal weights for latitude number 1.

Several options of SLHD smoothing are available, according to values of the keys **LSLHDQUAD** and **LSLHD_OLD**.

In part 11.2, cubic interpolations are written with conventional cubic weights: replace conventional cubic weights by their SLHD counterpart for diffusive cubic interpolations.

$*$ **Extension to 3D-turbulence:** not described in detail; additional cubic weights are computed if **L3DTURB**=T.

Vertical interpolation grid and weights for vertical cubic 4 points interpolations.

A 4 points vertical grid is defined as it is shown in figure 11.2. The interpolation point O (medium or origin point) is between T_{l+1} and T_{l+2}. The vertical weights are defined by:

$$ZCVE1 = \frac{(\eta_O - \eta_{T_l})(\eta_O - \eta_{T_{l+2}})(\eta_O - \eta_{T_{l+3}})}{(\eta_{T_{l+1}} - \eta_{T_l})(\eta_{T_{l+1}} - \eta_{T_{l+2}})(\eta_{T_{l+1}} - \eta_{T_{l+3}})}$$

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

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

$*$ **Computations:**

- The vertical weights $ZCVE1$, $ZCVE2$ and $ZCVE3$ are computed then stored in the array **PVINTW**. One can notice that the denominators of $ZCVE1$, $ZCVE2$ and $ZCVE3$ do not depend on coordinates of O and can be pre-computed in the subroutine **SUVERT**, in the array **VCUICO**.
- The level number l of T_l is stored in the array **KLEV**.
- Interpolations use data of points T_l, T_{l+1}, T_{l+2} and T_{l+3}.

$*$ **Remark:** The same formulae and computations are valid for half level data, simply replace the layer index l by the half level index $\bar{l}$, and use array **VCUICOH**.

$*$ **Extension to diffusive interpolations (SLHD):** Like we do for horizontal weights, the semi-Lagrangian diffusion can be taken into account by modifying vertical cubic weights as follow:

$$ZCVE1_{\text{slhd}} = ZCVE1 + ZINCRV1$$

$$ZCVE2_{\text{slhd}} = ZCVE2 + ZINCRV2$$

$$ZCVE3_{\text{slhd}} = ZCVE3 + ZINCRV3$$

Expressions giving vertical increments $ZINCRV1$ to $ZINCRV3$ have a shape similar to those of horizontal increments, and still depend on quantities like κ, $\kappa_{\min}$, $\kappa_{\max}$.

In part 11.2, cubic interpolations are written with conventional cubic weights: replace conventional cubic weights by their SLHD counterpart for diffusive cubic interpolations.

Vertical interpolation grid and weights for vertical cubic Hermite interpolations.

This part is valid only for interpolations of full level data.

A 4 points vertical grid is defined as it is shown in figure 11.2. 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 $\left(\dot{\eta}\frac{\partial X}{\partial \eta}\right)/\dot{\eta}$, but with additional approximations allowing to avoid horizontal interpolations for term $\left(\dot{\eta}\frac{\partial \Pi}{\partial \eta}\right)$.

- For layers other than the first or the last layer, discretisation follows:

$$\left(\frac{\partial X}{\partial \eta}\right)_{l+1} = \frac{0.5(X_{l+2} - X_{l})}{\eta_{\bar{l}+1} - \eta_{\bar{l}}} \tag{158}$$

- For layer $l = 1$, discretisation assumes that $\left(\dot{\eta}\frac{\partial \Pi}{\partial \eta}\right)_{\bar{l}=0} = 0$ (valid only if **LRUBC**=.F.); discretisation follows:

$$\left(\frac{\partial X}{\partial \eta}\right)_{l=1} = \frac{(X_{l=2} - X_{l=1})}{\eta_{\bar{l}=1} - \eta_{\bar{l}=0}} \tag{159}$$

- For layer $l = L$, discretisation assumes that $\left(\dot{\eta}\frac{\partial \Pi}{\partial \eta}\right)_{\bar{l}=L} = 0$; discretisation follows:

$$\left(\frac{\partial X}{\partial \eta}\right)_{l=L} = \frac{(X_{l=L} - X_{l=L-1})}{\eta_{\bar{l}=L} - \eta_{\bar{l}=L-1}} \tag{160}$$

The following weights are computed:

- For an interpolation point included between layers 2 and $L-1$ ($l \geq 1$ and $l \leq L-3$) :

$$VDERW11 = \frac{0.5(\eta_{l+2} - \eta_{l+1})}{\eta_{\bar{l}+1} - \eta_{\bar{l}}}$$

$$VDERW21 = \frac{0.5(\eta_{l+2} - \eta_{l+1})}{\eta_{\bar{l}+1} - \eta_{\bar{l}}}$$

$$VDERW12 = \frac{0.5(\eta_{l+2} - \eta_{l+1})}{\eta_{\bar{l}+2} - \eta_{\bar{l}+1}}$$

$$VDERW22 = \frac{0.5(\eta_{l+2} - \eta_{l+1})}{\eta_{\bar{l}+2} - \eta_{\bar{l}+1}}$$

- For an interpolation point included between layers 1 and 2:

$$VDERW11 = 0$$

$$VDERW21 = \frac{(\eta_{l=2} - \eta_{l=1})}{\eta_{\bar{l}=1} - \eta_{\bar{l}=0}}$$

$$VDERW12 = \frac{0.5(\eta_{l=2} - \eta_{l=1})}{\eta_{\bar{l}=2} - \eta_{\bar{l}=1}}$$

$$VDERW22 = \frac{0.5(\eta_{l=2} - \eta_{l=1})}{\eta_{\bar{l}=2} - \eta_{\bar{l}=1}}$$

  such 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:

$$VDERW11 = \frac{0.5(\eta_{l=L} - \eta_{l=L-1})}{\eta_{\bar{l}=L-1} - \eta_{\bar{l}=L-2}}$$

$$VDERW21 = \frac{0.5(\eta_{l=L} - \eta_{l=L-1})}{\eta_{\bar{l}=L-1} - \eta_{\bar{l}=L-2}}$$

$$VDERW12 = \frac{(\eta_{l=L} - \eta_{l=L-1})}{\eta_{\bar{l}=L} - \eta_{\bar{l}=L-1}}$$

$$VDERW22 = 0$$

  such 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.

* Computations:

- The vertical weights $VDERW11$, $VDERW21$, $VDERW12$ and $VDERW22$ are computed then stored in the array **PVDERW**.
- The weight $ZDVER$ is computed then stored in the array **PDVER** (see subsubsection 11.1).
- Functions $f_{\mathrm{H1}}(ZDVER)$ to $f_{\mathrm{H4}}(ZDVER)$ (involved in any Hermite cubic interpolation), where:
  - $f_{\mathrm{H1}}(\alpha) = (1-\alpha)^2(1+2\alpha)$
  - $f_{\mathrm{H2}}(\alpha) = \alpha^2(3-2\alpha)$
  - $f_{\mathrm{H3}}(\alpha) = \alpha(1-\alpha)^2$
  - $f_{\mathrm{H4}}(\alpha) = -\alpha^2(1-\alpha)$

  are computed and stored in array **PHVW**.
- The level number l of T_l is stored in the array **KLEV**.
- Interpolations use data of points T_l, T_{l+1}, T_{l+2} and T_{l+3}.

Vertical interpolation grid and weights for vertical cubic spline interpolations.

This part is valid only for interpolations of full level data.

A 4 points vertical grid is defined as it is shown in figure 11.2. The interpolation point O (medium or origin point) is between T_{l+1} and T_{l+2}. The algorithm of calculation of the vertical weights will be given in a later version of the documentation.

* Computations:

- Vertical interpolation is the product of two operators. The first one uses all the layers and is done in the unlagged part of the grid-point calculations by a routine **VSPLTRANS** and needs to compute top and bottom values and vertical derivatives of the field to be interpolated, and also the inversion of a tridiagonal matrix (routine **TRIDIA**); for this operation it is necessary to use some coefficients stored in the arrays **RVSPTRI** and **RVSPC** and pre-computed in the set-up subroutine **SUVSPLIP**; the original field is stored in the buffer **P(X)L9** and the intermediate result after this first part is stored in the buffer **P(X)SPL9**. The second part uses 4 points and is done in the interpolation routine itself.
- The vertical weights $ZCVE0$, $ZCVE1$, $ZCVE2$ and $ZCVE3$ necessary for the second part are computed then stored in the array **PVINTWS**. Some part of the calculations can be pre-computed in the set-up subroutine **SUVSPLIP** (arrays **RFAA**, **RFBB**, **RFCC** and **RFDD**).
- The level number l of T_l is stored in the array **KLEV**.
- Interpolations use data of points T_l, T_{l+1}, T_{l+2} and T_{l+3}, stored in the intermediate array **P(X)SPL9**.

Interpolation grid and weights for tri-linear interpolations.

A 64 points grid is defined as it is shown in figure 11.4, but only 8 of these 64 points are used in the interpolations. 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 subsubsection 11.1 corresponding to bi-linear horizontal interpolations for weights computations. For weights needed for vertical interpolations ($ZDVER$) see subsubsection 11.1 corresponding to linear vertical interpolations.

- The memory address (in SL arrays) of the data concerning the points $A_{0,l}$, $A_{1,l}$, $A_{2,l}$ and $A_{3,l}$ is computed then stored in the array **KL0**. Once knowing these addresses one can easily retrieve the other points addresses in the interpolations routines.
- Interpolations use data of points $B_{1,l+1}$, $C_{1,l+1}$, $B_{2,l+1}$, $C_{2,l+1}$, $B_{1,l+2}$, $C_{1,l+2}$, $B_{2,l+2}$ and $C_{2,l+2}$.

∗ **Remark:** The same formulae and computations are valid for half level data, simply replace the layer index l by the half level index $\bar{l}$.

Interpolation grid and weights for 32 points interpolations.

A 64 points grid is defined as it is shown in figure 11.4, but only 32 (48 if vertical spline cubic interpolations) of these 64 points are used in the interpolations. 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 subsubsection 11.1 corresponding to bi-linear horizontal interpolations for weights computations. For the two levels $l+1$ and $l+2$ see subsubsection 11.1 corresponding to 12 points horizontal interpolations for weights computations. For weights needed for vertical interpolations see subsubsection 11.1 for vertical cubic interpolations, see subsubsection 11.1 for vertical Hermite cubic interpolations, see subsubsection 11.1 for vertical spline cubic interpolations.

- The memory address (in SL arrays) of the data concerning the points $A_{0,l}$, $A_{1,l}$, $A_{2,l}$ and $A_{3,l}$ is computed then stored in the array **KL0**. Once knowing these addresses one can easily retrieve the other points addresses in the interpolations routines.
- Interpolations use data of the following 32 points $B_{1,l}$, $C_{1,l}$, $B_{2,l}$, $C_{2,l}$, $B_{0,l+1}$, $C_{0,l+1}$, $A_{1,l+1}$, $B_{1,l+1}$, $C_{1,l+1}$, $D_{1,l+1}$, $A_{2,l+1}$, $B_{2,l+1}$, $C_{2,l+1}$, $D_{2,l+1}$, $B_{3,l+1}$, $C_{3,l+1}$, $B_{0,l+2}$, $C_{0,l+2}$, $A_{1,l+2}$, $B_{1,l+2}$, $C_{1,l+2}$, $D_{1,l+2}$, $A_{2,l+2}$, $B_{2,l+2}$, $C_{2,l+2}$, $D_{2,l+2}$, $B_{3,l+2}$, $C_{3,l+2}$, $B_{1,l+3}$, $C_{1,l+3}$, $B_{2,l+3}$ and $C_{2,l+3}$.

∗ **Remarks:**

- The same formulae and computations are valid for half level data, simply replace the layer index l by the half level index $\bar{l}$.
- For vertical spline cubic interpolations a 12 points grid is used on each level l, $l+1$, $l+2$, $l+3$ (total = 48 points used).

Horizontal interpolation grid and weights for 16 points linear least-square fit interpolations.

A 16 points horizontal grid is defined as it is shown in figure 11.3. The interpolation point O (medium or origin point) is between B_1, C_1, B_2 and C_2. The interpolation is replaced by a linear least-square fit minimisation of a first order polynomial in each direction (first zonal interpolations, then meridian interpolations).

The weights used are the same ones as for the 4 points bilinear horizontal interpolation, but the zonal linear weights are required for the 4 latitudes of the 16 points grid ($ZDLO0$, $ZDLO1$, $ZDLO2$, $ZDLO3$, $ZDLAT$). The actual weights used in one direction (for example the meridian direction) are respectively: $0.4-0.3ZDLAT$, $0.3-0.1ZDLAT$, $0.2+0.1ZDLAT$, $0.1+0.3ZDLAT$.

∗ **Computations:**

- The weights $ZDLO0$, $ZDLO1$, $ZDLO2$ and $ZDLO3$ are computed then stored in the array **PDLO**.
- The meridian weight $ZDLAT$ is computed then stored in the array **PDLAT**.
- The memory address (in SL arrays) of the data concerning the points A_0, A_1, A_2 and A_3 are stored in the array **KL0** or **KLH0**. Once knowing these addresses one can easily retrieve the other points addresses in the interpolations routines.
- Interpolations use data of the following 16 points A_0, B_0, C_0, D_0, A_1, B_1, C_1, D_1, A_2, B_2, C_2, D_2, A_3, B_3, C_3, D_3.

Horizontal interpolation grid and weights for 32 points linear least-square fit interpolations.

A 32 points grid is defined as it is shown in figure 11.4 where only the two intermediate layers are retained. 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 subsubsection 11.1 corresponding to 16 points linear least-square fit horizontal interpolations for weights computations. For weights needed for vertical interpolations, only the linear weight $ZDVER$ is required; see subsubsection 11.1 for vertical linear interpolations.

- The memory address (in SL arrays) of the data concerning the points $A_{1,l}$ and $A_{2,l}$ is computed then stored in the array **KL0**. Once knowing these addresses one can easily retrieve the other points addresses in the interpolations routines.
- Interpolations use data of the following 32 points $A_{0,l+1}$, $B_{0,l+1}$, $C_{0,l+1}$, $D_{0,l+1}$, $A_{1,l+1}$, $B_{1,l+1}$, $C_{1,l+1}$, $D_{1,l+1}$, $A_{2,l+1}$, $B_{2,l+1}$, $C_{2,l+1}$, $D_{2,l+1}$, $A_{3,l+1}$, $B_{3,l+1}$, $C_{3,l+1}$, $D_{3,l+1}$, $A_{0,l+2}$, $B_{0,l+2}$, $C_{0,l+2}$, $D_{0,l+2}$, $A_{1,l+2}$, $B_{1,l+2}$, $C_{1,l+2}$, $D_{1,l+2}$, $A_{2,l+2}$, $B_{2,l+2}$, $C_{2,l+2}$, $D_{2,l+2}$, $A_{3,l+2}$, $B_{3,l+2}$, $C_{3,l+2}$, $D_{3,l+2}$.

* Remarks:

- The same formulae and computations are valid for half level data, simply replace the layer index l by the half level index $\bar{l}$.

Plane geometry (ALADIN).

All previous formulae for weight computation can be used for an irregular latitude spacing and a different number of points on each longitude. The ALADIN grid has a horizontal regular spacing, so the previous formulae can be simplified and array **RIPI** is no longer necessary. **ELASCAW** is called instead of **LASCAW** and is cheaper in CPU time.

11.2 Interpolations.

Bilinear interpolation (subroutine LAIDLI).

See figure 11.1 and subsubsection 11.1 for definition of $ZDLO1$, $ZDLO2$, $ZDLAT$ and points B_1, C_1, B_2 and C_2.

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 meridian linear interpolation:
  $X_{\text{interpolated}} = X_1 + ZDLAT(X_2 - X_1)$.

Tri-linear interpolation (subroutine LAITLI).

For layers $l+1$ and $l+2$ (see figure 11.4) bilinear horizontal interpolations give two interpolated values X_{l+1} and X_{l+2} (see subsubsection 11.2). Then the final interpolated value is given by the following expression:

$$X_{\text{interpolated}} = X_{l+1} + ZDVER(X_{l+2} - X_{l+1})$$

$*$ **Remark:** The same formulae and computations are valid for half level data, simply replace the layer index l by the half level index $\bar{l}$.

Horizontal 12 points interpolation (subroutine LAIDDI).

See figure 11.3 and subsubsection 11.1 for definition of $ZDLO0$, $ZDLO1$, $ZDLO2$, $ZDLO3$, $ZCLA1$, $ZCLA2$ and $ZCLA3$ and points B_0, C_0, A_1, B_1, C_1, D_1, A_2, B_2, C_2, D_2, B_3 and C_3.

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 points interpolation on the longitude number 1:
  $X_1 = X_{A_1} + ZCLO11(X_{B_1} - X_{A_1}) + ZCLO12(X_{C_1} - X_{A_1}) + ZCLO13(X_{D_1} - X_{A_1})$.
- a cubic 4 points interpolation on the longitude number 2:
  $X_2 = X_{A_2} + ZCLO21(X_{B_2} - X_{A_2}) + ZCLO22(X_{C_2} - X_{A_2}) + ZCLO23(X_{D_2} - X_{A_2})$.
- a linear interpolation on the longitude number 3:
  $X_3 = X_{B_3} + ZDLO3(X_{C_3} - X_{B_3})$.
- a meridian cubic 4 points interpolation:
  $X_{\text{interpolated}} = X_0 + ZCLA1(X_1 - X_0) + ZCLA2(X_2 - X_0) + ZCLA3(X_3 - X_0)$.

There is a shape-preserving option: after cubic 4 points 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 points interpolation $X_{\text{interpolated}}$ is bounded between X_1 and X_2. Use of switches **LQMW** (momentum equation), **LQMT** (temperature equation), **LQMP** (continuity equation), **LQMSPD** (pressure departure variable equation), **LQMSVD** (vertical divergence equation), **Y[X]_NL%LQM** for GFL variables allow to use shape-preserving option.

Cubic 4 points vertical interpolation.

See figure 11.2 and subsubsection 11.1 for definition of $ZCVE1$, $ZCVE2$ and $ZCVE3$. The cubic 4 points vertical interpolation gives the final interpolated value:

$$X_{\text{interpolated}} = X_l + ZCVE1(X_{l+1} - X_l) + ZCVE2(X_{l+2} - X_l) + ZCVE3(X_{l+3} - X_l)$$

$*$ **Remark:** The same formulae and computations are valid for half level data, simply replace the layer index l by the half level index $\bar{l}$.

Cubic Hermite vertical interpolation.

This part is valid for interpolations of full level variables. See figure 11.2 and subsubsection 11.1 for definition of $VDERW11$, $VDERW21$, $VDERW12$ and $VDERW22$. See subsubsection 11.1 for definition of $ZDVER$. See subsubsection 11.1 for definition of functions f_{H1} to f_{H4}. The cubic Hermite vertical interpolation gives the final interpolated value:

$$X_{\text{interpolated}} = f_{\text{H1}}(ZDVER)X_{l+1} + f_{\text{H2}}(ZDVER)X_{l+2}$$
$$+f_{\text{H3}}(ZDVER)(VDERW11(X_{l+1} - X_l) + VDERW21(X_{l+2} - X_{l+1}))$$
$$+f_{\text{H4}}(ZDVER)(VDERW12(X_{l+2} - X_{l+1}) + VDERW22(X_{l+3} - X_{l+2}))$$

Spline cubic 4 points vertical interpolation.

This part is valid for interpolations of full level variables. See figure 11.2 and subsubsection 11.1 for definition of $ZCVE0$, $ZCVE1$, $ZCVE2$ and $ZCVE3$. The spline cubic 4 points vertical interpolation gives the final interpolated value:

$$X_{\mathrm{interpolated}} = ZCVE0{*}XRP_l{+}ZCVE1{*}XRP_{l+1}{+}ZCVE2{*}XRP_{l+2}{+}ZCVE3{*}XRP_{l+3}$$

where XRP is the re-profiled version of X available in **P(X)SPL9**. A monotonic constraint can be added, bounding $X_{\mathrm{interpolated}}$ between X_{l+1} and X_{l+2}.

32 points 3D interpolation with vertical cubic 4 points interpolation (subroutine LAITRI).

For layers l and $l+3$ (see figure 11.4) bilinear horizontal interpolations give two interpolated values X_l and X_{l+3} (see subsubsection 11.2). For layers $l+1$ and $l+2$ (see figure 11.4) 12 points horizontal interpolations give two interpolated values X_{l+1} and X_{l+2} (see subsubsection 11.2). The final interpolated value $X_{\mathrm{interpolated}}$ is a cubic 4 points vertical interpolation of X_l, X_{l+1}, X_{l+2} and X_{l+3} (see subsubsection 11.2).

There are shape-preserving options for horizontal or both horizontal and vertical interpolations.

Use of switches **LQMW** (momentum equation), **LQMT** (temperature equation), **LQMP** (continuity equation), **LQMSPD** (pressure departure variable equation), **LQMSVD** (vertical divergence equation), **Y[X]_NL%LQM** (GFL equations), allows to use shape-preserving option for both horizontal and vertical interpolations.

Use of switches **LQMHW** (momentum equation), **LQMHT** (temperature equation), **LQMHP** (continuity equation), **LQMHSPD** (pressure departure variable equation), **LQMHSVD** (vertical divergence equation), **Y[X]_NL%LQMH** (GFL equations), allows to use shape-preserving option for horizontal interpolations.

* **Remark:** The same formulae and computations are valid for half level data, simply replace the layer index l by the half level index $\bar{l}$.

32 points 3D interpolation with vertical cubic Hermite interpolation (subroutine LAIHVT).

This part is valid for interpolations of full level variables. For layers l and $l+3$ (see figure 11.4) bilinear horizontal interpolations give two interpolated values X_l and X_{l+3} (see subsubsection 11.2). For layers $l+1$ and $l+2$ (see figure 11.4) 12 points horizontal interpolations give two interpolated values X_{l+1} and X_{l+2} (see subsubsection 11.2). The final interpolated value $X_{\mathrm{interpolated}}$ is a cubic Hermite vertical interpolation of X_l, X_{l+1}, X_{l+2} and X_{l+3} (see subsubsection 11.2).

There are shape-preserving options for horizontal or both horizontal and vertical interpolations.

Use of switch **Y[X]_NL%LQM** (GFL equations), allows to use shape-preserving option for both horizontal and vertical interpolations.

Use of switch **Y[X]_NL%LQMH** (GFL equations), allows to use shape-preserving option for horizontal interpolations.

48 points 3D interpolation with vertical cubic spline interpolation (subroutine LAITVSPCQM).

This part is valid for interpolations of full level variables. This type of interpolation is activated for GFL variables when **Y[X]_NL%LVSPLIP**=.T. in **NAMGFL** (currently for ozone only). Contrary to what is done for the other 32 points interpolations routines, the vertical interpolations are performed first. 12 vertical interpolations are done on the verticals matching the following points: $B_{0,l+1}$, $C_{0,l+1}$, $A_{1,l+1}$, $B_{1,l+1}$, $C_{1,l+1}$, $D_{1,l+1}$, $A_{2,l+1}$, $B_{2,l+1}$, $C_{2,l+1}$, $D_{2,l+1}$, $B_{3,l+1}$, $C_{3,l+1}$. A monotonic constraint is added for the lower levels (currently the 9 lower levels). The projection horizontal plane on the level of the interpolation point provides a 12-points grid. A 12 points interpolation is done on this projection (see part 11.2). A final monotonic constraint is added: the interpolated value of X is bounded between the value of X at the points $B_{1,l+1}$, $C_{1,l+1}$, $B_{2,l+1}$, $C_{2,l+1}$, $B_{1,l+2}$, $C_{1,l+2}$, $B_{2,l+2}$, $C_{2,l+2}$, and the overshoots/undershoots found are dispatched on upper levels to ensure as possible it can be done conservation properties.

Horizontal 16 points linear least-square fit interpolation.

See figure 11.3 and subsubsection 11.1 for definition of $ZDLO0$, $ZDLO1$, $ZDLO2$, $ZDLO3$, $ZDLAT$ and points A_0, B_0, C_0, D_0, A_1, B_1, C_1, D_1, A_2, B_2, C_2, D_2, A_3, B_3, C_3, D_3. Let us define:

- $f_1(\alpha) = 0.4 - 0.3\alpha$
- $f_2(\alpha) = 0.3 - 0.1\alpha$
- $f_3(\alpha) = 0.2 + 0.1\alpha$
- $f_4(\alpha) = 0.1 + 0.3\alpha$

For a quantity X, are computed successively:

- a linear least-square fit 4 points interpolation on the longitude number lon for $lon = 0, 1, 2, 3$:
$X_{lon} = +f_1(ZDLO_{lon})X_{A_{lon}} + f_2(ZDLO_{lon})X_{B_{lon}} + f_3(ZDLO_{lon})X_{C_{lon}} + f_4(ZDLO_{lon})X_{D_{lon}}$.
- a meridian linear least-square fit 4 points interpolation:
$X_{\text{interpolated}} = +f_1(ZDLAT)X_0 + f_2(ZDLAT)X_1 + f_3(ZDLAT)X_2 + f_4(ZDLAT)X_3$.

32 points 3D interpolation with linear least-square fit horizontal interpolations and vertical linear interpolations (subroutine LAISMOO).

For layers $l+1$ and $l+2$ (see figure 11.4) 16 points linear least-square fit horizontal interpolations give two interpolated values X_{l+1} and X_{l+2} (see subsubsection 11.2). The final interpolated value is then given by the following expression:

$$X_{\text{interpolated}} = X_{l+1} + ZDVER(X_{l+2} - X_{l+1})$$

∗ **Remark:** In the current usage of this interpolation (for $\dot{\eta}$ in the upper stratosphere) there is an additional smoothing done in routine **LAISMOA** before the interpolation by **LAISMOO**.

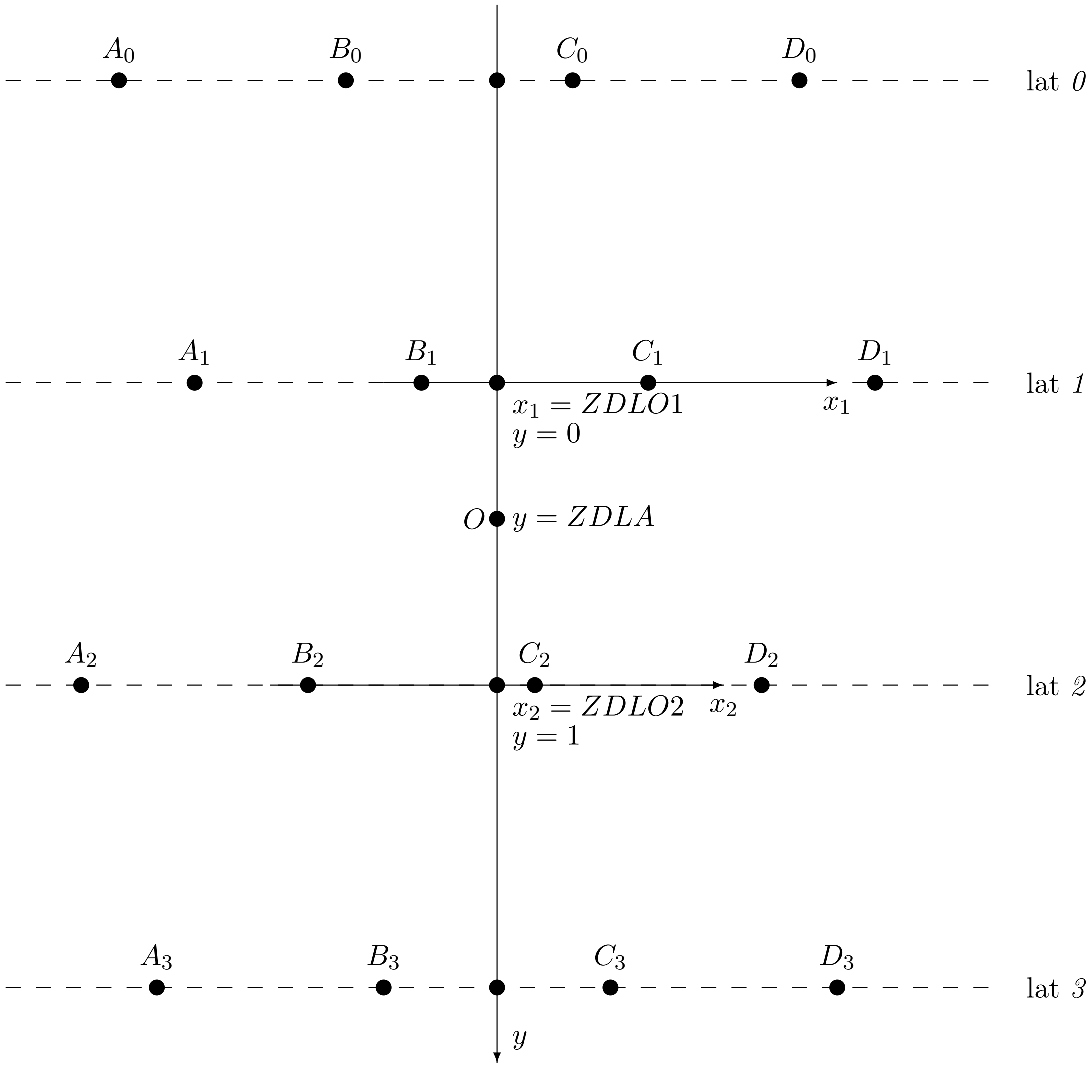
∗ **For more information:** See the internal paper (IDSVTSM).

11.3 Code structures to store weights.

Two structures have been created in module intdyn_mod.F90:

- structure **TLSCAW** for linear weights.
- structure **TRSCAW** for non-linear weights.

Figure 11.1: Interpolation horizontal grid for bilinear interpolations.



$x_1(B_1) = x_2(B_2) = 0$; $x_1(C_1) = x_2(C_2) = 1$.

Figure 11.2: Interpolation vertical grid for linear and cubic vertical interpolations.

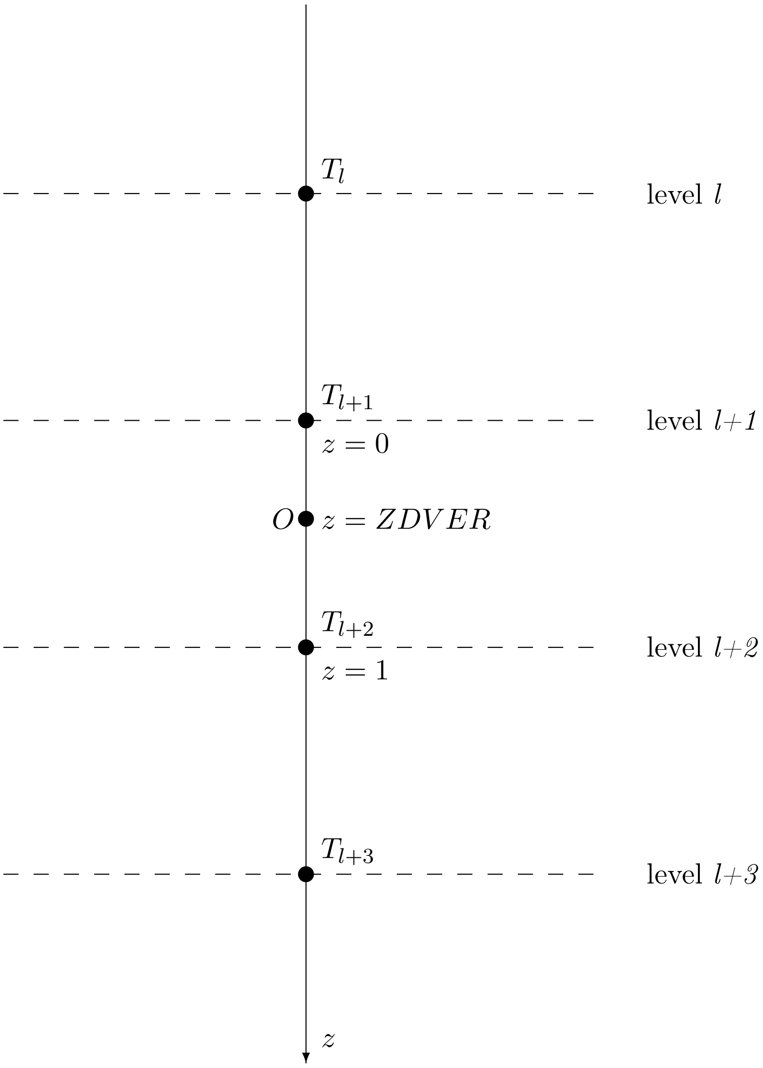
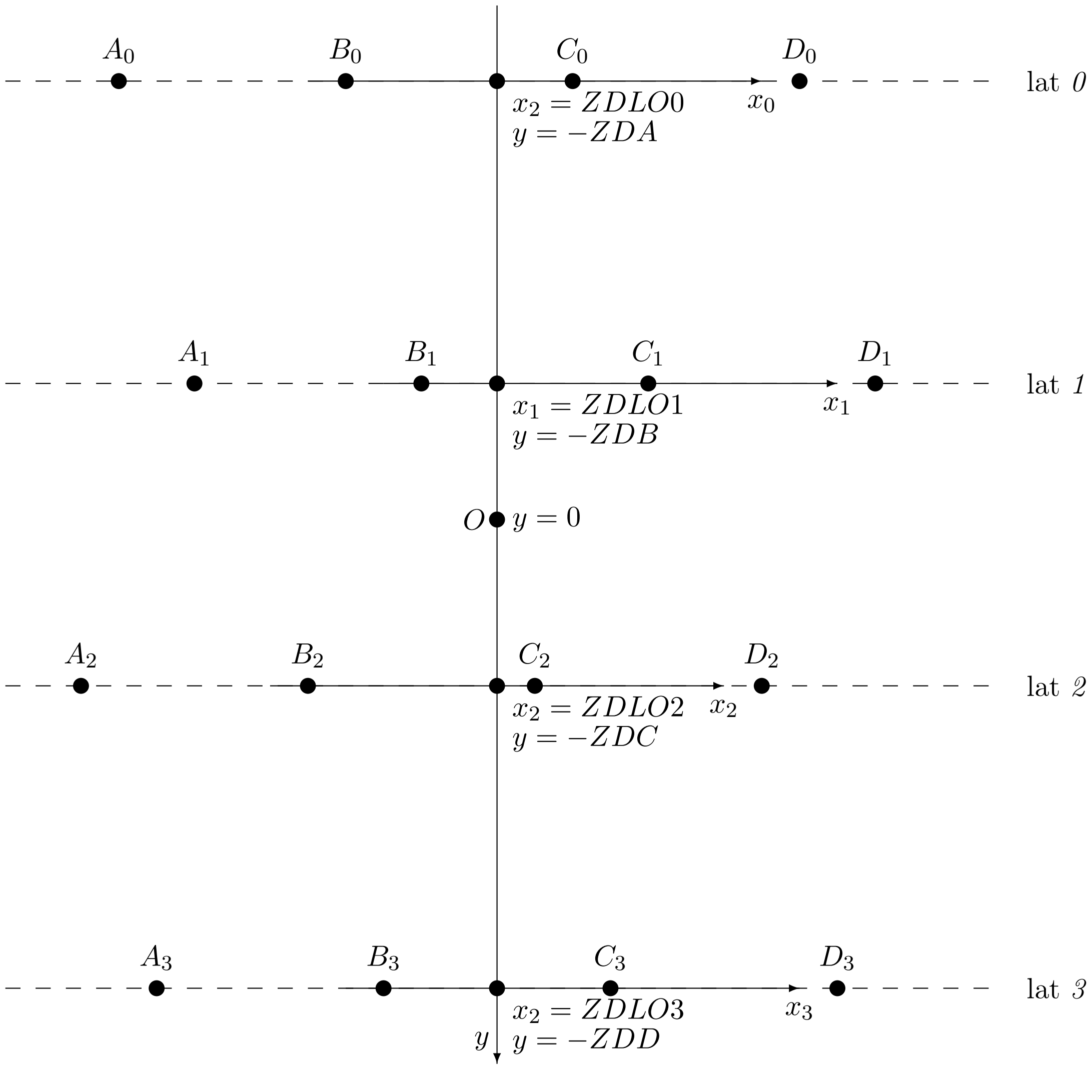

T_l
level l
T_{l+1}
level l+1
$z = 0$
O
$z = ZDVER$
T_{l+2}
level l+2
$z = 1$
T_{l+3}
level l+3
z

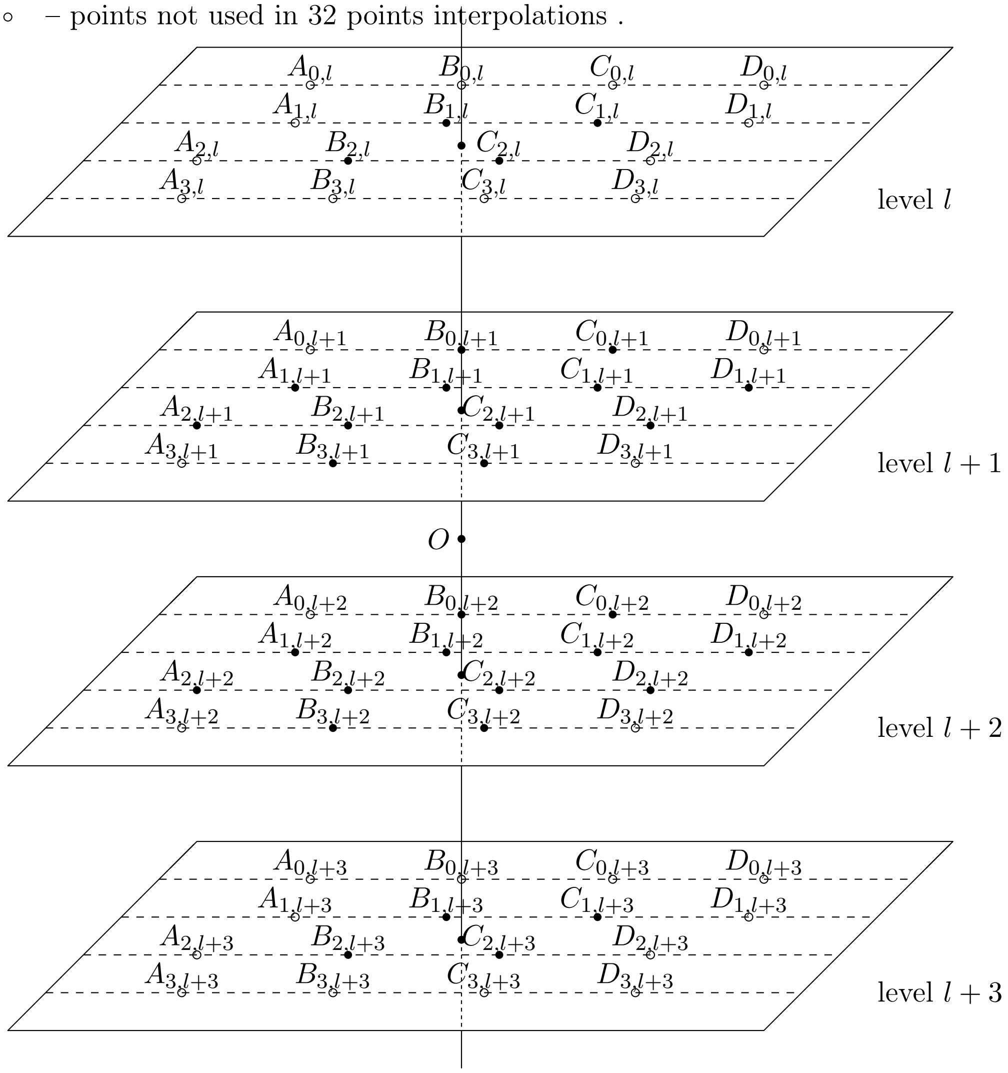
Figure 11.3: Interpolation horizontal grid for 12 points interpolations.



$x_0(B_0) = x_1(B_1) = x_2(B_2) = x_3(B_3) = 0$; $x_0(C_0) = x_1(C_1) = x_2(C_2) = x_3(C_3) = 1$.

Figure 11.4: Interpolation grid for tri-linear and 32 points interpolations.

● – points used in 32 points interpolations.
○ – points not used in 32 points interpolations .



12 Computation of $\dot{\eta}$ at full levels.

∗ **General expression:** $\dot{\eta}$ is needed to find the height of the medium and origin points.

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

$$\dot{\eta} = \left(\dot{\eta}\frac{\partial \Pi}{\partial \eta}\right)\frac{\partial \eta}{\partial \Pi} \tag{161}$$

∗ **Discretisation at full levels for LVERTFE=.F.:** $\left(\dot{\eta}\frac{\partial \Pi}{\partial \eta}\right)$ is provided at half levels, and $\Delta\eta$ and $\Delta\Pi$ are provided at full levels. Discretisation of (161) is:

$$\dot{\eta}_l = 0.5\left[\left(\dot{\eta}\frac{\partial \Pi}{\partial \eta}\right)_{\bar{l}} + \left(\dot{\eta}\frac{\partial \Pi}{\partial \eta}\right)_{\bar{l}-1}\right]\frac{[\Delta\eta]_l}{[\Delta\Pi]_l} \tag{162}$$

∗ **Discretisation at full levels for LVERTFE=.T.:** $\left(\dot{\eta}\frac{\partial \Pi}{\partial \eta}\right)$ is provided at full levels, and $\Delta\eta$ and $\Delta\Pi$ are provided at full levels. Discretisation of (161) is:

$$\dot{\eta}_l = \left(\dot{\eta}\frac{\partial \Pi}{\partial \eta}\right)_l\frac{[\Delta\eta]_l}{[\Delta\Pi]_l} \tag{163}$$

13 Lateral boundary conditions.

13.1 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**+**SLCOMM2A** which exchange data with other processors).

13.2 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**.

13.3 Vertical boundary conditions in the 3D model.

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

∗ **Vertical cubic 4 points 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 points 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.

∗ **Vertical cubic 4 points interpolations for half level variables at the origin point:** When the origin point is above the half level number 1 (resp. $L-1$), the vertical cubic 4 points interpolations using data of the half levels number -1, 0, 1, 2 (resp. $L-2$, $L-1$, L and $L+1$) are degenerated into linear interpolations between the half levels numbers 0 and 1 (resp. $L-1$ and L).

∗ **Vertical 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 itself 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. For more details see subsubsection 11.1.

∗ **Vertical cubic spline interpolations for layer variables at the origin point:** Some top and bottom values are computed (the algorithm of computation will be provided in a later version of this documentation) and the vertical interpolation always uses 4 points.

14 Some distributed memory features.

14.1 Case LEQ_REGIONS=F.

Calculations packets.

$*$ **Grid-point computations:** The total number of processors involved in the A-level parallelisation is **NPRGPNS**. The total number of processors involved in the B-level parallelisation is **NPRGPEW**. One processor treats **NGPTOT** points (a part of the Gaussian grid points). The total amount of grid-points for all the processors is **NGPTOTG**. The maximum value of **NGPTOT** is **NGPTOTMX**. In the grid-point space there is a subdivision of the current processor grid-points into **NGPBLKS**=int[(**NGPTOT**+**NPROMA**-1)/**NPROMA**] packets of length **NPROMA** (the useful number of values in each packet is lower or equal than **NPROMA**). These packets do not contain neither extra-longitudes nor polar or extra-polar latitudes data. A **NPROMA**-packet does not always contain a set of complete latitudes. This subdivision into **NPROMA**-packet both concern not lagged and lagged computations. Currently all the not lagged calculations are made for the int[(**NGPTOT**+**NPROMA**-1)/**NPROMA**] packets of length **NPROMA** before calling the lagged computations for the int[(**NGPTOT**+**NPROMA**-1)/**NPROMA**] packets of length **NPROMA**. int[(**NGPTOT**+**NPROMA**-1)/**NPROMA**] is stored in the variable **NGPBLKS**. More details will be given later for the data transmission for horizontal interpolations. One 2D field has **NGPTOTG** points divided into **NPRGPNS*NPRGPEW** sets of **NGPTOT** points treated by each processor. **NGPTOT** does not take account of the extra-longitudes and the extra-polar latitudes. All these variables take account of the reduced Gaussian grid. It is assumed and hardcoded that there are one western extra-longitude and two eastern extra-longitudes. The DM-global longitude $jlon = 1$ is always the “Greenwich" meridian of the computational sphere. All the vertical levels and the variables corresponding to a same grid-point are treated by the same processor. There are necessary transpositions (reorganisation of data) between grid point computations and Fourier transforms because Fourier transforms need complete latitudes.

$*$ **Additional remarks about the LEQ_REGIONS environment variables.** Variables **N_REGIONS_NS**, **N_REGIONS** and **N_REGIONS_EW** are used even when **LEQ_REGIONS**=F but in this case:

- **N_REGIONS_NS**=**NPRGPNS**.
- **N_REGIONS**=**NPRGPEW** everywhere.
- **N_REGIONS_EW**=**NPRGPEW**.

Transmission of data necessary for semi-Lagrangian horizontal interpolations from the non lagged grid-point computations towards the lagged grid-point computations: interpolation buffers.

Description is done for the 3D model (the 2D model uses an obsolescent dataflow structure and does not work any longer).

First one associates to each interpolation point, which is generally not a model grid-point, an “associated" model grid-point which currently satisfies to the following rule: for the semi-Lagrangian scheme one associates to the interpolation point the corresponding arrival point of the SL trajectory. This associated model grid-point is always between the latitudes 1 and **NDGLG** (it is never a pole or an extra-polar latitude point). The processor which treats the interpolation point is the processor which treats this associated model grid-point.

Interpolations use data of points which are not necessary on the same latitude and longitude as the interpolation point and the arrival grid-point of the semi-Lagrangian trajectory. Thus interpolation routines need to have access to a limited number of surrounding latitudes and longitudes which are not necessary treated by the current processor. First the not lagged computations are done for the int[(**NGPTOT**+**NPROMA**-1)/**NPROMA**] packets of length **NPROMA**, then the lagged computations are done for the int[(**NGPTOT**+**NPROMA**-1)/**NPROMA**] packets of length **NPROMA** interpolation points. The number of surrounding latitudes and longitudes rows necessary for interpolations but which do not belong to the current processor is precomputed in the subroutine **SUSC2B** (variable **NSLWIDE**). This is a sort of “halo" belonging to some other processors. Due to the “halo" there is still need to split calculations into not lagged ones and lagged ones. Quantities to be interpolated are computed in the non-lagged part and interpolations are performed in the lagged part. In the non-lagged part, only data of the current processor (without any extra-longitudinal data nor polar and extra-polar data) are computed. For all the **NPROMA**-packages treated by the current processor these data are stored in the arrays **PB1** in **CPG** (corresponding to local array **ZSLBUF1** in the routine **GP_MODEL**). The first dimension of **PB1** is **NASLB1**, which is the total number of points one needs for the interpolations (**NASLB1** is greater than **NGPTOT**). The second dimension of **PB1** is **NFLDSLB1**: this is the number of 2D quantities to be interpolated. Inside **CPG**, an intermediate array **ZSLBUF1AU** is used and data are transferred in **PB1**: this memory transfer uses a precomputed intermediate array **NSLCORE**. Then some communication routines are called in **CALL_SL** to constitute the halo. **SLCOMM** does processor communication to fill the halo (receives and sends data from some other processors). When all the **NASLB1** dataset is constituted, the lagged part **CALL_SL** is called which do horizontal interpolations for all the data to be interpolated for the current processor.

$*$ **Particular case of the “on demand" processor communication:** **SLCOMM** and **SLEXTPOL** are still called with specific options, and do the communications only on the part of the buffer which contains the information to find the semi-Lagrangian displacement. For the RHS of equations, the interpolations need to communicate a subset of points, this subset can be known only when all the interpolations grids have been computed (in **LASCAW**): communications are currently done by **SLCOMM2A** and **SLEXTPOL** called by **CALL_SL** just before the interpolations. **SLCOMM2A** and **SLEXTPOL** have less points to exchange so the model integration is slightly less expensive.

Terms which must be evaluated at F are stored in **GFLT1**, **GMVT1**, **GMVT1S** (for the RHS of equations) and **PB2** (**ZSLBUF2** in **GP_MODEL**) for additional intermediate terms. See documentation (IDEUL) for more details about **PB2** and the dataflow in the grid-point calculations.

14.2 Case LEQ_REGIONS=T.

This case is relevant only when **NPRGPEW**>1 (B-level parallelisation at least in the grid-point calculations). This is an optimised version of the LEQ_REGIONS=F case which is well designed for reduced Gaussian grid and it improves the load balance in this case. A comprehensive description can be found in (Mozdzynski, 2006). To sum-up, we can say that:

- the A-level grid-point distribution splits the Earth into **N_REGIONS_NS** bands. **N_REGIONS_NS** can be slightly different from **NPRGPNS**.
- for each band *jroca*, the B-level grid-point distribution splits the band into **N_REGIONS**(*jroca*) zones: the minimum value of **N_REGIONS** is at the poles of the computational sphere (equal to 1 in the examples provided by Mozdzynski); the maximum value of **N_REGIONS** is at the equator of the computational sphere and this maximum is equal to **N_REGIONS_EW**. The meridian variations of **N_REGIONS** are highly correlated to those of **NLOENG**.
- In the examples provided by Mozdzynski, **NPRGPNS**=**NPRGPEW**=**NPRTRW**=**NPRTRV** and we notice that **N_REGIONS_NS** is slightly below **NPRGPNS**, and that **N_REGIONS_EW** is slightly below 2∗**NPRGPEW**.

5

Semi-implicit spectral computations and predictor-corrector schemes

1 Introduction.

1.1 Interest of semi-implicit and iterative centred-implicit schemes.

For both hydrostatic and non-hydrostatic models it is necessary to treat implicitly the linear terms source of (fast moving) gravity waves to ensure a good stability; hence the resolution of equations involve the inversion of a linear system leading to a Helmholtz equation: inversion of such a system is more convenient to do in spectral space. For non-hydrostatic models the semi-implicit scheme is generally not sufficient (especially for a two-time level semi-Lagrangian scheme) and some non-linear terms have also to be treated implicitly; for that one uses an iterative centred-implicit (abbreviated into “ICI") scheme. The iterative centred-implicit schemes are often called “predictor-corrector" schemes, but in a theoretical point of view one has normally to reserve this appellation for a subset of iterative centred-implicit schemes with only one iteration. The iterative centred-implicit scheme can have an incremental formulation or a non-incremental formulation. The one which is coded in ARPEGE/ALADIN (for both hydrostatic and non-hydrostatic models) and which will be retained and described is a non-incremental one. All the additional calculations generated by an iterative centred-implicit scheme are mainly done in the grid-point calculations and in the spectral transforms.

1.2 Distributed memory.

Some distributed memory features are now introduced in the code and will be briefly described. For convenience one uses some generic appellations.

- Expression “DM-local" for a quantity means “local to the couple of processors (*proca*,*procb*)": each processor has its own value for the quantity. Expression “DM-local computations" means that the computations are done independently in each processor on “DM-local" quantities, leading to results internal to each processor, which can be different from a processor to another one.
- Expression “DM-global" for a quantity means that it has a unic value available in all the processors. Expression “DM-global computations" means that the computations are either done in one processor, then the results are dispatched in all the processors, or the same computations are done in all the processors, leading to the same results in all the processors.
- In a routine description the mention “For distributed memory computations are DM-local" means that all calculations done by this routine are DM-local; the mention “For distributed memory computations are DM-global" means that all calculations done by this routine are DM-global; when no information is provided it means that a part of calculations are DM-local and the other part is DM-global.
- Expression “main" processor currently refers to the processor number 1: (*proca*,*procb*)=(1,1).

1.3 The different models described.

- 2D shallow-water model.
- 3D primitive equations model (denoted as HYD).

1.4 Other restrictions of this documentation.

- Linear systems are written for thin layer equations: in practical, they don’t change for deep layer equations because the radius r is linearised around a reference value equal to a.

2 Notations.

- M is the mapping factor. $\overline{M}$ is a reference mapping factor for semi-implicit computations. $\overline{M} = c$ (stretching factor) if semi-implicit scheme with reduced divergence (**LSIDG**=.F. in **YOMDYN**). $\overline{M} = M$ (mapping factor) if semi-implicit scheme with unreduced divergence (**LSIDG**=.T. in **YOMDYN**).
- a is the Earth mean radius.
- r is the radius. The reference value of r for the linearisation is the mean radius a. In the thin layer equations, $r = a$ everywhere. In the **LVERCOR**=T (White and Bromley, 1995) deep-layer equations, r is replaced by a pseudo-radius $r_{\rm s}$ depending only on the hydrostatic pressure. All the equations involving the radius will be written with the denotation r.
- $\mathbf{V}$ is the horizontal geographical wind. Its zonal component is U. Its meridian component is V.
- D is the unreduced divergence of horizontal wind, $D^{'}$ is the reduced divergence. D and $D^{'}$ are linked by the relationship $D = (a/r) * M^2 * D^{'}$.
- ζ is the unreduced vorticity of horizontal wind, $\zeta^{'}$ is the reduced vorticity. ζ and $\zeta^{'}$ are linked by the relationship $\zeta = (a/r) * M^2 * \zeta^{'}$.
- 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 and in some non-hydrostatic equations. Default value is 300 K or 350 K according to configuration (for more details see subsection **??** for variable **SITR**). IF **LSPRT**=.T. (use of virtual temperature in spectral transforms instead of real temperature), T^* is used as a reference virtual temperature (same default value).

  $T^*_{\rm a}$ is a cold vertically-constant reference temperature which is used in the semi-implicit scheme in the NH vertical divergence equation; it is recommended to have $T^*_{\rm a}$ lower than the current temperature.
- q is the humidity.
- Π is the hydrostatic pressure, $\Pi_{\rm s}$ is the hydrostatic surface pressure. Π^* is a reference hydrostatic pressure and ${\Pi_{\rm s}}^*$ is a reference hydrostatic surface pressure, which are used in the semi-implicit scheme and in some non-hydrostatic equations. These reference quantities are vertically dependent and “horizontally" (i.e. on η surfaces) constant. Default value of ${\Pi_{\rm s}}^*$ is generally between 800 hPa and 1000 hPa. $\Delta\Pi^*$ are layer depths corresponding to a surface hydrostatic pressure equal to ${\Pi_{\rm s}}^*$.
- ${\Pi_{\rm s}}^{st}$ is a reference hydrostatic pressure equal to the surface pressure of the standard atmosphere (variable **VP00**). Default value is 101325 Pa.
- $\omega = \frac{d\Pi}{dt}$ is the total temporal derivative of the hydrostatic pressure (vertical velocity in hydrostatic pressure coordinate).
- p is the pressure, $p_{\rm s}$ is the surface pressure.
- $\hat{Q}$ is the pressure departure variable. Expression of $\hat{Q}$ is:

$$\hat{Q} = \log \frac{p}{\Pi} \tag{1}$$

- gz is the geopotential height.
- Φ is the total geopotential (equivalent height in the shallow-water model), $\Phi_{\rm s}$ is the surface geopotential (i.e. the orography). In the thin layer equations, $\Phi = gz$. $\Phi_{\rm s}$ is assumed to be always equal to $gz_{\rm s}$. Φ^* is a reference equivalent height which is only used in the shallow-water model (semi-implicit scheme). Default value of Φ^* is 100000 J/kg. $\Delta\Phi^*$ is a reference geopotential depth computed on model levels.

- $\mathbf{\Omega}$ is the Earth rotation angular velocity.
- $\mathbf{r}$ is the vector directed upwards, the length of which is the Earth radius a.
- g is the gravity acceleration constant, assumed to be vertically constant in the current documentation. For the (Wood and Staniforth, 2003) deep-layer NH equations with vertical variations of g, only the reference value of g (vertically constant) is taken into account in the semi-implicit scheme.
- R is the gas constant for air and R_{d} the gas constant for dry air.
- c_{p} is the specific heat at constant pressure for air and $c_{\mathrm{p_d}}$ is the specific heat at constant pressure for dry air.
- c_{v} is the specific heat at constant volume for air and $c_{v\,d}$ is the specific heat at constant volume for dry air.
- ∇ is the unreduced first order horizontal gradient on η-surfaces. $\nabla^{'}$ is the reduced first order horizontal gradient. These two operators are linked by the relationship $\nabla = (a/r) * M * \nabla^{'}$.
- D_3 is the true 3D divergence. In the thin layer equations, expression of D_3 is:

$$D_3 = \nabla\vec{V} + \frac{p}{\frac{\partial \Pi}{\partial \eta} RT} \nabla\Phi \left(\frac{\partial \mathbf{V}}{\partial \eta} \right) - \frac{gp}{\frac{\partial \Pi}{\partial \eta} RT} \left(\frac{\partial w}{\partial \eta} \right) \tag{2}$$

- d is the vertical divergence. In the thin layer equations, the relationship between d and the height-coordinate vertical velocity w is:

$$d = -\frac{gp}{\frac{\partial \Pi}{\partial \eta} R_{\mathrm{d}} T} \left(\frac{\partial w}{\partial \eta} \right) \tag{3}$$

- Variable $d_4 = d + \frac{p}{\frac{\partial \Pi}{\partial \eta} RT} \nabla\Phi \left(\frac{\partial \mathbf{V}}{\partial \eta} \right)$ can be also used as prognostic variable.
- L: number of layers of the model.
- A, B define hydrostatic pressure on the η levels ($\Pi = A + B\Pi_{\mathrm{s}}$, where Π_{s} is the hydrostatic surface pressure).
- β coefficient for the semi-implicit scheme (between 0 and 1).
- γ, τ, ν, μ, ∂^*, $\mathbf{L}^*$ and $\mathbf{T}^*$ are generic notations for linear operators (see subsection 4.2).
- H, C, N are intermediate constants used in the semi-implicit scheme of the non-hydrostatic model. Definitions are respectively:

$$H = \frac{R_d T^*}{g} \tag{4}$$

$$C = R_d T^* \frac{c_{\mathrm{p_d}}}{c_{v\,d}} \tag{5}$$

$$N = \frac{g}{\sqrt{c_{\mathrm{p_d}} T^*}} \tag{6}$$

- For a variable X defined at full levels, $\langle X \rangle$ is the vector of coordinates $(X_1; ...; X_l; ...; X_L)$.
- $\mathcal{R}_{inte}$ is the vertical integration operator used in the case **LVERTFE**=.T. :
  - $\int_{\eta=0}^{\eta=1} X d\eta$ is discretised by $[\mathcal{R}_{inte}]_{(top,surf)} \langle X \rangle$.
  - $\int_{\eta=0}^{\eta=\eta_l} X d\eta$ is discretised by $[\mathcal{R}_{inte}]_{(top,l)} \langle X \rangle$.
  - $\int_{\eta=\eta_l}^{\eta=1} X d\eta$ is discretised by $[\mathcal{R}_{inte}]_{(l,surf)} \langle X \rangle$.
- $\mathcal{R}_{\mathrm{deri}}$ is the vertical first-order derivative operator used in the case where VFE are also applied to derivatives.

3 General considerations.

3.1 Advection schemes.

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

$$\frac{\partial X}{\partial t} = -\mathbf{U}.\nabla_3 X + \mathcal{A} + \mathcal{F} \tag{7}$$

where $\mathbf{U}$ is the 3D wind, ∇_3 is the 3D gradient operator, $\mathcal{A}$ is the dynamical contribution, and $\mathcal{F}$ is the physical contribution. $X(t+\Delta t)$ is computed knowing $X(t-\Delta t)$ at the same grid point.

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

$$\frac{dX}{dt} = \mathcal{A} + \mathcal{F} \tag{8}$$

In a three-time level semi-Lagrangian scheme (abbreviated into 3TL SL 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 $t-\Delta t$. In a two-time level semi-Lagrangian scheme (abbreviated into 2TL SL 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 t.

3.2 Semi-implicit treatment of linear terms (case where there is no iterative centred-implicit scheme).

$*$ **Adding of a semi-implicit correction:** In all cases the linear terms source of gravity waves must be treated implicitly, in order to allow time-steps compatible with an operational use of the model. Expression of the linear terms is obtained assuming a definition of a reference state. The reference state is defined by a dry resting isotherm atmosphere in hydrostatic balance, reference orography is zero. Equations (7) and (8) become respectively (9) and (10):

- Eulerian scheme:
$$\frac{\partial X}{\partial t} = -\mathbf{U}.\nabla_3 X + \mathcal{A} + \mathcal{F} + [SIcorrection] \tag{9}$$
- Semi-Lagrangian scheme:
$$\frac{dX}{dt} = \mathcal{A} + \mathcal{F} + [SIcorrection] \tag{10}$$

$*$ **Discretisation of equations (9) and (10):** Equations (9) and (10) give the following discretized equations, where Δt is the time step, $\mathcal{B}$ is the linear term source of gravity waves, β is a tunable parameter ($\beta = 0$ corresponds to an explicit formulation, $\beta = 1$ to an implicit formulation):

- Eulerian scheme:

$$[SIcorrection] = -\beta\mathcal{B}^t + \frac{\beta}{2}\mathcal{B}^{t-\Delta t} + \frac{\beta}{2}\mathcal{B}^{t+\Delta t} \tag{11}$$

$$X^{t+\Delta t} - \beta\Delta t\mathcal{B}^{t+\Delta t} = X^{t-\Delta t} + 2\Delta t(-\mathbf{U}.\nabla_3 X + \mathcal{A} + \mathcal{F}) - 2\beta\Delta t\mathcal{B}^t + \beta\Delta t\mathcal{B}^{t-\Delta t} \tag{12}$$

  all computations are done at the same grid point.

- Three-time level semi-Lagrangian (3TL SL) scheme (without uncentering factor):

$$[SIcorrection] = -\beta\mathcal{B}^t + \frac{\beta}{2}\mathcal{B}^{t-\Delta t} + \frac{\beta}{2}\mathcal{B}^{t+\Delta t} \tag{13}$$

$$X^{t+\Delta t} - \beta\Delta t\mathcal{B}^{t+\Delta t} = X^{t-\Delta t} + 2\Delta t(\mathcal{A} + \mathcal{F}) - 2\beta\Delta t\mathcal{B}^t + \beta\Delta t\mathcal{B}^{t-\Delta t} \tag{14}$$

  where $X^{t+\Delta t} - \beta\Delta t\mathcal{B}^{t+\Delta t}$ is computed at the final grid point of the semi-Lagrangian trajectory, $X^{t-\Delta t}$ and $\beta\Delta t\mathcal{B}^{t-\Delta t}$ are computed at the origin point of the semi-Lagrangian trajectory, $-2\beta\Delta t\mathcal{B}^t$ is computed as an average between the origin and final points of the trajectory, $\mathcal{A}$ is computed either at the medium point or as an average between the origin and final points of the trajectory. If there is a uncentering factor ϵ replace Δt by $(1-\epsilon)\Delta t$ for terms at the origin point, Δt by $(1+\epsilon)\Delta t$ for terms at the final point. For more details see documentation (IDSL) about the semi-Lagrangian scheme.

- Two-time level semi-Lagrangian (2TL SL) scheme (without uncentering factor):

$$[SIcorrection] = -\beta\mathcal{B}^{t+0.5\Delta t} + \frac{\beta}{2}\mathcal{B}^t + \frac{\beta}{2}\mathcal{B}^{t+\Delta t} \tag{15}$$

$$X^{t+\Delta t} - 0.5\beta\Delta t\mathcal{B}^{t+\Delta t} = X^t + \Delta t(\mathcal{A} + \mathcal{F}) - \beta\Delta t\mathcal{B}^{t+0.5\Delta t} + 0.5\beta\Delta t\mathcal{B}^t \tag{16}$$

  where $X^{t+\Delta t} - 0.5\beta\Delta t\mathcal{B}^{t+\Delta t}$ is computed at the final grid point of the semi-Lagrangian trajectory, X^t and $0.5\beta\Delta t\mathcal{B}^t$ are computed at the origin point of the semi-Lagrangian trajectory, $-\beta\Delta t\mathcal{B}^{t+0.5\Delta t}$ and $\mathcal{A}$ are computed either at the medium point or as an average between the origin and final points of the trajectory. If there is a first-order uncentering factor ϵ replace Δt by $(1-\epsilon)\Delta t$ for terms at the origin point, Δt by $(1+\epsilon)\Delta t$ for terms at the final point. For more details see documentation (IDSL) about the semi-Lagrangian scheme.

$\mathcal{B}^{t+0.5\Delta t}$, $\mathcal{B}^t$ and $\mathcal{B}^{t-\Delta t}$ are computed in grid point space. The right-hand side members of equations (12), (14) and (16) are computed in grid point space, then transformed into spectral space. Entering spectral space a system of equations of the following type must be solved:

$$X^{t+\Delta t} - \beta\Delta t\mathcal{B}^{t+\Delta t} = \mathcal{X}^* \tag{17}$$

for a leap-frog scheme, and:

$$X^{t+\Delta t} - 0.5\beta\Delta t\mathcal{B}^{t+\Delta t} = \mathcal{X}^* \tag{18}$$

for a two-time level semi-Lagrangian scheme, where $\mathcal{X}^*$ is known and $X^{t+\Delta t}$ is unknown. Now the spectral computations to solve this system of equations are described for a primitive equations 3D model, a 2D shallow water model and several NH 3D models.

3.3 Iterative centred-implicit schemes and combination with semi-implicit schemes.

Purpose.

In some cases (especially in the non-hydrostatic models), the model with a semi-implicit treatment of linear terms may remain unstable, hence a treatment by an iterative centred-implicit scheme may be necessary. In the following description one sticks to non-incremental formulations.

Iterative centred-implicit scheme.

$*$ **Algorithm:** The total number of iterations is denoted by N_{siter}.

- The iteration number ($i = 0$) computes an estimation $X^{t+\Delta t}_{(i=0)}$ of $X^{t+\Delta t}$ with a normal semi-implicit scheme. Horizontal diffusion can be done optionally at this stage.
- Iterations ($i > 0$): the i-th iteration ($i > 0$) computes $X^{t+\Delta t}_{(i)}$ (after inversion of Helmholtz equation) knowing $X^{t+\Delta t}_{(i-1)}$. Horizontal diffusion is always done at the last iteration, it can be done optionally at the other iterations. The final value of $X^{t+\Delta t}$ is equal to $X^{t+\Delta t}_{(i=N_{\mathrm{siter}})}$.
- In the cycle 37T1 of ARPEGE/IFS and the cycle AL37T1 of ALADIN, this scheme is controlled by the key **LPC_FULL**=.T. . and is coded in the Eulerian scheme and the two-time level semi-Lagrangian scheme only. It is compulsory for the non-hydrostatic model with a SL2TL scheme to ensure stability (but not compulsory with the hydrostatic model with the two-time level semi-Lagrangian scheme).
- For unlagged physics, the physics has to be computed for the iteration ($i = 0$) only. For lagged physics, there are several possible options in test, but in practical the only one which works is:
  - adiabatic treatment of iterations 0 to $N_{\mathrm{siter}} - 1$.
  - diabatic treatment of iteration N_{siter}.

$*$ **Discretisation of algorithm:**

- First iteration ($i = 0$): One has to start from the discretisations of equations for a model with no iterative centred-implicit formulation (see documentations (IDEUL) and (IDSL)). For a leap-frog scheme the calculations are the same ones. For a two-time level semi-Lagrangian scheme, $(\mathcal{A} - \beta\Delta t\mathcal{B})^{t+0.5\Delta t}_{(i=0)}$ is assumed to be equal to $(\mathcal{A} - \beta\Delta t\mathcal{B})^t$ if no extrapolation is done (case **LPC_NESC**=.T.), and to $1.5(\mathcal{A} - \beta\Delta t\mathcal{B})^t - 0.5(\mathcal{A} - \beta\Delta t\mathcal{B})^{t-\Delta t}$ if extrapolation is done (case **LPC_NESC**=.F.). For a SL2TL case with no uncentering factor that yields the following discretisations (physics is assumed to be unlagged):
  - no extrapolation:

    $$(X^{t+\Delta t}_{(i=0)} - 0.5\Delta t\beta\mathcal{B}^{t+\Delta t}_{(i=0)})_F = [0.5\Delta t\mathcal{A}^t - 0.5\Delta t\beta\mathcal{B}^t]_F + [X^t + 0.5\Delta t\mathcal{A}^t - 0.5\Delta t\beta\mathcal{B}^t + 0.5\Delta t\beta\mathcal{B}^t + \Delta t\mathcal{F}^t]_{O(i=0)}$$

    ($O(i = 0)$ and F are respectively the origin and final points of the semi-Lagrangian trajectory), which can be rewritten:

    $$(X^{t+\Delta t}_{(i=0)} - 0.5\Delta t\beta\mathcal{B}^{t+\Delta t}_{(i=0)})_F = [0.5\Delta t\mathcal{A}^t - 0.5\Delta t\beta\mathcal{B}^t]_F + [X^t + 0.5\Delta t\mathcal{A}^t + \Delta t\mathcal{F}^t]_{O(i=0)}$$

  - extrapolation: discretisation is identical to the case with no iterative centred-implicit scheme and conventional extrapolation (type **LSETTLS**=.F.); the RHS terms other than X^t and $\mathcal{F}^t$ can be replaced by a "spatio-temporal" average; see documentation (IDSL).

- Following iterations ($i > 0$): The general iteration writes (no uncentering, unlagged physics):
  - Eulerian scheme (ADV stands for advection terms):

$$X_{(i)}^{t+\Delta t} - \Delta t\beta\mathcal{B}_{(i)}^{t+\Delta t}$$

$$= X^{t-\Delta t} + 2\Delta t ADV^t + [\Delta t\mathcal{A}_{(i-1)}^{t+\Delta t} - \Delta t\beta\mathcal{B}_{(i-1)}^{t+\Delta t}] + [\Delta t\mathcal{A}^{t-\Delta t} - \Delta t\beta\mathcal{B}^{t-\Delta t}] + \Delta t\beta\mathcal{B}^{t-\Delta t} + 2\Delta t\mathcal{F}^{t-\Delta t}$$

  which can be rewritten:

$$X_{(i)}^{t+\Delta t} - \Delta t\beta\mathcal{B}_{(i)}^{t+\Delta t} = \Delta t\mathcal{A}_{(i-1)}^{t+\Delta t} - \Delta t\beta\mathcal{B}_{(i-1)}^{t+\Delta t} + [X^{t-\Delta t} + 2\Delta t ADV^t + \Delta t\mathcal{A}^{t-\Delta t} + 2\Delta t\mathcal{F}^{t-\Delta t}]$$

  - Three-time level semi-Lagrangian scheme (without uncentering factor):

$$[X_{(i)}^{t+\Delta t} - \Delta t\beta\mathcal{B}_{(i)}^{t+\Delta t}]_F$$

$$= X_{O(i)}^{t-\Delta t} + [\Delta t\mathcal{A}_{(i-1)}^{t+\Delta t} - \Delta t\beta\mathcal{B}_{(i-1)}^{t+\Delta t}]_F + [\Delta t\mathcal{A}^{t-\Delta t} - \Delta t\beta\mathcal{B}^{t-\Delta t}]_{O(i)} + [\Delta t\beta\mathcal{B}^{t-\Delta t} + 2\Delta t\mathcal{F}^{t-\Delta t}]_{O(i)}$$

  which can be rewritten:

$$[X_{(i)}^{t+\Delta t} - \Delta t\beta\mathcal{B}_{(i)}^{t+\Delta t}]_F = X_{O(i)}^{t-\Delta t} + [\Delta t\mathcal{A}_{(i-1)}^{t+\Delta t} - \Delta t\beta\mathcal{B}_{(i-1)}^{t+\Delta t}]_F + [\Delta t\mathcal{A}^{t-\Delta t} + 2\Delta t\mathcal{F}^{t-\Delta t}]_{O(i)}$$

  The iterative centred-implicit algorithm also applies to re-compute the semi-Lagrangian trajectory (see documentation (IDSL) for more details), the position of the origin point at the i-th (resp $i-1$-th) iteration is $O(i)$ (resp. $O(i-1)$).

  - Two-time level semi-Lagrangian scheme (without uncentering factor):

$$[X_{(i)}^{t+\Delta t} - 0.5\Delta t\beta\mathcal{B}_{(i)}^{t+\Delta t}]_F$$

$$= X_{O(i)}^{t} + [0.5\Delta t\mathcal{A}_{(i-1)}^{t+\Delta t} - 0.5\Delta t\beta\mathcal{B}_{(i-1)}^{t+\Delta t}]_F + [0.5\Delta t\mathcal{A}^{t} - 0.5\Delta t\beta\mathcal{B}^{t}]_{O(i)} + [0.5\Delta t\beta\mathcal{B}^{t} + \Delta t\mathcal{F}^{t}]_{O(i)}$$

  which can be rewritten:

$$[X_{(i)}^{t+\Delta t} - 0.5\Delta t\beta\mathcal{B}_{(i)}^{t+\Delta t}]_F = X_{O(i)}^{t} + [0.5\Delta t\mathcal{A}_{(i-1)}^{t+\Delta t} - 0.5\Delta t\beta\mathcal{B}_{(i-1)}^{t+\Delta t}]_F + [0.5\Delta t\mathcal{A}^{t} + \Delta t\mathcal{F}^{t}]_{O(i)}$$

  The iterative centred-implicit algorithm also applies to re-compute the semi-Lagrangian trajectory (see documentation (IDSL) for more details), the position of the origin point at the i-th (resp $i-1$-th) iteration is $O(i)$ (resp. $O(i-1)$). Remark: when the iterative centred-implicit scheme is used, the stable extrapolation **LSETTLS**=.T. is never involved.

$*$ **Cheap version of this algorithm:** In a semi-Lagrangian scheme, it is possible not to iterate the position of the origin point O (i.e $O(i) = O(i = 0)$): this cheap version is activated if **LPC_CHEAP**=.T. . It is coded only for a non-extrapolating SL2TL scheme. In this case the quantity to be interpolated (i.e. $[0.5\Delta t\mathcal{A}^t + \Delta t\mathcal{F}^t]_O$) needs to be interpolated at the predictor step only. It is then stored in a buffer and re-used at the corrector steps without any interpolation.

3.4 Introduction of uncentering for semi-Lagrangian schemes.

Averages along the semi-Lagrangian trajectory will be weighted by $(1-\epsilon)$ at the origin point and $(1+\epsilon)$ at the final point. If the uncentering coefficient ϵ is horizontally constant the algorithms remain valid, replacing β by $(1+\epsilon)\beta$.

3.5 Plane geometry (ALADIN).

Particular features of ALADIN when differing from ARPEGE/IFS are not described in detail, only brief comments are mentioned. For more details see ALADIN documentation. Concerning the semi-implicit and iterative centred-implicit algorithms one can consider that the major part of this documentation is still valid for ALADIN; the main differences with ARPEGE/IFS are:

- The shallow-water model is not coded in ALADIN.
- Option **LESIDG**=.T. replaces **LSIDG**=.T. in ALADIN, and is available only with the tilted-rotated Mercator projection. This option is useful only when the horizontal variations of the mapping factor are significant. See (IDESIDG) about its implementation.
- Option **LIMPF**=.T. is not coded in ALADIN.
- In the hydrostatic (resp. NH) model, spectral part of the semi-implicit scheme is performed in routine **ESPCSI** instead of **SPCSI** (resp. **ESPNHSI** instead of **SPNHSI**) in ALADIN; the algorithm is the same as in ARPEGE but the truncation of the spectral representation is elliptic and not triangular.

3.6 Finite elements on the vertical.

The option with finite element vertical discretisations is coded for the hydrostatic model and partly for the NH models. For VFE, the main modifications in the semi-implicit scheme are the following ones:

- The discretisation of Π, α and δ at full levels is different.
- The model avoids as possible to compute quantities at half levels; all vertical integrals directly provide quantities at full levels.
- The vertical integrals contained in some linear operators (γ, τ and ν) are discretised differently, as a matricial multiplication with special coefficients (contained in the matrix $\mathcal{R}_{inte}$) computed in the setup routine **SUVERTFE1** or **SUVERTFE3**; the vertical integration is done by routine **VERINT**.

4 Prognostic variables and quantities involved in the semi-implicit scheme.

4.1 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 the components of the horizontal wind **V**, temperature T, and the two additional non-hydrostatic variables in a non-hydrostatic model.
- 3D “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 includes for example humidity q, liquid water, ice, cloud fraction, ozone, and some extra fields.
- 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).

Only the GMV and GMVS variables appear in the semi-implicit scheme. In the shallow-water 2D model, only GMV variables exist, this class of variables includes the components of the horizontal wind **V**, and the equivalent height $\Phi - \Phi_{\rm s}$ (continuity equation).

4.2 Quantities used for vertical discretisations and linear operators.

$*$ **Operators “alpha" and “delta":** these operators are used for discretisations of some vertical integrals. They have a different expression according to the value of variables **NDLNPR**, **LVERTFE**.

- For **LVERTFE**=.F., **NDLNPR**=0 and a layer l between 2 and L (and also $l=1$ if the pressure at the top of the model is not zero), α^* and δ^* are discretised as follows at full levels:

$$\alpha^*_l = 1 - \frac{\Pi^*_{\bar{l}-1}}{\Delta\Pi^*_l} \log\left(\frac{\Pi^*_{\bar{l}}}{\Pi^*_{\bar{l}-1}}\right) \tag{19}$$

$$\delta^*_l = \log\left(\frac{\Pi^*_{\bar{l}}}{\Pi^*_{\bar{l}-1}}\right) \tag{20}$$

- For **LVERTFE**=.F., **NDLNPR**=0 and the layer $l=1$ if the pressure at the top of the model is zero:
  - $\alpha^*_{l=1} = 1$ at METEO-FRANCE.
  - $\alpha^*_{l=1} = \log(2)$ at ECMWF.
  - $\delta^*_{l=1}$ has in theory an infinite value, but in the code it is computed with a top pressure equal to 0.1 Pa to provide a finite value.

- For **LVERTFE**=.F., **NDLNPR**=1 and a layer l between 2 and L (and also $l = 1$ if the pressure at the top of the model is not zero), α^* and δ^* are discretised as follows at full levels:
$$\alpha_l^* = 1 - \sqrt{\frac{\Pi^*_{\bar{l}-1}}{\Pi^*_{\bar{l}}}} = 1 - \frac{\Pi^*_l}{\Pi^*_{\bar{l}}} \tag{21}$$
$$\delta_l^* = \frac{\Delta\Pi_l^*}{\Pi_l^*} = \frac{\Delta\Pi_l^*}{\sqrt{\Pi^*_{\bar{l}-1}\Pi^*_{\bar{l}}}} \tag{22}$$
Π^* is discretised as follows at full levels:
$$\Pi^*{}_l = \sqrt{\Pi^*{}_{\bar{l}}\Pi^*{}_{\bar{l}-1}} \tag{23}$$
Notation $\alpha^*_{\bar{l}-1}$ is used for quantity $1 - \frac{\Pi^*_{\bar{l}-1}}{\Pi^*_l}$ instead of notation β_l^* of (Bubnová et al., 1995).
- For **LVERTFE**=.F., **NDLNPR**=1 and the layer $l = 1$ if the pressure at the top of the model is zero:
  - $\alpha^*_{l=1} = 1$ and $\alpha^*_{\bar{l}=0} = 1$.
  - $\delta^*_{l=1} = 1 + \frac{c_{\mathrm{p}_\mathrm{d}}}{R_\mathrm{d}}$.
  - $\Pi^*_{l=1} = \frac{\Delta\Pi^*_{l=1}}{\delta^*_{l=1}}$.
- For **LVERTFE**=.T., **NDLNPR**=0 and a layer l between 1 and L, α^* and δ^* are discretised as follows at full levels:
$$\alpha_l^* = \frac{\Pi^*_{\bar{l}} - \Pi^*_l}{\Pi^*_l} \tag{24}$$
$$\delta_l^* = \frac{\Delta\Pi_l^*}{\Pi_l^*} \tag{25}$$
where $\Pi^*_{\bar{l}} = A_{\bar{l}} + B_{\bar{l}}\Pi_\mathrm{s}{}^*$. See documentation (IDEUL) for computation of $A_{\bar{l}}$ and $B_{\bar{l}}$ in this case. Formulae (24) and (25) provide finite values of α_1^* and δ_1^* even if the pressure at the top of the model is zero. α_l^* is not used in the SI scheme in this case because vertical integrals are directly provided at full levels without the intermediate state of interlayer data.

$*$ **Linear operator "γ":** this operator is applied to temperature and pressure departure variable to compute linear term in momentum equation. Denotation γ has the same meaning as the denotation $R_\mathrm{d} * \mathbf{G}^*$ of (Bubnová et al., 1995) or (IDNH2.1). For a variable Z, (γZ) is a discretisation of vertical integral
$$\int_\eta^1 \frac{\left(\frac{\partial\Pi^*}{\partial\eta}\right)}{\Pi^*} R_\mathrm{d} Z d\eta$$
Expression of this discretisation is:

- Case **LVERTFE**=.T.:
$$(\gamma Z)_l = [\mathcal{R}_{inte}]_{(l,surf)} \left\langle \frac{R_\mathrm{d} Z \delta^*}{\Delta\eta} \right\rangle \tag{26}$$
- Case **LVERTFE**=.F.:
$$(\gamma Z)_l = \alpha_l^* R_\mathrm{d} Z_l + \sum_{k=l+1}^{L} R_\mathrm{d} Z_k \delta_k^* \tag{27}$$

Remark: if **LSPRT**=.T., γ is applied to virtual temperature instead of real temperature.

$*$ **Linear operator "τ":** this operator is applied to divergence to compute linear term in temperature equation. Denotation τ has the same meaning as the denotation $\frac{R_{\mathrm{d}}T^*}{c_{\mathrm{p_d}}} * \mathbf{S}^*$ of (Bubnová et al., 1995) and (IDNH2.1). For a variable Z, (τZ) is the discretisation of vertical integral $\frac{R_{\mathrm{d}}T^*}{c_{\mathrm{p_d}}\Pi^*} \int_0^\eta \frac{\partial \Pi^*}{\partial \eta} Z d\eta$. Expression of (τZ) is:

- Case **LVERTFE**=.T.:

$$(\tau Z)_l = \frac{R_{\mathrm{d}}T^*}{c_{\mathrm{p_d}}} \frac{\delta_l^*}{\Delta \Pi_l^*} [\mathcal{R}_{inte}]_{(top,l)} \left\langle \frac{\Delta \Pi^* Z}{\Delta \eta} \right\rangle \tag{28}$$

  Remark: according to the expression of δ_l^* in this case, this equation can be rewritten:

$$(\tau Z)_l = \frac{R_{\mathrm{d}}T^*}{c_{\mathrm{p_d}}} \frac{1}{\Pi_l^*} [\mathcal{R}_{inte}]_{(top,l)} \left\langle \frac{\Delta \Pi^* Z}{\Delta \eta} \right\rangle$$

- Case **LVERTFE**=.F.:

$$(\tau Z)_l = \frac{R_{\mathrm{d}}T^*}{c_{\mathrm{p_d}}} \left[\alpha_l^* Z_l + \frac{\delta_l^*}{\Delta \Pi_l^*} \sum_{k=1}^{l-1} \Delta \Pi_k^* Z_k \right] \tag{29}$$

Remark: if **LSPRT**=.T., $R_{\mathrm{d}}T^*$ becomes $\frac{R_{\mathrm{d}}^2}{R} T^*$ in expression of (τZ).

$*$ **Linear operator "ν":** this operator is applied to divergence to compute linear term in continuity equation. Denotation ν has the same meaning as the denotation $\Pi_{\mathrm{s}}{}^* \mathbf{N}^*$ of (Bubnová et al., 1995) and (IDNH2.1). For a variable Z, definition of (νZ) is:

- Case **LVERTFE**=.T.:

$$(\nu Z) = \frac{1}{\Pi_{\mathrm{s}}{}^*} [\mathcal{R}_{inte}]_{(top,surf)} \left\langle \frac{\Delta \Pi^* Z}{\Delta \eta} \right\rangle \tag{30}$$

- Case **LVERTFE**=.F.:

$$(\nu Z) = \frac{1}{\Pi_{\mathrm{s}}{}^*} \sum_{l=1}^{L} \Delta \Pi_l^* Z_l \tag{31}$$

(νZ) is the discretisation of vertical integral $\frac{1}{\Pi_{\mathrm{s}}{}^*} \int_0^1 \frac{\partial \Pi^*}{\partial \eta} Z d\eta$.

$*$ **Linear operator "μ":** This operator is applied to $\log(\Pi_{\mathrm{s}})$ to compute linear term in momentum equation. For a variable Z, Definition of (μZ) is:

$$(\mu Z) = R_{\mathrm{d}}T^* Z \tag{32}$$

(μZ) is applied to $\log(\Pi_{\mathrm{s}})$.

Remarks:

- if **LSPRT**=.T., $R_{\mathrm{d}}T^*$ becomes $\frac{R_{\mathrm{d}}^2}{R} T^*$ in expression of (μZ).
- In some documentations one sometimes finds the expanded expression $R_{\mathrm{d}}T^*\mathtt{I}$ instead of μ (where $\mathtt{I}$ is the identity matrix).

4.3 Relationships between some linear operators.

In the NH-PDVD model, operators ν, γ and τ SHOULD be linked by the following relationship if one wants the complete elimination of variables in order to provide a "one variable" Helmholtz equation computing the vertical divergence, and all the following parts of this documentation assumes that this condition is fulfilled. For more details see section 4 of (IDNH2.1) where this condition is denoted by "C1". The adimensioned formulation of constraint "C1" writes:

$$\nu = \frac{1}{R_{\mathrm{d}}}\gamma + \frac{c_{\mathrm{p_d}}}{R_{\mathrm{d}}T^*}\tau - \frac{c_{\mathrm{p_d}}}{R_{\mathrm{d}}^2T^*}\gamma\tau \qquad (33)$$

This equation can be multiplied by $c_{\mathrm{p_d}}/c_{v\,d}$ and rewritten as follows:

$$COR = 0 \qquad (34)$$

where:

$$COR = \frac{c_{v\,d}}{R_{\mathrm{d}}^2T^*}\gamma\tau - \frac{c_{v\,d}}{R_{\mathrm{d}}c_{\mathrm{p_d}}}\gamma - \frac{c_{v\,d}}{R_{\mathrm{d}}T^*}\tau + \frac{c_{v\,d}}{c_{\mathrm{p_d}}}\nu \qquad (35)$$

Note that this condition is fulfilled at least for **LVERTFE**=.F. (FD discretisation of operators ν, γ and τ) and **NDLNPR**=1 (this is the reason of using the **NDLNPR**=1 discretisation in a NH model), and is SURELY NOT fulfilled for **LVERTFE**=.F. and **NDLNPR**=0.

For **LVERTFE**=.T., this condition is not fulfilled. This is the reason why there is a possibility to keep **LVERTFE**=.F. and **NDLNPR**=1 in the linear model, even if VFE are switched on in the explicit model.

Additional studies (see for example Bénard, 2004) show that, when the constraint C1 is not fulfilled, it is not possible to fulfill it simply by changing the definition of one of the operators ν, γ or τ.

- Changing the definition of ν to match formula (33) does not ensure any longer the fact that ν is a 2D operator.
- Changing the definition of γ to match $COR = 0$ does not work because that provides an expression with the inverse of a non-invertible matrix.
- Changing the definition of τ to match $COR = 0$ has none of the shortcomings of the two previous proposals, but provides a linear operator generating spurious noise.

5 Semi-implicit scheme, no Coriolis term in the semi-implicit scheme.

Equations are written for a leap-frog scheme (Eulerian scheme of three-time level semi-Lagrangian scheme). For a two-time level semi-Lagrangian scheme replace Δt by $0.5\Delta t$.

5.1 3D hydrostatic primitive equations model.

$*$ Expression of the linear term $\mathcal{B}$ for GMV and GMVS variables:

- Continuity equation ($X = \log(\Pi_{\rm s})$):

$$\mathcal{B} = -\nu(\overline{M}^2 D^{'}) \qquad (36)$$

- Divergence equation ($X = D^{'}$):

$$\mathcal{B} = -\nabla^{'2}(\gamma T + \mu \log \Pi_{\rm s}) \qquad (37)$$

- Vorticity equation ($X = \zeta^{'}$):

$$\mathcal{B} = 0 \qquad (38)$$

- Temperature equation ($X = T$):

$$\mathcal{B} = -\tau(\overline{M}^2 D^{'}) \qquad (39)$$

$*$ System to be solved: Equations are written for $\log(\Pi_{\rm s})$ as a prognostic variable for continuity equation.

$$\log(\Pi_{\rm s})_{t+\Delta t} + \beta\Delta t\overline{M}^2 \nu D^{'}_{t+\Delta t} = \mathcal{P}^* \qquad (40)$$

$$D^{'}_{t+\Delta t} + \beta\Delta t\nabla^{'2}(\gamma T_{t+\Delta t} + \mu \log(\Pi_{\rm s})_{t+\Delta t}) = \mathcal{D}^{'*} \qquad (41)$$

$$T_{t+\Delta t} + \beta\Delta t\overline{M}^2 \tau D^{'}_{t+\Delta t} = \mathcal{T}^* \qquad (42)$$

$\mathcal{P}^*$, $\mathcal{D}^{'*}$, $\mathcal{T}^*$ correspond to $\mathcal{X}^*$ defined in equation (17) and are available in spectral arrays (**SPSP**, **SPDIV**, **SPT**) at the beginning of the spectral computations. Equations (40) to (42) yield (43) (Helmholtz equation):

$$(\mathtt{I} - \beta^2\Delta t^2 \ \mathtt{B} \ \nabla^{'2}\overline{M}^2)D^{'}_{t+\Delta t} = \mathcal{D}^{'*} - \beta\Delta t\nabla^{'2}(\gamma\mathcal{T}^* + \mu\mathcal{P}^*) \qquad (43)$$

where $\mathtt{B} = \gamma\tau + \mu\nu$ is a matricial operator $L * L$ (precomputed in routines **SUDYN, SUBMAT** and stored in the array **SIB**).

When $\overline{M} = M$ (cases **LSIDG**=.T. or **LESIDG**=.T.) it is more convenient to rewrite equation (43) as:

$$(\nabla^{'-2} - \beta^2\Delta t^2 \ \mathtt{B} \ M^2)D^{'}_{t+\Delta t} = \nabla^{'-2}\mathcal{D}^{'*} - \beta\Delta t(\gamma\mathcal{T}^* + \mu\mathcal{P}^*) \qquad (44)$$

which shows a symmetric matricial operator in the left hand side.

$*$ Spectral computations to solve system of equations (40) to (42).

Algorithm works zonal wave number by zonal wave number m ($\mid m \mid$ varies between 0 and the truncation $N_{\rm s}$) and performed in the routine **SPCSI** before all horizontal diffusion schemes. For a given zonal wave number m:

- After a preliminary memory transfer the right-hand side member of equation (43) is computed for all total wave numbers n between m and $N_{\rm s}$.
- Inversion of Helmholtz equation for the case "reduced divergence" (case **LSIDG**=.F. and **LESIDG**=.F.) and method via a diagonalisation in the eigenmodes space.
  - First the diagonalisation of $\mathtt{B}$ is used: $\mathtt{B} = \mathtt{Q}^{-1}\ \mathtt{A}\ \mathtt{Q}$, where $\mathtt{A}$ is a diagonal $L * L$ matrix, the diagonal coefficients a_l of which are stored in the array **SIVP**. $\mathtt{Q}$ is a $L * L$ matrix stored in the array **SIMI**, $\mathtt{Q}^{-1}$ is stored in the array **SIMO**. Note that the vertical operators ν, μ, τ, γ, $\mathtt{B}$, $\mathtt{Q}$ commute with the horizontal operator $\nabla^{'2}$.
  - Helmholtz equation (43) becomes, for each eigenmode l:

$$(\mathtt{I} - \beta^2\Delta t^2 a_l \nabla^{'2}\overline{M}^2)\mathtt{Q}D^{'}_{t+\Delta t} = \mathtt{Q}(\mathcal{D}^{'*} - \beta\Delta t\nabla^{'2}(\gamma\mathcal{T}^* + \mu\mathcal{P}^*)) \quad (45)$$

  - For each eigenmode l and each zonal wave number m: $(\mathtt{I} - \beta^2\Delta t^2 a_l \nabla^{'2}\overline{M}^2)$ is a diagonal matricial operator $(N_{\rm s} + 1- \mid m \mid) * (N_{\rm s} + 1- \mid m \mid)$: spectral coefficients of the right-hand side member of (45) are simply divided by the diagonal coefficients of this matrix. The result is then multiplied by $\mathtt{Q}$.
- Inversion of Helmholtz equation for the case "unreduced divergence" (case **LSIDG**=.T. in ARPEGE, **LESIDG**=.T. in ALADIN): in this case $\overline{M} = M$. Inversion of Helmholtz equation is more complicated than in the case of semi-implicit scheme with reduced divergence because the left-hand side member of Helmholtz equation contains values of the divergence for all levels and five total wave numbers ($n-2$ to $n+2$). Of course M^2 is a symmetric pentadiagonal matrix, for a given zonal wave number m. Pay attention to the fact that M^2 does not commute with the diagonal operator $\nabla^{'2}$.
  - First the diagonalisation of $\mathtt{B}$ is used: $\mathtt{B} = \mathtt{Q}^{-1}\ \mathtt{A}\ \mathtt{Q}$, where $\mathtt{A}$ is a diagonal $L * L$ matrix, the diagonal coefficients a_l of which are stored in the array **SIVP**. $\mathtt{Q}$ is a $L * L$ matrix stored in the array **SIMI**, $\mathtt{Q}^{-1}$ is stored in the array **SIMO**. Note that the vertical operators ν, μ, τ, γ, $\mathtt{B}$, $\mathtt{Q}$ commute with the horizontal operators $\nabla^{'2}$ and M^2.
  - Helmholtz equation (43) (resp. (44)) becomes, for each eigenmode l:

$$(\mathtt{I} - \beta^2\Delta t^2 a_l \nabla^{'2} M^2)\mathtt{Q}D^{'}_{t+\Delta t} = \mathtt{Q}(\mathcal{D}^{'*} - \beta\Delta t\nabla^{'2}(\gamma\mathcal{T}^* + \mu\mathcal{P}^*)) \quad (46)$$

  resp.:

$$(\nabla^{'-2} - \beta^2\Delta t^2 a_l M^2)\mathtt{Q}D^{'}_{t+\Delta t} = \mathtt{Q}(\nabla^{'-2}\mathcal{D}^{'*} - \beta\Delta t(\gamma\mathcal{T}^* + \mu\mathcal{P}^*)) \quad (47)$$

  - For each eigenmode l and each zonal wave number m: $(\nabla^{'-2} - \beta^2\Delta t^2 a_l M^2)$ is a symmetric pentadiagonal matricial operator $(N_{\rm s}+1- \mid m \mid)*(N_{\rm s}+1- \mid m \mid)$. The factorisation $\mathtt{LU}$ of this matrix is computed, where $\mathtt{L}$ is a lower triangular tridiagonal matrix, $\mathtt{U}$ is a upper triangular tridiagonal matrix with coefficients equal to 1 on the main diagonal. All useful coefficients of $\mathtt{L}$, $\mathtt{U}$ are computed in the set-up routine **SUHEG** and stored in the array **SIHEG**.
  - The right-hand side member of (47) is computed, then multiplied by the inverse of the symmetric pentadiagonal operator $(\nabla^{'-2} - \beta^2\Delta t^2 M^2 a_l)$ (resolution of two tridiagonal triangular systems by routine **MXTURS**). That yields $\mathtt{Q}D^{'}_{t+\Delta t}$. Multiplying by $\mathtt{Q}^{-1}$ one obtains $D^{'}_{t+\Delta t}$.

  - For the zonal wave number $m = 0$ equation (46) is used rather than (47) because, for the total wave number $n = 0$, $\nabla^{'2}$ is equivalent to a multiplication by 0 and $\nabla^{'-2}$ is equivalent to a division by 0. The only difference is that the pentadiagonal but non-symmetric operator $(\mathtt{I} - \beta^2 \Delta t^2 a_l \nabla^{'2} M^2)$ is factorised and inverted. All useful coefficients of `L`, `U` are computed in the set-up routine **SUHEG** and stored in the arrays **SIHEG** and **SIHEG2**.

- Once known $D^{'}_{t+\Delta t}$ equation (40) provides $\log(\Pi_{\mathrm{s}})_{t+\Delta t}$ and equation (42) provides $T_{t+\Delta t}$. For the case **LSIDG**=.T. only (resp. **LESIDG**=.T. in ALADIN), spectral multiplications by M^2 are performed by the product of a symmetric pentadiagonal matrix of dimensions $(N_{\mathrm{s}} + 1- \mid m \mid) * (N_{\mathrm{s}} + 1- \mid m \mid)$ (useful coefficients computed in routine **SUSMAP** (resp. **SUESMAP** in ALADIN) and stored in the array **SCGMAP** (resp. **ESCGMAP** in ALADIN)) by a vector containing spectral coefficients (m, n) for n varying from $\mid m \mid$ to N_{s}.
- Semi-implicit scheme ends by a final memory transfer.

5.2 2D shallow-water model.

$*$ Expression of the linear term $\mathcal{B}$:

- Continuity equation ($X = \Phi$):

$$\mathcal{B} = -\Phi^* \overline{M}^2 D^{'} \tag{48}$$

- Divergence equation ($X = D^{'}$):

$$\mathcal{B} = -\nabla^{'2}(\Phi) \tag{49}$$

- Vorticity equation ($X = \zeta^{'}$):

$$\mathcal{B} = 0 \tag{50}$$

$*$ System to be solved:

$$\Phi_{t+\Delta t} + \beta \Delta t \overline{M}^2 \Phi^* D^{'}_{t+\Delta t} = \mathcal{H}^* \tag{51}$$

$$D^{'}_{t+\Delta t} + \beta \Delta t \nabla^{'2}(\Phi_{t+\Delta t}) = \mathcal{D}^{'*} \tag{52}$$

$\mathcal{H}^*$, $\mathcal{D}^{'*}$ correspond to $\mathcal{X}^*$ defined in equations (17) and are available in spectral arrays (**SPSP**, **SPDIV**) at the beginning of the spectral computations. Equations (51) and (52) yield (53) (Helmholtz equation):

$$(1 - \beta^2 \Delta t^2 \Phi^* \nabla^{'2} \overline{M}^2) D^{'}_{t+\Delta t} = \mathcal{D}^{'*} - \beta \Delta t \nabla^{'2}(\mathcal{H}^*) \tag{53}$$

When $\overline{M} = M$ (case **LSIDG**=.T.) it is more convenient to rewrite equation (53) as:

$$(\nabla^{'-2} - \beta^2 \Delta t^2 \Phi^* M^2) D^{'}_{t+\Delta t} = \nabla^{'-2} \mathcal{D}^{'*} - \beta \Delta t (\mathcal{H}^*) \tag{54}$$

which shows a symmetric matricial operator in the left hand side.

$*$ **Spectral computations to solve system of equations (51) and (52).** Algorithm works zonal wave number by zonal wave number m ($\mid m \mid$ varies between 0 and the truncation $N_{\rm s}$) and performed in the routine **SPC2** before all horizontal diffusion schemes. For a given zonal wave number m:

- After a preliminary memory transfer the right-hand side member of equation (53) is computed for all total wave numbers n between m and $N_{\rm s}$.
- Inversion of Helmholtz equation for the case "reduced divergence" (case **LSIDG**=.F.): For each zonal wave number m: $(1-\beta^2\Delta t^2\Phi^*\nabla^{'2}\overline{M}^2)$ is a diagonal matricial operator $(N_{\rm s}+1- \mid m \mid)*(N_{\rm s}+1- \mid m \mid)$. Spectral coefficients of the right-hand side member of (53) are simply divided by the diagonal coefficients of this matrix.
- Inversion of Helmholtz equation for the case "unreduced divergence" (case **LSIDG**=.T.): Inversion of Helmholtz equation is more complicated than in the case of semi-implicit scheme with reduced divergence because the left-hand side member of Helmholtz equation contains values of the divergence for all levels and five total wave numbers ($n-2$ to $n+2$).
  - For each zonal wave number m: $(\nabla^{'-2}-\beta^2\Delta t^2 M^2\Phi^*)$ is a symmetric pentadiagonal matricial operator $(N_{\rm s}+1- \mid m \mid)*(N_{\rm s}+1- \mid m \mid)$. The factorisation `LU` of this matrix is computed, where `L` is a lower triangular tridiagonal matrix, `U` is a upper triangular tridiagonal matrix with coefficients equal to 1 on the main diagonal. All useful coefficients of `L`, `U` are computed in the set-up routine **SUHEG** and stored in the array **SIHEG**.
  - The right-hand side member of (54) is multiplied by the inverse of the symmetric pentadiagonal operator $(\nabla^{'-2}-\beta^2\Delta t^2\Phi^* M^2)$ (resolution of two tridiagonal triangular systems by routine **MXTURS**). That yields $D^{'}_{t+\Delta t}$.
  - For the zonal wave number $m=0$ equation (53) is used preferably than (54) because, for the total wave number $n=0$, $\nabla^{'2}$ is equivalent to a multiplication by 0 and $\nabla^{'-2}$ is equivalent to a division by 0. The only difference is that the pentadiagonal but non-symmetric operator $(1-\beta^2\Delta t^2\Phi^*\nabla^{'2}M^2)$ is factorised and inverted. All useful coefficients of `L`, `U` are computed in the set-up routine **SUHEG** and stored in the arrays **SIHEG** and **SIHEG2**.
- Once known $D^{'}_{t+\Delta t}$ equation (51) provides $\Phi_{t+\Delta t}$. For the case "unreduced divergence" (case **LSIDG**=.T.) only, spectral multiplications by M^2 are performed by the product of a symmetric pentadiagonal matrix of dimensions $(N_{\rm s}+1- \mid m \mid) * (N_{\rm s}+1- \mid m \mid)$ (useful coefficients computed in routine **SUSMAP** and stored in the array **SCGMAP**) by a vector containing spectral coefficients (m,n) for n varying from $\mid m \mid$ to $N_{\rm s}$.
- Semi-implicit scheme ends by a final memory transfer.

5.3 Plane geometry.

For 3D models, semi-implicit calculations are done in **ESPCSI**, **ESPNHSI**, **ESPNHSI_GEOGW** instead of **SPCSI**, **SPNHSI** and **SPNHSI_GEOGW**.

Option **LSIDG**=.T. has an equivalent **LESIDG**=.T. in the cycle AL37T1 of ALADIN, coded only for a tilted-rotated Mercator projection.

5.4 Shortcomings of the formulation of the semi-implicit scheme with "reduced divergence" (LSIDG=.F.) in case of stretching.

$*$ **ARPEGE/IFS:** In the grid points computations for some equations (for example temperature and continuity equation in the 3D hydrostatic model), the linear term $\mathcal{B}$

contains the reduced quantity $(\overline{M}^2 D^{'})$. This quantity is added to geographical quantities. That is no problem near the high resolution pole. This reduced quantity becomes very large near the low resolution pole: if the stretching coefficient is c, $\frac{\overline{M}^2}{M^2} = c^4$ at the low resolution pole, which is equal to 33.2 if $c = 2.4$. Thus the order of magnitude of the semi-implicit correction tendency becomes too high and physically absurd in the low resolution zone (gravity waves are no longer treated implicitly). That leads to instabilities in regions of the low resolution zone with high orography, in adiabatic eulerien runs, or in semi-Lagrangian runs with time-steps above the limit imposed by the Courant-Friedrich-Levy condition. In Eulerian runs with physics, the combination of physics and small time-steps inhibits this instability (at least in the hydrostatic model), but scores are degraded, especially far from the high resolution pole. In order to avoid this instability, we have implemented a new formulation of the semi-implicit scheme which allows to avoid mixing of reduced and geographical quantities in the grid-point computations, and which gives an implicit treatment of the gravity waves everywhere on the sphere and not only near the high resolution pole. This formulation is a formulation with unreduced divergence (simply by replacing the quantity $\overline{M}$ by the mapping factor M).
Remark: in the deep layer equations, the implicit treatment involves in this case the quantity $M^2 D^{'}$ and not D. The small residual term $D - M^2 D^{'} = (a/r - 1) M^2 D^{'}$ has an explicit treatment.

$*$ **ALADIN:** The previous point is not an issue in most applications of ALADIN where the mapping factor M has small variations in the forecast domain (which is not too large). It can become an issue for large domains (use of large domains with a tilted-rotated Mercator projection), and option **LESIDG** becomes useful in this case.

6 Inclusion of Coriolis term in the semi-implicit scheme.

Although this term is non linear, it is added to linear terms in the semi-implicit scheme. Equations are written for a leap-frog scheme (Eulerian scheme of three-time level semi-Lagrangian scheme). For a two-time level semi-Lagrangian scheme replace Δt by $0.5\Delta t$. In ARPEGE/IFS this option is coded for both Eulerian and semi-Lagrangian schemes (3D model and shallow-water model). This option is available only in unstretched untilted spherical geometry, in setting **LIMPF**=.T. in namelist **NAMDYN**. The reason to use such option is accuracy for longe range forecasts (but not stability issues: **LIMPF**=.F. is as stable as **LIMPF**=.T. but may lead to sligntly degraded scores for long-range forecasts).

6.1 Semi-implicit scheme including Coriolis term in the 3D hydrostatic model.

Thin layer equations.

∗ **Expression of the linear term $\mathcal{B}$:** Equations (37) and (38) become respectively:

- Divergence equation ($X = D^{'}$):

$$\mathcal{B} = -\nabla^{'2}(\gamma T + \mu \log \Pi_{\rm s}) - 2\nabla^{'}(\mathbf{\Omega} \wedge \mathbf{V}) \tag{55}$$

- Vorticity equation ($X = \zeta^{'}$):

$$\mathcal{B} = -2\mathbf{k}.[\nabla^{'} \wedge (\mathbf{\Omega} \wedge \mathbf{V})] \tag{56}$$

∗ **System to be solved:** Equations (40) and (42) are unchanged. Equation (41) is replaced by the two following equations, for divergence and vorticity.

$$D^{'}_{t+\Delta t} + \beta\Delta t \nabla^{'2}(\gamma T_{t+\Delta t} + \mu \text{log}(\Pi_{\rm s})_{t+\Delta t}) + \beta\Delta t(2\nabla^{'}(\mathbf{\Omega} \wedge \mathbf{V})) = \mathcal{D}^{'*} \tag{57}$$

$$\zeta^{'}_{t+\Delta t} + \beta\Delta t(2\mathbf{k}.[\nabla^{'} \wedge (\mathbf{\Omega} \wedge \mathbf{V})]) = \zeta^{'*} \tag{58}$$

$\mathcal{D}^{'*}$, ζ^{*} correspond to $\mathcal{X}^{*}$ defined in equation (17) and are available in spectral arrays (**SPDIV**, **SPVOR**) at the beginning of the spectral computations.

∗ **Restriction to not stretched and not tilted model:** Inclusion of Coriolis term im the semi-implicit scheme will be treated only in not stretched and not tilted geometry for different reasons, including the following considerations:

- In not stretched and not tilted geometry, Coriolis parameter $f = 2\Omega \sin\theta$ writes as a first degree polynomial of the sinus of computational sphere latitude. This property leads to invert pentadiagonal matrices in the algorithm which will be described.
- In stretched and not tilted geometry, Coriolis parameter $f = 2\Omega \sin\theta$ writes an homographical function of the sinus of computational sphere latitude. This property leads to invert full matrices in the algorithm which will be described (which in our case is very expensive in memory).
- In stretched and tilted geometry, Coriolis parameter $f = 2\Omega \sin\theta$ depends on the computational sphere longitude and latitude. This property leads to a coupling between all spectral coefficients and obliges to solve the semi-implicit system in the spectral space of geographical sphere, which is very expensive in memory and cost.

- There is another way to treat implicitly Coriolis term in the semi-Lagrangian scheme which works as well in the two-time level semi-Lagrangian scheme, replacing the prognostic variable $\mathbf{V}$ by $\mathbf{V} + 2\mathbf{\Omega} \wedge \mathbf{r}$, where $\mathbf{r}$ is the vertical vector from Earth centre to computational point ($\mathbf{r} = a\mathbf{k}$), and keeping the semi-implicit scheme unchanged. Contrary to inclusion of Coriolis term in the semi-implicit scheme, this method does not increase difficulty in stretched or tilted geometry.

In not stretched and not tilted geometry, equations (57) and (58) become:

$$D_{t+\Delta t} + \beta\Delta t\nabla^2(\gamma T_{t+\Delta t} + \mu \log(\Pi_{\rm s})_{t+\Delta t}) + \beta\Delta t(2\nabla(\mathbf{\Omega} \wedge \mathbf{V})) = \mathcal{D}^* \tag{59}$$

$$\zeta_{t+\Delta t} + \beta\Delta t(2\mathbf{k}.[\nabla \wedge (\mathbf{\Omega} \wedge \mathbf{V})]) = \zeta^* \tag{60}$$

∗ **Divergence and vorticity in spherical geometry:** For a vector $\mathbf{Y}$ of horizontal components $Y_{\rm x}$ and $Y_{\rm y}$, the divergence and vertical component of vorticity write as:

$$\nabla\mathbf{Y} = \frac{1}{a\cos\theta}\left[\frac{\partial Y_{\rm x}}{\partial\lambda} + \frac{\partial(Y_{\rm y}\cos\theta)}{\partial\theta}\right] \tag{61}$$

$$\mathbf{k}.(\nabla \wedge \mathbf{Y}) = \frac{1}{a\cos\theta}\left[\frac{\partial Y_{\rm y}}{\partial\lambda} - \frac{\partial(Y_{\rm x}\cos\theta)}{\partial\theta}\right] \tag{62}$$

∗ **Helmholtz equation:** Using relations (61) and (62) in equations (59) and (60) lead to the following equations:

$$D_{t+\Delta t}+\beta\Delta t\nabla^2(\gamma T_{t+\Delta t}+\mu \log(\Pi_{\rm s})_{t+\Delta t})+\beta\Delta t(-2\Omega\sin\theta)\zeta_{t+\Delta t}+\beta\Delta t\left(\frac{2\Omega\cos\theta}{a}\right)U_{t+\Delta t} = \mathcal{D}^* \tag{63}$$

$$\zeta_{t+\Delta t} + \beta\Delta t(2\Omega\sin\theta)D_{t+\Delta t} + \beta\Delta t\left(\frac{2\Omega\cos\theta}{a}\right)V_{t+\Delta t} = \zeta^* \tag{64}$$

The four following expressions are used, for each complex spectral coefficient:

- Relationship between divergence and velocity potential χ:

$$D = \nabla^2\chi \tag{65}$$

- Relationship between vorticity and stream function ψ:

$$\zeta = \nabla^2\psi \tag{66}$$

- (65) and (66) can be rewritten, for each complex spectral component:

$$D_{(m,n)} = -\frac{n(n+1)}{a^2}\chi_{(m,n)} \tag{67}$$

$$\zeta_{(m,n)} = -\frac{n(n+1)}{a^2}\psi_{(m,n)} \tag{68}$$

- Relationship between U, ψ and χ:

$$(Ua\cos\theta)_{(m,n)} = im\chi_{(m,n)}+(n-1)e_{(m,n)}\psi_{(m,n-1)}-(n+2)e_{(m,n+1)}\psi_{(m,n+1)} \tag{69}$$

- Relationship between V, ψ and χ:

$$(Va\cos\theta)_{(m,n)} = im\psi_{(m,n)} - (n-1)e_{(m,n)}\chi_{(m,n-1)} + (n+2)e_{(m,n+1)}\chi_{(m,n+1)} \tag{70}$$

where $e_{(0,0)} = 0$ and:

$$e_{(m,n)} = \sqrt{\frac{n^2 - m^2}{4n^2 - 1}} \tag{71}$$

Equations (67) to (70) allow to eliminate $U_{t+\Delta t}$ and $V_{t+\Delta t}$ in equations (63) and (64).

$$\begin{aligned}\left[1 - i\tfrac{2\Omega\beta\Delta tm}{n(n+1)}\right] D_{(m,n),t+\Delta t} + \beta\Delta t\nabla^2(\gamma T_{(m,n),t+\Delta t} + \mu\log(\Pi_\mathrm{s})_{(m,n),t+\Delta t}) - \beta\Delta t(2\Omega(\sin\theta)\zeta)_{(m,n),t+\Delta t} \\ -\beta\Delta t\tfrac{2\Omega e_{(m,n)}}{n}\zeta_{(m,n-1),t+\Delta t} + \beta\Delta t\tfrac{2\Omega e_{(m,n+1)}}{n+1}\zeta_{(m,n+1),t+\Delta t} = \mathcal{D}^*_{(m,n)}\end{aligned} \tag{72}$$

$$\begin{aligned}\left[1 - i\tfrac{2\Omega\beta\Delta tm}{n(n+1)}\right] \zeta_{(m,n),t+\Delta t} + \beta\Delta t(2\Omega(\sin\theta)D)_{(m,n),t+\Delta t} + \beta\Delta t\tfrac{2\Omega e_{(m,n)}}{n}D_{(m,n-1),t+\Delta t} \\ -\beta\Delta t\tfrac{2\Omega e_{(m,n+1)}}{n+1}D_{(m,n+1),t+\Delta t} = \zeta^*_{(m,n)}\end{aligned} \tag{73}$$

Multiplication by $\sin\theta$ is eliminated by using the following relationship valid for any variable X, in not stretched and not tilted geometry:

$$[(\sin\theta)X]_{(m,n)} = e_{(m,n)}X_{(m,n-1)} + e_{(m,n+1)}X_{(m,n+1)} \tag{74}$$

Equations (72) and (73) become:

$$\begin{aligned}\left[1 - i\tfrac{2\Omega\beta\Delta tm}{n(n+1)}\right] D_{(m,n),t+\Delta t} + \beta\Delta t\nabla^2(\gamma T_{(m,n),t+\Delta t} + \mu\log(\Pi_\mathrm{s})_{(m,n),t+\Delta t}) - \beta\Delta t\tfrac{2\Omega e_{(m,n)}(n+1)}{n}\zeta_{(m,n-1),t+\Delta t} \\ -\beta\Delta t\tfrac{2\Omega e_{(m,n+1)}n}{n+1}\zeta_{(m,n+1),t+\Delta t} = \mathcal{D}^*_{(m,n)}\end{aligned} \tag{75}$$

$$\begin{aligned}\left[1 - i\tfrac{2\Omega\beta\Delta tm}{n(n+1)}\right] \zeta_{(m,n),t+\Delta t} + \beta\Delta t\tfrac{2\Omega e_{(m,n)}(n+1)}{n}D_{(m,n-1),t+\Delta t} + \beta\Delta t\tfrac{2\Omega e_{(m,n+1)}n}{n+1}D_{(m,n+1),t+\Delta t} \\ = \zeta^*_{(m,n)}\end{aligned} \tag{76}$$

ζ is eliminated in equation (75) by using (76) in replacing n by $n-1$ then by $n+1$. Equation (75) becomes:

$$\begin{aligned}\left[\mathtt{I} - i\tfrac{2\Omega\beta\Delta tm}{n(n+1)} + \tfrac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta tm}{(n-1)n}}e^2_{(m,n)}\tfrac{(n-1)(n+1)}{n^2} + \tfrac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta tm}{(n+1)(n+2)}}e^2_{(m,n+1)}\tfrac{(n)(n+2)}{(n+1)^2}\right] D_{(m,n),t+\Delta t} \\ +\tfrac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta tm}{(n-1)n}}e_{(m,n)}e_{(m,n-1)}\tfrac{(n+1)}{(n-1)}D_{(m,n-2),t+\Delta t} + \tfrac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta tm}{(n+1)(n+2)}}e_{(m,n+1)}e_{(m,n+2)}\tfrac{n}{(n+2)}D_{(m,n+2),t+\Delta t} \\ +\beta\Delta t\nabla^2(\gamma T_{(m,n),t+\Delta t} + \mu\log(\Pi_\mathrm{s})_{(m,n),t+\Delta t}) \\ = \mathcal{D}^*_{(m,n)} + \tfrac{(2\beta\Delta t\Omega)}{1-i\frac{2\Omega\beta\Delta tm}{(n-1)n}}e_{(m,n)}\tfrac{(n+1)}{n}\zeta^*_{(m,n-1)} + \tfrac{(2\beta\Delta t\Omega)}{1-i\frac{2\Omega\beta\Delta tm}{(n+1)(n+2)}}e_{(m,n+1)}\tfrac{n}{(n+1)}\zeta^*_{(m,n+1)}\end{aligned} \tag{77}$$

$T_{t+\Delta t}$ and $\log(\Pi_{\rm s})_{t+\Delta t}$ are eliminated by using equations (40) and (42). That leads to Helmholtz equation (78):

$$\begin{aligned}
&\left[\mathtt{I} - i\tfrac{2\Omega\beta\Delta t m}{n(n+1)} + \tfrac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta t m}{(n-1)n}} e^2_{(m,n)} \tfrac{(n-1)(n+1)}{n^2} + \tfrac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta t m}{(n+1)(n+2)}} e^2_{(m,n+1)} \tfrac{(n)(n+2)}{(n+1)^2} - \beta^2\Delta t^2\ \mathtt{B}\ \nabla^2\right] D_{(m,n),t+\Delta} \\
&+\tfrac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta t m}{(n-1)n}} e_{(m,n)} e_{(m,n-1)} \tfrac{(n+1)}{(n-1)} D_{(m,n-2),t+\Delta t} + \tfrac{(2\beta\Delta t\Omega)^2}{1-i\frac{2\Omega\beta\Delta t m}{(n+1)(n+2)}} e_{(m,n+1)} e_{(m,n+2)} \tfrac{n}{(n+2)} D_{(m,n+2),t+\Delta t} \\
&= \mathcal{D}^*_{(m,n)} - \beta\Delta t\nabla^2(\gamma\mathcal{T}^*_{(m,n)} + \mu\mathcal{P}^*_{(m,n)}) \\
&+\tfrac{(2\beta\Delta t\Omega)}{1-i\frac{2\Omega\beta\Delta t m}{(n-1)n}} e_{(m,n)} \tfrac{(n+1)}{n} \zeta^*_{(m,n-1)} + \tfrac{(2\beta\Delta t\Omega)}{1-i\frac{2\Omega\beta\Delta t m}{(n+1)(n+2)}} e_{(m,n+1)} \tfrac{n}{(n+1)} \zeta^*_{(m,n+1)} \qquad (
\end{aligned}$$

Equation (78) is solved in eigenmodes space. The right-hand side member needs a multiplication by a tridiagonal complex matrix, for each zonal wave number m. Inversion of Helmholtz equation is equivalent to invert a pentadiagonal complex matrix (done in routine **SIMPLICO**), for each zonal wave number m. All computations are currently (in cycle 37T1) done in **SPCSI**. For $m = 0$ complex operators become real operators.

$*$ **Determination of other quantities at $t + \Delta t$:** Equation (77) is used to compute $\zeta_{t+\Delta t}$ (for each zonal wave number m, multiplication by a complex tridiagonal matrix). Then equation (40) is used to compute $\log(\Pi_{\rm s})_{t+\Delta t}$ and equation (42) is used to compute $T_{t+\Delta t}$.

Deep layer equations (according to White and Bromley, 1995).

Equations (55) to (58) remain valid. Only the horizontal part of the Coriolis term ($-2\mathbf{\Omega}\wedge\mathbf{V}$) can be included in the semi-implicit scheme. The term ($-2\mathbf{\Omega}\wedge W\mathbf{k}$) remains explicit. Equations (59) to (78) remain valid, replacing D by $(r_{\rm s}/a)D$, ζ by $(r_{\rm s}/a)\zeta$, ∇ by $(r_{\rm s}/a)\nabla$. The code of spectral computations is unchanged.

6.2 Semi-implicit scheme including Coriolis term in the 2D shallow-water model.

Such an algorithm is also coded in the 2D shallow-water model in not stretched and not tilted geometry.

- Replace $(\gamma T + \mu\log(\Pi_{\rm s}))$ by Φ in equation (55).
- Replace $(\gamma T_{t+\Delta t} + \mu\log(\Pi_{\rm s})_{t+\Delta t})$ by $\Phi_{t+\Delta t}$ in equations (57), (59) and (63).
- Replace $(\gamma T_{(m,n),t+\Delta t} + \mu\log(\Pi_{\rm s})_{(m,n),t+\Delta t})$ by $\Phi_{(m,n),t+\Delta t}$ in equations (72), (75) and (77).
- Replace $(\gamma\mathcal{T}^*_{(m,n)} + \mu\mathcal{P}^*_{(m,n)})$ by $\mathcal{H}^*_{(m,n)}$ in equation (78).
- Once computed $D^{'}_{t+\Delta t}$ equation (51) provides $\Phi_{t+\Delta t}$.

6.3 Conclusion.

The simplest configuration to treat the case **LIMPF**=.T. is the case where the unknown is the horizontal divergence in the Helmoltz equation: this is the case in the hydrostatic model and the shallow-water model, and treatment of implicit Coriolis term is similar in all these cases, with only one call to **SIMPLICO**.

7 Spectral multiplications by polynomial expressions of the mapping factor.

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

- $a_{\mathrm{c}} = 0.5(c + \frac{1}{c})$
- $b_{\mathrm{c}} = 0.5(c - \frac{1}{c})$
- $e_{(0,0)} = 0$
- $e_{(m,n)} = \sqrt{\frac{n^2-m^2}{4n^2-1}}$

Expression of M is:

$$M = a_{\mathrm{c}} + b_{\mathrm{c}}\xi \tag{79}$$

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

$$[MX]_{(m,n)} = b_{\mathrm{c}}e_{(m,n)}X_{(m,n-1)} + a_{\mathrm{c}}X_{(m,n)} + b_{\mathrm{c}}e_{(m,n+1)}X_{(m,n+1)} \tag{80}$$

It is easy from (80) to retrieve the coefficients of spectral multiplication by any first degree polynomial of M. This is equivalent to a multiplication by a tridiagonal symmetric matrix in spectral space.

∗ Expression of M^2 in spectral space in ARPEGE.

$$\begin{aligned}[M^2X]_{(m,n)} = {} & b_{\mathrm{c}}^2 e_{(m,n)}e_{(m,n-1)}X_{(m,n-2)} + 2a_{\mathrm{c}}b_{\mathrm{c}}e_{(m,n)}X_{(m,n-1)} + (a_{\mathrm{c}}^2 + b_{\mathrm{c}}^2(e_{(m,n)}^2 + e_{(m,n+1)}^2))X_{(m,n)} \\ & +2a_{\mathrm{c}}b_{\mathrm{c}}e_{(m,n+1)}X_{(m,n+1)} + b_{\mathrm{c}}^2 e_{(m,n+1)}e_{(m,n+2)}X_{(m,n+2)} \end{aligned} \tag{81}$$

This is equivalent to a multiplication by a pentadiagonal symmetric matrix in spectral space.

∗ Expression of M and M^2 in spectral space in ALADIN. This formula is only valid on a tilted-rotated Mercator projection, and it assumes that the reference latitude of the projection is at the middle of sub-domain C+I.

If we assume that the ALADIN plane coordinates will be not slanted relatively to the longitudes and latitudes of the Mercator projection, the mapping factor M always depends only on the y coordinate and never vary along the x coordinate.

$$M = \frac{1}{\cos\theta} = \frac{1}{\sqrt{1-\mu^2}} = \cosh(y/a)$$

where $\mu = \sin\theta$, the y coordinate (this is an distance measured on the plane projection) assumes that $y = 0$ at the apparent equator, and a is the mean Earth radius. M is not a low order polynomial function of y so even in this case it needs to be approximated. The approximation used is, for a Fourier decomposition of M^2, to have only two harmonics.

6

Horizontal diffusion

1 Introduction.

∗ **General considerations for ARPEGE/IFS:** In the code of ARPEGE/IFS 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 ALADIN:** 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 ARPEGE one. There are specific routines for spectral calculations.

∗ **Rayleigh friction:** there is a Rayleigh friction in the grid-point space computations which acts only in upper atmosphere for the U-component of the horizontal wind (properly coded only for untilted spherical geometry or untilted plane geometry).

∗ **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 conveniency, 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.

∗ **Distributed memory:** some distributed memory features are now introduced in the code and will be briefly described. For convenience one uses some generic appellations.

- Expression “DM-local" for a quantity means “local to the couple of processors (*proca*,*procb*)": each processor has its own value for the quantity. Expression “DM-local computations" means that the computations are done independently in each processor on “DM-local" quantities, leading to results internal to each processor, which can be different from a processor to another one.
- Expression “DM-global" for a quantity means that it has a unique value available in all the processors. Expression “DM-global computations" means that the computations are either done in one processor, then the results are dispatched in all the processors, or the same computations are done in all the processors, leading to the same results in all the processors.
- In a routine description the mention “For distributed memory computations are DM-local" means that all calculations done by this routine are DM-local; the mention “For distributed memory computations are DM-global" means that all calculations done by this routine are DM-global; when no information is provided it means that a part of calculations are DM-local and the other part is DM-global.
- Expression “main" processor currently refers to the processor number 1: (*proca*,*procb*)=(1,1).

2 Formulation of the horizontal diffusion schemes.

2.1 Main horizontal diffusion scheme.

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

$$\frac{\partial X}{\partial t} = -K_X^{''} M \vec{\nabla}'^{r} X \tag{1}$$

where $K_X^{''}$ is a vertically dependent and horizontally constant coefficient. $K_X^{''}$ is generally complex: $\exp(0.5\pi i r)$ multiplied by a real positive coefficient. r is the power of the horizontal diffusion scheme. M is the mapping factor.

In ARPEGE:

$$K_X^{''} = \exp(-0.5\pi i r) \left[\sqrt{\frac{N_{\rm s}(N_{\rm s}+1)}{a^2}} \right]^{-r} \Omega\ h_X\ g \tag{2}$$

(a is the mean Earth radius, $N_{\rm s}$ 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 ($h_{\rm EXT}$), 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:

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

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

In ALADIN:

$$K_X^{''} = \exp(-0.5\pi i r) \left[(2\pi) \sqrt{\frac{1}{\frac{L_{\mathrm{x}}^2}{N_{\mathrm{ms}}^2} + \frac{L_{\mathrm{y}}^2}{N_{\mathrm{s}}^2}}} \right]^{-r} \Omega\ h_X\ g \tag{4}$$

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

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

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 ARPEGE and ALADIN, for the other 3D upper air fields:

$$\frac{1}{\Omega h_X} = \frac{r_{\mathrm{dampX}}}{r_{\mathrm{dampdiv}}} \frac{1}{\Omega h_D} \tag{6}$$

which can be rewritten:

$$(\Omega h_X) = \frac{r_{\mathrm{dampdiv}}}{r_{\mathrm{dampX}}} (\Omega h_D) \tag{7}$$

$*$ 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 $\frac{1}{\Omega h_{\mathrm{hds}X}} = \frac{1}{r_{\mathrm{damphds}}} \frac{1}{\Omega h_X}$, i.e. $(\Omega h_{\mathrm{hds}X}) = r_{\mathrm{damphds}} (\Omega h_X)$).
  - $K_X^{''}$ becomes $\mathcal{K}_{\mathrm{hds}X}$.
  - the power r becomes s (s is generally above r).

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 ARPEGE, $M=1$. In ALADIN, the horizontal variations of M are generally neglected and M is replaced by a constant hidden in the coefficients $K_X^{''}$ and $\mathcal{K}_{\mathrm{hds}\,X}$. So the ALADIN diffusion looks like the unstretched ARPEGE one. This approximation may become not very appropriate for large domains (used particularly with the rotated-tilted Mercator projection).

2.2 Rayleigh friction.

The following scheme is applied on the U-component of the momentum equation (i.e. the zonal component on the Gaussian grid).

$$\left(\frac{\partial U}{\partial t}\right)_{\mathrm{fric}} = -K_{\mathrm{fric}} U \tag{8}$$

$-K_{\mathrm{fric}}U$ is a Rayleigh friction which is activated if **LRFRIC**=.T. in **NAMCT0**. K_{fric} is a vertical dependent coefficient which is non-zero only for a standard pressure lower than a threshold stored in variable **RRFPLM**.

2.3 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.

Main horizontal diffusion scheme in unstretched ARPEGE/IFS.

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

$$X^{+}_{(m,n)} - X^{-}_{(m,n)} = -\Omega\ h_X\ g(l) f(n, N, n_0(X), x_0, r) Dt X^{+}_{(m,n)} \tag{9}$$

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), x_0 is a threshold between 0 and 1. $\nabla^{'}$ 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 (9) becomes (10):

$$X^+_{(m,n)} = \frac{1}{1 + K_X(n)Dt} X^-_{(m,n)} \tag{10}$$

where:

$$K_X(n) = \Omega \; h_X \; g(l) f(n, N_{\mathrm{s}}, n_0(X), x_0, r) \tag{11}$$

Expression of $f(n, N, n_0(X), x_0, r)$ is currently:

$$f(n, N, n_0(X), x_0, r) = \max \left(0., \min \left(1., \left(\frac{\left(\frac{\max(0, n(n+1) - n_0)}{\max(0, N(N+1) - n_0)} \right)^{\frac{1}{2}} - x_0}{1 - x_0} \right)^r \right) \right) \tag{12}$$

An exact discretisation of equation (1) would give:

$$f(n, N, n_0(X), x_0, r) = f(n, N_{\mathrm{s}}, 0, 0, r) \tag{13}$$

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

$*$ **Vertical dependency of g:** $g(l)$ only depends on the altitude. For the layer number l, expression of $g(l)$ is:

$$g(l) = \min \left(y_0 \frac{\Pi_{ref}}{\Pi_{ST}(l)}, \frac{1}{y_3} \right) \tag{14}$$

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. In the 2D model, $g(l)$ can only take one value equal to 1. Definition of $g(l)$ is different in ECMWF configuration (**LECMWF**=.T.).

Main horizontal diffusion scheme in ALADIN.

Compared to ARPEGE, 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 conveniency one will let n_0 in the following formulae).
- Equation (9) becomes:

  $$X^+_{(m,n)} - X^-_{(m,n)} = -\Omega \; h_X \; g(l) f(m, n, N_{\mathrm{m}}, N, n_0(X), x_0, r) Dt X^+_{(m,n)} \tag{15}$$

  where N_{m} is a reference zonal wavenumber.

- Equation (10) becomes:

$$X^{+}_{(m,n)} = \frac{1}{1 + K_X(m,n)Dt} X^{-}_{(m,n)} \tag{16}$$

- Equation (11) becomes:

$$K_X(m,n) = \Omega\ h_X\ g(l) f(m,n,N_{\mathrm{ms}},N_{\mathrm{s}},n_0(X),x_0,r) \tag{17}$$

- Expression of $f(m,n,N_{\mathrm{m}},N,n_0(X),x_0,r)$ is currently:

$$f(m,n,N_{\mathrm{m}},N,n_0(X),x_0,r) = \max\left(0., \min\left(1., \left(\frac{\left(\frac{n^2}{N_{\mathrm{s}}^2} + \frac{m^2}{N_{\mathrm{ms}}^2}\right)^{\frac{1}{2}} - x_0}{1 - x_0}\right)^r\right)\right) \tag{18}$$

- An exact discretisation of equation (1) would give:

$$f(m,n,N_{\mathrm{m}},N,n_0(X),x_0,r) = f(m,n,N_{\mathrm{ms}},N_{\mathrm{s}},0,0,r) \tag{19}$$

Main horizontal diffusion scheme in stretched ARPEGE/IFS.

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

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

Expression of M is:

$$M = a_{\mathrm{c}} + b_{\mathrm{c}}\mu \tag{20}$$

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

$$[MX]_{(m,n)} = b_{\mathrm{c}} e_{(m,n)} X_{(m,n-1)} + a_{\mathrm{c}} X_{(m,n)} + b_{\mathrm{c}} e_{(m,n+1)} X_{(m,n+1)} \tag{21}$$

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:

$$\begin{gathered} X^{-}_{(m,n)} = \\ + b_{\mathrm{c}} e_{(m,n)} \Omega\ h_X\ g(l) Dt f(n-1, N, n_0(X), x_0, r) X^{+}_{(m,n-1)} \\ + (1 + a_{\mathrm{c}} \Omega\ h_X\ g(l) Dt f(n, N, n_0(X), x_0, r)) X^{+}_{(m,n)} \\ + b_{\mathrm{c}} e_{(m,n+1)} \Omega\ h_X\ g(l) Dt f(n+1, N, n_0(X), x_0, r) X^{+}_{(m,n+1)} \end{gathered} \tag{22}$$

Equation (22) 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 matrixes are performed in the set-up routine **SUHDU**. At each time step these two triangular bidiagonal matrixes 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 $g(l)$:** Expression of $g(l)$ is now modified; new expression of $g(l)$ is:

$$g(l) = \min\left(y_0 \frac{\Pi_{ref}}{\Pi_{ST}(l)}, \frac{1}{y_3} \right) - s_{\mathrm{dred}} \tag{23}$$

∗ **Unstretched ARPEGE/IFS:** A double diffusion is applied on vorticity and divergence. It writes:

$$X^{+}_{(m,n)} = \left[\frac{1}{1 + K_{\mathrm{hds}X}(n)Dt} \right] \left[\frac{1}{1 + K_X(n)Dt} \right] X^{-}_{(m,n)} \tag{24}$$

where:

$$K_{\mathrm{hds}X}(n) = \Omega\ h_{\mathrm{hds}X}\ g_{\mathrm{hds}}(l) f(n, N_{\mathrm{s}}, n_0(X), x_0, s) \tag{25}$$

∗ **ALADIN:** A double diffusion is applied on vorticity and divergence. It writes:

$$X^{+}_{(m,n)} = \left[\frac{1}{1 + K_{\mathrm{hds}X}(m,n)Dt} \right] \left[\frac{1}{1 + K_X(m,n)Dt} \right] X^{-}_{(m,n)} \tag{26}$$

where:

$$K_{\mathrm{hds}X}(m,n) = \Omega\ h_{\mathrm{hds}X}\ g_{\mathrm{hds}}(l) f(m, n, N_{\mathrm{ms}}, N_{\mathrm{s}}, n_0(X), x_0, s) \tag{27}$$

∗ **Stretched ARPEGE/IFS:** A double diffusion is applied on vorticity and divergence. It writes:

$$\begin{gathered} X^{-}_{(m,n)} = \\ + b_{\mathrm{c}} e_{(m,n)} \Omega\ h_X\ g(l) Dt f(n-1, N, n_0(X), x_0, r) X^{++}_{(m,n-1)} \\ + (1 + a_{\mathrm{c}} \Omega\ h_X\ g(l) Dt f(n, N, n_0(X), x_0, r)) X^{++}_{(m,n)} \\ + b_{\mathrm{c}} e_{(m,n+1)} \Omega\ h_X\ g(l) Dt f(n+1, N, n_0(X), x_0, r) X^{++}_{(m,n+1)} \end{gathered} \tag{28}$$

then:

$$\begin{gathered} X^{++}_{(m,n)} = \\ + b_{\mathrm{c}} e_{(m,n)} \Omega\ h_{\mathrm{hds}X}\ g_{\mathrm{hds}}(l) Dt f(n-1, N, n_0(X), x_0, s) X^{+}_{(m,n-1)} \\ + (1 + a_{\mathrm{c}} \Omega\ h_{\mathrm{hds}X}\ g_{\mathrm{hds}}(l) Dt f(n, N, n_0(X), x_0, s)) X^{+}_{(m,n)} \\ + b_{\mathrm{c}} e_{(m,n+1)} \Omega\ h_{\mathrm{hds}X}\ g_{\mathrm{hds}}(l) Dt f(n+1, N, n_0(X), x_0, s) X^{+}_{(m,n+1)} \end{gathered} \tag{29}$$

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

$*$ **Expression of $g_{\mathrm{hds}}(l)$:** Expression of $g_{\mathrm{hds}}(l)$ is:

$$g_{\mathrm{hds}}(l) = \min\left(yS_0\frac{\Pi_{ref}}{\Pi_{ST}(l)}, \frac{1}{y_3}\right) \tag{30}$$

3.3 Particular use of a second reference truncation N_2 for standard pressure levels $\Pi_{ST}(l)$ below $y_0 * \Pi_{ref}$.

In ARPEGE, it is possible in the code to modify $g(l)$ in order to simulate a replacing of $f(n, N_{\mathrm{s}}, n_0(X), x_0, r)$ by $f(n, N_2, n_0(X), x_0, r)$, where $N_2 < N_{\mathrm{s}}$, for standard pressures $\Pi_{ST}(l) < y_1 * \Pi_{ref}$, where $y_1 < y_0$. In this case $g(l)$ also depends on n for standard pressures below $y_0 * \Pi_{ref}$.
If $\Pi_{ST}(l)/\Pi_{ref}$ is below y_1 (y_3 is assumed to be lower than y_1):

$$g(l,n) = \min\left(y_0\frac{\Pi_{ref}}{\Pi_{ST}(l)}, \frac{1}{y_3}\right)\frac{f(n, N_2, n_0(X), x_0, r)}{f(n, N_{\mathrm{s}}, n_0(X), x_0, r)} \tag{31}$$

If $\Pi_{ST}(l)/\Pi_{ref}$ is over y_1 and below y_0:

$$g(l,n) = \frac{y_0 * \Pi_{ref} - b(n)}{\Pi_{ST}(l) - b(n)} \tag{32}$$

where:

$$b(n) = \frac{(y_0 * \Pi_{ref})(y_1 * \Pi_{ref})(f(n, N_2, n_0(X), x_0, r) - f(n, N_{\mathrm{s}}, n_0(X), x_0, r))}{(y_0 * \Pi_{ref})f(n, N_2, n_0(X), x_0, r) - (y_1 * \Pi_{ref})f(n, N_{\mathrm{s}}, n_0(X), x_0, r)} \tag{33}$$

One can check that (32) and (33) yield $g(l,n) = 1$ for $\Pi_{ST} = y_0 * \Pi_{ref}$ and that (31) and (32) are continuous for $\Pi_{ST} = y_1 * \Pi_{ref}$. When taking $N_2 = N_{\mathrm{s}}$, one retrieves formula (14). Choice of $N_2 < N_{\mathrm{s}}$ is useful in climate modelisation.

The same type of modification can be extended to g_{hds}.

This option is not coded in ALADIN.

3.4 Application of diffusion to each variable.

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

For GMV variables, horizontal diffusion is applied to:

- “reduced" vorticity $\zeta^{'}$ 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 3.4), with $n_0(T) = 0$.
- $\log(\Pi_{\mathrm{s}})$ 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.

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 $T-\alpha\log(\Pi_{\rm s})$ with a "good" formulation of α. $T-\alpha\log(\Pi_{\rm s})$ 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_{sST} 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_{tST} the tropopause temperature in a standard atmosphere (217.15 K), and Π_{ref} a reference temperature equal to 1013.25 hPa. If:

$$(T_{sST})(\Pi/\Pi_{ref})^{(R/g)[\frac{dT}{dz}]_{ST}} \ > \ (T_{tST})$$

the expression of α_l for the layer number l is:

$$\alpha_l = -B_l(R/g)\left[\frac{dT}{dz}\right]_{ST}(T_{sST})(\Pi/\Pi_{ref})^{-((R/g)[\frac{dT}{dz}]_{ST}+1)} \tag{34}$$

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

$$\alpha_l = 0 \tag{35}$$

Temperature horizontal diffusion writes as:

$$(T^{+}_{(m,n)} - \alpha\log(\Pi_{\rm s})^{-}_{(m,n)}) = \frac{1}{1+K_T(n)Dt}(T^{-}_{(m,n)} - \alpha\log(\Pi_{\rm s})^{-}_{(m,n)}) \tag{36}$$

Equation (36) yields (37):

$$T^{+}_{(m,n)} = (T^{-}_{(m,n)} + K_T(n)Dt\alpha\log(\Pi_{\rm s})^{-}_{(m,n)})/(1+K_T(n)Dt) \tag{37}$$

For temperature $n_0(T)=0$.

Application to GMV variables in 2D models.

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

3.5 Other spectral calculations.

∗ Numerical horizontal diffusion for post-processed fields by FULL-POS. See documentation (IDFPOS) about FULL-POS. This diffusion is done in routine **SPOS**.

∗ Rayleigh friction (ARPEGE/IFS only). The following scheme is discretised in grid-point space for the U-component of the momentum equation (i.e. the zonal component on the Gaussian grid). In equation (8) $-K_{\rm fric}U$ is a Rayleigh friction which is activated if **LRFRIC**=.T. in **NAMCT0**. $K_{\rm fric}$ is a vertical dependent coefficient which is non-zero only for a standard pressure lower than $\Pi_{\rm rfric}$. $K_{\rm fric}$ is computed in **SURAYFRIC** according the following formula for a layer l of standard pressure $\Pi_{ST}(l)$ lower than $\Pi_{\rm rfric}$:

$$K_{\rm fric}(l) = \frac{1-\frac{1}{7.7}\tanh\left(R_{\rm fz1} - 7\log(100000/\Pi_{ST}(l))\right)}{3*86400} \tag{38}$$

$R_{\rm fz1}$ is stored in variable **RRFZ1** in **YOMDYN**. $K_{\rm fric}$ is stored in variable **RKRF** in **YOMDYN**. This scheme has never been used at METEO-FRANCE, but only at ECMWF. It is suited for models having layers in the high stratosphere, with an untilted geometry (i.e. **NSTTYP**=1).

7

Radiative fluxes

This chapter describes the salient features concerning the radiation scheme, used in NWP as well as in ARPEGE-CLIMAT. The present chapter is based on the IFS Documentation-Cy37r2. The detailed original ECMWF scientific documentation is available online at `http://www.ecmwf.int/sites/default/files/elibrary/2012/9239-part-iv-physical-processes.pdf`. The package calculates the radiative fluxes taking into account absorption-emission of longwave radiation and reflection, scattering and absorption of solar radiation by the earth's atmosphere and surfaces. The longwave radiation scheme is based on that of the ECMWF model, the Rapid Radiation Transfer Model (RRTM). The shortwave part of the scheme, originally developed by Fouquart and Bonnel (1980), solves the radiation transfer equation and integrates the fluxes over the whole shortwave spectrum between 0.2 and 4 mm. Note that, since 2007, a new approach to the inclusion of the cloud effects on radiation fields (the Monte-Carlo Independent Column Approximation, McICA) has been introduced in the ECMWF radiation schemes , but not used yet in the ARPEGE NWP (where tests have been done) and Climate models. The radiative heating rate is computed as the divergence of net radiation fluxes F so that

$$(\frac{\partial T}{\partial t})_{rad} = -\frac{g}{c_p}\frac{\partial F}{\partial p} \tag{1}$$

where F is a net flux: i.e. $F = F^{\uparrow} + F^{\downarrow}$ sum of the upward $F^{\uparrow}$ and downward $F^{\downarrow}$ fluxes, and a total flux: i.e. $F = F_{LW} + F_{SW}$ sum of the solar or shortwave F_{SW} and atmospheric or longwave F_{LW} fluxes and c_p is the specific heat at constant pressure of moist air.

Sections 1 and 2 describe the computation of the shortwave and longwave radiative fluxes respectively. A description of the inputs, in particular the climatologically defined quantities of radiative importance is given in Section 3. In the CMIP6 version, the full radiation scheme is called every hour (corresponding to NRADFR=-1). There is no modification of the ECMWF code done in the ARPEGE code, except the choices made for optical parameters and overlap hypothesis. Note also that, since 2007, a new approach to the inclusion of the cloud effects on radiation fields (the Monte-Carlo Independent Column Approximation, McICA approximation) has been introduced in the ECMWF radiation schemes, but is not used in the ARPEGE NWP and Climate current versions.

1 Shortwave radiation

1.1 First glance

The rate of atmospheric warming by absorption and scattering of shortwave radiation is:

$$\frac{\partial T}{\partial t} = \frac{g}{c_p} \frac{\partial F_{SW}}{\partial p} \tag{2}$$

where F_{SW} is the net total shortwave flux, expressed in W m^{-2} and positive when downward.

$$F = \int_0^\infty d\nu \int_0^{2\pi} d\phi \int_{-1}^{+1} \mu \, L_\nu(\delta, \mu, \phi) \, d\mu \, d\phi \tag{3}$$

L_ν is the diffuse radiance at wavenumber ν, in a direction given by ϕ, the azimuth angle and ϑ the zenith angle such as $\mu = \cos\vartheta$. In (3), we assume a plane parallel atmosphere with the optical depth δ, as a convenient vertical coordinate when the energy source is outside the medium

$$\delta(p) = \int_p^0 \beta_\nu(p) \, dp \tag{4}$$

where $\beta_\nu^{ext}(p)$ is the extinction coefficient equal to the sum of the scattering coefficient β_ν^{sca} of the aerosol and cloud particle absorption coefficient β_ν^{abs} and of the purely molecular absorption coefficient k_ν. The diffuse radiance L_ν is governed by the radiation transfer equation

$$\begin{aligned} \mu \, \frac{dL_\nu(\delta, \mu, \phi)}{d\delta} \quad &= \quad L_\nu(\delta, \mu, \phi) - \frac{\overline{\omega}_\nu(\delta)}{4} \, P_\nu(\delta, \mu, \phi, \mu_0, \phi_0) \, E_\nu^0 e^{-\frac{\delta}{\mu_0}} && (5) \\ & - \frac{\overline{\omega}_\nu(\delta)}{4} \int_0^{2\pi} \int_{-1}^{+1} P_\nu(\delta, \mu, \phi, \mu', \phi') \, L_\nu(\delta, \mu', \phi') \, d\mu' \, d\phi'. && (6) \end{aligned}$$

E_ν^0 is the incident solar irradiance in the direction $\mu_0 = cos\vartheta_0$, $\overline{\omega}_\nu$ is the single scattering albedo ($= \beta_\nu^{sca}/k_\nu$) and $P_\nu(\delta, \mu, \phi, \mu', \phi')$ is the scattering phase function which defines the probability that radiation coming from direction (μ', ϕ') is scattered in direction (μ, ϕ). The shortwave part of the scheme, originally developed by Fouquart and Bonnel (1980), solves the radiation transfer equation and integrates the fluxes over the whole shortwave spectrum between 0.2 and 4 μm. Upward and downward fluxes are obtained from the reflectances and transmittances of the layers, and the photon-path-distribution method allows to separate the parametrization of the scattering processes from that of the molecular absorption.

1.2 Spectral integration

Solar radiation is attenuated by absorbing gases, mainly water vapor, uniformly mixed gases (oxygen, carbon dioxide, methane, nitrous oxide) and ozone, and scattered by molecules (Rayleigh scattering), aerosols and cloud particles. Since scattering and molecular absorption occur simultaneously, the exact amount of absorber along the photon path length is unknown, and band models of the transmission function cannot be used directly as in the longwave radiation transfer. The approach of the photon path distribution method is to calculate the probability $p(U) \, dU$ that a photon contributing to the flux F_c in the conservative case (i.e. no absorption, $\overline{\omega}_\nu = 1, k_\nu = 0$) has encountered an absorber amount between U and $U + dU$. With this distribution, the radiative flux at wavenumber ν is related to F_c by

$$F_\nu = F_c \int_0^\infty p(U)\,\exp(-k_\nu U)\,dU \tag{7}$$

and the flux averaged over the spectral interval $\Delta\nu$ can then be calculated with the help of any band model of the transmission function $t_{\Delta\nu}$

$$F = \frac{1}{\Delta\nu}\int_{\Delta\nu} F_\nu\,d\nu = F_c \int_0^\infty p(U)\,t_{\Delta\nu}(U)\,d\nu. \tag{8}$$

To find the distribution function $p(U)$, the scattering problem is solved first, by any method, for a set of arbitrarily fixed absorption coefficients k_l, thus giving a set of simulated fluxes F_{k_l}. An inverse Laplace transform is then performed on (7) to get $p(U)$ (Fouquart 1974). The main advantage of the method is that the actual distribution $p(U)$ is smooth enough that (7) gives accurate results even if $p(U)$ itself is not known accurately. In fact, $p(U)$ needs not be calculated explicitly as the spectrally integrated fluxes are, in the two limiting cases of weak and strong absorption:

$$F = F_c\,t_{\Delta\nu}(< U >) \qquad \text{where} \qquad < U > = \int_0^\infty p(U)\,U\,dU \tag{9}$$

$$F = F_c\,t_{\Delta\nu}(< U^{\frac{1}{2}} >) \qquad \text{where} \qquad < U^{\frac{1}{2}} > = \int_0^\infty p(U)\,U^{\frac{1}{2}}\,dU \tag{10}$$

respectively. The atmospheric absorption in the water vapor bands is generally strong and the scheme determines an effective absorber amount U_e between $< U >$ and $< U^{\frac{1}{2}} >$ derived from

$$U_e = \frac{1}{k_e}\,\ln(\frac{F_{k_e}}{F_c}) \tag{11}$$

where k_e is an absorption coefficient chosen to approximate the spectrally averaged transmission of the clear-sky atmosphere:

$$k_e = (\frac{U_{tot}}{\mu_0})^{-1}\,\ln(t_{\Delta\nu}\frac{U_{tot}}{\mu_0}) \tag{12}$$

with U_{tot} the total amount of absorber in a vertical column and $\mu_0 = \cos\vartheta_0$. Once the effective absorber amounts of H_2O and uniformly mixed gases are found, the transmission functions are computed using Padé approximants:

$$t_{\Delta\nu}(U) = \frac{\sum\limits_{i=0}^{N} a_i U^{i-1}}{\sum\limits_{j=0}^{N} b_j U^{j-1}}. \tag{13}$$

Absorption by ozone is also taken into account, but since ozone is located at low pressure levels for which molecular scattering is small and Mie scattering is negligible, interactions between scattering processes and ozone absorption are neglected. Transmission through ozone is computed using (13) where the amount of ozone U_{O_3} is:

$$U_{O_3}^d = M \int_p^0 dU_{O_3}$$

for the downward transmission of the direct solar beam, and:

$$U_{O_3}^u = r \int_{p_s}^p dU_{O_3} + U_{O_3}^d(p_s)$$

for the upward transmission of the diffuse radiation with $r = 1.66$ the diffusivity factor and M, the magnification factor (Rodgers 1967) used instead of μ_0 to account for the sphericity of the atmosphere at very small solar elevations:

$$M = \frac{35}{\sqrt{\mu_0^2 + 1}}. \tag{14}$$

To perform the spectral integration, it is convenient to discretize the solar spectrum into subintervals in which the surface reflectance, molecular absorption characteristics, and cloud optical properties can be considered as constants. One of the main causes for such a spectral variation is the sharp increase in the reflectivity of the vegetation in the near-infrared. Also, water vapour does not absorb below 0.69 μm nor do liquid water clouds. Till June 2000, the ECMWF shortwave scheme considered only two spectral intervals, one for the visible (0.2/0.69 μm), one for the near-infrared (0.69/4.00 μm) parts of the solar spectrum. From June 2000 to April 2002, the near-infrared interval was subdivided into three intervals (0.69/1.19/2.38/4.00 μm) to account better for the spectral variations of the cloud optical properties. Till April 2002, all the molecular absorption coefficients (for O3, H2O, uniformly mixed gases) were derived from statistical models of the transmission function using spectroscopic parameters derived from various versions of the HITRAN database (Rothman et al.,1987,1992). In April 2002, following the recomputation of all the molecular absorption coefficients from an updated version of the shortwave lineby- line model of Dubuisson et al. (1996) using spectroscopic data from HAWKS (2000, `http://www.hitran.com`) the ultraviolet and visible part of the spectrum are now considered in three spectral intervals (0.20/0.25/0.69 μm) making the scheme having a total of six spectral intervals over which the aerosol and cloud optical properties are also defined. The cut-off at 0.69 μm allows the scheme to be more computational efficient, in as much as the interactions between gaseous absorption (by water vapour and uniformly mixed gases) and scattering processes are accounted for only in the near-infrared interval(s).

1.3 Vertical integration

Considering an atmosphere where a fraction C_{tot} (as seen from the surface or the top of the atmosphere) is covered by clouds (the fraction C_{tot} depends on which cloud overlap assumption is assumed for the calculations), the final fluxes are given as

$$F^{\downarrow} = C_{tot}\, F^{\downarrow}_{cloudy} \; + \; (1 - C_{tot})\, F^{\downarrow}_{clear} \tag{15}$$

with a similar expression holding for the upward flux. Contrarily to the scheme of Geleyn and Hollingsworth (1979), the fluxes are not obtained through the solution of a system of linear equations in a matrix form. Rather, assuming an atmosphere divided into N homogeneous layers, the upward and downward fluxes at a given interface j are given by:

$$\begin{aligned} F^{\downarrow}(j) &= F_0 \prod_{k=j}^{N} T_b(k), \qquad (16) \\ F^{\uparrow}(j) &= F^{\downarrow}(j)\, R_t(j-1), \qquad (17) \end{aligned}$$

where $R_t(j)$ and $T_b(j)$ are the reflectance at the top and the transmittance at the bottom of the j^{th} layer. Computations of R_t's start at the surface and work upward, whereas those of T_b's start at the top of the atmosphere and work downward. R_t and T_b account for the presence of cloud in the layer:

$$\begin{aligned} R_t &= C_j\, R_{cdy} + (1 - C_j)\, R_{clr}, \qquad (18) \\ T_b &= C_j\, T_{cdy} + (1 - C_j)\, T_{clr}. \qquad (19) \end{aligned}$$

The subscripts $_{clr}$ and $_{cdy}$ respectively refer to the clear-sky and cloudy fractions of the layer with C_j the cloud fraction of the layer j.

Cloudy fraction of the layers

$R_{t_{cdy}}$ and $T_{b_{cdy}}$ are the reflectance at the top and transmittance at the bottom of the cloudy fraction of the layer calculated with the Delta-Eddington approximation. Given δ_c, δ_a and δ_g, the optical thicknesses for the cloud and the aerosol, and the molecular absorption of the gases, g_c and g_a the cloud and aerosol asymmetry factors, $R_{t_{cdy}}$ and $T_{b_{cdy}}$ are calculated as functions of:

- the total optical thickness of the layer δ^*:

  $$\delta^* = \delta_c + \delta_a + \delta_g$$

- the total single scattering albedo:

  $$\omega^* = \frac{\delta_c + \delta_a}{\delta_c + \delta_a + \delta_g} \tag{20}$$

- the total asymmetry factor:

$$g^* = \frac{\delta_c}{\delta_c + \delta_a} g_c + \frac{\delta_a}{\delta_c + \delta_a} g_a \tag{21}$$

of the reflectance $R\_$ of the underlying medium (surface or layers below the j^{th} interface) and of the effective solar zenith angle $\mu_e(j)$ which accounts for the decrease of the direct solar beam and the corresponding increase of the diffuse part of the downward radiation by the upper scattering layers.

The effective solar zenith angle $\mu_e(j)$ is equal to:

$$\mu_e(j) = \left[\frac{(1 - C_{cld}^{eff}(j))}{\mu} + r \, C_{cld}^{eff}(j) \right]^{-1}, \tag{22}$$

with $C_{cld}^{eff}(j)$ the effective total cloudiness over level j and r the diffusivity factor.

$$C_{cld}^{eff}(j) = 1 - \prod_{i=j+1}^{N} (1 - C_{cld}(i)E(i))$$

and

$$E(i) = 1 - \exp \left[- \frac{(1 - \omega_c(i) g_c(i)^2) \delta_c(i)}{\mu} \right] \tag{23}$$

where $\delta_c(i)$, $\omega_c(i)$ and $g_c(i)$ are the optical thickness, single scattering albedo and asymmetry factor of the cloud in the ith layer.

The scheme follows the Eddington approximation, first proposed by Shettle and Weiman (1970), then modified by Joseph et al. (1976) to account more accurately for the large fraction of radiation directly transmitted in the forward scattering peak in case of highly asymmetric phase functions. Eddington's approximation assumes that, in a scattering medium of optical thickness δ^*, of single scattering albedo ω, and of asymmetry factor g, the radiance L entering (5) can be written as:

$$L(\delta, \mu) = L_0(\delta) + \mu \, L_1(\delta). \tag{24}$$

In that case, when the phase function is expanded as a series of associated Legendre functions, all terms of order greater than one vanish when (5) is integrated over μ and ϕ. The phase function is therefore given by

$$P(\theta) = 1 + \beta_1(\theta) \, \cos\theta,$$

where θ is the angle between incident and scattered radiances. The integral in (5) thus becomes

$$\int_0^{2\pi} \int_{-1}^{+1} P(\mu, \phi, \mu', \phi') \, L(\mu', \phi') \, d\mu' \, d\phi' = 4\pi \, (L_0 + \pi L_1) \tag{25}$$

where g, the asymmetry factor identifies as

$$g = \frac{\beta_1}{3} = \frac{1}{2}\int_{-1}^{+1} P(\theta)\, \cos\theta\, d(\cos\theta).$$

Using (25) in (5) after integrating over μ and dividing by 2π, we get

$$\mu\, \frac{d(L_0+\mu L_1)}{d\delta} = -(L_0+\mu L_1) + \omega\,(L_0+g\mu L_1) + \frac{1}{4}\omega\, F_0\, \exp(\frac{-\delta}{\mu_0})\,(1+3g\mu_0\,\mu).(26)$$

where ω is the single scattering albedo.

We obtain a pair of equations for L_0 and L_1 by integrating (26) over μ:

$$\frac{d(L_0)}{d\delta} = -3(1-\omega)\,L_0 + \frac{3}{4}\omega\,F_0\,\exp(\frac{-\delta}{\mu_0}), \tag{27}$$

$$\frac{d(L_1)}{d\delta} = -(1-\omega g)\,L_1 + \frac{3}{4}\omega\,g\,\mu_0\,F_0\,\exp(\frac{-\delta}{\mu_0}). \tag{28}$$

For the cloudy layer assumed non-conservative ($\omega < 1$), the solutions to (27) and (28) are, in the range $0 \le \delta \le \delta^*$:

$$\begin{aligned} L_0(\delta) &= C_1\,\exp(-k\delta) + C_2\,\exp(+k\delta) - \alpha\,\exp(\frac{-\delta}{\mu_0}), & (29)\\ L_1(\delta) &= p(C_1\,\exp(-k\delta) - C_2\,\exp(+k\delta)) - \beta\,\exp(\frac{-\delta}{\mu_0}), & (30)\end{aligned}$$

where

$$\begin{aligned} k &= \left[3\,(1-\omega)\,(1-\omega g)\right]^{\frac{1}{2}}\\ p &= \left[3\,(1-\omega)/(1-\omega g)\right]^{\frac{1}{2}}\\ \alpha &= 3\,\omega\,F_0\,\mu_0\,\frac{\left[1+3g\,(1-\omega)\right]}{4\,(1-k^2\mu_0^2)}\\ \beta &= 3\,\omega\,F_0\,\mu_0\,\frac{\left[1+3\,g\,(1-\omega)\,\mu_0^2\right]}{4\,(1-k^2\mu_0^2)}. \end{aligned}$$

The two boundary conditions allow to solve the system for C_1 and C_2. First, the downward directed diffuse flux at the top of the layer is zero

$$F^{\downarrow}(0) = \left[L_0(0) + \frac{2}{3}L_1(0)\right] = 0,$$

which translates into

$$(1 + \frac{2p}{3})\, C_1 + (1 - \frac{2p}{3})\, C_2 = \alpha + \frac{2\beta}{3}. \tag{31}$$

For the second condition, one assumes that the upward directed flux at the bottom of the layer is equal to the product of the downward directed diffuse and direct fluxes by the corresponding diffuse and direct reflectances (R_d and $R\_$, respectively) of the underlying medium

$$F^{\uparrow}(\delta^*) = \left[L_0(\delta^*) - \frac{2}{3}\, L_1(\delta^*)\right] = R\_ \left[L_0(\delta^*) + \frac{2}{3}\, L_1(\delta^*)\right] + R_d\, \mu_0\, F_0\, \exp(\frac{-\delta^*}{\mu_0}),$$

which translates into

$$(1 - R\_ - \frac{2p}{3}\,(1 + R\_))\, C_1\, \exp(-k\,\delta^*) + (1 - R\_ + \frac{2p}{3}\,(1 + R\_))\, C_2 \quad \exp(+k\,\delta^*) \tag{32}$$
$$= ((1 - R\_)\alpha - \frac{2}{3}\,(1 + R\_)\beta + R_d\, \mu_0\, F_0) \quad \exp(\tfrac{-\delta^*}{\mu_0}) \tag{33}$$

In the Delta-Eddington approximation, the phase function is approximated by a Dirac delta function (forward scatter peak) and a two-term expansion of the phase function

$$P(\theta) = 2f\,(1 - \cos\theta) + (1 - f)\,(1 + 3g'\,\cos\theta),$$

where f is the fractional scattering into the forward peak and g' the asymmetry factor of the truncated phase function. As shown by Joseph et al. (1976), these parameters are:

$$f = g^2 \tag{34}$$
$$g' = \frac{g}{1 + g}. \tag{35}$$

The solution of the Eddington's equations remains the same provided that the total optical thickness, single scattering albedo and asymmetry factor entering (26)-(32) take their transformed values:

$$\delta^{'} = (1 + \omega f)\,\delta^*, \tag{36}$$
$$\omega^{'} = \frac{(1 - f)\,\omega}{1 - \omega f}. \tag{37}$$

Practically, the optical thickness, single scattering albedo, asymmetry factor, and solar zenith angle entering (26)-(31) are δ^*, ω^*, g^* and u_e defined in (21) and (22).

Clear-sky fraction of the layers

In the clear-sky fraction of the layers, the shortwave scheme accounts for scattering and absorption by molecules and aerosols. The following calculations are practically done twice, once for the clear-sky fraction $(1-C_{cld}^{tot})$ of the atmospheric column μ with equal to μ_0, simply modified for the effect of Rayleigh and aerosol scattering, the second time for the clear-sky fraction of each individual layer within the fraction C_{cld}^{tot} of the atmospheric column containing clouds, with μ equal to μ_e.

As the optical thickness for both Rayleigh and aerosol scattering is small, $R_{clr}(j-1)$ and $T_{clr}(j)$ the reflectance at the top and transmittance at the bottom of the j^{th} layer can be calculated using respectively a first and a second-order expansion of the analytical solutions of the two-stream equations similar to that of Coakley and Chylek (1975). For Rayleigh scattering, the optical thickness, single scattering albedo and asymmetry factor are respectively δ_R, $\omega_R = 1$ and $g_R = 0$, so that

$$R_R = \frac{\delta_R}{2\,\mu + \delta_R}, \tag{38}$$

$$T_R = \frac{2\mu}{2\,\mu + \delta_R}. \tag{39}$$

The optical thickness δ_R of an atmospheric layer is simply:

$$\delta_R = \delta_R^* \frac{(p(j) - p(j-1))}{p_{surf}},$$

where δ_R^* is the Rayleigh optical thickness of the whole atmosphere parameterized as a function of solar zenith angle (Deschamps et al. 1983):

$$\delta_R^* = \sum_{i=0}^{5} a_i\, \mu_0^{i-1}.$$

For aerosol scattering and absorption, the optical thickness, single scattering albedo and asymmetry factor are respectively δ_a, ω_a (with $1-\omega_a \ll 1$) and g_a so that:

$$den = 1 + (1-\omega_a + back(\mu_e)\,\omega_a)\,\frac{\delta_a}{\mu_e} + (1-\omega_a)\,(1-\omega_a + 2\,back(\mu_e)\,\omega_a)\,\frac{{\delta_a}^2}{{\mu_e}^2}$$

$$R(\mu_e) = \frac{back(\mu_e)\,\omega_a\,\delta_a/\mu_e}{den} \tag{40}$$

$$T(\mu_e) = \frac{1}{den}$$

where $back(\mu_e) = (2 - 3\mu_e g_a)/4$ is the backscattering factor.

Practically, R_{clr} and T_{clr} are computed using (40) and the combined effect of aerosol and Rayleigh scattering comes from using modified parameters corresponding to the addition of the two scatters with provision for the highly asymmetric aerosol phase function through a Delta-Eddington approximation of the forward scattering peak (as in (34)-(36)):

$$\delta^+ = \delta_R + \delta_a(1 - \omega_a g_a^2)$$

$$g^+ = \frac{g_a}{1+g_a}\frac{\delta_a}{(\delta_R+\delta_a)}$$

$$\omega^+ = \frac{\delta_R}{\delta_R+\delta_a}\omega_R + \frac{\delta_a}{\delta_R+\delta_a}\frac{\omega_a(1-g_a^2)}{1-\omega_a g_a^2}$$

As for their cloudy counterparts, R_{clr} and T_{clr} must account for the multiple reflections due to the layers underneath:

$$\begin{aligned} R_{clr} &= R(\mu_e) + \frac{T(\mu_e)}{1 - R^* \, R\_} \, R\_, & (41) \\ T_{clr} &= \frac{T(\mu_e)}{(1 - R^* \, R\_)}, & (42) \end{aligned}$$

with $R^* = R(1/r)$, $T^* = T(1/r)$, $R\_ = R_t(j-1)$ is the reflectance of the underlying medium and r is the diffusivity factor.

Since interactions between molecular absorption and Rayleigh and aerosol scattering are negligible, the radiative fluxes in a clear-sky atmosphere are simply those calculated from (16) and (41) attenuated by the gaseous transmissions (13).

1.4 Multiple reflections between layers

To deal properly with the multiple reflections between the surface and the cloud layers, it should be necessary to separate the contribution of each individual reflecting surface to the layer reflectances and transmittances in as much as each such surface gives rise to a particular distribution of absorber amount. In case of an atmosphere including N cloud layers, the reflected light above the highest cloud consists of photons directly reflected by the highest cloud without interaction with the underlying atmosphere and of photons that have passed through this cloud layer and undergone at least one reflection on the underlying atmosphere. In fact, (8) should be written

$$F = \sum_{l=0}^{N} F_{cl} \int_0^\infty p_l(U)\, t_{\Delta\nu}(U)\, d\nu, \qquad (43)$$

where F_{cl} and $p_l(U)$ are the conservative fluxes and the distributions of absorber amount corresponding to the different reflecting surfaces.

Fouquart and Bonnel (1980) have shown that a very good approximation to this problem is obtained by evaluating the reflectance and transmittance of each layer (using (32) and (41)), assuming successively a non-reflecting underlying medium ($R\_ = 0$), then a reflecting underlying medium ($R\_ \neq 0$). First calculations provide the contribution to reflectance and transmittance of those photons interacting only with the layer into consideration, whereas the second ones give the contribution of the photons with interactions also outside the layer itself.

From these two sets of layer reflectances and transmittances (R_{t_0}, T_{t_0}) and ($R_{t_{\neq}}, T_{t_{\neq}}$) respectively, effective absorber amounts to be applied to computing the transmission functions for upward and downward fluxes are then derived using (11) and starting from the surface and working the formulas upward:

$$U^{\downarrow}_{e_0} = \frac{1}{k_e} \ln(\frac{T_{b_0}}{T_{b_c}}),$$

$$U^{\downarrow}_{e_{\neq}} = \frac{1}{k_e} \ln(\frac{T_{b_{\neq}}}{T_{b_c}}),$$

$$U^{\uparrow}_{e_0} = \frac{1}{k_e} \ln(\frac{R_{t_0}}{R_{t_c}}),$$

$$U^{\uparrow}_{e_{\neq}} = \frac{1}{k_e} \ln(\frac{R_{t_{\neq}}}{R_{t_c}}),$$

where R_{t_c} and T_{b_c} are the layer reflectance and transmittance corresponding to a conservative scattering medium. Finally the upward and downward fluxes are obtained as:

$$F^{\uparrow}(j) = F_0 \left[R_{t_0}\, t_{\Delta\nu}(U^{\uparrow}_{e_0}) + (R_{t_{\neq}} - R_{t_0})\, t_{\Delta\nu} U^{\uparrow}_{e_{\neq}}) \right] \tag{44}$$

$$F^{\downarrow}(j) = F_0 \left[T_{b_0}\, t_{\Delta\nu}(U^{\downarrow}_{e_0}) + (T_{b_{\neq}} - T_{b_0})\, t_{\Delta\nu} U^{\downarrow}_{e_{\neq}}) \right] \tag{45}$$

2 Longwave radiation: the RRTM scheme

The main characteristics of RRTM are:

- Solution of radiative transfer equation: Two-stream method
- Number of spectral intervals: 16
- Absorbers : H_2O, CO_2, $O3$, CH_4, N_2O, $CFC11$, $CFC12$, aerosols
- Spectroscopic data base: HITRAN 1996
- Absorption coefficient: From LBLRTM line-by-line model
- Cloud handling: True cloud fraction

- Cloud optical properties: 16-band spectral emissivity
- Cloud overlap assumption: Maximum random
- References : Mlawer et al. (1997)

As stated in Mlawer et al. (1997), the objective in the development of RRTM has been to obtain an accuracy in the calculation of fluxes and heating rates consistent with the best line-by-line models. It utilizes the correlated-k methode and shows its filiation to the Atmospheric and Environmental Research, Inc. (AER) line-by-line model (LBLRTM; Clough et al. 1989, 1992; Clough and Iacono 1995) through its use of absorption coefficients for the relevant k-distributions derived from LBLRTM. Therefore the k-coefficients in RRTM include the effect of the CKD2.2 water vapour continuum (Clough et al. 1989).

The main point in the correlated-k method (Lacis and Oinas 1991; Fu and Liou 1992) is the mapping of the absorption coefficient $k(\nu)$ from the spectral space (where it varies irregularly with wavenumber ν) to the g-space (where $g(k)$ is the probability distribution function, i.e. the fraction of the absorption coefficients in the set smaller than k). The effect of this reordering is a rearrangement of the sequence of terms in the integral over wavenumber in the radiative transfer equation (RTE), which makes it equivalent to what would be done for monochromatic radiation.

In the ECMWF (hence, ARPEGE-CLIMAT) model, no provision is presently taken for scattering in the longwave. Therefore, in order to get the downward radiance, the integration over the vertical dimension is simply done starting from the top of the atmosphere, going downward layer by layer. At the surface, the boundary condition (in terms of spectral emissivity, and potential reflection of downward radiance) is computed, then, in order to get the upward radiance, the integration over the vertical dimension is repeated, this time from the surface upward.

The spectrally averaged radiance (between ν_1 and ν_2) emerging from an atmospheric layer is

$$\overline{R} = \frac{1}{(\nu_1 - \nu_2)} \int_{\nu_2}^{\nu_1} d\nu \left\{ R_0(\nu) + \int_{t_\nu}^{1} \left[B(\nu, T(t'_\nu)) - R_0(\nu) \right] dt' \right\} \tag{46}$$

where R_0 is the incoming radiance to the layer, $B(\nu, T)$ is the Planck function at wavenumber ν and temperature T, t_ν is the transmittance for the layer optical path, and t'_ν is the transmittance at a point along the optical path in the layer. Under the mapping $\nu \rightarrow g$, this becomes

$$\overline{R} = \int_0^1 dg \left\{ B_{\text{eff}}(g, T_g) + [R_0(g) - B_{\text{eff}}(g, T_g)] \exp\left[-k(g, P, T) \frac{\rho \delta z}{\cos \phi} \right] \right\} \tag{47}$$

where $B_{\text{eff}}(g, T)$ is an effective Planck function for the layer that varies with the layer's transmittance such as to ensure continuity of flux across layer boundaries for opaque conditions. The dependence of the transmittance is now written in terms of the absorption coefficient $k(g, P, T)$ at layer pressure P and temperature T, the absorber density ρ, the vertical thickness of the layer δz, and the angle ϕ of the optical path.

For a given spectral interval, the domain of the variable g is partitioned into subintervals (see Table 7.1, number of g-points), each corresponding to a limited range of $k(g)$ values and for which a characteristic value κ_j of the absorption coefficient is chosen. These κ_j are then used to compute the outgoing radiance

$$\overline{R} = \Sigma_j W_j \left[B_{\text{eff}_j} + (R_0(g) - B_{\text{eff}_j}) \exp\left(-\kappa_j \frac{\rho \delta z}{\cos \phi} \right) \right] \tag{48}$$

where W_j is the size of the sub-intervals ($\Sigma W_j = 1$).

The accuracy of these absorption coefficients has been established by numerous and continuing high-resolution validations of LBLRTM with spectroscopic measurements, in particular those from the Atmospheric Radiation Measurement program (ARM). Compared to the original RRTM (Mlawer et al. 1997), the version used at ECMWF (hence ARPEGE-CLIMAT) has been slightly modified to account for cloud optical properties and surface emissivity defined for each of the 16 bands over which spectral fluxes are computed. For efficiency reason, the original number of g-points ($256 = 16 \times 16$) has been reduced to 140 (see Table 7.1). Other changes are the use of a diffusivity approximation (instead of the three-angle integration over the zenith angle used in the original scheme) to derive upward and downward fluxes from the radiances, and the modification of the original cloud random overlapping assumption to include (to the same degree of approximation as used in the operational SW scheme) a maximum-random overlapping of cloud layers. Given the monochromatic form of the RTE, the vertical integration is simply carried out one layer at a time from the top-of-the-atmosphere to the surface to get the downward fluxes. The downward fluxes at the surface are then used with the spectral surface emissivities and the surface temperature to get the upward longwave fluxes in each of the 140 subintervals. Then the upward fluxes are obtained in a similar fashion from the surface to the top of the atmosphere.

For the relevant spectral intervals of the RRTM schemes, ice cloud optical properties are derived from Ebert-Curry (1992), and water cloud optical properties from Smith and Shi (1992) . Whereas in the previous operational scheme the cloud emissivity used to compute the effective cloud cover is defined over the whole LW spectrum from spectrally averaged mass absorption coefficients and the relevant cloud water and/or ice paths, in RRTM, the cloud optical thickness is defined as a function of spectrally varying mass absorption coefficients and relevant cloud water and ice paths, and is used within the true cloudy fraction of the layer. Alternate sets of cloud optical properties are also available for RRTM (Section 3).

Table 7.1: Spectral distribution of the absorption by atmospheric gases in RRTM

Spectral intervals	Number	Gases included	
cm^{-1}	g-points	Troposphere	Stratosphere
10-250	8	H2O	H2O
250-500	14	H2O	H2O
500-630	16	H2O, CO2	H2O, CO2
630-700	14	H2O, CO2	O3, CO2
700-820	16	H2O, CO2, CCl4	O3, CO2, CCl4
820-980	8	H2O, CFC11, CFC12	CFC11, CFC12
980-1080	12	H2O, O3	O3
1080-1180	8	H2O, CFC12, CFC22	O3, CFC12, CFC22
1180-1390	12	H2O, CH4	CH4
1390-1480	6	H2O	H2O
1480-1800	8	H2O	H2O
1800-2080	8	H2O	
2080-2250	4	H2O, N2O	
2250-2380	2	CO2	CO2
2380-2600	2	N2O, CO2,	
2600-3000	2	H2O , CH4	

3 Input to the radiation scheme

3.1 Solar irradiance data

Depending on the experiment, the solar constant can be specified on a yearly basis in the model. The recommendations for CMIP6 solar forcing data are available there: `http://solarisheppa.geomar.de/cmip6`

3.2 Clouds

Cloud fraction, and liquid/ice water content are provided in all layers by the cloud scheme.

3.3 Ground albedo and emissivity

Ground albedo and emissivities are calculated in the SURFEX module (see chapter “Surfex processes schemes”). The main features are the following ones:

1. one emissivity by type of vegetation (coming from the Ecoclimap 1 km resolution data base) and for ocean and sea-ice.
2. Albedo:

- over continental areas. It is deduced from the Ecoclimap data base, one by type of vegetation, the total mean albedo on a mesh is weighted with the vegetation fraction. This version is updated by the annual mean MODIS cycle (Carrer et al.,RSE, 2014), introducing annual cycles of vegetation, and also soil albedo.
- over ocean: a new parametrization of the ocean surface albedo has been developed in collaboration with LMD. It includes dependencies on wavelength, surface wind speed, chlorophyll content as well as the distribution of solar zenith angle. The albedo is decomposed into a direct and a diffuse contribution, each of them being divided according to the reflection process, i.e a Fresnel surface albedo and an ocean volume albedo. The computation is done in the routine albedors14.F90.
- over snow: albedo varies with the age of snow following Douville et al. (1995)

3.4 Aerosols

Aerosols are included in ARPEGE-Climat using either monthly 2D aerosol optical depth (AOD) files or an interactive aerosol scheme providing 3D concentrations. In both cases, five tropospheric aerosol types are taken into account: desert dust, sea-salt, sulfate, black carbon and organic matter.

In the first case, these AOD data come from ARPEGE-Climat simulations using the interactive aerosol scheme (TACTIC), a running average on 11 years is calculated to smooth interannual variability. The 2D optical thicknesses are vertically distributed in the RADAER subroutine according to the aerosol type.

In the second case, the aerosol vertical profile of extinction is directly computed in the aerosol scheme. In both cases, these 5 tropospheric aerosol types are redistributed in the 4 following types, to be taken into consideration in the radiation scheme :

- Continental : Organic + Sulfate
- Maritime (sea-salt aerosols)
- Desert (soil dust type aerosols)
- Urban (black-carbon type aerosols)

Two additional types are also included in ARPEGE-Climat to represent volcanic and stratospheric aerosols:

- Volcanic class (background) and Stratospheric (background) + volcanic eruptions

Direct forcing of aerosols

Through their optical properties, aerosols can scatter and/or absorb solar and thermal radiation. The following optical properties, namely normalized optical depth (RTAUA), single scattering albedo (RPIZA) and asymmetry factor (RCGA), all defined in SUAERSN and used in SWCLR, are used in the SW radiation scheme (6 spectral bands), depending on the value of NCOEFAERO (noted NCA in the tables below). The values of RPIZA and RCGA have been updated according to Mie calculations (see Nabat et al., (2013) for more details). Note that the spectral dependance of all these optical properties is for the moment not taken into consideration inside the visible spectrum (i.e. the first three bands have the same coefficients), but it should be the case in a future version of the model.

- NCOEFAERO=0 : updated values for continental, sea-salt, desert dust and black carbon
- NCOEFAERO=1 : updated values for desert dust only, old coefficients (with no more than three significant numbers)
- NCOEFAERO=2 : old coefficients

Normalized optical depth (RTAUA) :

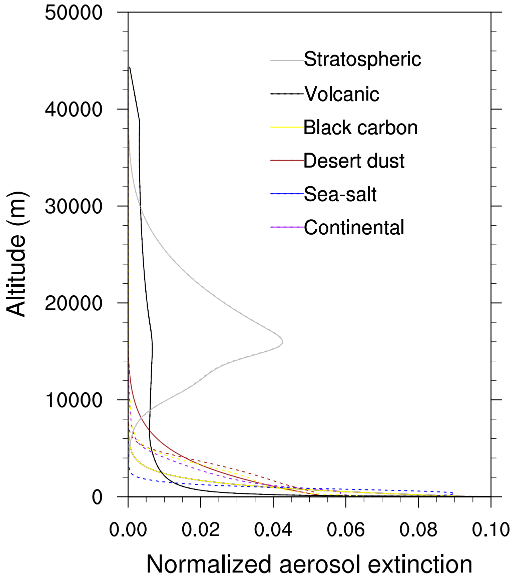


Figure 1: Average global profile for each aerosol type in simulations using either 2D AOD data (full lines) or the interactive aerosol scheme (dotted lines). Note that the full lines for black carbon, sea-salt and continental aerosols are superimposed. Altitudes have been calculated for standard atmospheric conditions.

Aerosols	NCA	0.185	0.25	0.44	0.69	1.19	2.38
		-	-	-	-	-	-
		0.25	0.44	0.69	1.19	2.38	4.0
		μm	μm	μm	μm	μm	μm
Continental	0/1/2	1.69446	1.69446	1.69446	0.52838	0.20543	0.10849
Sea-salt	0/1/2	1.11855	1.11855	1.11855	0.93285	0.84642	0.66699
Desert dust	0/1/2	1.09212	1.09212	1.09212	0.93449	0.84958	0.65255
Black carbon	0/1/2	1.72145	1.72145	1.72145	0.53078	0.21673	0.11600
Volcanic	0/1/2	1.03858	1.03858	1.03858	0.67148	0.28270	0.06529
Stratospheric	0/1/2	1.12044	1.12044	1.12044	0.46608	0.10915	0.04468

Single scattering albedo (RPIZA) :

Aerosols	NCA	0.185	0.25	0.44	0.69	1.19	2.38
		-	-	-	-	-	-
		0.25	0.44	0.69	1.19	2.38	4.0
		μm	μm	μm	μm	μm	μm
Continental	0	0.99	0.99	0.99	0.99	0.99	0.99
	1	0.91	0.91	0.91	0.897	0.828	0.52
	2	0.9148907	0.9148907	0.9148907	0.8970131	0.8287144	0.5230504
Sea-salt	0	0.996	0.996	0.996	0.995	0.994	0.99
	1	0.996	0.996	0.996	0.998	0.994	0.79
	2	0.9956173	0.9956173	0.9956173	0.9984940	0.9949396	0.7868518
Desert dust	0/1	0.90	0.90	0.90	0.93	0.94	0.95
	2	0.7504584	0.7504584	0.7504584	0.9245594	0.9279543	0.8531531
Black carbon	0	0.32	0.32	0.32	0.27	0.24	0.20
	1	0.81	0.81	0.81	0.777	0.677	0.405
	2	0.8131335	0.8131335	0.8131335	0.7768385	0.6765051	0.4048149
Volcanic	0/1	0.94	0.94	0.94	0.953	0.946	0.875
	2	0.9401905	0.9401905	0.9401905	0.9532763	0.9467578	0.8748231
Stratospheric	0/1	0.999	0.999	0.999	0.999	0.995	0.235
	2	0.9999999	0.9999999	0.9999999	0.9999999	0.9955938	0.2355667

Asymmetry factor (RCGA) :

Aerosols	NCA	0.185	0.25	0.44	0.69	1.19	2.38
		-	-	-	-	-	-
		0.25	0.44	0.69	1.19	2.38	4.0
		μm	μm	μm	μm	μm	μm
Continental	0	0.69	0.69	0.69	0.64	0.62	0.60
	1	0.73	0.73	0.73	0.67	0.64	0.70
	2	0.729019	0.729019	0.729019	0.668431	0.636342	0.700610
Sea-salt	0	0.72	0.72	0.72	0.73	0.73	0.73
	1	0.80	0.80	0.80	0.79	0.80	0.82
	2	0.803129	0.803129	0.803129	0.788530	0.802467	0.818871
Desert dust	0/1	0.785	0.785	0.785	0.75	0.75	0.75
	2	0.784592	0.784592	0.784592	0.698682	0.691305	0.702399
Black carbon	0	0.43	0.43	0.43	0.31	0.30	0.30
	1	0.71	0.71	0.71	0.66	0.63	0.69
	2	0.712208	0.712208	0.712208	0.657422	0.627497	0.689886
Volcanic	0/1	0.70	0.70	0.70	0.673	0.611	0.463
	2	0.7008249	0.7008249	0.7008249	0.6735182	0.6105750	0.4629866
Stratospheric	0/1	0.727	0.727	0.727	0.652	0.476	0.191
	2	0.7270548	0.7270548	0.7270548	0.6519706	0.4760794	0.1907639

The dependance of these optical properties on relative humidity can be activated using the key LHUMDEP : it concerns only sea-salt (for RCGA) and continental (for RPIZA and RCGA) aerosols. In the case LHUMDEP=False, values in the tables above are used (corresponding to an average relative humidity of 50%). In the case LHUMDEP=True, RCGA (for sea-salt and continental aerosols) and RPIZA (for continental aerosols) are replaced by the following values:

- Single scattering albedo (RPIZA) for continental aerosols:

RH	0.185 - 0.25 μm	0.25 - 0.44 μm	0.44 - 0.69 μm	0.69 - 1.19 μm	1.19 - 2.38 μm	2.38 - 4.0 μm
0-15%	0.988	0.988	0.988	0.988	0.988	0.988
15-25%	0.989	0.989	0.989	0.989	0.989	0.989
25-35%	0.989	0.989	0.989	0.989	0.989	0.989
35-45%	0.990	0.990	0.990	0.990	0.990	0.990
45-55%	0.991	0.991	0.991	0.991	0.991	0.991
55-65%	0.993	0.993	0.993	0.993	0.993	0.993
65-75%	0.994	0.994	0.994	0.994	0.994	0.994
75-85%	0.995	0.995	0.995	0.995	0.995	0.995
85-92.5%	0.997	0.997	0.997	0.997	0.997	0.997
92.5-97%	0.998	0.998	0.998	0.998	0.998	0.998
97-100%	0.999	0.999	0.999	0.999	0.999	0.999

- Asymmetry factor (RCGA) for sea-salt aerosols:

RH	0.185 - 0.25 μm	0.25 - 0.44 μm	0.44 - 0.69 μm	0.69 - 1.19 μm	1.19 - 2.38 μm	2.38 - 4.0 μm
0-15%	0.69	0.69	0.69	0.70	0.70	0.70
15-25%	0.70	0.70	0.70	0.71	0.71	0.71
25-35%	0.71	0.71	0.71	0.72	0.72	0.72
35-45%	0.72	0.72	0.72	0.73	0.73	0.73
45-55%	0.73	0.73	0.73	0.74	0.74	0.74
55-65%	0.74	0.74	0.74	0.75	0.75	0.75
65-75%	0.75	0.75	0.75	0.76	0.76	0.76
75-85%	0.76	0.76	0.76	0.77	0.77	0.77
85-92.5%	0.78	0.78	0.78	0.78	0.78	0.78
92.5-97%	0.79	0.79	0.79	0.79	0.79	0.79
97-100%	0.82	0.82	0.82	0.79	0.79	0.79

- Asymmetry factor (RCGA) for continental aerosols:

RH	0.185 - 0.25 μm	0.25 - 0.44 μm	0.44 - 0.69 μm	0.69 - 1.19 μm	1.19 - 2.38 μm	2.38 - 4.0 μm
0-15%	0.66	0.66	0.66	0.38	0.38	0.38
15-25%	0.67	0.67	0.67	0.39	0.39	0.39
25-35%	0.68	0.68	0.68	0.40	0.40	0.40
35-45%	0.68	0.68	0.68	0.41	0.41	0.41
45-55%	0.69	0.69	0.69	0.43	0.43	0.43
55-65%	0.70	0.70	0.70	0.45	0.45	0.45
65-75%	0.71	0.71	0.71	0.47	0.47	0.47
75-85%	0.72	0.72	0.72	0.51	0.51	0.51
85-92.5%	0.74	0.74	0.74	0.58	0.58	0.58
92.5-97%	0.76	0.76	0.76	0.64	0.64	0.64
97-100%	0.80	0.80	0.80	0.74	0.74	0.74

The following optical properties (absorption coefficients) are used in the LW radiation scheme (16 spectral bands, the aerosol coefficients, defined in SUAERL, are gathered in 5 classes and used in RRTM_ECRT_140GP)

Aerosols	NCA	Spectral bands					
		1/2	3/4/5	6/8/9	7	10-16	
Continental	0/1/2	0.036271	0.030153	0.017343	0.015002	0.008806	0.006865
Sea-salt	0/1/2	0.026561	0.032657	0.017977	0.014210	0.016775	0.022123
Desert dust	0/1	0.1	0.1	0.1	0.1	0.1	0.1
	2	0.014897	0.016359	0.019789	0.030777	0.013341	0.014321
Black carbon	0/1/2	0.001863	0.002816	0.002355	0.002557	0.001774	0.001780
Volcanic	0/1/2	0.011890	0.016142	0.021105	0.028908	0.011890	0.011890
Stratospheric	0/1/2	0.013792	0.026810	0.052203	0.066338	0.013792	0.013792

Indirect forcing of the sulfate aerosols

The aerosols can act as cloud condensation nuclei. At constant cloud liquid water, increasing aerosol concentration leads to a larger concentration of cloud droplets of small radius and increases cloud reflectivity. This effect, referred as the first aerosol indirect effect (Twomey, 1974), can be taken into account depending on the value of NRADLP used in RADLSW to calculate the effective radius of cloud droplets:

- NRADLP=2 : no aerosol indirect effect
- NRADLP=3 : A simple parametrization from Quaas and Boucher (2005) simulates this effect in the case of sulfate aerosols. The cloud droplet concentration CDN (in cm^{-3}) is given by:
  lnCDN = 1.7 +0.2 ln m$_{SO_4^{2-}}$
  with $m_{SO_4^{2-}}$expressed in μgm^{-3} (calculated from the sulfate aerosol optical depth). The mean cloud droplet radius is then calculated from the cloud liquid water content q_l and liquid water density ρ_l by:
  $r_v = \sqrt[3]{\frac{3q_l}{4\pi\rho_l CDN}}$
  and the effective radius r_e =1.1 r$_v$.
  The sulfate aerosols are available as output of the routine RADAER in the 3D array ZAERINDS.
- NRADLP=4 : the same as 3, but relative humidity is taken into account in the calculation of $m_{SO_4^{2-}}$.
- NRADLP=5 : organic matter and sea-salt aerosols are also taken into account in the calculation of the cloud droplet concentration according to Menon et al. (2002).
  lnCDN = 2.41 + 0.5 ln m$_{SO_4^{2-}}$ + 0.05 ln m$_{SS}$ + 0.13 ln m$_{OM}$

  Only the hydrophilic part of organic matter and the fine mode of sea-salt aerosols are considered for this calculation (climatological fractions, respectively 77.64% and 1.37%).

Note that the second aerosol indirect effect (modification of cloud life time and precipitation by aerosols) is not included in ARPEGE-Climat.

3.5 Radiatively active compounds

Concentrations of carbon dioxide, methane, nitrous oxide, CFC-11, CFC-12 and chlorine are given in namelist NAERAD. During the simulation, values are evolving yearly following the IPCC data. The CFC-12 compound also includes the CFC-12 equivalent of halocarbon species. Outputs from the CNRM REPROBUS chemical model, simulating coefficients which describe 3D ozone distributions, are used as input of a linear interactive scheme (Cariolle and Teyssèdre, 2007). The value NVCLIS = 7 (in NAMDPHY) means that seven of the climatological coefficients of the REPROBUS outputs are used. Evolution of ozone chemistry is due to chlorine and meteorological parameters (temperature, humidity) evolutions.

3.6 Cloud optical properties

For the SW radiation, the cloud radiative properties depend on three different parameters: the optical thickness, the asymmetry factor and the single scattering albedo of particles. For LW the cloud properties are linked with emissivity and spectral optical thickness. All these properties are defined for water and ice particles in the RADLSW routine. They depend on liquid (or ice) water path or on the effective radius of the particles. These characteristic parameters can be calculated by different ways according to NRADLP, NRADIP, NLIQOPT, NICEOPT parameters. During the tuning of the CMIP6 model, corrections of some options have been done. New combinations of optical properties have been coded: see comments on NICEOPT/NLIQOPT and NRADLP/NRADIP below.

Cloud liquid particles

- NRADLP = 0; cloud water effective radius is calculated as a function of pressure (old parameterisation).
- NRADLP = 1; the cloud effective radius is equal to 10 μm over land and equal to 13 μm over the ocean.
- NRADLP = 2; parameterisation based on Martin et al. (1994).
- From NRADLP=3, a parametrization of the first indirect effect of aerosols is proposed: **Effective radius of cloudy droplets**
  The effective radius of cloudy droplets writes:
  $$r_e = 100 * RTUNHRM * (\frac{3 * ZLWC}{4 * \pi ZCCDNC})^{1/3} \tag{49}$$
  where r_e is the effective radius in μm, RTUNHRM is a parameter (RTUNHRM=1.1), $ZLWC$= $1000 * \rho_{air} * PQLWP$ (unit: g/m^3) with $PQLWP$ input of radlsw.F90 (unit: kg/kg) and $ZCCDNC$ the number of CCN (cm^{-3}). If NRADLP=5, the min value of r_e is 2 μm and the max one is 24 μm.
- NRADLP = 3: the effective radius of cloud particle is calculated taking into account the indirect effects of sulfate aerosols only. The number of cloud condensation nuclei is computed by Quaas, pers. comm.
- NRADLP = 4: as NRADLP =3 but with tabulated extinction coefficients vs the aerosol types and relative humidity. CNN number of Quaas.
- NRADLP = 5: effect of sea salt and organic aerosols. Tabulated extinction coefficient as for NRADLP=4, Menon's (2002) parameter for the computation of CCN number.

Cloud ice particles

- NRADIP = 0; the ice particle effective radius is fixed at 40 μm.
- NRADIP = 1; ice particle effective radius calculated as f(T) from Ou and Liou (1995).
- NRADIP = 2; ice particle effective radius calculated as f(T) from Ou and Liou (1995) and fixed between 30 and 60 microns .
- NRADIP = 3; Ice effective radius calculated as a function of temperature and ice water content from Sun and Rikus (1999) and revised by Fu and Sun (2001).
- NRADIP = 4 is equivalent to NRADIP = 2 for LW computations and NRADIP = 3 for SW computations.

SW radiation

Cloud water optical properties depend on NLIQOPT:

- NLIQOPT different from zero, refers to Slingo (1989).

- NLIQOPT = 0, refers to Parol et al. (1991).

Ice water optical properties depend on NICEOPT:

- NICEOPT lower than or equal to 1 , refers to Ebert and Cury (1992).
- NICEOPT = 2, refers to Fu and Liou (1993).
- NICEOPT = 3 or NICEOPT =4, refers to Fu (1996).

LW radiation

Cloud water emissivity depends on NLIQOPT:

- NLIQOPT = 0 or greater than or equal to 3, refers to Smith and Shi (1992).
- NLIQOPT = 1 , refers to Savijarvi and Raisanen (1997).
- NLIQOPT = 2 , refers to Lindner and Li (2000).
- NLIQOPT > ou = 3 , refers to Smith and Shi(1992). NLIQOPT = 4 is equivalent to NLIQOPT= 0 in LW computations.

Ice cloud emissivity depends on NICEOPT:

- NICEOPT = 0, refers to Smith and Shi (1992).
- NICEOPT = 1, refers to Ebert-Curry (1992).
- NICEOPT = 2, refers to Fu and Liou (1993).
- NICEOPT = 3, refers to Fu et al. (1998) including parametrisation for LW scattering effect .
- NICEOPT = 4, equivalent to NICEOPT = 1 in LW computations.

Inhomogeneity factors

The cloudy inhomogeneity factors are RSWINHF for SW and RLWINHF for LW. During the tuning of the model, a difference between the liquid and ice clouds (RSWINHF$_{LIQ}$ and RSWINHF$_{ICE}$ for the SW and RLWINHF$_{LIQ}$/RLWINHF$_{ICE}$ for the LW) has been introduced for computation of optical depth and spectral emissivity. According to J.J Morcrette (pers. comm.) the SW coefficients can be chosen in the range (0.6-0.8) and the LW one in the range (0.8-1), depending also on the water condensate vertical, horizontal distribution and effective radius. These parameters have been used to obtain closed TOA and surface budgets.

The current values of the parameters are:

- NRADLP = 5,
- NRADIP = 4,
- NLIQOPT = 4
- NICEOPT = 4, Note that the couple (NLIQOPT=4, NICEOPT=4) is equivalent to the couple (2,3) in SW computations and (0,1) in the LW computations.
- RTUNHRM=1.1,
- NCOEFAERO =0,
- LHUMDEP=.T.,
- RSWINHF=1.,
- RLWHINF=1.,
- RSWINHF$_{LIQ}$=0.74,
- RSWINHF$_{ICE}$=0.74,
- RLWINHF$_{LIQ}$=0.95,
- RLWINHF$_{ICE}$=0.76.

3.7 Cloud overlap assumption

The cloud overlap assumption is used for the determination of C_{tot} (see Section 1). It is determined by the NOVLP key: NOVLP = 1 means "maximum-random overlap", NOVLP = 2 means "maximum overlap " and NOVLP = 3 means "random overlap". An option NOVLP=6 has been added in the code for CMIP6 following the work of Thouron (Thouron et al 2017) which proposes to use different overlap assumptions for the computation of clear-sky fraction and for total efficiency of diffusion. This option corresponds to a max-random overlap for the computation of the clear-sky fraction and a random one for the computation of the solar zenithal angle. This formulation is particularly adapted to multi-layers case.

3.8 Interactions with the SURFEX module

To obtain consistent radiative fluxes in ARPEGE and in SURFEX, downward fluxes calculated by ARPEGE and provided to the SURFEX scheme have to be cut into spectral bands. Moreover, for the long-wave radiative fluxes, additional corrections have to be made in the APLPAR subroutine to ensure equal radiative net budgets in ARPEGE as well as in SURFEX.

4 Simplified radiation scheme

Given the computation cost of the radiation scheme, it seems interesting to call it only at some time steps, named radiative time steps. Nevertheless, it is necessary to produce radiative flows at each time step. One thus needs a way to calculate these fluxes at low cost. This is carried out in Arpege-climat by calling subroutine RADHEAT.

The idea of this routine is to save between two radiative time steps the solar transmissivity and long-wave emissivity. Let F_{SW} (PFRSO in APLPAR) and F_{LW} (PFRTH) be total net fluxes of solar and infrared radiation calculated at the last radiative time step. The approximation consists in maintaining constant the values of transmissivities t_0 (PTRSOL) and emissivities (PEMTD) ϵ_0 calculated at this time:

$$t_0 = \frac{F_{SW0}}{\mu_0 E^0} \tag{50}$$

$$\epsilon_0 = \frac{F_{LW0}}{\sigma T_0^4} \tag{51}$$

Then, in the next time steps, one recomposes solar and infrared fluxes by modifying only the solar incidence μ and temperature T:

$$F_{SW} = t_0 \mu_0 E^0 \tag{52}$$

$$F_{LW} = \epsilon_0 \sigma T^4 \tag{53}$$

Note that with the logical key LABSDIR=.TRUE., a modification to diagnose layers' radiative properties as function of infrared fluxes of the last radiative step is introduced (in APLPAR). The surface "Delta" (difference between the surface infrared flux with Ts from SURFEX and the surface infrared flux computed in the last call of the full radiative scheme) infrared flux is computed as function of Delta Ts. The modification consists in a progressive absorption of the "Delta IR" from the surface to the top of the atmosphere. It starts with a diagnostic of the IR absorption rate in each atmospheric layer and modifies the corresponding heating rate. The radiative budget is modified taking into account the delta IR upward flux reaching TOA.

8

The turbulence scheme

1 Introduction

1.1 Objectives

Among all the different processes which need to be represented in the numerical prediction model for the atmosphere, valid from the short term range forecasting to the simulation of the general climate, one of the most important one is the decreasing of the vertical gradients or heterogeneous feature in potential temperature, wind or humidity. These processes correspond to "vertical transport" or "dissipation", represented either by what is usually called "turbulence" or "convection".

The splitting of the mixing processes between "turbulent" or "convective" ones are made in terms of asymmetric buoyancy effects for the convection, whereas the turbulence is think in terms of symmetric and horizontal rolls, mainly located within the dry PBL, bellow the cloudy part of the atmosphere.

This separation was clear until the 80's and 90's. Since then, more and more moist effects are taken into account in the turbulent schemes, and more and more dry thermal convective processes are associated to the shallow convection (inside the cloud) and in the dry PBL (bellow the cloud base).

Nowadays, several unified "turbulent + mass-flux" schemes exist. They are often called "EDMF" (or "EDKF"), for "Eddy Diffusivity Mass-Flux". The aim of the scheme described in this documentation is to described only the vertical mixing of atmospheric variables due to turbulent processes, with the vapor and condensed waters taken into account. The dry and moist convection processes are not explicitly managed, though they impact in some way on the turbulent scheme via the two "turbulent" and "convective" tendencies which are fully combined for computing the next time-step, with possible covering of the two schemes on the vertical.

The aim the the parameterization is to compute the exchange coefficients K_m (valid for the momentum, i.e. the 2 wind components u and v) and K_h (valid for the potential temperature and the specific humidity). These exchange coefficients are used to compute the turbulent fluxes, used in the implicit solver for the vertical mixing and obtained by the inversion of a tri-diagonal matrix.

The choice of the present parameterization, implemented both in the NWP and the GCM versions of ARPEGE, has been mainly motivated by the hope to better represent the marine Strato-cumulus in the model. Indeed, there is a lack of strato-cumulus in most of the CMIP5 GCM and NWP models in the Eastern part of the tropical oceans, close to the coast. These low-level clouds are important features for determining even the sign of the local impacts associated with the Climate Change.

1.2 From CMIP5 to CMIP6

For CMIP5, a turbulent scheme, based on the stationary equation for the turbulent kinetic energy, using the 2.0 order scheme of Mellor and Yamada (1977, 1982), has been used in the Climate version of ARPEGE. The numerical computations of the exchange coefficients K_m and K_h are described more precisely in Ricard and Royer (1993). The moist processes are taken into account in the Mellor-Yamada set of equations, by using the subgrid variance of atmospheric humidity, following the ideas of Sommeria, Deardorff, Bougeault (1976, 1977, 1982) where a certain statistical law is prescribed (an asymmetric and exponential fixed PDF).

The turbulent kinetic energy variable is defined on the full-levels of ARPEGE (same location as for the wind or the temperature). As a consequence the advective processes could be activated, but they are switched-off since it is a stationary (diagnostic) turbulent kinetic energy equation. A lot of half-summations are made to go from the full-levels to the half-levels, and vice versa. With the crude spacing of the vertical levels, such half-summations processes create an artificial and strong mixing at the top of the PBL and at the inversion. It is probably the reason why the shallow convection processes are - in some way - represented by this moist diagnostic turbulent scheme, via a synergy between the moist subgrid representation and the (too) strong vertical mixing.

The present prognostic scheme for CMIP6 corresponds to a continuous series of developments, starting in 1998-99.

The different processes represented in this new set of parameterizations are

- the turbulent kinetic energy (e) is computed with a prognostic equation where the horizontal and vertical advections are switched-off, with the vertical mixing of e, the dissipation, the shear (dynamical) production and the buoyancy (thermal) production occur ;
- the turbulent kinetic energy (e) is computed on the “half-levels”, the levels where the exchange coefficients K_m and K_T, also the vertical velocity, are computed (in between the “half-levels” where are computed the wind components, the temperature and the specific humidity) ;
- the turbulent kinetic energy (e) has a minimum value (typically of $10^{-6}\ m^2\ s^{-2}$) ;
- the formulations for the turbulent fluxes are given in Redelsperger and Sommeria (1981) and Cuxart, Bougeault, Redelsperger (2000) ;
- the “moist” versions of the fluxes are computed by using the Lilly (1968) potential temperature (i.e. θ_{vl}) and the Betts (1973) variables (i.e. for θ_l and q_t), for the vertical mixing of the conservative variables ;
- the computations of the sub-grid variance of cloud liquid water are made with the hypotheses of Bougeault (1982) and Bechtold (1995), by using some mixed symmetric (Gaussian) and asymmetric (Exponential), in order to represent the Cumulus and the Strato-Cumulus, respectively ;
- the mixing and dissipation lengthes are given by the non-local formulation of Bougeault and LacarrÃ¨re (1989) ;
- the surface layer value for the turbulent kinetic energy is given by AndrÃ© et al.(1978) ;
- the surface layer values for the exchange coefficients are given by Louis (1979, 1981) ;
- the vertical mixing of the turbulent kinetic energy is made in an implicit way, by taking into account a linearized version of the dissipation term, i.e. the non-linear term $(\overline{e})^{3/2}$, leading to the weighting factors $\alpha_{exp} = -0.5$ and $\alpha_{imp} = 1.5$;
- the top-PBL vertical entrainment is parameterized following the ideas of Grenier and Bretherton (2000) and Grenier (2002).

2 The CBR scheme

2.1 The turbulent kinetic energy equation

The turbulent kinetic energy equation gives the change in time of the grid-cell mean value ($\overline{e}$).

$$\begin{aligned}
\frac{\partial \overline{e}}{\partial t} &= [\,\text{Advect.}\,] + \text{Diff}_{\text{vert}} + P_{\text{dyn.}} + P_{\text{ther.}} - D\text{iss}\,, & (1)\\
\text{Diff}_{\text{vert}} &= -\,\frac{1}{\rho}\,\frac{\partial}{\partial z}\left(\rho\,\overline{e'w'}\right)\,, & (2)\\
P_{\text{dyn.}} &= -\,\left[\overline{u'w'}\,\frac{\partial \overline{u}}{\partial z} + \overline{v'w'}\,\frac{\partial \overline{v}}{\partial z}\right]\,, & (3)\\
P_{\text{ther.}} &= \beta\,\overline{w'\theta'_{vl}}\,, & (4)\\
D\text{iss} &= C_\epsilon\,\frac{\overline{e}\,\sqrt{\overline{e}}}{L_\epsilon}\,. & (5)
\end{aligned}$$

Except the (neglected) advective part (Advect.), it is the sum of 4 terms. The first term ($\text{Diff}_{\text{vert}}$) is the vertical mixing (or diffusion) of $\overline{e}$. It represents the change of $\overline{e}$ by the turbulent processes plus, in some way, the impact of the (unknown) presso-correlation term. There are two production terms: the dynamical production $P_{\text{dyn.}}$ is always positive and represents the impact of the shear of the wind components, the thermal production $P_{\text{ther.}}$ can be positive or negative, depending on the vertical fluxes of the Lilly (1968) potential temperature θ_{vl}, computed in a complex way by using the fluxes of the conservative Betts (1973) variables θ_l and q_t. The last dissipation term (Diss) depends on a constant C_ϵ and on a dissipation length L_ϵ.

For the thermal production, the formulation for β and $\overline{\theta}_{vl}$ writes:

$$\begin{aligned}
\beta &= \frac{g}{\overline{\theta}}\,, & (6)\\
\overline{\theta}_{vl} &= \overline{\theta}\,(1 + 0.608\,\overline{q}_v - \overline{q}_c)\ . & (7)
\end{aligned}$$

The potential temperature of Lilly depends on the grid-cell average of both the water vapor $\overline{q}_v$ and the condensed water $\overline{q}_c$ (either liquid or solid). The main difficulty is to compute the fluxes of these quantities, whereas the moist fluxes are only known for the Betts variables $\overline{w'\theta'_l}$ and $\overline{w'q'_t}$.

The conservative variables of Betts write in terms of the mean water vapor ($\overline{q}_v$), the mean liquid cloud water ($\overline{q}_l$), the mean solid cloud water ($\overline{q}_i$) and the mean total condensed cloud water ($\overline{q}_c = \overline{q}_l + \overline{q}_i$) :

$$\begin{aligned}
\overline{\theta}_l &= \overline{\theta}\left(1 - \frac{L_v\,\overline{q}_l + L_f\,\overline{q}_i}{c_p\,T}\right)\,, & (8)\\
\overline{q}_t &= \overline{q}_v + \overline{q}_c\,. & (9)
\end{aligned}$$

The two terms L_v and L_f are the latent heats of vaporization and fusion, respectively.

The first order turbulent fluxes are written in terms of the vertical gradients of the mean variables, following the formulations of Redelsperger and Sommeria (1981 ; or RS81), also of Cuxart, Bougeault and Redelsperger (2000 ; or CBR00).

$$\begin{aligned}
\overline{u'w'} &= -\,C_m\,L_m\,\sqrt{\overline{e}}\,\frac{\partial \overline{u}}{\partial z} \qquad ;\ K_m \;=\; C_m\,L_m\,\sqrt{\overline{e}}\,, & (10)\\
\overline{v'w'} &= -\,C_m\,L_m\,\sqrt{\overline{e}}\,\frac{\partial \overline{v}}{\partial z} \qquad ;\ K_m \;=\; C_m\,L_m\,\sqrt{\overline{e}}\,, & (11)
\end{aligned}$$

$$\overline{e'w'} \;=\; -\,C_e\,L_m\,\sqrt{\overline{e}}\;\frac{\partial\,\overline{e}}{\partial z} \qquad ;\; K_e \;=\; C_e\,L_m\,\sqrt{\overline{e}}\,, \tag{12}$$

$$\overline{w'\theta_l'} \;=\; -\,C_\theta\,L_m\,\sqrt{\overline{e}}\;\frac{\partial\,\overline{\theta}_l}{\partial z}\;\phi_3 \qquad ;\; K_T \;=\; C_\theta\,L_m\,\sqrt{\overline{e}}\,\phi_3\,, \tag{13}$$

$$\overline{w'q_t'} \;=\; -\,C_q\,L_m\,\sqrt{\overline{e}}\;\frac{\partial\,\overline{q}_t}{\partial z}\;\psi_3 \qquad ;\; K_q \;=\; C_q\,L_m\,\sqrt{\overline{e}}\,\psi_3\,. \tag{14}$$

Note that the unit of turbulent exchange coefficients computed in acturb.F90 is $kg.m^{-2}.s^{-1}$. Indeed, in the relation:

$$\frac{\partial\Psi}{\partial t} = -g\frac{\partial(-\rho\overline{w'\Psi'})}{\partial p} = -g\frac{\partial(-\rho*(-K\frac{\partial\Psi}{\partial z}))}{\partial p} \tag{15}$$

unit of K is m^2s^{-1}. The computed turbulent exchange coefficients (PKTROV, PKQROV, PKQLROV,PKUROV) are $\frac{g\rho K}{\Delta\Psi}$.

The first order turbulent fluxes and the associated exchange coefficients (10) to (14) depend on four unknown constants (C_m, C_e, C_θ, C_q), with the mixing length denoted by L_m and the dissipation length by L_ϵ.

This turbulent scheme includes two stability functions (ϕ_3 and ψ_3), which are found to be equal in that case of an "1D-vertical" turbulence, leading to

$$\phi_3 \;=\; \psi_3 \;=\; \frac{1}{1+\mathrm{C}\,(R_\theta+R_q)}\,. \tag{16}$$

The term $\mathrm{C} = C_\theta C_{\epsilon_\theta}$ is another unknown constant. The stability functions ϕ_3 and ψ_3 describe the enhancement or inhibition of turbulent transfers by stability effects.

Hypothesis 1 : In the Meso-NH and AROME LAM, the true computations of $R_\theta + R_q$ need to keep some pseudo-prognostic variables (kept in memory from one time step for the next one). In the CBR00 scheme implemented in ARPEGE, the computation of (16) is made with a more straightforward method, without pseudo-prognostic variables but with a direct computation of the vertical gradient of $\overline{\theta}_{vl}$ instead, with the assumption

$$R_\theta + R_q \;\approx\; \beta\;\frac{L_m\,L_\epsilon}{\overline{e}}\;\frac{\partial\overline{\theta}_{vl}}{\partial z}\,. \tag{17}$$

The vertical gradient is computed from (7), with no hypothesis concerning the sub-grid variability of the humidity.

The stability function $\phi_3 = \psi_3$ varies from 0.78 (for the stables cases) to possible large or even infinite values for unstable cases, for instance if $\mathrm{C}(R_\theta+R_q) = -1$. As a practical rule, values for $\phi_3 = \psi_3$ are limited to the maximum value of ACBRPHIM = 2.2 (or possibly less), available in the NAMELIST of ARPEGE.

2.2 The constants

The set of constants used in ARPEGE are different from the one used in Meso-NH and AROME. Three of the independent constants defined in Meso-NH and AROME are C_{pv}, $C_{p\theta}$ and C_{ϵ_θ}, leading to the four ARPEGE constants C_m, C_θ, C_q and C, according to

$$C_m \;=\; \frac{4}{15\,C_{pv}}\,, \tag{18}$$

$$C_\theta \;=\; C_q \;=\; \frac{2}{3\,C_{p\theta}}\,, \tag{19}$$

$$\mathrm{C} \;=\; \frac{C_\theta}{C_{\epsilon_\theta}} \;=\; \frac{2}{3\,C_{p\theta}\,C_{\epsilon_\theta}}\,. \tag{20}$$

In addition to C_{pv}, $C_{p\theta}$ and C_{ϵ_θ}, the other independent constants are C_ϵ in (5) and C_e in (12).

There exist different sets of values for these five independent constants, depending on different papers or different tuning of the models. The set of constants proposed by Cheng, Canuto and Howard (2002)(below CCH02) is used here.

The choice of the Cheng et al. constants provides an enhancement of the momentum fluxes (a decrease in C_{pv}), an increase of the dissipation (a decrease in C_ϵ) and with almost the same thermal and moisture fluxes (not so different values for C_e, $C_{p\theta}$ and C_{ϵ_θ}), compared with other tested sets of constants (e.g those of Cuxart, Bougeault, Redelsperger (2000) ou Redelsperger, Sommeria (1981)). The constant values are: $C_\epsilon = 0.845$, $C_e = 0.34$, C_{pv}= 2.11, $C_{p\theta}$= 4.65 and $C_{\epsilon_\theta} = 1.01$.

The corresponding values for C, C_m and C_θ write: C = $2/(3\,C_{p\theta}C_{\epsilon_\theta})$ =0.143, $C_m = 4/(15\,C_{pv})$= 0.126, $C_\theta = C_q = 2/(3\,C_{p\theta})$= 0.143.

In ARPEGE, there are five constants which are all set in the NAMELIST of the model. In addition to the same constant C used in (16) for the definition of $\phi_3 = \psi_3$, the four other constants are AKN, α_T, α_e and ALD, corresponding to

$$\begin{array}{lclcl}
K_m & = & \text{AKN}\ L_m\,\sqrt{\overline{e}} & \Rightarrow & \text{AKN} \;=\; C_m \;=\; \dfrac{4}{15\,C_{pv}} \quad ; \quad C_{pv} \;=\; \dfrac{4}{15\ \text{AKN}}\ ,(21)\\
K_T & = & K_q \;=\; \alpha_T\,K_m\,\phi_3 & \Rightarrow & \alpha_T \;=\; \dfrac{C_\theta}{C_m} \;=\; \dfrac{5}{2}\,\dfrac{C_{pv}}{C_{p\theta}} \quad ; \quad C_{p\theta} \;=\; \dfrac{2}{3\ \text{AKN}\ \alpha_T}\ ,(22)\\
K_e & = & \alpha_e\,K_m & \Rightarrow & \alpha_e \;=\; \dfrac{C_e}{C_m} \;=\; \dfrac{15}{4}\,C_{pv}\,C_e \quad ; \quad C_e \;=\; \alpha_e\ \text{AKN}\ ,(23)\\
D\text{iss} & = & \dfrac{\overline{e}\,\sqrt{\overline{e}}}{\text{ALD}\ \ L_\epsilon} & \Rightarrow & \text{ALD} \;=\; \dfrac{1}{C_\epsilon} \quad ; \quad C_\epsilon \;=\; \dfrac{1}{\text{ALD}}\ .(24)
\end{array}$$

The values used in this version are AKN = $C_m = 4/(15\,C_{pv})$= 0.126;ALPHAT = $\alpha_T = (5\,C_{pv})\,/(2\,C_{p\theta})$= 1.13 and ALD = $1/C_\epsilon$= 1.18 and ALPHAE = $\alpha_e = C_e/C_m = 2.7$.

2.3 The vertical mixing and the (dynamic+thermal) productions

The dissipation term is given directly by (5). The vertical mixing term and the dynamical production term are computed by putting (10) and (12) into (2) and (3), with $dp = -\,\rho\,g\,dz \;=\; -\,\rho\,d\phi$, leading to

$$\text{Diff}_{\text{vert}} \;=\; -\,g\,\frac{\partial}{\partial p}\left(\rho\,g\,K_e\,\frac{\partial\overline{e}}{\partial\phi}\right) \;=\; -\,g\,\frac{\partial}{\partial p}\left(\rho\,g\,C_e\,L_m\,\sqrt{\overline{e}}\,\frac{\partial\overline{e}}{\partial\phi}\right)\ , \qquad (25)$$

$$P_{\text{dyn.}} \;=\; K_m\,\left[\left(\frac{\partial\overline{u}}{\partial z}\right)^2 + \left(\frac{\partial\overline{v}}{\partial z}\right)^2\right] \;=\; C_m\,L_m\,\sqrt{\overline{e}}\,\left[\left(\frac{\partial\overline{u}}{\partial z}\right)^2 + \left(\frac{\partial\overline{v}}{\partial z}\right)^2\right]\ . \qquad (26)$$

The thermal production term (4) is computed differently. It is not computed as the product of an exchange coefficient with the associated gradient, i.e.

$$\overline{w'\theta'_{vl}} \;\neq\; -\,C_\theta\,L_m\,\sqrt{\overline{e}}\,\frac{\partial\,\overline{\theta}_{vl}}{\partial z}\,\phi_3\ . \qquad (27)$$

The usual method is rather to express $P_{\text{ther.}}$ in terms of the conservative Betts variables, given by (8) and (9), leading to a formulation

$$P_{\text{ther.}} \;\equiv\; \beta\,\overline{w'\theta'_{vl}} \;=\; \beta\,E_\theta\,\overline{w'\theta'_l} \;+\; \beta\,E_q\,\overline{w'q'_t}\ , \qquad (28)$$

where the two terms E_θ and E_q are two non-trivial coefficients to be determined, which depends on the normalized saturation deficit and to the the sub-grid variance of the humidity (cloud water).

The two following additional hypotheses are made in (28)

$$\overline{w'\theta'_l} \;=\; -\,C_\theta\,L_m\,\sqrt{\bar{e}}\,\frac{\partial\,\overline{\theta}_l}{\partial z}\,\phi_3\;, \qquad (29)$$

$$\overline{w'q'_t} \;=\; -\,C_q\,L_m\,\sqrt{\bar{e}}\,\frac{\partial\,\overline{q_t}}{\partial z}\,\psi_3\;, \qquad (30)$$

with $\phi_3 = \psi_3$ given by (16) and (17).

2.4 A3-d: The sub-grid variability of cloud water

The two coefficients E_θ and E_q are to be determined in order to allow the computation of the moist thermal production (28). To do so, some additional hypotheses must be done, concerning the sub-grid variability of the humidity. It is the reason why the moist turbulent scheme is so dependent on the way to represent the moist processes.

We will describe the same method that has been already used from the old version 3 to the actual version 6 of the ARPEGE-GCM (see Ricard and Royer, 1993). Let us define the parameter "s" which depends on the sub-grid departure terms θ'_l and q'_t of the conservative Betts variables :

$$s \;=\; \frac{a}{2}\,\left(q'_t \,-\, \alpha_1\,\theta'_l\right)\;, \qquad (31)$$

$$a \;=\; \left[1+\frac{L_{v/f}}{c_p}\,\left(\frac{\partial\,q_{sat}}{\partial\,T}\right)_{(T=T_l)}\right]^{-1}\;, \qquad (32)$$

$$\alpha_1 \;=\; \frac{T}{\theta}\,\left(\frac{\partial\,q_{sat}}{\partial\,T}\right)_{(T=T_l)}\;. \qquad (33)$$

Both a and α_1 depend on the derivative with respect to T of the saturating water vapor q_{sat}, where the derivative is computed at the value T_l.

Let us denote by Q_1 the normalized saturation deficit, which depends both on a and on the standard deviation of s, denoted by σ_s, leading to

$$Q_1 \;=\; a\,\left[\frac{q_t - q_{sat}(T_l)}{2\,\sigma_s}\right]\;. \qquad (34)$$

The standard deviation of s, i.e. σ_s, writes in terms of a, α_1 and the second order fluxes, giving

$$\sigma_s \;=\; \frac{a}{2}\left[\,\overline{(q'_t)^2}\;-\;2\,\alpha_1\,\overline{(q'_t\,\theta'_l)}\;+\;(\alpha_1)^2\,\overline{(\theta'_l)^2}\,\right]^{1/2}\;. \qquad (35)$$

From Cuxart et al. (2000), the second order fluxes write

$$\overline{(q'_t)^2} \;=\; \mathrm{C}\,(L_m)^2\,\psi_3\,\left(\frac{\partial\overline{q_t}}{\partial z}\right)^2\;, \qquad (36)$$

$$\overline{(\theta'_l)^2} \;=\; \mathrm{C}\,(L_m)^2\,\phi_3\,\left(\frac{\partial\overline{\theta}_l}{\partial z}\right)^2\;, \qquad (37)$$

$$\overline{(q'_t\,\theta'_l)} \;=\; \mathrm{C}\,(L_m)^2\,\frac{(\phi_3+\psi_3)}{2}\,\left(\frac{\partial\overline{q_t}}{\partial z}\right)\left(\frac{\partial\overline{\theta}_l}{\partial z}\right)\;. \qquad (38)$$

Let us replace the second order fluxes (36) to (38) into (35), with the property $\phi_3 = \psi_3$ valid for the present "1D-column" version of the scheme. The result can be factorized into the square of a quantity expressed in terms of the vertical gradients of $\overline{\theta}_l$ and $\overline{q_t}$, leading to

$$\sigma_s \;=\; \frac{a}{2}\,\sqrt{\mathrm{C}\,(L_m)^2\,\phi_3}\;\left|\;\frac{\partial\overline{q_t}}{\partial z}\;-\;\alpha_1\,\frac{\partial\overline{\theta}_l}{\partial z}\;\right|\;. \tag{39}$$

It is possible to write differently the term $\sqrt{\mathrm{C}\,(L_m)^2\,\phi_3}$, in order to express the result in terms of the exchange coefficient K_T given by (13), the constant C_{ϵ_θ} given by (20), the mixing length L_m and the turbulent kinetic energy $\overline{e}$, leading to

$$\sqrt{\mathrm{C}\,(L_m)^2\,\phi_3} \;=\; \frac{1}{\sqrt{C_{\epsilon_\theta}}}\,\sqrt{\frac{L_m\,K_T}{\sqrt{\overline{e}}}}\;. \tag{40}$$

The coefficient $1/\sqrt{C_{\epsilon_\theta}}$ (ARSB2 in the code) equals 0.833.

2.5 The sub-grid variability of cloud water

The method, described in Mellor (1977), starts with the definition of (7), (8) and (9) for the temperature of Lilly and the conservative variables of Betts.

In order to get an equation like (28) for the flux of θ_{vl}, let us write θ_{vl} in terms of the Betts variables θ_l and q_t :

$$\theta_{vl} \;=\; \theta_l \;+\; D_1\,q_t \;+\; D_2\,q_c\;, \tag{41}$$

$$D_1 \;=\; 0.608\,\theta\;, \tag{42}$$

$$D_2 \;=\; \left(\frac{L_{v/f}}{c_p\,T}\;-\;1.608\right)\,\theta\;. \tag{43}$$

From (41), the flux of the Lilly temperature writes

$$\overline{w'\theta'_{vl}} \;=\; \overline{w'\theta'_l} \;+\; D_1\,\overline{w'q'_t} \;+\; D_2\,\overline{w'q'_c}\;. \tag{44}$$

The last terms of (41) or (44), i.e. $D_2\,q_c$ or $D_2\,\overline{w'q'_c}$, cannot be computed without making additional hypotheses. There are 3 such hypotheses, all involving the fluxes, not directly the variables.

Hypothesis 2 : established by Mellor (1977) for the Gaussian distributions, it has been extended for non-Gaussian cases by Bougeault (1982). From Bougeault (1982), Cuijpers and Bechtold (1995), Bechtold et al. (1995), it is assumed that the residual term $\overline{w'q'_c}$ in (44) can be expressed in terms of 's' defined by (31), leading to

$$\overline{w'q'_c} \;=\; \overline{w's'}\;\;\lambda_3(Q_1,A_S)\;\left\{\frac{\overline{s'q'_c}}{(\sigma_s)^2}\right\}\;, \tag{45}$$

$$\text{where}\;\;\overline{w's'} \;\equiv\; \frac{a}{2}\;\left[\,\overline{s'q'_t}\;-\;\alpha_1\;\overline{s'\theta'_l}\,\right]\;, \tag{46}$$

where λ_3 is a function of the two variables (Q_1, A_S), function defined hereafter, depending of the normalized saturation deficit Q_1 and the asymmetry factor A_S.

Hypothesis 3 : from Bougeault (1982), it is assumed that the last term of (45) can be written as

$$\frac{\overline{s'q'_c}}{(\sigma_s)^2} \;=\; 2\;\;F_2(Q_1,A_S)\;, \tag{47}$$

Table 8.1: Value of A_S and λ_3 in terms of Q_1.

Exp / Asym. (Cu)	Mixed regime (Cu / Sc)	Gaussian (Sc)
$Q_1 < -2$	$-2 < Q_1 < 0$	$Q_1 > 0$
$A_S = 2$	$A_S = -Q_1$	$A_S = 0$
$\lambda_3 = 3$	$\lambda_3 = 1 - Q_1$	$\lambda_3 = 1$

where F_2, like λ_3, is a function of the two variables (Q_1, A_S).

Finally, from (44) to (47), the flux of θ_{vl} can be rewritten as

$$\overline{w'\theta'_{vl}} \;=\; \left[\,\overline{w'\theta'_l} \,+\, D_1\,\overline{w'q'_t}\,\right] \,-\, \{a\,D_2\,F_2\,\lambda_3\}\left[\,\alpha_1\overline{w'\theta'_l} \,-\, \overline{w'q'_t}\,\right]. \tag{48}$$

It is the sum of two terms. The first term into brackets doesn't depend on F_2 or the variability in cloud water and moisture. The second term directly depends on the variability in cloud water and moisture, via the product of the four terms $a\,D_2\,F_2\,\lambda_3$.

Hypothesis 4 : from Bougeault (1982), it is assumed that the probability density function $G(s)$ is a possibly mixed symetric (Gaussian) and asymmetric (exponential) function. This PDF gives the average values over the grid-cell for the cloud cover $\overline{N}_s$, for the cloud water content $\overline{q_c}_s$ and the normalized second order flux $\overline{s'q'_c}\;/\;[\,2\,(\sigma_s)^2\,]$. It results

$$\overline{N}_s \;=\; F_0(Q_1, A_S) \;=\; \int_{-Q_1}^{+\infty} G(t)\,dt\,, \tag{49}$$

$$\frac{\overline{q_c}_s}{2\,\sigma_s} \;=\; F_1(Q_1, A_S) \;=\; \int_{-Q_1}^{+\infty} (Q_1 + t)\,G(t)\,dt\,, \tag{50}$$

$$\frac{\overline{s'q'_c}}{2\,(\sigma_s)^2} \;=\; F_2(Q_1, A_S) \;=\; \int_{-Q_1}^{+\infty} t\,(Q_1 + t)\,G(t)\,dt\,. \tag{51}$$

For the given PDF, i.e. $G(s)$, all the functions F_0, F_1, F_2 and λ_3 can be analytically determined and numerically computed. They depend on the two variables (Q_1, A_S) and they determine the flux (48), and thus the thermal production term (28).

The Table (8.1) gives the definitions for A_S and λ_3 in terms of the normalized saturation deficit Q_1. The symmetric Gaussian distributions regime ($Q_1 > 0$) is supposed to represent the Strato-Cumulus. The asymmetric Exponential regime ($Q_1 < -2$) is supposed to represent the Cumulus regime. The regime in between, a mix of the Gaussian and the Exponential PDFs, has been pre-computed and tabulated in the Meso-NH model.

The "large-scale" (or "stratiform") cloud cover $\overline{N}_s$ is computed at each time step and for each grid-point with (49). Similarly, the "large-scale" cloud water content $\overline{q_c}_s$ is computed with (50), with $2\,\sigma_s$ given by (39) and (40).

These "large-scale" cloud cover and cloud water content are merged with the "convective" quantities, with some overlapping assumptions, in order to transmit them to the radiation code.

Also, the "large-scale" cloud water content is transmitted to the Bulk prognostic scheme of condensation and precipitation (Lopez's scheme).

The identification of (28) with (48) leads to the following formulas for E_θ et E_q :

$$E_\theta \;=\; 1 \,-\, \alpha_1\,\{\,a\,D_2\,F_2\,\lambda_3\,\}\,, \tag{52}$$

$$E_q \;=\; D_1 \,+\, \{\,a\,D_2\,F_2\,\lambda_3\,\}\,, \tag{53}$$

2.6 Min and Max Q1 limitations: LECTQ1=.TRUE.

During the development of the high-resolution version of the model (T359) for CMIP6, it appears some instabilities, for which we have decided to reactivate the limitation of Q1, already used in CMIP5 configuration. This limitation is run with the logical key LECTQ1= .TRUE. in the acturb.F90 subroutine. It corresponds both to a minimum and maximum value of Q1. As in CMIP5 Ricard-Royer scheme, it appears essential to limit the occurence of negative moistures (otherwise unrealistic values of the mixing length can appear) through a condition on the standard deviation of q_w. The following inequality is met:

$$|Q1| > \frac{A|\Delta\overline{q}||\frac{\partial\overline{q_w}}{\partial z}|}{\overline{q_w}|\frac{\partial\overline{q_w}}{\partial z} - \alpha_1\frac{\partial\overline{\theta_l}}{\partial z}|} = Q1_{min} \tag{54}$$

In the code, $Q1_{min}$ is ZQ1MIN, A is STTBMIN (STTBMIN=1.73), $|\Delta\overline{q}|$ is ZDELTQH, $|\frac{\partial\overline{q_w}}{\partial z}|$ is ABS(ZDQW), $\overline{q_w}$ is ZQWF, and $|\frac{\partial\overline{q_w}}{\partial z} - \alpha_1\frac{\partial\overline{\theta_l}}{\partial z}|$ is ZDIFFC. This minimum Q1 value gives a maximum value for the mixing length. For Q1 max value, firstly the minimum value for Φ_3 (neutral cases, see Cuxart et al.), is computed, in a consistent manner with the computation done in ACBL89.F90, by:

$$\phi_{3min} = \frac{1}{1 + 2.ARSC1} \tag{55}$$

Then the maximum of Q1 (ZQ1MAX in the code) is computed as:

$$Q_{1max} == \frac{q_t - q_{sat}(T_l)}{\sqrt{\mathrm{C}\ (L_m)^2\ \phi_3}} \frac{\partial\overline{q_t}}{\partial z} - \alpha_1\frac{\partial\overline{\theta}_l}{\partial z} \tag{56}$$

2.7 The mixing and dissipation lengths

The mixing length L_m and the dissipation length L_ϵ are computed in a non-local way, following Bougeault et Lacarrère (1989). Starting from a level at the altitude z where the energy is set to the local turbulent kinetic energy $\overline{e}\,(z)$, the non-local lengths L_{up} and L_{down} are computed as the possible upward and downward displacements, respectively, until the energy $\overline{e}\,(z)$ is equal to the integral of the work of the buoyancy force, expressed in terms of $\overline{\theta}_{vl}$ give by (7).

It results

$$\overline{e}\,(z) \;=\; \int_{z}^{z+L_{up}} \beta\left[\,\overline{\theta}_{vl}(z')\;-\;\overline{\theta}_{vl}(z)\,\right]\,dz'\,, \tag{57}$$

$$\overline{e}\,(z) \;=\; \int_{z-L_{down}}^{z} \beta\left[\,\overline{\theta}_{vl}(z)\;-\;\overline{\theta}_{vl}(z')\,\right]\,dz'\,. \tag{58}$$

The formulation for L_m in ARPEGE-GCM is:

$$L_m \;=\; \left[\,\frac{1}{2}\left\{\,(L_{up})^{-2/3} + (L_{down})^{-2/3}\,\right\}\right]^{-3/2}\,. \tag{59}$$

Close to the ground, the Karman law $L_m \approx 0.4\,z$ is not set. Instead, following the theoretical arguments of Redelsperger, Mahé and Carlotti (2001), it could be normal to approach $L_m \approx 2.8\,z$ in true convective cases (values of $L_m \approx 2\,z$ are simulated in 1D cases as RICO or ARM-Cu).

For the stable regimes, the numerical scheme for the computation of L_{up} and L_{down} are made in a very accurate way, with a second order scheme, in order to be as close as possible to the Deardoff's length:

$$L_D \;=\; \sqrt{\frac{2\,\overline{e}}{\beta\,\left(\partial\overline{\theta}_{vl}/\partial z\right)}} \;=\; \frac{\sqrt{2}}{N}\,\sqrt{\overline{e}}\,, \tag{60}$$

where $N^2 = \beta\,\partial\overline{\theta}_{vl}/\partial z$ is the square of the Brunt Väisälä frequency.

The Deardoff's length can be very small for very stable cases ($L_m \approx L_D \ll 1$ m), because in that cases $\overline{e}$ is small, reaching the minimum value of 10^{-4} to 10^{-6} $m^2\ s^{-2}$, also because $\partial\,\overline{\theta}/\partial z \gg 0$ is large.

As a consequence, the small values for both L_m and $\overline{e}$ in the stable regions lead to small values for the exchange coefficients K_m to K_q given by (10) to (14), which all vary as $L_m\,\sqrt{\overline{e}}$.

The large scale ARPEGE-GCM must be able to manage all the cases, from the surface layer to the mesosphere, including the planetary boundary layer, the troposphere and the stratosphere. In order to maintain a minimum vertical mixing in all stable regions, some modifications have been included after the computation of L_m in (87).

$$L_m \;=\; \text{Max}\,[\,L_m\ ;\ \text{Min}\,(\lambda_E\ ;\ 0.4\,z)\,]\ . \tag{61}$$

In the code $g * L_m$ is ZGLMIX and $0.4gz$ is ZGLKARMN.

As a consequence, the asymptotic value λ_E replaces the Deardorff's length in the stable regions. Close to the ground, the formulae (61) is a security, leading to $L_m \geq 0.4\,z$ for $z < \lambda_E/0.4$ and $L_m \geq \lambda_E$ for $z > \lambda_E/0.4$. Typically, $\lambda_E = 10$ m (ALMAVE in the code) and $\lambda_E/0.4 = 25$ m. The mixing length L_m is PLMECT in the code and the dissipation length L_ϵ (1/PUSLE in the code) is computed as ALD*PLMECT, with ALD=1.18.

2.8 The turbulent kinetic energy in the surface layer

The values of $\overline{e}$ are given by the prognostic equation (1) for each upper-air half-levels, i.e. on the half-levels where the exchange coefficients K_m to K_q are computed, at the middle points between the full-levels where all the thermodynamics variables are available (temperature, wind, specific humidity, ...)

But the vertical mixing processes represented by (2) needs the knowledge of $\overline{e}$ at the groung level or in the surface layer, denoted by $\overline{e}_S$.

In the current version of the code (for .LECTREP=.TRUE.), $\overline{e_s}$ =$\overline{e}_{klev}$, value at the last vertical level. This hypothesis avoids strong and unrealistic values of $\overline{e_s}$ over mountains.

For the key LECTREP=.FALSE., $\overline{e}_S \equiv [\overline{(u'_S)^2} + \overline{(v'_S)^2} + \overline{(w')^2}]/2$ is computed following André et al. (1978), see their Eq.(29), p.1866, with the term $(-\zeta)^{2/3}\,(u_*)^2$ dropped in the instable case, leading to :

$$\overline{e}_S \;=\; 3.75\,(u_*)^2 \;+\; 0.3\,(w_*)^2\,(1-\delta_{stab})\,. \tag{62}$$

The first part, valid for both stable and instable conditions, depends on the friction velocity u_*. The values of u_* are given by the "Bulk" scheme of Louis (1979, 1982), previously computed in ACHMT in ARPEGE. The second part, only valid for the instable (convective) cases, i.e. for $1-\delta_{stab} = 1$, and including the term $0.3\ (w_*)^2$, depends on the convective velocity w_* (Deardorff, 1980), defined by :

$$w_* \;=\; (\,\beta\,H_{PBL}\,Q_0\,)^{1/3}\ , \tag{63}$$

$$\text{where}\ Q_0 \;=\; (\overline{w'\theta'})_{surf} \;=\; -\,C_h\,|u_N|\,(\Delta\theta)_N\ . \tag{64}$$

In stable regime, (62) reduces to $\overline{e}_S \ = \ 3.75 \ (u_*)^2$.

The term H_{PBL} in (63) is a diagnostic PBL height. It is computed starting from the ground as the height of the first level where an important decrease in $\overline{e}$ occurs, corresponding to the top of the PBL and the beginning of the stable layers located above the PBL.

The term C_h in (64) is the thermal surface Drag coefficient given by (67). The term $|u_N|$ is the norm of the wind speed at the first upper-air level above the ground. The term $(\Delta\theta)_N$ is the difference in θ between the ground level and the first upper-air level above the ground.

2.9 The "Bulk" formulations in the surface layer

The neutral, dynamical and thermal versions of the surface Drag are denoted by C_{dn},C_d and C_h, respectively. They are computed following the "Bulk" scheme of Louis (1979, 1982) :

$$\begin{aligned}
C_{dn}(z, z_0) &= \left(\frac{0.4}{\log(1 + z/z_0)} \right)^2 , & (65) \\
C_d(R_i, z, z_0) &= f_d(R_i) \ C_{dn}(z, z_0) \ , & (66) \\
C_h(R_i, z, z_0) &= f_h(R_i) \ C_{dn}(z, z_0) \ . & (67)
\end{aligned}$$

The functions $f_d(R_i)$ and $f_h(R_i)$ only depend on the Richardson's number R_i. The coefficient C_h given by (67) is the one used to define Q_0 in (64), in order to compute w_* by (63).

2.10 The vertical mixing and the dissipation term

The dissipation term (5) writes $C_\epsilon \ \overline{e} \ \sqrt{\overline{e}}/L_\epsilon$. It is numerically computed by using either explicit or implicit methods, depending of the value of two coefficients α_{exp} (ADISE in the code) and α_{imp} (ADISI). In the current version, ADISE=-0.5 and ADISI=1.5.

The term $\sqrt{\overline{e}}$ in the numerator of $C_\epsilon \overline{e} \ \sqrt{\overline{e}}/L_\epsilon$ always appears in the explicit form $\sqrt{\overline{e}(t)}$. The two coefficients α_{exp} and α_{imp} determine how the other term $\overline{e}$ is computed, leading to the following discretization for the term $(\overline{e})^{3/2}$, in terms of the time steps t and $t + dt$:

$$\sqrt{\overline{e}(t)} \ [\ 1.5 \ \overline{e}(t + dt) \ - \ 0.5 \ \overline{e}(t) \ ] \ . \qquad (68)$$

The last formulation is obtained when the term $X^{3/2} = (\overline{e})^{3/2}$ is written as the following Taylor expansion (personal communication of V. Masson)

$$\begin{aligned}
X^{3/2} &= (X_0)^{3/2} \ + \ 1.5 \, (X_0)^{1/2} \, (X - X_0) \, , & (69) \\
(\overline{e})^{3/2}(t + dt) &= \overline{e}(t) \sqrt{\overline{e}(t)} \ + \ 1.5 \sqrt{\overline{e}(t)} \ [\, \overline{e}(t + dt) - \overline{e}(t) \,] \, , & (70)
\end{aligned}$$

where $1.5 \, (X_0)^{1/2}$ is the derivative of $X^{3/2}$ at X_0, leading indeed to (68).

2.11 The Top-PBL vertical entrainment: LPBLE=.TRUE.

The properties of stratocumulus clouds are greatly affected by cloud-top entrainment process corresponding to the mixing of free-tropospheric air above the inversion with the turbulent cloudy air. Stratocumulus cloud layers are maintained by an energetic balance between radiative cooling and surface and cloud top entrainment warming, and a moisture balance between moisture fluxes and drying from cloud top entrainment. Cloud top entrainment of dry, warm air from the free troposphere in stratocumuls is strongly aided by the evaporation of cloud water into a thin layer just above cloud, which cools the air parcels in the mixing region, making them sufficiently negatively buoyant to sink into the cloud layer. Strength of cloud top entrainment increases with stratocumulus cloud thickness, as thicker stratocumuli have more cloud-top condensate than thinner ones. Cloud top entrainment mixes in warm, dry air into the boundary layer, raising the cloud base and thereby thinning the cloud layer. A review on stratocumulus can be found there:

$http : //www.atmos.washington.edu/ robwood/teaching/535/StratusStratocumulus_Wood_July22.pdf$
A historical review up to 1986 can be found in Reuter (1986).

Although the importance of entrainment in affecting cloud properties was realized early (Lilly 1968), the parameterization of entrainment for stratocumulus cloud is yet to be fully resolved (Wood 2012; Bretherton and Blossey 2014). The first effort to understand cloud-top entrainment can be found in the classic work by Lilly (1968), in which a budget for turbulence kinetic energy (TKE) was applied to the whole BL. Subsequent treatments of entrainment rate closures considered the TKE budget of the inversion layer (e.g., Tennekes 1973; Tennekes and Driedonks 1981). More recently, substantial theoretical, modeling, and observational work has focused on the application of a simplified TKE budget to relate entrainment rates to vertical velocity variance in the cloud-topped BL and the convective velocity scale w* (e.g., Nicholls 1984; Nicholls and Turton 1986; Bretherton et al. 1999; Lock et al. 2000). The vertical resolution of the large scale models in GCM mode, remains rather coarse. Above an altitude of 1000 to 1500 m height, the layer depths are typically as high as $\Delta z > 200$ to 300 m.

With such coarse vertical resolutions (above the critical value of $(\Delta z)_{crit} \approx 20$ m to 50 m) it is not possible to represent in a realistic way the vertical entrainment in the Stratocumulus. This vertical entrainment occurs at the very top of the PBL, where the dry air above the Top-PBL is mixed into the underlying cloudy air, located below the Top-PBL height.

In order to parameterize this sub-grid process (sub-grid on the vertical), the old ideas of Tenekes (1973), revisited by Nicholls and Turton (1986) or Grenier et Bretherton (2001), and summarized in the note of Grenier (2002), has been implemented in the CBR00 version of ARPEGE.

The activation of top-PBL entrainment is also specific of the climate version (LPBLE=.TRUE) vs the NWP one (LPBLE=.FALSE.) for positive impact in terms of stratocumulus cover in the eastern part of the oceans.

The first step is to compute the inversion height z_{inv}. It could be defined as a jump in, either $\partial\overline{\theta}/\partial z$ or θ, or when the Richardson number R_i becomes greater than a threshold value $(R_i)_{crit}$. It is rather computed in the code starting from the ground as the height of the first level where an important decrease in $\overline{e}$ occurs, corresponding to the top of the PBL and the beginning of the stable layers located above the PBL.

The second step is to replace the exchange coefficients previously computed by (10) to (14) by another coefficient, computed at the inversion, denoted by K_{inv} and verifying

$$\begin{aligned} (\overline{w'\theta_{vl}{}'})(z_{inv}) &= -\, w_{ent}\, \Delta_{inv}(\theta_{vl})\,, && (71) \\ &= -\, K_{inv}\, \frac{\Delta_{inv}(\theta_{vl})}{\Delta_{inv}(z)} \quad \Rightarrow \quad K_{inv} \;=\; w_{ent}\, \Delta_{inv}(z)\,, && (72) \end{aligned}$$

where w_{ent} is called "entrainment velocity" (at the inversion) and where $\Delta_{inv}(\theta_{vl})$ is the buoyancy jump in θ_{vl} and across the layer surrounding this inversion $\Delta_{inv}(z)$.

Following the results of Grenier and Bretherton (2001) and some unpublished ideas of Grenier (Workshop EUROCS at Utrecht, in April 2002), the Top-PBL entrainment exchange coefficient K_{inv} is defined at the inversion level by

$$K_{inv} \quad = \quad w_{ent}\, \Delta_{inv}(z) \;=\; A_{inv}\, \frac{<\overline{e}>^{3/2}}{L_{inv}\; N^2_{inv}} \;. \tag{73}$$

Presently, there is no Prandtl number and all the exchange coefficients (10) to (14) are replaced by the same value (73).

Grenier (2002) has introduced the average value over the whole PBL of the turbulent kinetic energy, denoted by

$$<\overline{e}> \quad = \quad \frac{1}{z_{inv}} \int_0^{z_{inv}} \overline{e}(z)\, dz \;. \tag{74}$$

The unknown parameters in (73) are the Richardson number at the inversion N^2_{inv}, the mixing length at the inversion L_{inv} and an adimensional coefficient A_{inv}. These parameters are defined by:

$$N^2_{inv} \quad = \quad \beta \left[\frac{\Delta_{inv}(\theta_{vl})}{\Delta_{inv}(z)} \right]_{(z=z_{inv})} \;, \tag{75}$$

$$L_{inv} \quad = \quad 0.085\; z_{inv} \;, \tag{76}$$

$$A_{inv} \quad = \quad A_1 \left[1 + A_2\, \frac{L_{v/f}\;\; <\overline{q_c}>}{c_p\; \Delta_{inv}(\theta_{lt})} \right] \;. \tag{77}$$

for positive values of AGREF. And for negative values of AGREF, as proposed by GuÃ©rÃ©my (2011):

$$A_{inv} \quad = \quad -A_1 * AGREF \left[1 - (1 + \frac{1}{AGREF}) * sin^2(f(inversion)) \right] \;. \tag{78}$$

For AGREF <0 (the current choice is AGREF=-0.036), the sinus is zero for strong inversions ($\Delta_{inv}(\theta_{lt}) > 1.5K$ corresponding to stratocumulus case, whereas the sinus equals 1 for smaller inversion (convective case). If the sinus = 0, A_{inv} = - AGREF*AGRE1 (0.2 with the current values, and AGRE1 (=5.5) for the convective cases. Then the ratio for top PBL entrainment between convective clouds and stratocumulus is around 27. In the code, N^2_{inv} is ZBI (acturb.F90) There is a security for the computation of K_{inv} in (73) where, from (75), the division by N^2_{inv} leads to a division by the term $\Delta_{inv}(\theta_{vl})$ which can be tiny or even equal to 0. A minimum value of 1.5 K or so is managed by setting AJBUMIN= 0.005 in the NAMELIST, with $\Delta_{inv}(\theta_{vl}) >$ AJBUMIN $\star\, \theta$.

The part of the entrainment coefficient A_{inv} which is controlled by A_2 is called the "evaporative enhancement of entrainment" in Grenier and Bretherton (2001). It is not used in the current version. The jump in potential temperature $\Delta_{inv}(\theta_{lt})$ corresponds to another moist potential temperature, different from the Lilly one θ_{vl} given in (7). This conservative potential temperature θ_{lt} is a mix of the Betts variables, defined by

$$\theta_{lt} \quad = \quad \overline{\theta}_l\, (1 + 0.608\, \overline{q}_t) \;. \tag{79}$$

The average value over the whole PBL (liquid + solid) $<\overline{q_c}>$ is defined by (ZQCBLK in the code):

$$<\overline{q_c}> \quad = \quad \frac{1}{z_{inv}} \int_0^{z_{inv}} \overline{q_c}(z)\, dz \;. \tag{80}$$

During CMIP6 development, the choice for the numerical values of A_1 and A_2 has been made with analysis of 1D cases and checking the stratocumulus cover on the eastern side of oceans.

It is possible to rewrite K_{inv} given by (73) in a form similar to (13) or (14), in order to define a kind of "stability function" (the term into brackets).

$$K_{inv} \quad = \quad \frac{A_{inv}}{2} \, L_{inv} < \overline{e} >^{1/2} \left[\frac{2 < \overline{e} >}{(L_{inv})^2 \, N_{inv}^2} \right] . \tag{81}$$

The term into brackets is a kind of bulk stability function $[\, (L_D)_{inv}^2 \,/\, (L_{inv})^2 \,]$, similar to the functions ϕ_3 and ψ_3 defined in (13) and (14), where $(L_D)_{inv}^2 \equiv 2 < \overline{e} > / N_{inv}^2$ is the square of a kind of bulk Deardorff length, similar to (60).

2.12 Explicit diffusion of conservative variables: LDIFCEXP=.T.

Note that this option is specific of climate simulations (GuÃ©rÃ©my, 2011) and corresponds to the logical keys LDIFCEXP=.T. and LDIFCONS=.F.. In the NWP version, LDIFCEXP=.F. and LDIFCONS=.T.

The conservative variables are:

$$s_l = s - \frac{L}{C_p} q_l - \frac{L}{C_p} q_i = c_p T_l + gz \qquad\qquad q_t = q_v + q_l + q_i \tag{82}$$

, with $s = C_p T + gz$, the dry static energy, s_l the moist static energy, q_t the total water specific content, and $T_l = T - \frac{L}{C_p} q_l$ (Bougeault, 1982 for example). Using the subgrid distribution of q_l, the definition of T_l and the s variable, it can be shown that the equation for turbulent fluxes of q_l and of conservative variables is of the type:

$$\overline{w' q'_l} = a \lambda_3 F2(Q1) * (\overline{w' q'_t} - \alpha_1 \overline{w' T'_l}) \tag{83}$$

In the code: a corresponds to ZAH variable, $\lambda_3 F2(Q1)$ to PL3F2 and α_1 to ZQSLTLH.

For computing the corresponding turbulent exchange coefficients,K_{ql} (ZGKQLH in the code), K_s (ZGKTAH in the code), and K_{qv} (ZGKQH in the code), the following relations are used: $\frac{K_{ql} \Delta q_l}{\Delta \Phi} = a \lambda_3 F2(Q1) \frac{K_{qt}}{\Delta \Phi} * (\Delta q_t - \alpha_1 \frac{\Delta s_l}{c_p})$ because the turbulent exchange coefficient for moist static energy K_s equals that for the total water content K_{qt}.

For computing K_{qv}, as $q_t = q_v + q_l + q_i$, we have:

$$K_{qv} \Delta qv = K_{qt} \Delta qt - K_{ql} \Delta ql \tag{84}$$

and, for K_s,

$$K_s \Delta s = K_{sl} \Delta sl + L_v K_{ql} \Delta ql \tag{85}$$

The subroutine ACTURB computes the turbulent coefficients of conservative variables (s_l, q_t) and the transformation of model variables (s,q_v and q_l) into conservative variables. As output of the routine, when LDIFCEXP =.TRUE., the turbulent coefficients of the model variables are computed: the generic relation $F_\psi = -K_\psi * \frac{\partial \psi}{\partial z}$ is inverted, starting with ql (equation 83) and then s and q_v with equation 82.

The subroutines ACDIFV1 and ACDIFV2 then compute the turbulent fluxes (implicit resolution) with as input model variables and their turbulent coefficient (if LDIFCEXP=.TRUE.) or the conservative variables and their exchange coefficients for LDIFCONS=.TRUE..

2.13 Enhanced turbulence due to convection, LCVTURB=.TRUE.

At high levels, in presence of convection and when the grid is saturated, the turbulence is generally insufficient. As a consequence, due to the formulation of cloudiness, the grid is too often overcast. There is a need to add an extra source of turbulence in order to decrease the cloudiness. The source of turbulence due to convection could be introduced by induced gravity wave or by a convective R_i (from convective transport of dynamics and thermodynamics). For sake of simplicity a first attempt has been designed using a minimum value of σ_s ($= 5.10^{-5}$ kg/kg) in case of presence of convective condensates and grid saturation above 600 hPa (mean level of beginning of detrainment for deep convection).

3 Eddy-Diffusivity Mass Flux parameterization

The algorithmic treatment of transport is based on the "EDMF" concept and uses an implicit combined resolution. The "EDMF" concept (Hourdin et al 2002, SOares et al 2004, Siebesma 2007) is used in all NWP and climates models in MF (Arome, ARPEGE/ALADIN):

$$\overline{w'\phi'} = -K\frac{\partial\overline{\phi}}{\partial z} + \frac{M_u}{\rho}(\phi_u - \overline{\phi}) \tag{86}$$

with $K = cL_{BL89}\sqrt{TKE}$ and

$$L_{BL89]} \;=\; L_\epsilon \;=\; \left[\,\frac{1}{2}\,\{\,(L_{up})^{-2/3} + (L_{down})^{-2/3}\,\}\,\right]^{-3/2}\;. \tag{87}$$

where L_{up} and L_{down} are computed using dry buoyancy following Bougeault and LacarrÃ¨re (1989). Due to numerical stability problems encountered with EDKF (shallow convection scheme of EDMF type; Pergaud et al., 2009) in ARPEGE with "large time-steps", a common implicit solver for Eddy-Diffusivity and Mass Flux part has been developed, enabling stability for large time-steps.

1.Implicit treatment of the mass flux equations:

$$\left\{\begin{array}{rcl} F_\psi &=& \rho\overline{w'\psi'} = M(\psi_u - \overline{\psi}) \\ \left(\frac{\partial\psi}{\partial t}\right)_{MF} &=& \frac{1}{\rho}\frac{\partial}{\partial z}F_\psi \end{array}\right. \tag{88}$$

The second equation is solved implicitely for $F_\psi = (1-z_i)F_\psi^- + z_iF_\psi^+$,

$$F_\psi^+ = F_\psi^- + \delta F_\psi = F_\psi^- + \frac{\partial F_\psi}{\partial\psi}\delta\psi = F_\psi^- - M(\tilde{\psi^+} - \tilde{\psi^-}) \tag{89}$$

then:

$$\left(\frac{\partial\psi}{\partial t}\right)_{MF} = \frac{1}{\rho}\frac{\partial}{\partial z}(F_\psi^- - \underbrace{z_iM(\tilde{\psi^+} - \tilde{\psi^-})}) \tag{90}$$

with upward increasing j levels and F_ψ, M defined at levels where $\tilde{\psi}(j) = 0.5(\psi(j) + \psi(j-1))$ The second rhs term corresponds to the implicit correction.

We obtain:

$$\left\{\begin{array}{l} \psi^{+}(j)-\psi^{-}(j)= \\ \frac{\Delta t}{\rho\Delta z}(F_{\psi}^{-}(j+1)-F_{\psi}^{-}(j) \\ -z_iM(j+1)(0.5\psi^{+}(j+1)+0.5\psi^{+}(j)-0.5\psi^{-}(j+1)-0.5\psi^{-}(j)) \\ +z_iM(j)(0.5\psi^{+}(j)+0.5\psi^{+}(j-1)-0.5\psi^{-}(j)-0.5\psi^{-}(j-1))) \end{array}\right. \tag{91}$$

Grouping $'+'$ terms in the lhs of the equation the following tridiagonal system is obtained:

$$\left\{\begin{array}{l} \psi^{+}(j+1)(0.5\frac{\Delta t}{\rho\Delta z}z_iM(j+1)) \\ +\psi^{+}(j)(1+0.5\frac{\Delta t}{\rho\Delta z}z_iM(j+1)-0.5\frac{\Delta t}{\rho\Delta z}z_iM(j)) \\ -\psi^{+}(j-1)(0.5\frac{\Delta t}{\rho\Delta z}z_iM(j))=\psi^{-}(j)+\frac{\Delta t}{\rho\Delta z}(F_{\psi}^{-}(j+1)-F_{\psi}^{-}(j)) \\ +0.5\frac{\Delta t}{\rho\Delta z}z_iM(j+1)(\psi^{-}(j+1)+\psi^{-}(j)) \\ -0.5\frac{\Delta t}{\rho\Delta z}z_iM(j)(\psi^{-}(j)+\psi^{-}(j-1)) \end{array}\right. \tag{92}$$

2.Implicit treatment of the Eddy-Diffusivity equation. The Eddy diffusivity equation:

$$\left(\frac{\partial\psi}{\partial t}\right)_{eddy}=-\frac{1}{\rho}\frac{\partial}{\partial z}(k\frac{\partial\psi}{\partial z}) \tag{93}$$

is discretized as follows:

$$\psi^{+}(j)-\psi^{-}(j)=-\frac{\Delta t}{\rho\Delta z(j)}(\frac{k(j+1)}{\Delta z(j+1)}(\psi^{+}(j+1)-\psi^{+}(j))-\frac{k(j)}{\Delta z(j)}(\psi^{+}(j)-\psi^{+}(j-1))) \tag{94}$$

This yields to the simple tridiagonal system:

$$\left\{\begin{array}{l} \psi^{+}(j+1)(\frac{\Delta t}{\rho\Delta z(j)}\frac{k(j+1)}{\Delta z(j+1)}) \\ +\psi^{+}(j)(1-\frac{\Delta t}{\rho\Delta z(j)}(\frac{k(j+1)}{\Delta z(j+1)}+\frac{k(j)}{\Delta z(j)})) \\ +\psi^{+}(j-1)(\frac{\Delta t}{\rho\Delta z(j)}\frac{k(j)}{\Delta z(j)})=\psi^{-}(j) \end{array}\right. \tag{95}$$

3. Common implicit resolution of the EDMF equation The discretization of the full EDMF equation:

$$\left(\frac{\partial\psi}{\partial t}\right)_{edmf}=\frac{1}{\rho}\frac{\partial}{\partial z}(-k\frac{\partial\psi}{\partial z}+M(\psi_u-\overline{\psi})) \tag{96}$$

yields to the following tridiagonal system:

$$\left\{\begin{array}{l} \psi^{+}(j+1)(\frac{\Delta t}{\rho\Delta z(j)}(\frac{k(j+1)}{\Delta z(j+1)}+0.5M(j+1)))+ \\ \psi^{+}(j)(1-\frac{\Delta t}{\rho\Delta z(j)}(\frac{k(j+1)}{\Delta z(j+1)}+\frac{k(j)}{\Delta z(j)}+0.5M(j+1)-0.5M(j)))+ \\ \psi^{+}(j-1)((\frac{\Delta t}{\rho\Delta z(j)}(\frac{k(j)}{\Delta z(j)}+0.5M(j))) \\ =\psi^{-}(j)+\frac{\Delta t}{\rho\Delta z(j)}(\psi^{-}(j+1)-(\psi^{-}(j))+0.5\frac{\Delta t}{\rho\Delta z(j)}M(j+1)(\psi^{-}(j+1)+\psi^{-}(j)) \\ -0.5\frac{\Delta t}{\rho\Delta z(j)}M(j)(\psi^{-}(j)+\psi^{-}(j-1)) \end{array}\right. \tag{97}$$

The value of z_i in the code is 0, meaning that a combined (convection, turbulence) CHECK treatment is used but the coefficient for implicitation of convection equals 0. Sensitivity tests around this choice have been run.

This work is done in subroutines acdifv1/acdifv2.

4 Algorithmics

The monitor of the ARPEGE physics is APLPAR. The prognostic TKE scheme is called by the main subroutine ACTKE, with the TKE variable "PTKE" as input array and the flux of the tendencies "PFTKE" as output array.

The list of subroutines called if LVDIF.AND.LECT=.TRUE. is:

- APLPAR : monitor of the ARPEGE physics
- ACHMTLS : general surface layer computations, PCPS (chaleur massique Ã pression constante de l'air au sol), surface latent heat...
- ACTKE : monitor of the TKE computations ; possible half<->full<->half averages (if LECTFL)
- ACBL89 : computes the upward and downward buoyancy lengths L_{up} and L_{down}, then the mixing length L_m and the dissipation length L_ϵ by (87) and (61) ; computes the stability functions $\phi_3 = \psi_3$ given by (16) and (17). It is assumed that the grid-cell mean liquid and solid cloud water contents are available as input of ACBL89 and ACTURB.
- ACTURB : compute $\overline{e}_S$ by (62) ; computes the exchange coefficients K_m and K_T; computes the moist thermal production term (28) with the help of (48) and (51) computes the grid-cell mean cloud cover $\overline{N}_s$, the grid-cell mean cloud water content $\overline{q_c}_s$ and the subgrid normalized second order flux $F_2(Q_1, A_S)$ with the use of (49) to (51) ; compute the Top-PBL height and the exchange coefficient at the Top-PBL inversion.
- ACEVOLET : numerical computation of the TKE equation (1), with an implicit solver for the vertical diffusion (25), with $\alpha_{imp} = 1.5$ and $\alpha_{exp} = -0.5$ given by (68). The dissipation term (5) is computed with the help of the same (1.5 ; −0.5) implicit scheme. The moist thermal production term is an input of the subroutine (computed in ACTURB). The dynamical production term is computed in this subroutine.

Note on the interaction with the radiation scheme.

The "large scale" variables $\overline{N}_s$ and $\overline{q_c}_s$ are computed in (49) and (50), in the turbulent scheme. They correspond to the grid-cell averaged values for the Cloud Cover and the Condensed Cloud Water. They are added to the same informations coming from the PCMT convections scheme, with some suitable assumptions for the overlapping of the cloud layers on the vertical (Maximum overlap for LGPCMT + LRNUMX).

5 Logical keys and main parameters in the current version

Logical keys

&NAMPHY

LDIFCEXP= .TRUE. : explicit diffusion of conservative variables

LDIFCONS= .FALSE. : use the Betts conservative variables (LDIFCONS and LDIFCEXP work together: if one is TRUE, the other is False)

LDISTUR= .TRUE. : alternative discretization in ACTURB (cf JEF)

LECT= .TRUE. : the main switch for the turbulent scheme

LECTFL= .FALSE. : no half<->full<->half averages in ACTKE

LECTQ1= .TRUE. : for limitation of Q1 in acturb

LNEBECT= .TRUE. : use of the Bougeault-Bechtod values (F_0, F_1, F_2)

LNSMLIS= .TRUE. : key for activating smoothing of Smith's cloudiness

LPBLE= .TRUE. : use of the Top-PBL vertical entrainment

Parameters **&NAMPHY0**

ACBRPHIM= 2.2 : a maximum value for (ϕ_3). ZPHI3MAX (in acbl89) is (1-ACBRPHIM)/ACBRPHIM.

ADISE= −0.5 : Dissipative parameter for TKE (explicit part)

ADISI= 1.5 : Dissipative parameter for TKE (implicit part)

AGREF= −0.036 Parameter for Top-PBL entrainment; $A_{inv} = AGRE1 * AGREF$

AGRE1= 5.5 : first Sc value A_1 – Top-PBL entrainment ; see the Table (**??**)

AGRE2= 0. : second Sc value A_2 – Top-PBL entrainment ; see the Table (**??**)

AJBUMIN= 0.00525 : a threshold value to avoid division by zero

AKN= 0.126 : CCH02 value for ($K_m/L_m/\sqrt{\overline{e}}$) – see the Table (**??**)

ALD= 1.18 : CCH02 value for ($1/C_\epsilon$) – see the Table (**??**)

ALMAV= 20. : momentum exchange asymptotic mixing length

ALMAVE= 10. : an asymptotic and minimum value for L_m

ALPHAE= 2.7 : CCH02 value for (K_e/K_m) – see the Table (**??**)

ALPHAT= 1.13 : CCH02 value for ($K_T/K_m/\phi_3$) – see the Table (**??**)

ARSB2= 0.833 : the coefficient $1/\sqrt{C_{\epsilon\theta}}$ in (40)

ARSC1= 0.1389 : parameter for ϕ_{3min}

ECTMIN= $1.0E-6$: a minimum value for ($\overline{e}$)

ECTMAX= 100 : a maximum value for ($\overline{e}$)

Top-PBL entrainment ; see the section (2.11)

STTBMIN= 1.73 : parameter for Q1 limitation (min)

9

Microphysics parameterizations scheme

1 Introduction

There exists two main types of microphysics scheme, Bulk and Bin schemes:

- spectral (Bin) microphysics aim to compute microphysics as accurately and generally as possible: it divides microphysical particles into bins for different sizes and compute evolution of each bin separately; in that case the particle size distribution (PSD, see below) is an output not an input, but these schemes are much too expensive for operational use.
- Bulk schemes are used to calculate microphysics with a semi-empirical PSD.
- There exist also hybrid schemes which aim the accuracy of bin schemes with efficiency of bulk schemes (for example, Onishi and Takahashi (2012) who used bin for warm processes and bulk for ice) but they remain too expensive for NWP and Climate models. There exist also bin-emulating schemes which calculate rates offline with complex bin scheme and develop lookup tables (Verlinde, 1990).

Microphysics processes are complex, but can be modelled very effectively with a bin approach, which is only a research tool, much too expensive for use in NWP or Climates.

Note also the definition of microphysical moments: for PSDs $f(m)$, m being the particle mass, the kth moment is:

$$M^{(k)} = \int_0^\infty m^k f(m) dm \tag{1}$$

It is possible to distinguish three types of schemes according to the "k" value:

- One-moment schemes (1M) correspond to k=1 (mass)
- Two-moments schemes (2M) correspond to k=0 and 1 (number concentration and mass)
- Three-moments schemes (3M) correspond to k=0 and 1 and 2 (number concentration, mass and radar reflectivity)

A partial history of schemes with main steps and papers is proposed below. Note the increase in complexity and reducing gap between bulk and bin methods:

- Kessler (1969) : first warm rain bulk parameterization
- Lin et al. (1983) : 1M, includes hail
- Cotton et al. (1986) : first bin parameterization (RAMS)
- Murakami (1990) : 1 M, snow includes crystals and aggregates
- Verlinde et al.(1990) : development of lookup tables
- Ferrier (1993) : 2M for ice and precipitating species
- Cohard and Pinty (2000) : 2M for warm microphysics
- Saleeby and Cotton (2004) : 2M bin-emulating bulk scheme. Fully interactive with prognostic CCN and IN aerosol schemes
- Morrison et al. (2005) : 2M scheme for droplets, cloud ice, rain and snow
- Milbrandt and Yau (2005) : 3M scheme for hail
- Lim and Hong (2010) : WRF 2M 6 classes; prognostic treatment of cloud condensation nuclei

Processes to account for are:

- Droplet nucleation (condensation)
- Droplet growth by vapour diffusion
- Collisions between droplets and between hydrometeors
- Sedimentation (differential motion)
- Freezing/Melting
- Ice multiplication
- Raindrop breakup
- Effects of aerosol on all these

Much "large-scale" rain originates from melting ice. Much convective rain and some large-scale rain originates from collision and coalescence of cloud droplets.

In bulk schemes different collisions can be described:

- self-collection (X+X–>X): droplet + droplet –> droplet
- self-collection: rain + rain —> rain
- autoconversion: droplet + droplet —> rain
- accretion: droplet + rain —> rain

Self-collection does occur due to fall speed differences between particles of different sizes but of same type.

2 The microphysics scheme

2.1 History of the Lopez Scheme

Mainly developed for variational assimilation of cloud and precipitation observations, the Lopez' scheme (2002) describes both the large-scale cloud water content and the large-scale precipitations (liquid and solid), by using prognostic equations and describing several physical processes acting between the different species. Since then, the Lopez has been tested, modified and improved, for both Climate and NWP mode (see Bouyssel, 2005), also in the ALADIN-ALARO team (see Gerard, 2005). This is also the microphysics scheme used in the current NWP ARPEGE model. This scheme has been deeply modified by Bouteloup et al. (2010) introducing a new statistical sedimentation scheme. Since 2009, this scheme has been used and tested in different research versions of the climate model, with a convection diagnostic scheme (Bougeault, 1985) or a prognostic one (PCMT, Piriou, 2007, GuÃ©rÃ©my, 2011). In the second case, the Lopez scheme is called twice, one time by the APLPAR routine for the large-scale part, and one time by the ACPCMT routine for the convective part.

2.2 Which processes are described in the scheme?

The physical processes described by the scheme of Lopez (2002) are based on a “bulk” assumption. They are close to the one described in Fowler et al. (1996).

- The cloud water content and the precipitations (4 prognostic variables, either for large-scale or for convective part = 8 prognostic variables) are supposed to be pronostic variables, available as input of the scheme.
- The cloud water content (liquid and ice) and the precipitations (rain and snow) form two separate distributions of dropplets. The smaller ones are the suspended cloud condensate, whereas the larger ones are precipitating species.
- The condensation/deposition process acts as a source for the liquid/ice condensed cloud water.
- The evaporation/sublimation process acts as sinks for the rain/snow precipitation content
- The condensed cloud water (liquid and ice) can transform into the precipitation content species (rain and snow) via the processes of auto-conversion and collections (accretion + aggregation + riming).
- The precipitation content species (rain and snow) fall with different fall speeds. The sedimentation processes is that of Bouteloup et al. (2010), described in chapter PCMT.
- The Freezing-Melting-Bergeron processes between cloud liquid water and cloud ice are not explicitly described. Only the snow melting is computed (between snow and rain). The detrainment from the convection schemes (neither shallow nor deep) are not taken into account.

2.3 "Bulk" equations

The either grid-averaged or convective condensed water species, input of the scheme, will be denoted below by

$\overline{q_l}$	liquid cloud water content
$\overline{q_i}$	ice cloud water content
$\overline{q_r}$	rain precipitation content
$\overline{q_s}$	snow precipitation content

The system of four prognostic equations corresponding to the physical processes described in the section (**??**) can be written as

$$\left\{\begin{array}{lcl} \partial(\overline{q_l})/\partial t &=& +C_l - A_l \quad - COL_{(l/r)} \quad - COL_{(l/s)} \\ \partial(\overline{q_i})/\partial t &=& +C_i \quad - A_i \quad - COL_{(i/s)} \\ \partial(\overline{q_r})/\partial t &=& -E_r + A_l \quad + COL_{(l/r)} \quad - F_r \\ \partial(\overline{q_s})/\partial t &=& -E_s \quad + A_i \quad + COL_{(i/s)} + COL_{(l/s)} \quad - F_s \end{array}\right. \tag{2}$$

The condensation processes C_l and C_i act as source terms of the cloud water contents (liquid and solid). The evaporation and the falling processes, i.e. E_r, E_s, F_r and F_s, act as sink terms for the precipitation contents (rain and snow). The auto-conversion terms A_l and A_i and the collection terms $COL_{(l/r)}$, $COL_{(i/s)}$ and $COL_{(l/r)}$ act as conversion terms, transforming one species into another.

The three collection processes parameterized in the scheme of Lopez are caused by the differential (constant) fall speed between the cloud species and the precipitating ones. They represent the capture of the cloud species by the precipitations.

$COL_{(l/r)}$	accretion	collection of cloud liquid water by rain
$COL_{(i/s)}$	aggregation	collection of cloud ice by snow
$COL_{(l/s)}$	riming	collection of cloud liquid water by snow

The fourth collection (of cloud ice by rain) is not parameterized. Indeed, the cloud ice regions are usually located above the rain regions.

2.4 The condensation/evaporation process

The large-scale cloud-cover and condensed cloud water are computed in the prognostic TKE scheme, using the Bougeault functions (LNEBECT=TRUE). If the statistical cloud water content is denoted by $(\overline{q_c})_{stat}$, the partitioning of the liquid part $(\overline{q_l})_{stat}$ and the solid part $(\overline{q_i})_{stat}$ is obtained with

$$\delta_{ice} \;=\; \begin{cases} 1 - \exp\left[\,(T - T_f)^2/(2\,\Delta T)^2\,\right] & \text{for } \; T < T_f \;, \\ 0 & \text{otherwise} \,. \end{cases} \tag{3}$$

The value of δ_{ice} is given by the function FONICE(T), contained in the header fctdoi.h, with ΔT = RDT∗RDTFAC, RDT= 11.82 K and RDTFAC= 0.5, leading to ΔT = 5.91 K.

The fluxes of condensation/evaporation are denoted by PFCSQL for the liquid cloud water and by PFCSQN for the solid cloud water. They are computed in ACPLUIZ, with $(\overline{q_c})_{stat}$ as input, following

$$g\,\frac{\Delta\,(\mathrm{PFCSQL})}{\Delta p} \;=\; \frac{(1-\delta_{ice})\;*\;(\overline{q_c})_{stat} - \overline{q_l}}{\Delta t}\;, \tag{4}$$

$$g\,\frac{\Delta\,(\mathrm{PFCSQN})}{\Delta p} \;=\; \frac{\delta_{ice}\;*\;(\overline{q_c})_{stat} - \overline{q_i}}{\Delta t}\;. \tag{5}$$

2.5 The auto-conversion processes

Rain is initiated in liquid water clouds by collision and coalescence of cloud droplets wherein larger droplets with higher settling velocities collect smaller droplets and become embryonic raindrops. This so-called autoconversion process is usually the dominant process that leads to the formation of drizzle in stratiform clouds.

Kessler (1969) proposed a simple parametrization that linearly relates the autoconversion rate to the cloud liquid water content, and this parametrization is widely used in cloud-related modeling studies because of its simplicity. This simple parametrization supposes that the autoconversion rate is a function of not only of the liquid water content but also the cloud droplet number concentration and the spectral shape of the cloud droplet size distribution (Berry, 1967; Berry and Reinhardt 1974).

Without loss of generality, all of the Kessler-type parametrizations can be written as(Liu et al., 2003):

$$P = cLH(y - y_c) \tag{6}$$

where P is the autoconversion rate (in $gcm^{-3}s^{-1}$, c is an empirial coefficient (in s^{-1}) and L is the cloud liquid water content (in gcm^{-3}). The Heaviside step function $H(y - y_c)$ is introduced to describe a threshold y_c below which the autoconversion is negligibly small. The meaning of y is different in different parametrizations. In our scheme, it represents the cloud liquid water content, as in the original Kessler parametrization. Liu et al. (2003) propose a reexamination of typical Kessler-type parametrizations and show that application of the generalized mean value theorem for integrals to the general equation for the autoconversion rate provides a rigourous theoretical basis for Kessler's formulation.

As shown in eq. 6, the auto-conversion processes start to occur when the values of the suspended condensed water content are larger than some threshold value q_{crit}. It is important to think in terms of the in-cloud values (close to the saturating state), and not with the grid-cell averages (often far from the saturating state).

Let us denote by $\overline{q_c}$ the grid-cell average of the condensed water content, and by $\overline{q_c^*}$ the corresponding in-cloud value, obtained from the cloud cover $N_s = \text{Max}[\ \varepsilon_N\ ;\ N_s\ ]$ as

$$\overline{q_c^*} \quad \equiv \quad \text{Max}\left[\ 0\ ;\ \frac{\overline{q_c}}{N_s}\ \right] . \tag{7}$$

The auto-conversion process can be represented as the in-cloud conversion rate A_c^* from one of the (liquid or solid) condensed water in-cloud value $\overline{q_c^*}$ to the corresponding (liquid or solid) precipitating in-cloud value $\overline{q_p^*}$, with the Kessler (1969) assumption, leading to

$$\left(\frac{\partial \overline{q_c^*}}{\partial t}\right)_{auto} \quad \equiv \quad -\ A_c^*\ , \tag{8}$$

$$\left(\frac{\partial \overline{q_c^*}}{\partial t}\right)_{auto} \quad \equiv \quad -\ \frac{\overline{q_c^*}\ -\ q_{crit}}{\tau_c}\ . \tag{9}$$

The threshold value q_{crit} and the time scale τ_c are supposed to be two constant terms. They correspond to the physical assumption that the dropplets smaller than a given radius remain suspended cloud condensate, whereas the dropplets larger than a given radius are becoming precipitating species, with a time scale τ_c. The threshold used in this "bulk" scheme is not a critical radius, but a critical specific content q_{crit} for the suspended cloud condensate.

The threshold values for q_{crit} are set from the NAMELIST of ARPEGE. They are not the same for the liquid and the solid species.

$$(q_l)_{crit} \quad = \quad \text{RQLCR}\ , \tag{10}$$

$$(q_i)_{crit}(T) \quad \in \quad [\ \text{RQICRMIN}\ ;\ \text{RQICRMAX}\ ]\ . \tag{11}$$

The threshold for the liquid water content is a constant, whereas for the solid water (ice), the formulation is more complex, with a variation between the two extremum values $(q_i)_{crit}^{min} = \text{RQICRMIN}$ and $(q_i)_{crit}^{max} = \text{RQICRMAX}$. Indeed, in order to allow the auto-conversion of ice even at very low temperature, $(q_i)_{crit}(T)$ is assumed to decrease with temperature, according to the formulae (A.1) of Lopez (2002):

$$(q_i)_{crit}(T) \quad = \quad (q_i)_{crit}^{max}\ -\ 0.5\left[\ (q_i)_{crit}^{max} - (q_i)_{crit}^{min}\right]\ \{\ 1\ +\ \tanh\left[\ \alpha\ (T - T_f) + \beta\ \right]\ \} \tag{12}$$

The coefficients α and β are two constant terms in Lopez (2002), see his Appendix-D where $\alpha = -0.1572$ K^{-1} and $\beta = -4.9632$. In the ARPEGE code, due to additional tunings made in the NWP team, α and β depend on the temperatures $T_1 = \text{RQICRT1}$ and $T_2 = \text{RQICRT2}$, both available in the NAMELIST:

$$\gamma_1 \quad = \quad -1\ +\ 2\ \frac{(q_i)_{crit}^{max}\ (1 - 0.999)}{(q_i)_{crit}^{max}\ -\ (q_i)_{crit}^{min}}\ , \tag{13}$$

$$\gamma_2 \quad = \quad -1\ +\ 2\ \frac{(q_i)_{crit}^{max}\ -\ 1.5\ (q_i)_{crit}^{min}}{(q_i)_{crit}^{max}\ -\ (q_i)_{crit}^{min}}\ , \tag{14}$$

$$\gamma_3 \quad = \quad 0.5\ \log\left[\ \text{abs}\ \left(\ \frac{1\ +\ \gamma_1}{1\ -\ \gamma_1}\ \right)\right]\ , \tag{15}$$

$$\gamma_4 \quad = \quad 0.5\ \log\left[\ \text{abs}\ \left(\ \frac{1\ +\ \gamma_2}{1\ -\ \gamma_2}\ \right)\right]\ . \tag{16}$$

From (13) to (16), α and β are defined by

$$\begin{aligned}\alpha &= \frac{\gamma_3 \; - \; \gamma_4}{T_2 \; - \; T_1} \;, && (17)\\ \beta &= \gamma_3 \; - \; T_2 \, \alpha \,. && (18)\end{aligned}$$

Over the ice-shell sea regions or over the snow-covered land areas, there are fewer Cloud Condensation Nuclei (CNC) than over the open-sea or free land areas, where more aerosol are available. A first attempt to take into account these differential icy/warm properties is made in the ACMICRO subroutine, by computing the term ZFACICE and by multiplying $(q_i)_{crit}(T)$ by 1−ZFACICE∗(1−RQICRSN).

The term ZFACICE is equal to 0 over the open-sea and free land areas, equal to 1 over the icy regions. For RQICRSN = 0.5 (available in the NAMELIST), $(q_i)_{crit}(T)$ is not modified over the open-sea and free land areas, whereas it is divided by a factor 2 over the icy regions. For RQICRSN = 1, $(q_i)_{crit}(T)$ is not modified over the icy regions.

The in-cloud auto-conversion rate A^*_c is computed with both the finite difference (19) plus the analytic method (20). These schemes are obtained by integrating (8) and (9) from the time t with the value $\overline{q_c}^{\,*(-)}$ to the time $t + \Delta t$ with the value $\overline{q_c}^{\,*(+)}$, considering q_{crit} and τ_c as two constant terms.

It results the two following formulations

$$\begin{aligned}A^*_c \;&\approx\; - \, \frac{\overline{q_c}^{\,*(+)} \; - \; \overline{q_c}^{\,*(-)}}{\Delta t} \;, && (19)\\ \frac{\overline{q_c}^{\,*(+)} \; - \; q_{crit}}{\overline{q_c}^{\,*(-)} \; - \; q_{crit}} \;&\approx\; \exp\left(- \, \frac{\Delta t}{\tau_c} \right) \,. && (20)\end{aligned}$$

They respectively correspond to discrete solutions of (8) and (9).

From (19) and (20), the auto-conversion rates for the liquid and the solid (ice) water content write

$$\begin{aligned}A^*_l \;&\approx\; \left[1 \; - \; \exp\left(- \, \frac{\Delta t}{\tau_l} \right) \right] \left[\frac{\overline{q_c}^{\,*(-)} \; - \; (q_l)_{crit}}{\Delta t} \right] , && (21)\\ A^*_i \;&\approx\; \left[1 \; - \; \exp\left(- \, \frac{\Delta t}{\tau_i} \right) \right] \left[\frac{\overline{q_c}^{\,*(-)} \; - \; (q_i)_{crit}(T)}{\Delta t} \right] . && (22)\end{aligned}$$

The terms A^*_l and A^*_i are denoted by ZDUM in ACMICRO.

The threshold values $(q_l)_{crit}$ and $(q_i)_{crit}(T)$ are given by (10) and (12). The two time constant terms τ_l and τ_i are denoted by ZCAUT in ACMICRO, they write

$$\begin{aligned}1/\tau_l &= \text{RAUTEFR} \,, && (23)\\ 1/\tau_i &= \text{RAUTEFS} \, * \, \exp[\, - \, (\beta)_{ice} \, (T - T_f) \,] \,, && (24)\end{aligned}$$

with RAUTEFR and RAUTEFS available in the NAMELIST.

The exponential term in (24) is the temperature dependant ice conversion efficiency of Pruppacher and Klett (1998), with the coefficient $(\beta)_{ice}$ = RAUTSBET = 0.025 available in the NAMELIST.

The final grid-cell average or convective auto-conversion rate for the liquid and solid species are computed from (21) and (22) by inverting (7) and by multiplying by the large-scale cloud cover N_s, for the large-scale contribution (ZNEBS in the code), and convective cloudiness ($ZNEBC_MICRO$ in acpcmt.F90).

$$\begin{aligned}A_l &= \text{Max} \{ \, 0 \, ; \, A^*_l \, * \, N_s \, \} \;, && (25)\\ A_i &= \text{Max} \{ \, 0 \, ; \, A^*_i \, * \, N_s \, \} \;. && (26)\end{aligned}$$

The terms A_l = PAUTOL and A_i = PAUTOI are available as output of ACMICRO.

2.6 The distribution of particle - Fall-velocities.

The classical equations for the collection of suspended cloud species by the precipitating ones are derived in Fowler et al. (1996) by using the Marshall and Palmer (1948) distribution for the rain and the Gunn and Marhall (1958) distribution for the snow. In the paper of Lopez (2002), the same distribution of Marshall and Palmer (1948) is used for the rain, with the fall speed given by Sachidananda and Zrnić (1986) and Foote and Du Toit (1969). For the ice particles, the ideas of Houze et al. (1979) and Cox (1988) have been taken into account.

For the rain, the assumed spectra of particle number $N_r(D)$, mass $M_r(D)$ and fall velocity $V_r(D)$ express in terms of the particle diameter D as follows:

$$\begin{aligned} N_r(D) &= N_{0r}\ \exp(-\,\lambda_r\ D)\ , \qquad (27)\\ M_r(D) &= \rho_w\ \frac{\pi\ D^3}{6}\ , \qquad (28)\\ V_r(D) &= \nu_1\ \left(\frac{\rho_0}{\rho}\right)^{0.4}\ D^{\nu_2}\ . \qquad (29)\end{aligned}$$

with N_{0r} the intercept parameter in raindrop size distribution ($N_{0r} = 8.10^6 m^{-4}$) , λ_r the slope parameter of the raindrop size distribution, ρ_w the liquid water density (ρ_w= 1000 kgm^{-3}) and ν_1 a coefficient in rain fall-speed distribution (ν_1 = 377.8 $m^{1/3}s^{-1}$). The slope λ_r of the Marshall and Palmer distribution is related to the grid-cell average rain content $\overline{q_r}$ by

$$\lambda_r \;=\; \left(\frac{\pi\ \rho_w\ N_{0r}}{\rho\ \overline{q_r}}\right)^{1/4}\ . \qquad (30)$$

For the snow, the equivalent spectra of particle number $N_s(D)$, mass $M_s(D)$ and fall velocity $V_s(D)$ express as follows (with D the maximum dimension of ice particles):

$$\begin{aligned} N_s(D) &= N_{0s}(T)\ \exp(-\,\lambda_s\ D)\ , \qquad (31)\\ M_s(D) &= \sigma_1\ D^{\sigma_2}\ , \qquad (32)\\ V_s(D) &= \tau_1\ \left(\frac{\rho_0}{\rho}\right)^{0.4}\ D^{\tau_2}\ . \qquad (33)\end{aligned}$$

with $N_{0s(T)}$ the intercept parameter in ice particle size distribution ($N_{0s(T)} = 2.10^6 exp(-0.1222(T-T_f))m^{-4}$, Houze et al. (1979)) , λ_s the slope parameter of the ice particle size distribution, σ_1 a coefficient in ice-mass distribution (σ_1 = 0.069 kgm^{-2}), τ_1 a coefficient in ice fall-speed distribution (τ_1 = 21 $m^{1/2}s^{-1}$) and ν_2 the exponent in snow fall-speed distribution ($\nu_2 = 2/3$). The mass-diameter relationship $M_s(D)$ is given by Cox (1988).

The slope λ_s of the distribution is related to the snow content $\overline{q_s}$ by:

$$\lambda_s \;=\; \left(\frac{\Gamma(1+\sigma_2)\ N_{0s}(T)\ \sigma_1}{\rho\ \overline{q_s}}\right)^{1/(1+\sigma_2)}\ . \qquad (34)$$

where $\sigma_2 = 2$, and the Gamma function $\Gamma(z)$ is defined for any real z by

$$\Gamma(z) \;=\; \int_0^{+\infty} t^{z-1} \exp(-t)\, dt, \qquad (35)$$

with $\Gamma(1) = 1$ and the usual properties $\Gamma(z+1) = z\,\Gamma(z)$, leading to $\Gamma(n+1) = n\,!$ valid for any positive integer n.

The mass-weighted average fall speed of particles can be computed as:

$$\overline{V} = \frac{\int_0^\infty N(D)M(D)V(D)dD}{\int_0^\infty N(D)M(D)dD} \tag{36}$$

and the derived associated fall speed of particles $\overline{V_r}$ for rain and $\overline{V_s}$ for snow should be given by

$$\overline{V_r} \;=\; 17.116 \;*\; \left(\frac{\rho_0}{\rho}\right)^{0.4} \;(\rho\,\overline{q_r})^{1/6} \;, \tag{37}$$

$$\overline{V_s} \;=\; 4.323 \;*\; \exp\left[\,0.0204\,(T - T_f)\,\right] \left(\frac{\rho_0}{\rho}\right)^{0.4} \;(\rho\,\overline{q_s})^{1/6} \;. \tag{38}$$

These expressions are very similar to the ones derived by Heymsfield (1977).

During the development of CMIP6 ARPEGE model, SCM and GCM tests of these formulations have been done with rather positive impacts in 1D, but detrimental ones in 3D. So constant values are used as in CMIP5: $\overline{V_r} =$ TFVR and $\overline{V_s} =$ TFVS are set equal to some "bulk" constant values, with both TFVR ≈ 5 m s^{-1} and TFVS ≈ 1 m s^{-1} available in the NAMELIST.

2.7 The collection processes

The distributions (27) to (29) for the rain spectra and (31) to (33) for the snow spectra lead the three collection rates (39) to (41) for $COL_{(l/r)}$, $COL_{(i/s)}$ and $COL_{(l/s)}$.

$$COL_{(l/r)} \;\equiv\; \text{accretion} \;=\; K_{\text{acc}}^{(l/r)}\; E_{\text{acc}}^{(l/r)}\; \overline{q_l}\; \overline{q_r}\;, \tag{39}$$

$$COL_{(i/s)} \;\equiv\; \text{aggregation} \;=\; K_{\text{agg}}^{(i/s)}\; E_{\text{agg}}^{(i/s)}(T)\; \overline{q_i}\; \overline{q_s}\;, \tag{40}$$

$$COL_{(l/s)} \;\equiv\; \text{riming} \;=\; K_{\text{rim}}^{(l/s)}\; E_{\text{rim}}^{(l/s)}\; \overline{q_l}\; \overline{q_s}\;. \tag{41}$$

They express in terms of the product of the 3 collection coefficients (K), the 3 collection efficiencies (E) and appropriate couples of grid-cell mean water species content, i.e. $\overline{q_l}\,\overline{q_r}$ or $\overline{q_i}\,\overline{q_s}$ or $\overline{q_l}\,\overline{q_s}$.

The collection efficiencies write

$$E_{\text{acc}}^{(l/r)} \;=\; \text{RACCEF}\,, \tag{42}$$

$$E_{\text{agg}}^{(i/s)}(T) \;=\; \text{RAGGEF} \;*\; \exp\left[\,-\,0.025\,(T - T_f)\,\right]\;, \tag{43}$$

$$E_{\text{rim}}^{(l/s)} \;=\; \text{RRIMEF}\,, \tag{44}$$

with RACCEF, RAGGEF and RRIMEF three constants, all available in the NAMELIST.

The collection coefficients write

$$K_{\text{acc}}^{(l/r)} \;=\; \frac{12.695\;\;\nu_1\;\;\Gamma(3+\nu_2)}{4\,\rho_w} \left(\frac{\rho_0}{\rho}\right)^{0.4} \;\approx\; 4.8108 \left(\frac{\rho_0}{\rho}\right)^{0.4}\,, \tag{45}$$

$$K_{\text{agg}}^{(i/s)} \;=\; K_{\text{rim}}^{(l/s)} \;=\; \frac{0.0485\;\;\tau_1\;\;\pi\;\;\Gamma(3+\tau_2)}{4\;\;\sigma_1\;\;[\,\Gamma(1+\sigma_2)\,]^{\frac{3+\tau_2}{1+\sigma_2}}} \left(\frac{\rho_0}{\rho}\right)^{0.4} \;\approx\; 17.1623 \left(\frac{\rho_0}{\rho}\right)^{0.4} \tag{46}$$

The numerical values 4.8108 and 17.1623 are the same as the values given in Lopez (2002, Appendix-D), providing that $\nu_1 = 377.8$ m$^{1/3}$ s^{-1}, $\nu_2 = 2/3$, $\tau_1 = 21$ m$^{1/2}$ s^{-1}, $\tau_2 = 0.5$, $\sigma_1 = 0.069$ kg m^{-2}, $\sigma_2 = 2$, $\rho_w = 1000$ kg m^{-3} and $\rho_0 = 1.2$ kg m^{-3}.

In the subroutine ADVPRCS of the ARPEGE code, the coefficients ZCOEFF1, ZCOEFF2 and ZCOEFF2b correspond to the products of the collection coefficients by the collection efficiencies for accretion, riming and aggregation resp.

The three collection tendencies are computed with an analytic scheme applied to the terms $\overline{q_l}$ and $\overline{q_i}$. As an exemple, let us derive the result for the accretion term. The equivalent of the auto-conversion results (8), (9), (20) and (21) write

$$\left(\frac{\partial \overline{q_l}}{\partial t}\right)_{\text{acc}} \approx \frac{\overline{q_l}^{\,(+)} - \overline{q_l}^{\,(-)}}{\Delta t} = -\,\frac{[\,\Delta \overline{q_r}\,]_{\text{acc}}}{\Delta t}\,, \tag{47}$$

$$\left(\frac{\partial \overline{q_l}}{\partial t}\right)_{\text{acc}} \equiv -\,K^{(l/r)}_{\text{acc}}\;E^{(l/r)}_{\text{acc}}\;\overline{q_l}\;\overline{q_r}\,, \tag{48}$$

$$\Rightarrow\quad \left(\frac{\partial \ln(\,\overline{q_l}\,)}{\partial t}\right)_{\text{acc}} = -\,K^{(l/r)}_{\text{acc}}\;E^{(l/r)}_{\text{acc}}\;\overline{q_r}\,, \tag{49}$$

$$\text{and so :}\quad \frac{\overline{q_l}^{\,(+)}}{\overline{q_l}^{\,(-)}} \approx \exp\left[\,-\,K^{(l/r)}_{\text{acc}}\;E^{(l/r)}_{\text{acc}}\;\overline{q_r}^{\,(-)}\;\Delta t\,\right]. \tag{50}$$

Since the collection processes correspond to conversion terms from the suspended cloud condensate $\overline{q_l}$ or $\overline{q_i}$ into the precipitating species $\overline{q_r}$ or $\overline{q_s}$, the following properties holds

$$[\,\Delta\overline{q_r}\,]_{\text{acc}} = \left[\,\overline{q_r}^{\,(+)} - \overline{q_r}^{\,(-)}\,\right]_{\text{acc}} = -\,[\,\Delta\overline{q_l}\,]_{\text{acc}} = -\,\left[\,\overline{q_l}^{\,(+)} - \overline{q_l}^{\,(-)}\,\right]_{\text{acc}}, \tag{51}$$

$$[\,\Delta\overline{q_s}\,]_{\text{agg}} = \left[\,\overline{q_s}^{\,(+)} - \overline{q_s}^{\,(-)}\,\right]_{\text{agg}} = -\,[\,\Delta\overline{q_i}\,]_{\text{agg}} = -\,\left[\,\overline{q_i}^{\,(+)} - \overline{q_i}^{\,(-)}\,\right]_{\text{agg}}, \tag{52}$$

$$[\,\Delta\overline{q_s}\,]_{\text{rim}} = \left[\,\overline{q_s}^{\,(+)} - \overline{q_s}^{\,(-)}\,\right]_{\text{rim}} = -\,[\,\Delta\overline{q_l}\,]_{\text{rim}} = -\,\left[\,\overline{q_l}^{\,(+)} - \overline{q_l}^{\,(-)}\,\right]_{\text{rim}}. \tag{53}$$

From (47) and (50), the final formulae for the decrease in $\overline{q_l}$ due to the accretion process, i.e. $-[\,\Delta\overline{q_l}\,]_{\text{acc}}$, is exacty equal to the corresponding opposite increase in $\overline{q_r}$, i.e. $[\,\Delta\overline{q_r}\,]_{\text{acc}}$. The impact of the accretion process is finally computed as a change in $\overline{q_r}$, given by (54)

$$[\,\Delta\overline{q_r}\,]_{\text{acc}} \equiv \left[\,\overline{q_r}^{\,(+)} - \overline{q_r}^{\,(-)}\,\right]_{\text{acc}} = \overline{q_l}^{\,(-)}\left\{1 - \exp\left[\,-\,K^{(l/r)}_{\text{acc}}\;E^{(l/r)}_{\text{acc}}\;\overline{q_r}^{\,(-)}\;\Delta t\,\right]\right\}, \tag{54}$$

$$[\,\Delta\overline{q_s}\,]_{\text{agg}} \equiv \left[\,\overline{q_s}^{\,(+)} - \overline{q_s}^{\,(-)}\,\right]_{\text{agg}} = \overline{q_i}^{\,(-)}\left\{1 - \exp\left[\,-\,K^{(i/s)}_{\text{agg}}\;E^{(i/s)}_{\text{agg}}\;\overline{q_s}^{\,(-)}\;\Delta t\,\right]\right\}, \tag{55}$$

$$[\,\Delta\overline{q_s}\,]_{\text{rim}} \equiv \left[\,\overline{q_s}^{\,(+)} - \overline{q_s}^{\,(-)}\,\right]_{\text{rim}} = \overline{q_l}^{\,(-)}\left\{1 - \exp\left[\,-\,K^{(l/s)}_{\text{rim}}\;E^{(l/s)}_{\text{rim}}\;\overline{q_s}^{\,(-)}\;\Delta t\,\right]\right\}. \tag{56}$$

The equivalent formulas for the aggregation and the riming processes are given in (55) and (56), with conversion of $\overline{q_l}$ or $\overline{q_i}$ into $\overline{q_s}$, generating positive values of $[\,\Delta\overline{q_s}\,]$.

From the general set of equations (2), and from (51) to (53), the accretion and riming processes cannot transform into the time step Δt more cloud liquid water than the available existing amount at time t, i.e. $\overline{q_l}$. Similarly, the aggregation process cannot transform more cloud ice water than $\overline{q_i}$.

It results the two following important limitations

$$[\,\Delta\overline{q_l}\,]_{\text{acc}} + [\,\Delta\overline{q_l}\,]_{\text{rim}} < \overline{q_l}\,, \tag{57}$$

$$[\,\Delta\overline{q_i}\,]_{\text{agg}} < \overline{q_i}\,. \tag{58}$$

2.8 The evaporation processes

The tendencies due to the evaporation of rain E_r and the sublimation of snow E_s are given by Eq.(9) in Lopez (2002). The distributions (27) and (29) for the rain spectra and (31) and (33) for the snow spectra are used, together with some linearized versions of (C.7) and (C.8) made in Lopez (2002).

Let us define the four constant terms c_3 to c_6 by

$$\begin{aligned} c_3 &= 2 \, * \, 0.78 \, * \, \sqrt{\pi} \, \approx \, 2.765 \, , & (59) \\ c_4 &= 2 \, * \, (\pi)^{(3-\nu_2)/8} \, * \, 0.31 \, * \, \sqrt{\nu_1} \, * \, (\rho_{ref})^{0.2} \, * \, \Gamma\Big(\frac{5+\nu_2}{2}\Big) \, \approx \, 30.1 \, , & (60) \\ c_5 &= \frac{4 \, * \, 0.65}{(2 \, * \, \sigma_1)^{2/3}} \, \approx \, 9.736 \, , & (61) \\ c_6 &= \frac{5.784 \, * \, 4 \, * \, 0.44}{(2 \, * \, \sigma_1)^{(5+\tau_2)/6}} \, * \, \sqrt{\tau_1} \, * \, (\rho_{ref})^{0.2} \, * \, \Gamma\Big(\frac{5+\tau_2}{2}\Big) \, \approx \, 478.1 \, . & (62) \end{aligned}$$

The coefficients c_3 to c_6 are computed in the subroutine ADVPRCS in the local variables ZCOEFF3 to ZCOEFF6.

Let us define the terms $f_3(p)$ and $f_4(T,p)$, two functions of the temperature and the pressure, by

$$\begin{aligned} f_3(p) &= \Big[\sqrt{1.669 \, 10^{-5}} \, * \, 2 \, 10^{-5} \, * \, (\, p_{ref}/p) \, \Big]^{1/3} \approx \, 4.34 \, 10^{-3} \, * \, \Big[\frac{p_{ref}}{p}\Big]^{1/3} & (63) \\ f_4(T,p) &= \frac{R_v \, T}{2 \, 10^{-5} \, * \, (\, p_{ref}/p)} \, \approx \, 230.8 \, 10^5 \, * \, T \, * \, \Big[\frac{p_{ref}}{p}\Big]^{-1} \, . & (64) \end{aligned}$$

The coefficients f_3 and f_4 are computed in the subroutine ADVPRCS in the local variables ZFACT3 and ZFACT4.

Let us define the evaporation and sublimation terms c_{evap} and c_{subl} by

$$\begin{aligned} c_{evap}(T,p) &= \frac{[\, 1 - \overline{q_l}/(q_{sat})_l(T) \,] \, [\, 1 - N_s \,]}{\rho \; f_5(T,p)} \, * \, \text{RNINTR} \, , & (65) \\ c_{subl}(T,p) &= \frac{[\, 1 - \overline{q_i}/(q_{sat})_i(T) \,] \, [\, 1 - N_s \,]}{\rho \; f_6(T,p)} \, * \, \text{RNINTS} \, * \, \exp[\, - \, 0.122 \, (T - T_f) \,] & (66) \end{aligned}$$

where RNINTR and RNINTS are both available in the NAMELIST, with f_5 and f_6 defined by

$$\begin{aligned} f_5(T,p) &= \frac{1}{2.31 \, 10^{-2} \, * \, R_v} \, * \, \left(\frac{L_v(T)}{T}\right)^2 + \frac{f_4(T,p)}{(e_{sat})_l(T)} \, , & (67) \\ f_6(T,p) &= \frac{1}{2.31 \, 10^{-2} \, * \, R_v} \, * \, \left(\frac{L_f(T)}{T}\right)^2 + \frac{f_4(T,p)}{(e_{sat})_i(T)} \, . & (68) \end{aligned}$$

The coefficient f_5 (resp. f_6) is computed in the subroutine ADVPRCS in the local variable ZCONDT+ZDIFFV, just before the computations of ZCEV= c_{evap} (resp. ZCSU= c_{subl}).

Let us define

$$\begin{aligned} c_{E1}(T,p) &= c_{evap}(T,p) \, * \, c_3 \, , & (69) \\ c_{E2}(T,p) &= c_{evap}(T,p) \, * \, c_4 \, / \, f_3(p) \, , & (70) \\ c_{S1}(T,p) &= c_{subl}(T,p) \, * \, c_5 \, , & (71) \\ c_{S2}(T,p) &= c_{subl}(T,p) \, * \, c_6 \, / \, f_3(p) \, . & (72) \end{aligned}$$

The coefficients c_{E1}, c_{E1}, c_{S1} and c_{S2} are computed in the subroutine ADVPRCS in the local variables ZCEV1, ZCEV2, ZCSU1 and ZCSU2, respectively.

For given amount of precipitations, either for rain $\overline{q_r}$ or for snow $\overline{q_s}$, the parts of them to be evaporated express in terms of the variables

$$
\begin{aligned}
<\overline{q_r}> &= \overline{q_r}\ /\ \{\,\text{RNINTR} \, * \, \rho_w \,\}\ , & (73)\\
<\overline{q_s}> &= \overline{q_s}\ /\ \{\,\text{RNINTS} \, * \, \exp[\,-\,0.122\,(T-T_f)\,]\,\}\ . & (74)
\end{aligned}
$$

In the subroutine ADVPRCS, $<\overline{q_r}>$ and $<\overline{q_s}>$ are denoted by ZQRNR and ZQRNS, respectively.

The final formulas (75) and (76), analogous of Eqs. (9), (C.7) and (C.8) in Lopez (2002), writes

$$
\begin{aligned}
\text{EVA} &= c_{E1} * <\overline{q_r}>^{1/2} + c_{E2} * <\overline{q_r}>^{17/24}\ , & (75)\\
\text{SUB} &= c_{S1} * <\overline{q_s}>^{2/3} + c_{S2} * <\overline{q_s}>\ . & (76)
\end{aligned}
$$

The flux terms EVA and SUB are denoted in ADVPRCS by ZEVAPPL and ZEVAPPN, respectively. The corresponding decrease in liquid or solid precipitations are denoted by $\Delta<\overline{q_r}>$=ZTQEVAPPL and $\Delta<\overline{q_s}>$=ZTQEVAPPN, respectively. They are computed by choosing the minimum values among three terms, leading to

$$
\begin{aligned}
\Delta<\overline{q_r}> &= \text{Min}\left\{ <\overline{q_r}>\ ;\ \Delta t\ \frac{\Delta p}{g}\ \text{EVA}\ ;\ \frac{\Delta p}{g}\ \frac{\text{EVA}}{\text{EVA}+\text{SUB}}\,[\,(q_{sat})_l(T)-\overline{q_v}\,] \right\} & (77)\\
\Delta<\overline{q_s}> &= \text{Min}\left\{ <\overline{q_s}>\ ;\ \Delta t\ \frac{\Delta p}{g}\ \text{SUB}\ ;\ \frac{\Delta p}{g}\ \frac{\text{SUB}}{\text{EVA}+\text{SUB}}\,[\,(q_{sat})_i(T)-\overline{q_v}\,] \right\} & (78)
\end{aligned}
$$

These limitations for the evaporations and sublimation processes are equivalent to the limitations (57) and (58), valid for the collection processes.

2.9 The melting

The snow melting is computed in ADVPRCS (under the logical key LLMELTS = .TRUE. at the beginning of the routine) in the term ZQMLTX,

$$
\text{ZQMLTX} = \text{Max}\left\{ 0\ ;\ \frac{\Delta p}{g}\ \frac{c_p\,(T-T_f)}{L_f-L_v} \right\}, \tag{79}
$$

where $L_f - L_v$ is the latent heat of fusion of snow into rain.

2.10 The sedimentation process

In the original Lopez scheme, a kind of semi-lagrangian method was applied for the sedimentation of precipitating species. It permits to use NWP and GCM longest time-step. A new statistical scheme for the sedimentation of precipitation has been introduced later by Bouteloup et al. (2010), (subroutine ADVPRCS) and enables, for example, to use the complete formulas for the fall speed of the liquid and solid particles instead of the "bulk" constant values TFVR and TFVS or to describe sedimentation of cloudy condensates (TFVI/TFVL larger than zero). This scheme is described in part ?. Note that the fall speed for liquid and solid precipitation can be either computed as a combined constant value for NSEDIM=0 (same value for liquid and solid), or as two different constant fall speeds, one for liquid precipitation, one for solid one (NSEDIM=1).

3 Algorithmics

As the same microphysics scheme is used for convective and large-scale condensates, the Lopez scheme is called twice:

- in aplpar.F90 subroutine, for the large-scale part, under the logical keys (LPROCLD.AND.LCONDWT)=.TRUE.
- in acpcmt.F90 subroutine, for the convective part.

The following sequence of subroutines is used (the example is given here for large-scale condensates; the sequence under ACPLUIZ is the same for the convective part). Only the sequence used in our options is described:

APLPAR : monitor of the ARPEGE physics
> **ACPLUIZ** : general call to the Lopez subroutine
> **in ACPLUIZ** : compute the condensation/evaporation of cloud water,
with the statistical cloud water coming as output from
the prognostic TKE scheme (LNEBECT=TRUE) for the stratiform part, and the convective cloud water coming as output of the convection scheme for the convective part
> **ACMICRO** : compute the auto-conversion fluxes
with PAUTOL and PAUTOI as output ;
> **ADVPRCS** : compute the vertical sedimentation of the precipitation,
plus the collection, the evaporation of precipitation
and the melting (snow<->rain) processes.

4 Logical keys and main parameters in the current version

Logical keys (in NAMPHY)

- LCOLLEC=.TRUE. to switch-on the collection processes
- LCONDWT=.TRUE. use of prog. condensed water in the Physics
- LEVAPP=.TRUE. to switch-on the evaporation of precipitations
- LNEBECT=.TRUE. use of Bougeault-Bechtold PDFs (<-> TKE-CBR)
- LPROCLD=.TRUE. to switch-on the Lopez scheme

Main parameters used in the current climate version (NAMPHY0). During the development of the atmospheric model for CMIP6, tunings of some not-well constrained parameters have been done to balance TOA and surface budgets. Following Mauritsen et al. (2012) and Hourdin et al. (2012), these tuned parameters are either microphysics ones (see below), or parameters describing clouds-radiation interaction (see chapter on radiation).

- NSEDIM= 1, sedimentation options in advprcs
- RACCEF=1.0, collection efficiencies (accretion)
- RAGGEF=0.3, collection efficiencies (aggregation)
- RAUTEFR = 10 E-04 efficiency for auto-conversion water –> rain: value for $1/\tau_l$ (unit in s^{-1})
- RAUTEFS = 52 E-04 auto-conversion: value for $1/\tau_l$ (unit in s^{-1})
- RAUTSBET=0.025 ; auto-conversion: the efficiency coefficient $(\beta)_{ice}$ for autoconversion ice –> snow
- RDTFAC=0.5, auto-conv.: tunes the width of δ_{ice}in term of RDT= 11.82 K
- REVASX=1 E-7., to limit (if > 0) the evaporation of precipitations (maximum evaporation rate)

- RNINTR=0.8 E+7, collection: a global tuning for c_{evap}
- RNINTS=0.2 E+7, collection: a global tuning for c_{subl}
- RQICRMAX=0.21 E-4, maxi auto-conv. threshold (solid) (kg/kg)
- RQICRMIN=0.14 E-6 ; mini auto-conv. threshold (solid) (kg/kg)
- RQICRSN=1., auto-conv.: tunes the impact of CNC concentration on $(q_i)_{crit}$
- RQICRT1=-80., auto-conversion: first temperature for $(q_i)_{crit}$ (T_1, in K)
- RQICRT2=0., auto-conversion: second temperature for $(q_i)_{crit}$ (T_2, in K)
- RQICVMAX=0.3 E-4, maximum critical ice content for autoconversion of convective cloud ice (kg/kg)
- RQICVMIN=0.2 E-6, minimum critical ice content for autoconversion of convective cloud ice (kg/kg)
- RQLCR=200. E-6, auto-conversion: threshold of liquid water content (liquid) (kg/kg)
- RQLCV=0.2 E-3, critical liquid water content for autoconversion of convective cloud water (kg/kg)
- RRAUTCS=1, ratio (convective/stratiform) of efficiency for autoconversion of water —> rain. In the convective case, the efficiency of autoconversion water —> rain is RAUTEFR*RRAUTCS, RAUTEFR in the stratiform case.
- RREVASXCS = 1, ratio of the maximum evaporation rate (convective/stratiform). In the convective case, the evaporation rate is REVASX*RREVASXCS, REVASX in the stratiform case,
- RRIMEF=1.3, collection efficiencies (riming)
- RSPCRR=1, supercooled rain switch (1= yes, 0= no)
- TFVR=5.0, Fall speed for rain (m/s)
- TFVI=0.08, Sedimentation speed of cloud ice water (m/s)
- TFVL=0.02, Sedimentation speed for cloud liquid water (m/s)
- TFVS=1., Fall speed for snow (m/s)

10

The CMIP6 dry and moist convection scheme: Prognostic Condensates Microphysics and Transport

1 Introduction: objectives of convective parametrization schemes

Following Arakawa (2004), the deep cumulus parameterization problem may be defined as the problem of formulating the statistical effects of moist convection to obtain a closed system for predicting weather and climate, trying to predict the grid-averaged effect of an ensemble of subgrid-scale drafts. Since cumulus parametrization is an attempt to formulate the statistical effects of cumulus convection without predicting individual clouds, it is a closure problem in which a limited number of equations that govern the statistics of a system with huge dimensions is looked for. The closure consists in a hypothesis that links the occurrence and overall intensity of cumulus activity (e.g., cloud-base mass flux) to large-scale processes.

1.1 Convective components and developments

Three main aspects for the causes of convection in the atmosphere can be highlighted: the buoyancy force (responsible for free convection), convergence (due to orography or frontal systems and that causes forced convection), and vertical wind shear (potential additional role in the formation of organized thunderstorms). In the convective development the first phase is the thermal, driven by the buoyancy, followed by the second phase with cumulus, when the water vapour condensation starts (cumulus humilis in the sky), then towering cumulus (cumulus congestus), and cumulonimbus at a late stage.

1.2 The apparent sources: the Q1, Q2, Q3 terms

The core of parameterization consists in evaluating the apparent source terms, as defined in Yanai (1973), looking for a closed expression for the apparent sources in terms of the known resolved-scale variables (grid-averaged quantities).

Following Yanai et al (1973), the total effect of convection on large-scale may be deduced from the basic equations of thermodynamic and dynamics: averaging these equations on a grid ($\bar{.}$), using the Reynolds axioms and assuming that the horizontal transport of subgrid scale quantities is small relative to the vertical transport, the large-scale tendencies due to convection for $s = CpT + gz$ dry static energy (unit: $m^2.s^{-2}$), q specific humidity (unit:$kg.kg^{-1}$), u, v zonal and meridional component of wind (unit: $m.s^{-1}$) can be written as:

$$\left\{ \begin{array}{lcl} \left(\frac{\partial \bar{s}}{\partial t}\right)_c & = & L_v(\bar{c} - \bar{e}) - \frac{\partial \overline{\omega' s'}}{\partial p} = Q_1 - \overline{Q_r} \\ \left(\frac{\partial \bar{q}}{\partial t}\right)_c & = & -(\bar{c} - \bar{e}) - \frac{\partial \overline{\omega' q'}}{\partial p} = \frac{-Q_2}{L_v} \\ \left(\frac{\partial \bar{u}}{\partial t}\right)_c & = & -\frac{\partial \overline{\omega' u'}}{\partial p} = Q_3 u \\ \left(\frac{\partial \bar{v}}{\partial t}\right)_c & = & -\frac{\partial \overline{\omega' v'}}{\partial p} = Q_3 v \end{array} \right. \tag{1}$$

with ω ($Pa.s^{-1}$) vertical velocity in pressure coordinate, Q_r ($W.kg^{-1}$) radiative heating rate, Q_1 ($W.kg^{-1}$) apparent source of heat, Q_2 ($W.kg^{-1}$) apparent sink of moisture, Q_3u, Q_3v ($m.s^{-2}$) apparent sources of momentum, L_v ($J.kg^{-1}$) water vaporization latent heat, c and e ($kg.kg^{-1}.s^{-1}$) condensation and evaporation rate.

Then, following Yanai et al (1973) :

$$\left\{ \begin{array}{lcl} Q_1 & = & \underbrace{\overline{Q_r}}_{radiative\ effect} + \underbrace{L_v(\bar{c} - \bar{e})}_{condensation/evaporation} \underbrace{-\frac{\partial \overline{\omega' s'}}{\partial p}}_{vertical\ transport\ by\ unresolved\ eddies} \\ Q_2 & = & \underbrace{L_v(\bar{c} - \bar{e})}_{condensation/evaporation} + \underbrace{L_v \frac{\partial \overline{\omega' q'}}{\partial p}}_{vertical\ transport\ by\ unresolved\ eddies} \\ Q_3 u & = & \underbrace{-\frac{\partial \overline{\omega' u'}}{\partial p}}_{vertical\ transport\ by\ unresolved\ eddies} \\ Q_3 v & = & \underbrace{-\frac{\partial \overline{\omega' v'}}{\partial p}}_{vertical\ transport\ by\ unresolved\ eddies} \end{array} \right. \tag{2}$$

These terms represent the mean effect of convective processes on large-scale, as a function of the grid-averaged variables. The apparent heat source, Q_1, depends on condensation/evaporation rate, radiative effects and dry static energy vertical transport by unresolved (i.e subgrid) eddies. $Q1$ >0 for atmospheric heating, $Q1$<0 for atmospheric cooling. Condensation/(evaporation) is a source/(sink) of heat ($Q1$>0/(<0)) and of drying ($Q2$>0/(<0)).

$Q2$ depends on condensation/evaporation rate and of vertical transport of water vapor by unresolved eddies. $Q3$ depends only on the vertical transport of momentum by unresolved eddies.

Some basic constraints for the apparent sources can be derived (Yanai et al 1973): first, the sum of the two first equations of 1 leads to a budget equation for the moist static energy $h = s + L_v q$:

$$Q_1 - Q_2 - \overline{Q_r} = -\frac{1}{\rho} \frac{\partial}{\partial z} \rho \overline{w' h'} \tag{3}$$

Furthermore, a vertical integral of the source terms can be expressed to a good approximation by:

$$C_p \langle Q_1 \rangle = \langle \overline{Q_r} \rangle + L_v P + F_s$$

$$\frac{C_p}{L_v}\langle Q_1\rangle = P - F_L$$

where $\langle . \rangle$ designates the vertical integral from the surface to the top of the atmosphere, P the precipitation rate, F_s the surface sensible heat flux and F_L is the surface evaporation rate. Any given parameterization must be consistent with those constraints but there are not strong enough to guide the construction of a sub-grid scale parameterization. Then, determining the vertical distribution of the terms $Q_1 - Q_r$, Q_2 and Q_3 is one of the objectives of the parameterization schemes.

Generally speaking, a convection parameterization scheme follows three main steps:
- definition of a triggering condition to describe when and where convection appears
- determine the vertical distribution of Q_1, Q_2, Q_3u and Q_3v
- determine the grid-averaged convective intensity, through the closure of the scheme

As CMIP5 one, CMIP6 deep convection parameterization scheme uses a mass-flux formulation, as the vast majority of NWP and Climate models. This formulation is quickly examined below.

1.3 The mass-flux parametrization schemes

The hypothesis used in the mass-flux formulation relies on a self-consistency argument for the mean heat-budget in the tropical atmosphere: the tropical mean heat budget can not be closed without assuming the existence of tall, isolated, convective vertical heat transport that happens without substantial interaction with the environment (named "hot tower" by (Riehl and Marcue (1958)). To represent such an entity as a subgrid-scale process, an "isolated" vertical transport process is required, which may be described by introducing a mass-flux, a measure of vertical transport rate (Yano and Plant (2016)). Once a mass-flux is defined, it is relatively straightforward to define the vertical transport of heat and moisture. Thus, the key issue of mass-flux parameterization is to define the mass-flux associated with convection. The mass-flux schemes allow to write the convective transport of the variables s, q, u and v from the mass flux and the Cloud/Environment differences of these variables. Each model grid is divided in two sub-domains: the convective one (cloudy) and the environmental one (clear sky). Then, for any variable χ:

$$\overline{\chi} = \sigma\chi_c + (1-\sigma)\chi_e \tag{4}$$

and for the vertical velocity ω:

$$\overline{\omega} = \sigma\omega_c + (1-\sigma)\omega_e \tag{5}$$

where $\overline{\chi}$ is the grid-averaged variable, $\overline{\omega}$ the averaged vertical velocity, σ the convective fraction in the grid, indices $_c$ and $_e$ stand for convective and environmental variables.

The subgrid vertical transport of χ is :

$$\overline{\omega'\chi'} = \overline{\omega\chi} - \overline{\omega}\,\overline{\chi} = \sigma\omega_c\chi_c + (1-\sigma)\omega_e\chi_e - (\sigma\omega_c + (1-\sigma)\omega_e)\overline{\chi} = \sigma(\omega_c - \omega_e)(\chi_c - \overline{\chi}) \tag{6}$$

Assuming $\omega_c >> \omega_e$, ω^* mass flux of the updraughts relative to their environment in the grid is :

$$\omega^* = \sigma(\omega_c - \omega_e) = \sigma\omega_c \tag{7}$$

, and the convective transport flux is: $\overline{\omega'\chi'} = \omega^*(\chi_c - \overline{\chi})$.

Assuming that the area with updraught are small ($\sigma << 1$), and steadiness of the cloud budget for mass, heat and water vapor (Yanai et al 1973), the apparent source terms can be written (as in Yanai et al., 1973):

$$\left\{\begin{array}{rcl} Q_1 & = & \overline{Q_r} + \omega^* \frac{\partial \overline{s}}{\partial p} + D(s_c - \overline{s}) \\ Q_2 & = & -L_v \omega^* \frac{\partial \overline{q}}{\partial p} - L_v D(q_c - \overline{q}) \\ Q_3 u & = & \omega^* \frac{\partial \overline{u}}{\partial p} + D(u_c - \overline{u}) \\ Q_3 v & = & \omega^* \frac{\partial \overline{v}}{\partial p} + D(v_c - \overline{v}) \end{array}\right. \tag{8}$$

with D $(unit : s^{-1})$ the cloudy air detrainment rate.

The convective tendencies on the large-scale are given as functions of convective mass flux ω^*, cloudy detrainment rate D and differences between cloudy and environmental variables. Then the impact of convection on the environment in terms of heating $(Q_1 - Q_r)$ is due to heating associated with compensating subsidence between cumulus clouds (term $\omega^* \frac{\partial \overline{s}}{\partial p}$) and detrainment of cloudy air in the environment (term $D(s_c - \overline{s})$). One of the drawback of mass flux formulation is the need of specifying not well-known parameters as entrainment/detrainment rates.

2 From CMIP5 to CMIP6

2.1 Limits of the CMIP5 convection scheme (Bougeault, 1985)

The objective here is not to describe the CMIP5 scheme, rather to give some ideas about the limits of the CMIP5 scheme and the way followed from CMIP5 to CMIP6 convection schemes. Some limits of the Bougeault's scheme are known: misrepresentation of diurnal cycle of convection on continents (too early) linked with the triggering condition, too easy triggering with large areas of simulated low precipitation. More precisely, during the EUROCS[1] project (from 2000 to 2003), with the purpose to improve the clouds simulation in RCM and GCM, it has been shown, in SCM configurations of deep convection triggering in a oceanic stratocumulus case (Duynkerke et al 2004), that the scheme includes a moisture convergence closure not well adapted to extreme moisture conditions (Derbyshire et al 2004), a simulated diurnal cycle of clouds depth and precipitation too linked with the maximum of solar heating and too small downdrafts (Guichard et al 2004).

In parallel of the development of the PCMT scheme, many tests have been done in a climate configuration with the prognostic physics used in NWP, i.e similar to that described in this documentation, except the deep and shallow convection schemes that are that of Bougeault for the deep convection and Kain-Fritsh-Bechtold (Kain and Fritsch 1993, Bechtold et al 2001) for the shallow convection (see description in table 10.1, column "NWP"). Even if many sensitivity tests have been run, this physics has never been chosen for CMIP exercise due to a very strong double ITCZ (even in forced simulations). Moreover, in CMIP5 configurations (left column), no specific scheme exists for shallow convection and thermals are not represented.

2.2 The CMIP6 convection scheme: an unified scheme, separating microphysic and transport in grid-scale equations

In a parametrization context, it is common to talk about "dry" convection as that produced by thermals which are not associated with cloud and which are confined within the boundary layer; to talk about "shallow" convection as relatively weak, non-precipitating cumuli; and to talk about "deep" convection as cumulonimbus clouds. These aspects are

[1] EUROpean Cloud Systems

Table 10.1: Table with main characteristics of CMIP5, NWP (2015 version) and CMIP6 atmospheric physics.

Parametrization	CMIP5	NWP	CMIP6
Shallow convection	No specific scheme, partly through moist PDF	Mass-flux scheme Bechtold et al 2001	PCMT (Gueremy 2011 , Piriou et al 2007)
Deep convection	Bougeault 1985	Bougeault 1985 with modifications	PCMT (Gueremy 2011, Piriou et al 2007)
Turbulence	Diagnostic TKE Ricard-Royer 1993	Prognostic TKE Cuxart et al 2000	Prognostic TKE (Cuxart et al 2000)
Mixing length	Quadratic profile Lenderink-Holtslag 2004	Mixing length Bougeault-LacarrÃ¨re 1989	Mixing length Bougeault-LacarrÃ¨re 1989
Clouds	Bougeault 1981 PDF	Bougeault 1981 PDF	Bougeault 1981 PDF
Microphysics	Diagnostic Smith 1990	Pronostic scheme Lopez 2002	Prognostic scheme Lopez 2002
Radiation	ECMWF scheme (Fouquart-Bonnel 1980 and Mlawer et al 1997)	ECMWF scheme (Fouquart-Bonnel 1980 and Mlawer et al 1997)	ECMWF scheme (Fouquart-Bonnel 1980 and Mlawer et al 1997)

often treated separately in numerical models, within the boundary-layer parameterization, the shallow convection parameterization and the deep-convection parameterization respectively. In the PCMT scheme an unified approach of the three types of convection is proposed. For example, a continous formulation of turbulent entrainment (see eq. 24) is used, allowing treatment of both shallow and deep convection. For the vertical profile, both dry (i.e starting from the absolute temperature) and moist (starting from the wet bulb temperature) adiabats are considered, the choice being done through the comparison of the temperatures at the top level.

The idea behind separating microphysics and transport is given in Piriou et al. 2007. The proposed separation consists in defining the role of convective parameterizations as "estimating the subgrid-scale rate of convective transport (subgrid-scale motions) and microphysical processes (condensation, evaporation, downdrafts...) as a feedback to resolved forcing". In this paper the separation is used both as a way to introduce into the parametrization a more explicit causal link between all involved processes and as a vehicle for an easier representation of convective cells. Doing that the equations of parameterization become closer to those of CRM and LES.

Actually, the "Microphysics-Transport" spirit consists in externalizing microphysical computations from the subgrid convection code, in order to share microphysical routines between subgrid and resolved convection, and use "up-to-date" microphysics (see chapter: "The CMIP6 microphysics scheme"). The grid-averaged convective tendencies at resolved scale are expressed in terms of change of phase (Microphysics) and Transport, enabling a more direct comparison with CRM or LES data. The generic equation for q, specific humidity, is written as $\frac{\partial q}{\partial t} = \frac{\partial (M(q_c - q))}{\partial p} - C$, separating transport and microphysics, instead of $\frac{\partial q}{\partial t} = -M\frac{\partial q}{\partial p} + D(q_c - q)$, with M the mass flux, C the condensation rate and D the detrainment rate.

3 The PCMT scheme

The PCMT scheme (Piriou 2007, Gueremy 2011) is a parametrization scheme in which the grid-scale budget equations of parametrization use separate microphysics and transport terms. The role of convective parameterizations is then to estimate the subgrid-scale rate of convective transport (subgrid-scale motions) and microphysical processes (condensation, evaporation, downdrafts...) as a feedback to resolved forcing. The prognostic microphysics scheme is that of LOpez (2002), largely modified by Bouteloup et al (2010). Some adjustments of microphysics parameters have been done (see Part Microphysics) for the convective part. In this documentation only the options for closure and entrainement used in the Climate version are described for simplicity. The closure hypothesis is based on CAPE relaxation. A prognostic equation is used for convective vertical velocity and the triggering condition is based on upward vertical velocity. A continous treatment of convection, from thermals to deep precipitating convection, is obtained through a continous formulation of entrainment/detrainment rates depending on convective vertical velocity.

3.1 Prognostic variables and geometry

Geometry

The variables q_l, q_i, q_r, q_s are prognostic, enabling to use the same microphysics for convective and resolved part. Vertical velocity is also a prognostic variable.

The gridpoint is divided in three parts: updraft (of area α_u), downdraft (of area α_d), environment (of area $1 - \alpha_u - \alpha_d$).

Table 10.2: Geometry of the three subgrid components of PCMT.

Component	updraft	downdraft	environment
Area fraction	α_u	α_d	$1 - \alpha_u - \alpha_d$
Vertical velocity	w_u	w_d	$\frac{-\alpha_u w_u - \alpha_d w_d}{1-\alpha_u-\alpha_d}$
Liquid water	q_{lc}	q_{lc}	q_{lr}
Precipitation	q_{rc}	q_{rc}	q_{rr}

Be careful that vertical velocities defined here are those for convection, then relative to the large-scale vertical velocity: the absolute velocity in the updraft is $w = \overline{w} + w_u$. At the moment the downdrafts are only diagnostic ones, not prognostic.

Tendency equations for q_v and s

For q_v, the tendency due to subgrid convection is (see table 10.3) :

$$(\frac{\partial}{\partial t}\overline{q_v})_c = -\frac{1}{\rho}\frac{\partial}{\partial z}\rho\alpha_u w_u(q_{vu}-\overline{q_v}) - \frac{1}{\rho}\frac{\partial}{\partial z}\rho\alpha_d w_d(q_{vd}-\overline{q_v}) - \text{Condens} + \text{Evap}_{q_{rc}} + \text{Evap}_{q_{sc}} \quad (9)$$

The enthalpy ($s = c_p T + \phi$) tendency due to the convection is (see also Table 10.4):

$$\begin{aligned}(\frac{\partial}{\partial t}\overline{s})_c \quad = \quad & -\tfrac{1}{\rho}\frac{\partial}{\partial z}\rho\alpha_u w_u(s_u - \overline{s}) \\ & -\tfrac{1}{\rho}\frac{\partial}{\partial z}\rho\alpha_d w_d(s_d - \overline{s}) \\ & +\text{CondensEvap}_{q_{lc}} + \text{CondensEvap}_{q_{ic}} + \text{Evap}_{q_{rc}} + \text{Evap}_{q_{sc}} \\ & +\text{MeltingIcing}_{q_{lc}}\end{aligned} \quad (10)$$

Table 10.3: Name of fortran variables in the code

Convective flux of specific humidity (not rain/snow)	$-\frac{1}{\rho}\frac{\partial}{\partial z}\rho\alpha_u w_u(q_{vu}-q_{ve})-\frac{1}{\rho}\frac{\partial}{\partial z}\rho\alpha_d w_d(q_{vd}-q_{ve})$	PDIFCQ
Convective condensation flux for liquid/ice water	$-$Condens	PFCCQL + PFCCQN
precipitation flux due to convective evaporation	+Evap$_{q_{rc}}$	PFPEVPCL
precipitation flux due to convective sublimation	+Evap$_{q_{sc}}$	PFPEVPCN PFPEVPCN

Table 10.4: Name of fortran variables in the code

Convective flux of enthalpy (not rain/snow) $-\frac{1}{\rho}\frac{\partial}{\partial z}\rho\alpha_d w_d(s_d-\overline{s})$	$-\frac{1}{\rho}\frac{\partial}{\partial z}\rho\alpha_u w_u(s_u-\overline{s})$ PDIFCS	
Convective condensation flux for liquid water/ice	$-$CondensEvap$_{q_{lc}}$ + CondensEvap$_{q_{ic}}$	PFCCQL/PFCCQN times L in CPTEND_NEW
Precipitation flux due to convective evaporation/sublimation	+Evap$_{q_{rc}}$ + Evap$_{q_{sc}}$	PFPEVPCL/PFPEVPCN times L in CPTEND_NEW
q-flux due to icing-melting following convective transport of convective liquid and ice water	+MeltingIcing$_{q_{lc}}$	PFIMCC times L_{fusion} in CPTEND_NEW

Table 10.5: Name of fortran variables in the code

Convective flux of liquid water (not rain/snow)	$-\frac{1}{\rho}\frac{\partial}{\partial z}\rho\left[-\alpha_u w_u - \alpha_d w_d\right] q_{lr}$	PDIFCQL
Entrainment/detrainment q_{lr} (liquid convective and resolved)	$-(E_u + E_d)\, q_{lr} + (D_u + D_d)\, q_{lc}$	-PFEDQLC
Stratiform condensation flux for liquid water	$\mathrm{CondensEvap}_{q_{lr}} + \mathrm{MeltingIcing}_{q_{lr}}$	-PFCSQL
Flux of resolved precipitation: the generation term	$-\mathrm{AutoconvColl}_{q_{lr}}$	PFPFPSL

Using p instead of z as vertical coordinate, equations include only a transport term and a condensation one, (microphysics, downdraughts and other phase changes (melting/freezing) are taken into account as induced processes), and with M the mass-flux defined as $M = -\alpha\sigma\omega_u$:

$$(\frac{\partial}{\partial t}\overline{q_v})_c = \frac{M(q_{vu} - \overline{q_v})}{\partial p} - M\frac{q_{vu}}{\partial p} \tag{11}$$

The last rhs term corresponds to the condensation term $C= M\frac{q_{vu}}{\partial p}$ being positive. And for enthalpy the tendency equation in p-coordinate writes:

$$(\frac{\partial}{\partial t}\overline{s})_c = \frac{M(s_u - \overline{s})}{\partial p} + L_v M\frac{q_{vu}}{\partial p} \tag{12}$$

Tendency equations for cloudy variables

The environment includes liquid water q_{lr}, (the same for ice q_{ir}), and precipitating condensates q_{rr} for rain and q_{sr} for snow, with grid-averaged value: $\overline{q_{lr}} = (1 - \alpha_u - \alpha_d)\, q_{lr}$. These condensates correspond to the large-scale condensation tendency, and to a contribution of entrainment/detrainment of updraft/downdraft, as can be seen in equation 13,for cloudy liquid water from resolved condensation (see part 2.5 CBR documentation) :

$$\begin{aligned}\frac{\partial}{\partial t}\overline{q_{lr}} \quad = \quad & \mathrm{Advec}(\overline{q_{lr}}) \\ & -\tfrac{1}{\rho}\frac{\partial}{\partial z}\rho\left[-\alpha_u w_u - \alpha_d w_d\right] q_{lr} \\ & -(E_u + E_d)\, q_{lr} + (D_u + D_d)\, q_{lc} \\ & +\mathrm{CondensEvap}_{q_{lr}} - \mathrm{AutoconvColl}_{q_{lr}} + \mathrm{MeltingIcing}_{q_{lr}}\end{aligned} \tag{13}$$

In this equation, the first four rhs terms correspond to transport (the third and fourth to horizontal one) and the last rhs terms to microphysics. Table 10.5 gives the name of the different terms in the code.

The updraft and downdraft contributions appear in the budget of subgrid convective liquid water q_{lc} (the same for ice q_{ic}, and precipitating condensates q_{rc} and q_{sc}), with grid-averaged value $\overline{q_{lc}} = (\alpha_u + \alpha_d)\, q_{lc}$. These condensates are due to condensation in the subgrid convective updraft. Equation 14 gives the tendency of cloudy convective subgrid liquid water:

$$\begin{aligned}\frac{\partial}{\partial t}\overline{q_{lc}} \quad = \quad & \mathrm{Advec}(\overline{q_{lc}}) \\ & -\tfrac{1}{\rho}\frac{\partial}{\partial z}\rho\left[\alpha_u w_u + \alpha_d w_d\right] q_{lc} \\ & +(E_u + E_d)\, q_{lr} - (D_u + D_d)\, q_{lc} \\ & +\mathrm{CondensEvap}_{q_{lc}} - \mathrm{AutoconvColl}_{q_{lc}} + \mathrm{MeltingIcing}_{q_{lc}}\end{aligned} \tag{14}$$

Table 3.1 gives the names of the different terms in the code.

Table 10.6: Name of fortran variables in the code

Convective transport flux of q_{lc} (liquid convective)	$-\frac{1}{\rho}\frac{\partial}{\partial z}\rho\left[\alpha_u w_u + \alpha_d w_d\right] q_{lc}$	PDIFCQLC
Entrainment/Detrainment of q_{lc} and q_{lr}	$+(E_u + E_d)\, q_{lr} - (D_u + D_d)\, q_{lc}$	PFEDQLC
Convective condensation flux for liquid water	$\text{CondensEvap}_{q_{lc}}$	-PFCCQL
Flux of convective precipitation: generation term	$-\text{AutoconvColl}_{q_{lc}}$	PFPFPCL
q flux due to icing-melting following convective transport of convective liquid and ice water	$+\text{MeltingIcing}_{q_{lc}}$	PFIMCC

Table 10.7: Name of fortran variables in the code

Convective precipitation as rain	$-\frac{1}{\rho}\frac{\partial}{\partial z}\rho\left[\alpha_u(w_u + w_s) + \alpha_d(w_d + w_s)\right] q_{rc}$	PFPLCL
Entrainment/detrainment of q_{rc} and q_{rr} (convective and resolved rain)	$+(E_u + E_d)\, q_{rr} - (D_u + D_d)\, q_{rc}$	PFEDQRC
Flux of convective precipitation: generation term	$+\text{AutoconvColl}_{q_{lr}}$	-PFPFPCL
precipitation flux due to convective evaporation	$-\text{Evap}_{q_{rc}} + \text{MeltingIcing}_{q_{rc}}$	PFPEVPCL

Tendency equations for prognostic precipitation

The equation for convective precipitation is:

$$\begin{aligned}
\frac{\partial}{\partial t}\overline{q_{rc}} \quad = \quad & \text{Advec}(\overline{q_{rc}}) \\
& -\tfrac{1}{\rho}\frac{\partial}{\partial z}\rho\left[\alpha_u(w_u + w_s) + \alpha_d(w_d + w_s)\right] q_{rc} \\
& +(E_u + E_d)\, q_{rr} - (D_u + D_d)\, q_{rc} \\
& +\text{AutoconvColl}_{q_{lr}} - \text{Evap}_{q_{rc}} + \text{MeltingIcing}_{q_{rc}}
\end{aligned} \tag{15}$$

and Table 3.1 gives the names of the different variables. The equation for large-scale precipitation from resolved + subgrid condensation Smith type (table 3.1) is:

$$\begin{aligned}
\frac{\partial}{\partial t}\overline{q_{rr}} \quad = \quad & \text{Advec}(\overline{q_{rr}}) \\
& -\tfrac{1}{\rho}\frac{\partial}{\partial z}\rho\left[-\alpha_u w_u - \alpha_d w_d + (1 - \alpha_u - \alpha_d) w_s\right] q_{rr} \\
& -(E_u + E_d)\, q_{rr} + (D_u + D_d)\, q_{rc} \\
& +\text{AutoconvColl}_{q_{lr}} - \text{Evap}_{q_{rr}} + \text{MeltingIcing}_{q_{rr}}
\end{aligned} \tag{16}$$

Table 10.8: Name of fortran variables in the code

Stratiform precipitation as rain	$-\frac{1}{\rho}\frac{\partial}{\partial z}\rho\left[-\alpha_u w_u - \alpha_d w_d + (1 - \alpha_u - \alpha_d) w_s\right] q_{rr}$	PFPLSL
Entrainment/Detrainment of q_{rc} and q_{rr} (convective and resolved rain)	$-(E_u + E_d)\, q_{rr} + (D_u + D_d)\, q_{rc}$	-PFEDQRC
Flux of resolved precipitation: generation term	$+\text{AutoconvColl}_{q_{lr}}$	-PFPFPSL
precipitation flux due to resolved evaporation	$-\text{Evap}_{q_{rr}} + \text{MeltingIcing}_{q_{rr}}$	PFPEVPSL

3.2 Vertical velocity prognostic equation and triggering conditions

Vertical velocity prognostic equation

The cloud vertical velocity is that of Gueremy(2011), derived from Simpson and Wiggert (1969) and Chen and Bougeault (1992) (corresponding to the logical key LCVNHD=F, see parameters below):

$$\frac{\partial \omega_u}{\partial t} = \underbrace{-\frac{1}{2}\frac{\partial {\omega_u}^2}{\partial p}}_{advection} \underbrace{-\frac{\rho\, g^2}{1+\gamma}\frac{(T_{vu}-\overline{T_v})}{\overline{T_v}}}_{buoyancy} + \underbrace{(\epsilon_t+\epsilon_0+K_d){\omega_u}^2}_{entrainment} \tag{17}$$

where:

- $T_{vu} = T_u(1+0.608q_u - q_{lu})$ (K) is the cloud virtual temperature, and $\overline{T_v} = \overline{T}(1+0.608\overline{q})$ (K) the grid averaged virtual temperature
- the total friction coefficient is the sum of turbulent (ϵ_t), organized (ϵ_0) entrainments and aerodynamic drag (K_d), unit of all three being Pa^{-1}, following Simpson-Wiggert (1969), Simpson et al (1965) and Simpson (1971).
- γ describes the buoyancy reduction due to non-hydrostatic effects ($\gamma = 0.5$ as in Simpson, 1971), analog to the impact of the motion of a solid in a fluid, this "virtual mass coefficient" has been introduced by Simpson and Wiggert (1969) to compensate for a too strong buoyancy acceleration in their model. Essentially this is a factor through which the effective mass of a bubble (an air parcel) is enhanced.

The evolution of ω_u depends on three terms: advection, buoyancy (source term) and sink term (entrainment).
An implicit time discretization is used following Chen and Bougeault (1992):

$$\frac{0.5(\omega(j-1)^+ + \omega(j)^+) - 0.5(\omega(j-1)^- + \omega(j)^-)}{\Delta t} = Buoyancy - \frac{1}{2}\frac{\omega(j)^{2+} - \omega(j-1)^{2+}}{\Delta p} + (\epsilon_t+\epsilon_0+K_d)(\tfrac{1}{4}(\omega(j-1)^+ + \omega(j)^+)^2) \tag{18}$$

where $\omega(j)$ is the vertical velocity at level j of the model (j decreasing upward) and the superscript stands for the temporal discretization. Then the cloud vertical velocity $\omega(j-1)^+$ is obtained as the negative root of a second order polynomial of $\omega(j-1)^+$. Buoyancy (second rhs term of equation 17) is a forcing term of the prognostic vertical velocity equation. This equation is then used to define triggering conditions, mass flux and entrainment/detrainment rates. Buoyancy is also used for the closure assumption, through its vertically integrated form: the CAPE. Unlike what has been done in part 1.3, where ω^* and D are given from steady cloud budget equations, no more steady hypothesis is assumed. The mass-flux M is written as :

$$M = -\alpha\sigma\omega_u \tag{19}$$

with ω_u $(Pa.s^{-1})$ the convective vertical velocity, σ the convective fraction and α the convective fraction at the cloud bottom, both unitless.

Entrainment/detrainment

Dilution of a cumulus cloud by entrainment of environmental air was described for the first time by Stommel (1947). Numerous observational studies of cumulus clouds using aircraft followed (e.g Warner, 1955), then more precise quantitative descriptions of entrainment originated from laboratory water tank experiments of thermal plumes (Morton et al. ,1956; Turner, 1962) describing an increasing upward mass flux M (with $M = \rho\omega_u\sigma_u$ (unit $kgm^{-2}s^{-1}$), ω_u the plume updraught velocity and the associated plume fractional area σ_c), with height:

$$\frac{1}{M}\frac{\partial M}{\partial z} = \epsilon$$

, where ϵ denotes the fractional entrainment. An important refinement of the entrainment formulation of this equation was first pointed out by Houghton and Cramer (1951), who made a distinction between dynamical -organized- entrainment due to larger scale organized inflow and turbulent entrainement caused by turbulent mixing at the cloud edge:

$$\frac{1}{M}\frac{\partial M}{\partial z} = \epsilon_o + \epsilon_t$$

The idea behind dynamical (or organized) entrainment is simple: there must be an inflow to a convective plume (or tower) proportional to a large-scale convergence. The same idea is termed by Tiedtke (1989) the "entrainment by organized flow" and Nordeng (1994) invoked an approximate relationship between the fractional entrainment and the fractional vertical divergence:

$$\epsilon_o = \frac{1}{w_u}\frac{\partial w_u}{\partial z} + \frac{1}{\rho}\frac{\partial \rho}{\partial z}$$

Assuming that the density stratification (second rhs term) is negligible, this relation reduces to:

$$\epsilon_o = \frac{1}{w_u}\frac{\partial w_u}{\partial z}$$

Whereas the organized entrainment has the characteristics of advective transport across the interface, turbulent entrainment is of a diffusive character. Since both fractional entrainment rates are by definition positive, they cause the mass flux to increase with height. For a cumulus cloud, due to the turbulent mixing at the cloud edge, a mixture of in-cloud and environmental air is made; this mixture can become negatively buoyant by evaporative cooling and will in this case detrain from the cloud, represented by a turbulent detrainment rate. The cloud or thermal itself can become negatively buoyant, stops rising and is usually dissolved into the environment. This process is called organized or dynamical detrainment, and is represented by an organized detrainment rate.

Qualitatively the following behaviours have to be represented by entrainment/detrainment:

- Two spatial scales of entrainment/detrainment: organized and turbulent. The organized entrainment is linked with the cloud-scale global velocity field (bottom entrainment, top detrainment, role of the vertical divergence of the mass flux), whereas the turbulent entrainment/detrainment is of smaller scale, located at the cloud edges and due to vertical shear of horizontal wind and horizontal shear of vertical wind.
- When in-cloud vertical velocity is large, the effect of turbulent entrainment/detrainment on the updraft is smaller as the entrainment/detrainment processes have less time to impact on a parcel.

- Wind shear is conducive to turbulent detrainment, as dry environmental air (saturation deficit $q_{sat} - q_v$), because liquid and ice water evaporation at the edge of the clouds is a source of cooling and then of turbulence.

Following Tiedtke (1989), the total entrainment/detrainment in PCMT is partitioned in these two parts, corresponding to two different scales at which the entrainment/detrainment processes occur : $\epsilon = \epsilon_0 + \epsilon_t$ where ϵ_0 is the meso-scale entrainment, and ϵ_t the turbulent one.

As the organized entrainment corresponds to the mixing due to the flow associated with large-scale divergence/convergence, the organized inflow is assumed to occur in the lower part of the cloud layer where there is large-scale moisture convergence, whereas the organized outflow is assumed to exist only in the top layer of the deepest clouds. The meso-scale entrainment is derived from the mass conservation in the convective ascent, modulated by a buoyancy-sorting process following Bretherton et al (2004). The buoyancy-sorting approach assumes that the lateral mixing of the updraught and its environment generates a spectrum of mixtures, and is used to determine which mixtures are incorporated into the updraught and which are rejected (negatively buoyant mixtures are assumed to detrain, whereas positively buoyant mixtures entrain). First, a maximum value of the fractional entrainment rate is defined as:

$$\epsilon_{ox} = |\frac{1}{\omega_u}\frac{\partial \omega_u}{\partial p}| \tag{20}$$

, derived from the classical equation (from Yanai 1973) for a cumulus ensemble budget of mass as

$$\frac{\partial M}{\partial p} = D - E \tag{21}$$

,M being the mass-flux, D and E the rates of mass detrainment/entrainment per unit pressure interval (unit: s^{-1}). To obtain eq. 20, convective fractional area coverage is assumed to be constant and no detrainment is assumed. Then, following Bretherton (2004), if μ is the mass-mixing fraction of environmental air in the mixtures, ranging from 0 for undiluted updraught to 1 for pure environmental air, μ_0 the critical mixing fraction partioning positive versus negative buoyancy mixtures, the fractional entrainment and detrainment rates are finally:

$$\epsilon_o = \epsilon_{ox}\mu_0^2 \tag{22}$$

$$\delta_o = \epsilon_{ox}(1-\mu_0)^2 \tag{23}$$

This buoyancy-sorting scheme is not applied as long as $\frac{1}{\omega_u}\frac{\partial \omega_u}{\partial P} > K_\epsilon \epsilon_t$, with K_ϵ a constant (parameter FENTRT in the code), to take into account the fact that some negatively buoyant parcels are not immediately rejected from the updraught and that the buoyancy-sorting scheme is not applied as long as the maximum fractional organized entrainment is significantly larger than its turbulent counterpart. The organized entrainment is then a function of ω_u but also of the thermodynamical state of the convective fraction and of environment (parameter μ).

The turbulent entrainment is also a function of ω_u (unit: Pas^{-1}) defined by $\epsilon_t = \epsilon_{tn} + (\epsilon_{tx} - \epsilon_{tn}) \times f_\epsilon(\omega_u, \omega_u x, \omega_u n)$ with ϵ_{tx} the maximum value of ϵ_t and ϵ_{tn} the minimum one (unit: $m^{-2}s^2$) or.

$$\epsilon_t(\omega_u) = \begin{cases} \epsilon_{tx} & if\ |\omega_u| > |\omega_x| \\ sin^2\left(\frac{\pi}{2}\frac{\omega_x - \omega_u}{\omega_x - \omega_n}\right) & if\ |\omega_n| \leq |\omega_u| \leq |\omega_x| \\ \epsilon_{tn} & if\ |\omega_u| < |\omega_n| \end{cases} \tag{24}$$

with :
$\epsilon_{tx} = 0,47.10^{-4}\ Pa^{-1}$ the maximum value of entrainment for small ω_u (TENTRX parameter in the code)
$\epsilon_{tn} = 0,5.10^{-5}\ Pa^{-1}$ the minimum value for large ω_u (TENTR in the code),
$\omega_x = -27,5\ Pa.s^{-1}$ the maximum threshold of ω_u to get minimum entrainment (VVX in the code).
$\omega_n = -2\ Pa.s^{-1}$ the minimum threshold of ω_u to get maximum entrainment (VVN in the code).

This formulation is consistent with: more intense ascent, more isolated updraft from the environment and smaller interaction. This formulation allows the treatment of both shallow and deep convection, giving a large value of entrainment for shallow clouds (small absolute value of ω) and a small value of entrainment for deep clouds (large absolute value of ω_c), using a continous transition.

The aerodynamic drag coefficient K_d follows similar equations as ϵ_t:

$$K_d(\omega_u) = \left\{ \begin{array}{ll} K_{dx} & if\ |\omega_u| > |\omega_x| \\ sin^2\left(\frac{\pi}{2}\frac{\omega_x - \omega_u}{\omega_x - \omega_n}\right) & if\ |\omega_n| \leq |\omega_u| \leq |\omega_x| \\ K_{dn} & if\ |\omega_u| < |\omega_n| \end{array} \right. \qquad (25)$$

with :
$K_{dx} = 3,57.10^{-4}\ Pa^{-1}$
$K_{dn} = 3,810^{-5}\ Pa^{-1}$

The lateral entrainment for a variable χ is given by:

$$\frac{\partial \chi_u}{\partial \phi} = \rho(\epsilon_0 + \epsilon_t)(\overline{\chi} - \chi_u) \qquad (26)$$

For a variable χ with an updraft value χ_u, a downdraft one χ_d, and an environmental value χ_r, the tendency of χ_u in the updraft associated with entrainment/detrainment is:

$$\frac{\partial \overline{\chi_u}}{\partial t} = (E_u + E_d)\chi_r - (D_u + D_d)\chi_u \qquad (27)$$

with, for a vertical flow of perimeter L, of section S, u_E the entrainment velocity and α the surface fraction,

$$\left\{ \begin{array}{lcl} E_u & = & u_{E_u}\frac{L_u}{S_u}\alpha_u \\ E_d & = & u_{E_d}\frac{L_d}{S_d}\alpha_d \\ D_u & = & u_{D_u}\frac{L_u}{S_u}\alpha_u \\ D_d & = & u_{D_d}\frac{L_d}{S_d}\alpha_d \end{array} \right. \qquad (28)$$

with unit of E_u, E_d, D_u, D_d: $kg.m^{-3}s^{-1}$.

The generic equation 27 is that used to give the entrainment/detrainment process in the PCMT resolved prognostic equations (as eq. 14). A protection against large time-step instability has been introduced: for any time-step the mixing process for the updraft variable $\overline{\psi_u}$ can neither give negative value of $\overline{\psi_u}$ nor exceeding $\overline{\psi_u} + \overline{\psi_r}$. So we have:

$$0 \leq \overline{\psi_u} + \frac{\partial \overline{\psi_u}}{\partial t}\Delta t \leq \overline{\psi_u} + \overline{\psi_r}$$

and the condition

$$-\overline{\psi_u} \leq \frac{\partial \overline{\psi_u}}{\partial t} \Delta t \leq \overline{\psi_r} \tag{29}$$

An equal rate for both turbulent entrainment and detrainment is assumed as in Tiedtke (1989). The ACMTENTR subroutine computes equation 27 and then limits the tendency in the range given by 29. $\overline{\psi_r}$ is the variable PPSIR whereas $\overline{\psi_u}$ is the variable PPSIC.

σ computation

Using equation 21, we can write:

$$\frac{1}{\sigma\omega_u}\frac{\partial \sigma\omega_u}{\partial p} = \epsilon_o - \delta_o \tag{30}$$

Note that the contribution of turbulent entrainment/detrainment is zero as the entrainment and detrainment rates are equal. The normalized convective fraction σ is computed thanks to this mass-conservation equations and decreases from value 1 at the surface. The computation is done in local variable ZSIGMA in acmtud.F90 subroutine. In the formula, for maximum organized entrainment rate, there is no variation of σ in the vertical.

3.3 Triggering condition

At this point of the algorithm, the triggering condition of the considered layer is determined by the sign of the convective vertical velocity at both flux levels surrounding the top variable level. If the sign of ω_u is negative, at least at one of the two flux levels, the layer is considered as convectively unstable. Then, in this scheme, it is possible to trigger convection even if CIN (Convective INhibition, corresponding to the amount of energy required to overcome the negatively buoyant energy the environment exerts on an air parcel, from the ground to the level of free convection) is not equal to zero. It is possible, using parameter GCVCINC (see below), to choose to add a threshold of CIN to trigger convection. Then only α needs to be known to close the system.

3.4 Closure hypothesis

A CAPE relaxation closure is used for that (corresponding to the choice $NCVCLOS = 1$, see below):

$$CAPE = \int_{P_{LFC}}^{P_{NET}} R_a(T_{vc} - \overline{T_v})\frac{dP}{P} \tag{31}$$

with P_{LFC} the pressure at the pressure level of free convection, P_{NET} the level of thermal equilibrium of the lifted parcel and R_a the gas constant of dry air. The closure hypothesis assumes an equilibrium between the $CAPE$ production by large-scale and sink by convection reached on a typical time-scale τ (see also part 3.5). We write:

$$\left(\frac{\partial CAPE}{\partial t}\right)_c = -\frac{CAPE}{\tau} \tag{32}$$

The relaxation time τ can be related to the adjustment time defined by Arakawa and Schubert (1974) as the "time required for convective processes to produce a neutral state, reducing the CAPE to zero, in the absence of large-scale forcing". Neglecting $\left(\frac{\partial T_{vc}}{\partial t}\right)_c$ (following Nordeng 1994)

$$\left(\frac{\partial CAPE}{\partial t}\right)_c = -R_a \int_{P_{NCL}}^{P_{NET}} \left((1+0.608\overline{q})(\frac{\partial \overline{T}}{\partial t})_c + 0.608\overline{T}(\frac{\partial \overline{q}}{\partial t})_c\right)\frac{dP}{P} \tag{33}$$

Then the equation 32 is linear with α (see equations 11 and 12) and CAPE is consumed by convection on a characteristic time proportional to the ratio of convective depth on the mean convective vertical velocity. The proportionality is defined with a linear function of the inverse of the resolution (i.e. linear function of the grid size) in order to keep the same magnitude between the mass flux and the resolved vertical velocity, which is itself proportional to the resolution (through the continuity equation).

$$\tau = f_\tau(resolution)\frac{\left(\int_{P_{NCL}}^{P_{NET}} dp\right)^2}{\int_{P_{NCL}}^{P_{NET}} |\omega_c| dp} \tag{34}$$

The resolution dependency is described below in more details below.

3.5 Resolution dependency

At higher horizontal resolution, convection becomes more and more explicitely resolved, and subgrid convection must decrease: mean subgrid mass flux on a given domain decreases to zero for infinite higher resolution.

In this approach, the convective response time τ (ZTAU in acmtud.F90 subroutine) writes:

$$\tau = \frac{1}{f_\tau}\frac{I_{12}^2}{I_{11}}$$

where : $\frac{1}{f_\tau}$ (ZFDXTAUX variable in acmtud.F90 subroutine) is a function of $TEQC$, PGM (scale factor) and $\frac{1}{\Delta x}$, tuned for climate resolutions (from T127 to T359):

$$\frac{1}{f_\tau} = TEQC.10^3(2.23.10^{-4}(\frac{PGM}{\Delta x})^2 - 0.425\frac{PGM}{\Delta x} + 252.5)$$

with :

- $TEQC$ (in m^{-1}) is linked with $REFLCAPE$ parameter through the mesh size (computed in sugem1b.F90 subroutine): $\texttt{TEQC} = \frac{\texttt{REFLCAPE}}{\Delta x}$
- $\frac{I_{12}^2}{I_{11}}$ is a response time, (unit s), I_{12} is the convective depth (variable ZS12 in the acmtud.F90 subroutine, in Pa) and I_{11} is the mean vertical velocity (in Pas^{-1}). Then the CAPE relaxation time τ is proportional to the convective depth and inversely proportional to the vertical velocity.

τ is used to compute the active fraction at the convective cloud bottom:

$$\alpha = min(ALFX, \frac{I_{15}}{I_{16}.\tau})$$

where I_{15} is the CAPE and I_{16} is such that the equation 32 linear versus α is $\left(\frac{1}{\tau}\frac{\partial CAPE}{\partial t}\right)_c = \alpha I_{16}$.

Then, for a constant horizontal resolution, for higher REFLCAPE, higher TEQC, smaller τ and higher α.

3.6 Parametrization of cumulus-scale downdraughts

A downdraught is a commonly observed feature of the circulation pattern of many convective systems: from an individual cumulus which may be associated with an adjacent area of rapidly descending air which may in turn be associated with strong surface winds and the formation of a pool of cold, low-lying air , to organized convective systems associated with coherent mesoscale streams of descending air. Downdraughts can have contrasting effects on the life cycle and organization of convective systems: on the one hand, it modifies the structure of the underlying boundary layer, which may dampen or inhibit convection. On the other, particularly when embedded within a sheared environment, the associated near-surface outflow can produce near-surface convergence and uplift that induces new convective elements. Convective cells generate precipitation that intensifies downdraughts by evaporation and by inertia. The downdraughts from each cell create a cold pool near the Earth's surface, which spreads out at the surface as a density current , triggering new convective cells and limiting the lifetime of the parent cells. Mesoscale subsidence is driven by evaporation of precipitation from the stratiform part (the anvil shield) and this reinforces the density current, which in turn helps to initiate a rear-to-front mid-level flow and the convective ascent with a jump updraught driven by a propagating positive pressure jump near the edge of the cold pool. The jump updraught is sensitive to the low-level vertical shear of the environment. A mass-flux scheme with an explicit cold-pool representation was created by Grandpeix and Lafore (2010). Circular cold pools (the wakes) are treated parametrically with cooling by the precipitating downdraughts while the outside area is warmed by the subsidence induced by the saturated draughts.

The current version of PCMT scheme includes only a diagnostic downdrafts parameterization described in (Gueremy 2011). The definition of the downdraught profile is very similar to that of the updraught: the bulk profile is defined starting from the top of the updraught with a saturated parcel having the thermodynamical properties of the environmental air. If this starting level is located below the level of minimum equivalent potential temperature, no downdraught is considered. In most cases, the downdraught starting level is actually the level of minimum equivalent potential temperature. Entrainment is taken into account and uses a turbulent entrainment rate equal to ϵ_{tx} and an organized entrainment rate defined with a value μ_0 equal to 1 if $\omega_{downdraft}$ is increasing with the pressure, 0 otherwise. Moist adiabatic process is considered until the updraught lifting condensation level where dry adiabat takes over. The vertical downdraught velocity is computed using the steady version of eq. 17. The downdraught triggering condition is the same as that of the updraught, with a change of sign. The same $\alpha\sigma$ value obtained for the updraught is used to get the downdraught mass flux. As the downdraught is taking its energy from evaporation of convective rain or condensate over a fractional area $\alpha\sigma/4$ (only one-fourth of the convective area is assumed to be affected by the evaporation process, according to SCM simulations, see parameter FEVAPC below), the downdraught mass flux is multiplied again by this value, and the value of the downdraught mass flux is equal to the opposite of $\omega_{downdraft}.(\alpha\sigma)^2/4$. This work is done in subroutine acmtddd.F90

3.7 Convective cloudiness

Convective cloudiness(Gueremy 2011) n_c is computed by $n_c = \alpha\,\sigma\,m$, where α is the convective fraction at the cloud bottom (obtained from the scheme closure), σ is the vertical variation of this convective buoyancy (linked with the buoyancy-sorting). The m constant describes the fact that, in convective clouds, cloudiness n_c is larger than the area occupied by the strongest ascents ($\alpha\,\sigma$).

Algorithmics

- In ACMTUD, convective cloudiness PNEBC is computed by PNEBC = $\alpha\,\sigma$ FNEBC (FNEBC = 10.). The cloudy condensate (liquid + ice) is PQLIC=ZLN * α * σ * FQLIC, where ZLN is the condensate issued from the diagnostic computation of ascent (FQLIC = 1),
- In ACPCMT, PNEBC and PQLIC are multiplied by PAIPCMT, a binary indicator: 1 if PCMT computes the shallow convection, 0 if an other shallow-convection scheme is used.

- In APLPAR: cloudiness named PNEBC in ACPCMT becomes ZNEBC0 in APLPAR. ZNEBC0 is used by ACNEBN. The condensate PQLIC in ACPCMT becomes ZQLIC_CVP in APLPAR.
- In ACNEBN: APLPAR/ZNEBC0 becomes ACNEBN/PNEBC. Two possible cases:
    1. where PAIPCMT equals 1, convective cloudiness PNEBC issued from PCMT is combined with resolved cloudiness, (logical key LGPCMT) according to max or random hypothesis (driven by logical key LRNUMX).
    2. where PAIPCMT equals 0., PNEB is computed as large-scale cloudiness + that of shallow convection.
- In APLPAR, total cloudiness PNEB may be multiplied by a constant value (GAEPS).
- Then PNEB is send to the radiation scheme to compute radiative effects.

3.8 Vertical transport

Vertical transport of prognostic subgrid and large-scale variables q_l and q_i by the subgrid updrafts, downdrafts, and their overturn circulation uses the scheme, developed by Yves Bouteloup et al. (2010) and coded in subroutine acadvec.F90. This semi-lagrangian scheme of probabilistic type enables to transport subgrid variables with CFL conditions greater than 1 in a stable and conservative manner, and constitutes a simplified approach of Geleyn (2008).

Description of the sedimentation scheme: equations of the scheme

As in Bouteloup et al. (2010), the reasoning is applied here to rain but could be extended to any precipitating species. The scheme appears as a generalization of the approach of Rotstayn (1997). The budget equation for rain water mass can be written as:

$$\frac{\partial q_r}{\partial t} = -\overrightarrow{V}.\overrightarrow{\nabla} q_r + \frac{1}{\rho}\frac{\partial}{\partial z}F_r + s_r^0 - s_r^i$$

with q_r the mixing ratio of rain, F_r the precipitation flux, ρ the density of air, $\overrightarrow{V}$ the wind and s_r^0 -s_r^i the sources and sinks of rain, respectively. For rain, sources are auto-conversion, collection and melting of snow (see chapter microphysics); sinks are evaporation and freezing. The advection term is computed by the dynamical core of the model. The equation to be solved by the microphysics scheme is:

$$\frac{\partial q_r}{\partial t} = \frac{1}{\rho}\frac{\partial}{\partial z}F_r + s_r^0 - s_r^i \tag{35}$$

If levels are supposed numbered from top to bottom (as in ARPEGE), the vertical and temporal discretization of eq. 35 gives:

$$q_r^+(j) = q_r^-(j) + \frac{\Delta t}{\rho \Delta z}(F_r(j-1) - F_r(j)) + S_r^0 - S_r^i$$

where Δz is the thickness of the current layer j, Δt the time-step, $S_r = \Delta t.s_r$, $F_r(j-1)$ the incoming precipitation flux at the top of the layer, $F_r(j)$ the output flux, q_r^- the value of the rain mixing ratio before microphysics (including sedimentation) and q_r^+ the value after microphysics. There are two unknown quantities in eq. 35, q_r and F_r; to solve the problem, F_r is generally related to q_r by:

$$F_r = \rho q_r V_r^t$$

where V_r^t is the mass-weighted bulk terminal fall velocity of rain. Then eq. 35 writes:

$$\frac{\partial q_r}{\partial t} = \frac{1}{\rho}\frac{\partial}{\partial z}(\rho q_r V_r^t) + s_r^0 - s_r^i$$

Equation 3.8 is not easy to solve because s_r^0, s_r^i and V_r^t are non-linear functions of q_r. Lopez (2002) gives a Lagragian solution. In many other schemes, eq. 3.8 is split into two equations, sequentially solved. The first one takes microphysical processes into account (without sedimentation):

$$\frac{\partial q_r}{\partial t}_{micro} = s_r^0 - s_r^i$$

while the second takes the sedimentation process alone into account:

$$\frac{\partial q_r}{\partial t}_{sed} = \frac{1}{\rho}\frac{\partial F_r}{\partial z}$$

All microphysical processes are computed before the sedimentation. This sequential approach requires small time-steps. Following Rotstayn (1997), it is possible to define a vertical discretized form of eq. 3.8 where the output flux is written as a function of q_r and V_r^t:

$$\frac{\partial q_r}{\partial t}(j) = \frac{F_r(j-1)}{\rho \Delta z} - q_r(j)\frac{V_r^t(j)}{\Delta z}$$

Assuming that the flux coming from the upper-level $F_r(j-1)$ is constant during the time-step, the integration of eq. 3.8 yields:

$$q_r^+(j) = q_r^-(j) exp(-C) + \frac{F_r(j-1)\Delta t}{\rho \Delta z}(1 - exp(-C))/C$$

where $C = (V_r^t \Delta t/\Delta z)$ is the sedimentation Courant number. It can be written as:

$$q_r^+(j) = (1 - P_1)q_r^-(j) + (1 - P_2)\frac{F_r(j-1)\Delta t}{\rho \Delta z}$$

with $P_1 = 1 - exp(-C)$ and $P_2 = 1 - P_1/C$. In the Bouteloup 's formulation, the vertical and temporal discretization of eq. 3.8 is written:

$$q_r^+(j) = q_r^-(j) + \frac{\Delta t}{\rho \Delta z}(F_r(j-1) - F_r(j))$$

Combining eqs. 3.8 et 3.8, a diagnostic equation for F_r can be obtained:

$$F_r(j) = P_1 \frac{\rho \Delta z}{\Delta t} q_r^-(j) + P_2 F_r(j-1)$$

Then the solution of the microphysics parameterization (microphysics + sedimentation) can be solved first by diagnostically computing F_r from the top to the bottom of the atmosphere without taking microphysical processes into account, using eq. 3.8, then by computing the temporal evolution of q_r through eq.3.8. So, for any precipitating species, the vertical transport flux is computed by vertical integration of:

$$g\frac{F(j-1) - F(j)}{\Delta p} = P_1 \frac{\psi(j)}{\Delta t} - (1 - P_2)\frac{g}{\Delta p}F(j) \qquad (36)$$

where P_1 is the probability for the top layer to receive, in a time-step, the total of the local quantity ($\psi_l \, \Delta p/g$, local source), and P_2 the probability for total transmission of the bottom flux to the top (non-local source).

Description of the sedimentation scheme: computation of the basic proportions.(Routine acadvec.F90)

If a set of raindrops with a terminal velocity V_r^t is considered, the proportion $P_0(z,t)$ of drops able to pass at least a vertical distance z during a time t is 0 if $\frac{V_r^t t}{z}$ less than 1, is 1 otherwise. For a layer of the model, containing rain and assuming an homogeneous vertical distribution, the proportion P_1 of rain which leaves the layer during the time-step Δt is $P_1 = \frac{1}{\Delta z}\int_0^{\Delta z} P_0(z,\Delta t)dz$. Secondly, if we consider drops of rain which are not in the layer under consideration at the beginning of the time-step but which come from the level above, the proportion P_2 of those which also leave the layer by the bottom during the time-step (assuming a constant incoming flux of rain mass from the level above) is: $P_2 = \frac{1}{\Delta t}\int_0^{\Delta t} P_0(\Delta z,t)dt$. Using the expression for P_0, it is found that:

$$P_1 = \min(1, C)$$

or

$$P_1 = \min(1, \frac{V_r^t\,\Delta t}{\Delta z})$$

with C the sedimentation Courant number. For P_2,

$$P_2 = \max(0, 1 - \frac{1}{C})$$

or

$$P_2 = \max(0,\ \min(e^{\frac{\Delta z\ \log\epsilon}{V_r^t\,\Delta t}},\ 1 - \frac{\Delta z}{V_r^t\,\Delta t}))$$

If .LADVLIM=.T., P_2 is limited to $e^{\frac{\Delta z\ \log\epsilon}{V_r^t\,\Delta t}}$ (with $\epsilon = 0.01$), to ensure that it is not possible to transport during one time-step on a vertical distance greater than $V_r^t\,\Delta t$. Then it is not possible to transport during one time-step on a vertical distance greater than $V_r^t\,\Delta t$ (LADVLIM=.FALSE. in our code).

3.9 Conclusion

Compared with Bougeault's scheme, the PCMT scheme provides an improvement of the distribution of precipitation regime and of convection diurnal cycle (Couvreux et al 2015), intensity and propagation of precipitation. Main defaults are a lack of triggering over continents with convective inhibition and an insufficient propagation, maybe linked with too large cumulated precipitation for intense convective events.

Table 3.9 gives the main characteristics of the different convection schemes used in MF models.

Table 10.9: Main characteristics of the deep and shallow convective schemes of MF

Convection scheme	Bougeault CMIP5	Bougeault NWP	PCMT
Closure of deep convection	Large-scale advection + moisture convergence = convective precipitation + detrainment	As Bougeault CMIP5	CAPE relaxation
Entrainment and Detrainment	Exponential decrease with height of entrainment;detrainment deduced from conservation of cloud moist static energy	As Bougeault CMIP5	Organized entrainment + turbulent entrainment
Triggering conditions	Moisture convergence + buoyancy	As Bougeault with a minimum cloud depth of 3 km	Prognostic equation for convective vertical velocity. Triggering if w>0
Downdraught	No	No	Yes-Diagnostic
Closure of shallow convection	No shallow scheme	CAPE relaxation (Kain-Fritsh-Betchold)	CAPE relaxation

4 Algorithmic

- Input of ACPCMT: prognostic convective variables of the preceeding time-step: w_u (variable PUDOM), w_d (variable PDDOM), α_u (variable PUDAL), α_d (variable PDDAL), q_{lc} (variable PQLCONV) , q_{ic} (PQICONV), q_{rc} (PQRCONV), q_{sc} (PQSCONV).
- From w_u, ACMTUD computes entrainment E and detrainment D, and their fractional analog ϵ and δ ($PENTR_U$, $PDETR_U$, $PENTR_D$,$PDETR_D$, entrainment/detrainment in updraft/downdraft in s^{-1}).
- ACMTUD uses ϵ and δ to compute the profile of a diagnostic ascent, the associated buoyancy, the new value of w_u through a prognostic equation, α_u through a prognostic equation, and gives the condensation rate of equation 14
- ACMTENTR computes the tendency of prognostic variables with the entrainment/detrainment process.
- ACPCMT updates the intra time-step prognostic variables with condensation and entrainment/detrainment
- ACPLUIZ computes microphysics (autoconversion, collection, precipitation evaporation).
- ACADVEC computes non-precipitating condensate transport: q_{lc} and q_{ic} subgrid and resolved.
- ACPCMT updates intra time-step prognostic variables with transport.
- ACPCMT computes freezing/melting of q_{lc}, q_{ic} after transport.
- ACPCMT updates all the fluxes of the prognostic equations for q_{lc}, q_{ic}, q_{rc}, q_{sc}, if a shallow convection scheme, different from PCMT, is activated (like that of current NWP ARPEGE -Kain-Fritsch-Bechtold- or PMMC09, Arome's one) and if the convective depth of the PCMT updraft is smaller than a threshold (THPCMT).
- As for the other parametrization schemes, all the processes are described as fluxes as output of the subroutines. These fluxes are read in the subroutin CPTEND_NEW which computes tendencies of prognostic variables.

5 Logical keys and main parameters in the current version

Logical keys

- NCVCLOS: type of convective closure (α computation: NCVCLOS=1, CAPE relaxation)
- NCVENTR: type of entrainment (NCVENTR=1, GuÃ©rÃ©my, Tellus, 2011)
- NCVSIG: type of σ computation (NCVSIG=1, equation 30)
- LADVLIM: in subroutine acadvec.F90, minimises $p2$ to the value such as the transport scheme can not transport beyond a distance of $w.\Delta t$ in a time-step (LADVLIM=.F.)
- LGDDD: activates diagnostic downdrafts (LGDDD=.TRUE.)
- LGMT: activates separate microphysics and transport resolved-scale tendencies (LGMT=.TRUE.)
- LGPCMT: for using PCMT scheme (LGPCMT=.T.)
- LGPSMI: activates Smith-Type convective precipitation (LGPSMI=.F.)
- LCVFEVV: feedback of evaporation on vertical velocity (LCVFEVV=.F.)
- LCVCONTAU: control response time of convection min/max values (LCVCONTAU=.F.)
- LCVEOD: activates large entrainment/detrainment rates below or over the cloud (LCVEOD=.F.)

- LCVIMPT: implicit transport (LCVIMPT=.F.)
- LCVNAUV: new algorithmics to compute U and V convective effects (LCVNAUV=.F.)
- LCVNHD: activates non-hydrostatic drafts dynamics (LCVNHD=.F.)
- LCVUVM: estimates the U and V updraft values as uniform, equal to the mean of PU (LCVUVM=.F.)
- LEDKFI: key for EDMF concept (LEDKFI=.TRUE.)
- LRNUMX: key for maximum overlap of adjacent radiative clouds (LRNUMX=.T.)
- LSBUO: smooth buoyancy (LSBUO=.F.)

Main parameters used in the current climate version

- ALFX: maximum convective updraft fraction (ALFX= 0.04)
- ECMNP: minimum critical thickness for precipitating clouds (ECMNP=8000 m)
- FENTRT: factor of turbulent entrainment at detrainment level (FENTRT=1.75)
- FEVAPC: factor for precipitation evaporation (FEVAPC=4)
- FNEBC: factor for convective cloudiness (FNEBC=10)
- FQLIC: factor for convective liquid water (FQLIC=1)
- GAEPS: constant used in APLPAR to multiply total cloudiness issued from ACNEBN (GAEPS=1).
- GAMAP1: virtual mass parameter +1: parameter γ+1 of equation 17 (GAMAP1= 1.5)
- GCLOEB: link between cloud depth and effective transport (GCLOEB=0)
- GCVCINC: CIN threshold for triggering condition (GCVINC=0 Jkg^{-1}, a threshold of 5 Jkg^{-1} between LCL and LFC corresponds to 0.5 K of difference between cloud and environment)
- GCVADET: fraction of zero buoyancy variable in the detrainment (GCVADET=1)
- GCVADS: Coefficient to switch from an equipressure adiabat computation to an equipotential one (GCVADS=0)
- GCVRE: Reduce convective entrainment (GCVRE=1)
- GFRIC: aerodynamic of wu and wd (GFRIC=$4.2 * 10^{-3}$, used only for diagnostic)
- GFSURF: fraction of surface vs lowest model in the computation of diagnostic updraft ascent initial state (GFSURF=0, meaning that the profile starts at lowest model level)
- GREDDRS: reduction of detrainment for convective rain and snow into the environment (parameter without unit between 0 and 1, GREDDRS=1 for no reduction).
- GREMAX: maximum value of the "reducing entrainment factor" (GREMAX=1.0, not used if GCLOEB=0)
- GREMIN:minimum value of the "reducing entrainment factor" (GREMIN=0,01, not used if GCLOEB =0)
- GSDMAX: maximum value of saturation deficit, to compute entrainment reduction (GSDMAX=0.13E-02 $kg.m^{-2}$), not used if GCLOEB=0
- GSDMIN: minimum value of saturation deficit, to compute entrainment reduction (GSDMIN=0.1E-02 $kg.m^{-2}$), not used if GCLOEB=0
- PAIPCMT: Binary indicator for treatment of shallow-convection (1 if PCMT)
- RDTFAC: Modulation factor of "RDT" in "FONICE" function (RDTFAC=0.5)
- REFLCAPE: Parameter for convective response time (REFLCAPE =1)
- RKDN: minimum aerodynamic friction ($RKDN = 3.8 * 10^{-5} Pa^{-1}$)
- SCO: threshold under which convective precipitation are not taken into account (SCO=-1 kgm^{-2})
- TENTR: convective updraft entrainment rate ($0.5 * 10^{-5} Pa^{-1}$)

- TENTRX: maximum convective updraft's entrainment rate ($0.47 * 10^{-4} Pa^{-1}$)
- THPCMT: threshold value, that defines the activation of PCMT versus shallow convection schemes (THPCMT=-1 as PCMT is used for both convections in this configuration)
- TUDGP: Updraught horizontal pressure gradient effect coefficient (Kershaw and Gregory) (TUDGP= 0)
- TVFC: parameter for convective precipitation (TVFC= 1, not used if LCVNAUV=.F.)
- VVN: minimum vertical velocity for turbulent entrainment (VVN=-2 Pa/s)
- VVX: maximum vertical velocity for turbulent entrainment (VVX= -27.5 Pa/s)

11

Gravity wave drag

1 Parametrization of non-orographic gravity wave drag

This parametrization of non-orographic gravity wave drag is mainly described in Lott *et al.* (2012) and is treated by subroutine ACNORGWD.

1.1 Formalism

A broadband spectrum of gravity waves (GWs) is represented via the superposition of a large ensemble of statistically independent monochromatic ones. To produce this large ensemble at a reasonable numerical cost, one launchs at each model time step δt a finite number M (typically 8) of waves with characteristics chosen randomly and compute the tendencies due to them. As they are independent realizations the averaged tendency they produce (the gravity wave drag $(\partial_t u)_{gw}$) is the average of these M tendencies. One takes into account that the lifetime of these waves (typically $\Delta t \sim 1$ day) exceeds the model time step (typically $\delta t < 1$h).

We then redistribute this averaged tendency over a longer time scale Δt by first rescaling it by $\delta t/\Delta t$ and second by using a lag-one auto-regressive (AR-1) relation between the GW tendencies at two successive time steps:

$$(\partial_t u)_{gw}^{t+\delta t} = \frac{\Delta t - \delta t}{\Delta t} (\partial_t u)_{gw}^{t} + \frac{\delta t}{\Delta t} \frac{1}{M} \sum_{n=1}^{M} \frac{1}{\rho} \frac{\partial F_n}{\partial z} \tag{1}$$

In other words, and at each time step, M new waves are emitted, and ones reduces the probabilities of all the others by the multiplicative factor $(\Delta t - \delta t)/\Delta t$. A few hundred waves are then active at each model step, which gives an excellent spectral resolution at a reasonable computational cost.

At each time step, the GWs horizontal wave number is choosed randomly within the interval $k_{min} \leq k \leq k_{max}$, with k_{max} and k_{min} are related to the grid dimensions. The intrinsic phase speed is also randomly chosen from a Gaussian distribution of mean 0 and standard deviation c_{max}.

1.2 Launch spectrum

The Eliassen-Palm (EP) momentum flux carried by each wave is specified at a given launching altitude z_l by

$$\mathbf{F}(z_l) = \rho_r G_p \left(\frac{R L_w}{\rho_r H c_p}\right)^2 \frac{k^2 e^{-(m\Delta z_p)^2}}{N\Omega^3} P^2 \frac{\mathbf{k}}{k} + G_f \frac{\Delta z_f}{4f} \int_0^{z_{top}} \rho_0(z) N(z) \zeta^2(z) e^{-\pi\sqrt{J(z)}} dz \frac{\mathbf{k}}{k} \quad (2)$$

The first term in the right-hand side of equation (2) represents the contribution from convective sources. The second term in the right-hand side represents the contribution from the background flux and accounts for GWs from nonconvective (and nonorographic) sources.

To relate the gravity waves to the convective forcing, the surface precipitation P is used. And to translate precipitation into diabatic heating, the latent heat flux produced by precipitation is distributed into the vertical over a Gaussian distribution with standard deviation Δz_p, that characterizes the heating depth (see Lott *et al.* (2013) for mathematical derivation). In (2), $\rho_r = 1$ kg.m^{-3} is the reference density, L_w is the latent heat of condensation, c_p is the heat capacity at constant pressure, H is the characteristic height of the atmosphere, N is the Brunt-Vaisala frequency, $k = ||\mathbf{k}||$ is the horizontal wavenumber, Ω is the intrinsic frequency $\Omega = \omega - \mathbf{k}.\mathbf{u}$ and m is the vertical wavenumer $m = Nk/\Omega$. G_p is a tuning parameter of order 1.

To relate the gravity waves to frontal sources, the potential vorticity ζ is used (see De la Camara *et al.* (2015) for details). In (2), f is the Coriolis parameter, z_{top} is the model top altitude, $\rho_0 = \rho_r e^{-z/H}$ is the background density, $J = N^2/\Lambda^2$ is the Richardson number (Λ is the vertical shear of the horizontal wind), Δz_f is the tunable depth of the frontal source, and G_f is also a tunable parameter of order 1 that controls the amplitude of the EP flux.

1.3 Vertical propagation

First, the momentum flux carried by gravity waves is set to zero where the waves encounter a critical level. In the absence of critical levels, the momentum flux remains nearly constant except for a small dissipation μ/ρ_0 in order to guarantee that the waves are finally erased over the last few model layers. The GW momentum flux is also limited by that produced by a saturated monochromatic wave. The passage of EP flux from one level z to the next $z + \delta z$ can be written as

$$\mathbf{F}(z+\delta z) = \frac{\mathbf{k}}{k} \Theta(\Omega(z)\Omega(z+\delta z)) \min\left(|\mathbf{F}(z)| e^{-2\frac{\mu m^3}{\rho_0 \Omega}\delta z}, \rho_r S_c \frac{|\Omega|^3 k_{min}^2}{N k^4}\right) \quad (3)$$

where the Heaviside function Θ handles critical levels and S_c is a tunable parameter controlling the saturated momentum flux.

1.4 Computation aspects

Subroutine ACNORGWD is active under LNORGWD logical key and calculates the gravity wave drag $(\mathrm{PTEND\_U}, \mathrm{PTEND\_V}) = ((\partial_t u)_{gw}, (\partial_t v)_{gw})$ from the vertical profiles of temperature PTT, zonal wind PUU, meridional wind PVV and potential vorticity PVOVO. The total (stratiform plus convective) precipitation PPREC is also used as a input of the subroutine.

The parameters of the scheme are defined in the module YOMNORGWD.

- NORGWD_RDISS is the dissipation coefficient (μ).

- NORGWD_SAT is the saturation parameter (S_c).
- NORGWD_KMIN is the mininal horizontal wavenumber (k_{min}).
- NORGWD_KMAX is the maximal horizontal wavenumber (k_{max}).
- NORGWD_CMAX is the 'maximal' intrinsic phase velocity (c_{max}).
- NORGWD_DELTAT is the time scale (Δt) of the life cycle of the waves parameterized
- NORGWD_DZ is the characteristic depth of the convective sources (Δz_p).
- NORGWD_RUWMAX is the tunable parameter that controls the amplitude of the EP flux from convective GWs (G_p).
- NORGWD_DZFRON : is the characteristic depth of the frontal sources (Δz_f).
- NORGWD_GFRON is the tunable parameter that controls the amplitude of the EP flux emitted by frontal GWs (G_f).

12

Ozone

1 Default ozone

When ozone is not a historical variable advected by dynamics and modified by photochemistry, it is simply specified by a climatological monthly file interpolated onto model levels from a climatology coming from University of Reading calculated by Li and Shine (1995). See:

http://badc.nerc.ac.uk/data/ugamp-o3-climatology/ugamp_help.html

2 Parameterization of photochemical ozone sources and sinks

The explicit representation of stratospheric photochemistry is too complex to be able to be introduced into a general circulation model. This is why one uses a linearization of the terms of ozone sources and sinks starting from a 2d latitude-pressure model (Cariolle and Déqué, 1986). The 2d model MOBIDYC (64 latitudes and 40 levels pressure) utilized a zonal circulation of stratosphere and 168 chemical reactions concerning 59 components (with in addition 51 reactions of photo-dissociation). The 2d model reaches an equilibrium at the end of 30 years of integration (certain reactions have characteristic times of several years). Stating at the equilibrium situation, one advances one time step ahead in each grid point of the photochemical model, after having disturbed by ±10% the ozone mixing ratio r_{O3}. One thus obtains the derivative of the term of ozone photochemical production or destruction $P-L$ versus r_{O3}. This derivative can be regarded as a relaxation coefficient of the ozone field. Time characteristic of this relaxation (calculated for each latitude and each level of the model 2d and each month of the year) varies from 0.1 day towards 1 hPa to 1 year at the tropopause. One calculates in same manner the derivative of $P-L$ versus temperature T. Indeed the rates of the chemical reactions (Chapman cycle) depend on temperature. One calculates finally the derivative with respect to ozone thickness above the grid point, Σ_{O3}. When this quantity decreases, ultraviolet flux reaching the point increases, and the production of ozone due to photo-dissociation increases.

The following linear model is thus considered:

$$\begin{aligned}\frac{\partial r_{O3}}{\partial t} &= \overline{P-L} + \overline{\frac{\partial P-L}{\partial r_{O3}}}\left[r_{O3} - \overline{r_{O3}}\right] \\ &\quad + \overline{\frac{\partial P-L}{\partial T}}\left[T - \overline{T}\right] + \overline{\frac{\partial P-L}{\partial \Sigma_{O3}}}\left[\Sigma_{O3} - \overline{\Sigma_{O3}}\right] \qquad (1)\end{aligned}$$

where the over-lined quantities were obtained using the 2d model MOBIDYC. This linear model was substituted for the photochemical model in the 2d model, and the results of the two versions were compared. The relative error remains lower than 10%; the maximum error is in low stratosphere where chemistry is strongly non-linear (just as in troposphere, but the concentrations are weak). The linearized model is thus validated compared to the full model from which it results.

Linear parameterization thus utilizes the 7 coefficients of Equation (1) which are regarded as two-dimensional fields as long as the poles are not tilted, three-dimensional otherwise.

Subroutine ACOZONE calculates initially the ozone quantity above each point:

$$\Sigma_{O3}(l) = \sum_{i=0}^{l-1} \frac{p(\widetilde{i}) - p(\widetilde{i-1})}{g} r_{O3}(i) \ + \ \frac{p(l) - p(\tilde{l}-1)}{g} r_{O3}(l)$$

The same algorithm is used for term $\overline{\Sigma_{O3}}$ starting from $\overline{r_{O3}}$ to ensure the coherence of the vertical discretization of the model. Then, one calculates the ozone tendency of Equation (1) which is written as:

$$\frac{\partial r_{O3}}{\partial t} = PK_2 + PK_3(r_{O3} - PK_1) + PK_5(T - PK_4) + PK_7(\Sigma_{O3} - \overline{\Sigma_{O3}})$$

with:

$$\overline{\Sigma_{O3}(l)} = \sum_{i=0}^{l-1} \frac{p(\widetilde{i}) - p(\widetilde{i-1})}{g} PK_1(i) \ + \ \frac{p(l) - p(\tilde{l}-1)}{g} PK_1(l)$$

In fact, the way it is written is less simple as one uses an implicit temporal discretization for the term in r_{O3} (but not for Σ_{O3} because it would be too complicated) to ensure numerical stability with large time steps.

Finally one calculates the ozone flow per vertical integration of the temporal tendencies which one has just calculated and with the assumption of null flow at the top.

3 The effect of chlorine on ozone

Parameterization above does not make it possible to calculate the effect of the application of the Montreal protocol on the reduction of Clx in stratosphere. Heterogeneous chemistry responsible for "ozone hole" is very complex and here a particularly simplified formulation is used. When the solar zenith angle is positive (*i.e.* during daylight) and when temperature is lower than TPSCLIM, namely 195 K, one adds to the PK_3 term the quantity $PK_6\,\mathrm{RCLX}^2$, where PK_6 is a new 2d field and RCLX is a parameter whose temporal evolution is controlled by `namelist`. There is no compatibility with the former versions of the scheme for which fields PK_1 to PK_7 corresponded to other coefficients of Equation (1) (caution: do not mix boundary conditions files!).

4 Parameterization of "mesospheric drag"

The goal of this parameterization is to mitigate the absence of physics in the highest levels of the model (*i.e.* in the mesosphere) when the stratospheric vertical discretization is high. This parameterization consists simply of a linear relaxation of the wind towards 0, of specific moisture towards $q_{min} = 3.725\,10^{-6}$ (RFMESOQ) to avoid excessive drying and represent chemical sources of H$_2$O and of the temperature towards the temperature of standard atmosphere T_{sta} (Fels, 1986) defined in subroutine SUSTA by:

$$\left[\begin{array}{ll}
z = 0 & T_{sta} = 288.15\,K \\
0 < z < 11.0\,km & \dfrac{\partial T_{sta}}{\partial z} = -6.5\,K\,km^{-1} \\
11.0\,km < z < 20.0\,km & \dfrac{\partial T_{sta}}{\partial z} = 0 \\
20.0\,km < z < 32.0\,km & \dfrac{\partial T_{sta}}{\partial z} = 1.00\,K\,km^{-1} \\
32.0\,km < z < 47.4\,km & \dfrac{\partial T_{sta}}{\partial z} = 2.75\,K\,km^{-1} \\
47.4\,km < z < 51.4\,km & \dfrac{\partial T_{sta}}{\partial z} = 0 \\
51.4\,km < z < 71.7\,km & \dfrac{\partial T_{sta}}{\partial z} = -2.75\,K\,km^{-1} \\
71.7\,km < z < 85.7\,km & \dfrac{\partial T_{sta}}{\partial z} = -1.97\,K\,km^{-1} \\
85.7\,km < z & \dfrac{\partial T_{sta}}{\partial z} = 0
\end{array}\right.$$

The profile of the relaxation coefficients $K(l)$ is defined in subroutine SUTOPH. One chooses for the wind a reference level pressure p_{ref} and a coefficient α. In the same way, for moisture and temperature, one takes another set of coefficients. Profile $K(l)$ is then:

$$K(l) = \alpha \ \max\left(\frac{p_{ref} - p_{sta}(l)}{p_{sta}(l)}, 0\right)$$

where $p_{sta}(l)$ is the standard air pressure at level l. In practice, these two levels are selected above 1 hPa and the constants are adjusted in order to obtain time-constants of a few hours in the highest level.

The writing of linear parameterization is a little complicated by the fact that on the one hand an implicit discretization is used, and that on the other hand one must calculate a flux instead of a tendency. For example, the enthalpy flux at inter-level l is given by:

$$\text{PFRMH} = -\frac{1}{g}\sum_{i=l+1}^{KLEV} \delta p(i)\frac{K(i)}{1+\Delta t K(i)} c_p(i)(T(i) - T_{sta}(i))$$

where c_p is the specific heat of the air, g gravity, $KLEV$ the number of levels, $\delta p(i)$ the thickness of the layer, Δt the time step of physics and $T(i)$ the temperature. Contrary to fluxes produced by the other parameterizations, this flux is zero at surface. The momentum and energy exchange is done with space and not with earth.

References

- Arakawa, A. (2004). The cumulus parameterization problem: Past,present,and future. *J. Climate*, 17:2493–2525.
- Arakawa, A. and Schubert, W. H. (1974). Interaction of a cumulus cloud ensemble with the large-scale environment,part i. *J. Atmos. Sci.*, 31:674–701.
- Bechtold, P., Bazile, E., Guichard, F., Mascart, P., and Richard, E. (2001). A mass-flux convection scheme for regional and global models. *Quart. J. Roy. Meteor. Soc.*, 127:869–886.
- Berry, E. (1967). Cloud drop growth by coalescence. *J. Atmos. Sci.*, 24:688–701.
- Berry, E. and Reinhardt, R. (1974). An analysis of cloud drop growth by collection. part i and ii. *J. Atmos. Sci.*, 31:1814–1831.
- Bougeault, P. (1981). Modeling the trade-wind cumulus boundary layer. part i: testing the ensemble cloud relations against numerical data. *J. Atmos. Sci.*, 38:2414–2428.
- Bougeault, P. (1985a). A simple parameterization of the large-scale effects of cumulus convection. *Mon. Wea. Rev.*, 113:2108–2121.
- Bougeault, P. (1985b). A simple parameterization of the large-scale effects of deep cumulus convection. *Mon. Weather Rev.*, 113:2108–2121.
- Bougeault, P. and Lacarrère, P. (1989). Parametrization of orography-induced turbulence in a mesobetascale model. *Mon. Weather Rev.*, 117:1872–1890.
- Bouteloup, Y., Seity, Y., and Bazile, E. (2010). Description of the sedimentation scheme used operationally in all météo-france nwp models. *Tellus*, 63A:300–311.
- Bouyssel, F., Bouteloup, Y., and Marquet, P. (2005). Toward an operational implementation of lopezś prognostic large scale cloud and precipitation scheme in arpege/aladin nwp models. In *HIRLAM workshop on convection and clouds,Tartu,24-26 January 2005.*
- Bretherton, C. S., McCaa, J. R., and Grenier, H. (2004). A new parameterization for shallow cumulus convection and its application to marine subtropical cloud-topped boundary layers. part i: description and 1d results. *Mon. Weather Rev.*, 132:864–882.
- Cariolle, D. and Déqué, M. (1986). Southern hemisphere medium-scale waves and total ozone disturbances in a spectral general circulation model. *J. Geophys. Res.*, 91:10825–10846.
- Cariolle, D. and Teyssèdre, H. (2007). A revised linear ozone photochemistry parameterization for use in transport and general circulation models: multi-annual simulations. *Atmospheric chemistry and physics*, 7:2183–2196.
- Carrer, D., Meurey, C., Ceamanos, X., Roujean, J.-L., Calvet, J.-C., and Liu, S. (2014). Dynamic mapping of snow-free vegetation and bare soil albedos at global 1km scale from 10-year analysis of modis satellite products. *Remote Sensing of Environment*, 140:420–432.
- Clough, S. and Iacono, M. (1995). Line-by-line calculation of atmospheric fluxes and cooling rates. application to carbon dioxide, ozone, methane, nitrous oxide and the halocarbons. *J. Geophys. Res.*, 100D:16519–16536.

- Clough, S., Iacono, M., and Moncet, J. (1992). Line-by-line calculation of atmospheric fluxes and cooling rates: Application to water vapor. *J. Geophys. Res.*, 97D:15761–15786.
- Clough, S., Kneizys, F., and Davies, R. (1989). Line shape and the water vapor continuum. *Atmos. Res.*, 23:229–241.
- Coakley, J. and Chylek, J. (1975). The two-stream approximation in radiation transfer: Including the angle of the incident radiation. *J. Atmos. Sci.*, 32:409–418.
- Cohard, J.-M. and Pinty, J.-P. (2000). A comprehensive two-moment warm microphysical bulk scheme.part i and ii. *Quart. J. Roy. Meteor. Soc.*, 126:1843–1859.
- Cotton, W., Tripoli, G., and Rauber, R.M.and Mulvihill, E. (1986). Numerical simulation of the effects of varying ice crystal nucleation rates and aggregation processes on orographic snowfall. *J. Clim. Appl.Meteorol.*, 25:1658–1680.
- Courtier, P. and Geleyn, J. (1988). A global numerical weather prediction model with variable resolution: Application to the shallow water equations. *Quart. J. Roy. Meteor. Soc.*, 114:1321–1346.
- Couvreux, F., Roehrig, R., Rio, C., Lefebvre, M., Caian, M., Komori, T., Derbyshire, S., Guichard, F., Favot, F., DAndrea, F., Bechtold, P., and Gentine, P. (2015). Daytime moist convection over the semi-arid tropics : impact of parametrizations used in cmip5 and other models. *QJRLS*, 141:2220–2236.
- Cox, G. P. (1988). Modeling precipitation in frontal rainbands. *Quart. J. Roy. Meteor. Soc.*, 114:115–127.
- Cuxart, J., Bougeault, P., and Redelsperger, J.-L. (2000). A turbulence scheme allowing for mesoscale and large-eddy simulations. *Quart. J. Roy. Meteor. Soc.*, 126:1–30.
- de la Camara, A. and Lott, F. (2015). A parameterization of gravity waves emitted by fronts and jets. *Geophys. Res. Lett.*, 42.
- Derbyshire, S. H., Beau, I., Bechtold, P., Grandpeix, J.-Y., Piriou, J.-M., Redelsperger, J.-L., and Soares, P. M. M. (2004). Sensitivity of moist convection to environmental humidity. *Quart. J. Roy. Meteor. Soc.*, 130:3055–3079.
- Deschamps, P., Herman, M., and Tanré, D. (1983). Modélisation du rayonnement solaire réfléchi par l'atmosphère et la terre, entre 0,35 et 4 microns. *Rapport ESA 4393/80/F/DD(SC)*, ??:156 pp.
- Douville, H., Royer, J., and Mahfouf, J. (1995). A new snow parameterization for the météo-france climate model. Part I: Validation and stand alone experiment. *Climate Dynamics*, 12:21–35.
- Dubuisson, P., Buriez, J., and Fouquart, Y. (1996). High spectral resolution solar radiative transfer in absorbing and scattering media: application to the satellite simulation. *J. Quant. Spectrosc. Radiat. Transfer*, 55:103–126.
- Duynkerke, P. G., de Roode, S. R., Zanten, M. C. V., Calvo, J., Cuxart, J., Cheinet, S., Chlond, A., Grenier, H., Jonker, P. J., Köhler, M., Lenderink, G., Lewellen, D., Lappen, C.-L., Lock, A. P., Moeng, C.-H., Müller, F., Olmeda, D., Piriou, J.-M., Sanchez, E., and Sednev, I. (2004). Observations and numerical simulations of the diurnal cycle of the eurocs stratocumulus case. *Quart. J. Roy. Meteor. Soc.*, 130:3269–3296.
- Ebert, E. and Curry, E. (1992). A parameterization of ice cloud optical properties for climate models. *J. Geophys. Res.*, 97D:3831–3836.
- Fels, S. (1986). Analytic representations of standard atmosphere temperature profiles. *J. Atmos. Sci.*, 43:219–221.
- Ferrier, B. (1993). A double-moment multiple-phase four-class bulk ice scheme. part 1: Description. *J. Atmos. Sci.*, 51:249–280.
- Foote, G. B. and Du Toit, P. S. (1969). Terminal velocity of raindrops aloft. *J. Appl. Meteor.*, 8:249–253.
- Fouquart, Y. (1974). Utilisation des approximants de padé pour l'étude des largeurs équivalentes des raies formes en atmosphèrere diffusante. *J. Quant. Spectrosc. Radiat. Transfer*, 14:497–506.

- Fouquart, Y. and Bonnel, B. (1980). Computations of solar heating of the earth's atmosphere: A new parametrization. *Beitr. Phys. Atmosph.*, 53:35–62.
- Fowler, L., Randall, D., and Rutledge, S. (1996). Liquid and ice cloud microphysics in the csu general circulation model. part i: model description and simulated microphysical processes. *J. Climate*, 9:489–529.
- Fu, Q. (1996). An accurate parameterization of the solar radiative properties of cirrus clouds for climate models. *J. Climate*, 9:2058–2082.
- Fu, Q. and Liou, K. (1992). On the correlated k-distribution method for radiative transfer in non homogeneous atmospheres. *J. Atmos. Sci.*, 49:2139–2156.
- Fu, Q. and Liou, K. (1993). Parameterization of the radiative properties of cirrus clouds. *J. Atmos. Sci.*, 50:2008–2008.
- Fu, Q. and Sun, W. (2001). Retrieval of cirrus particle sizes using a split-window technique: a sensitivity study. *J. Quant. Spectrosc. Radiat. Transfer*, 68:725–736.
- Fu, Q., Yang, P., and Sun, W. (1998). An accurate parametrization of the infrared radiative properties of cirrus clouds of climate models. *J. Climate*, 11:2223–2237.
- Geleyn, J. and Hollingsworth, A. (1979). An economical analytic method for the computation of the interaction between scattering and line absorption of radiation. *Beitr. Phys. Atmosph.*, 52:1–16.
- Geleyn, J.-F., Catry, B., Bouteloup, Y., and Brozkova, R. (2008). A statistical approach for sedimentation inside a microphysical precipitation scheme. *Tellus*, 60:649–662.
- Gerard, L. and Geleyn, J.-F. (2005). Evolution of a subgrid deep convection parameterization in a limited area model with increasing resolution. *Quart. J. Roy. Meteor. Soc.*, 131:2293–2312.
- Grandpeix, J.-Y. and Lafore, J.-P. (2009). A density current parameterization coupled with emanuel's convection scheme. part i: The models. *Journal of the Atmospheric Sciences,accepted*.
- Guichard, F., Petch, J. C., Redelsperger, J.-L., Bechtold, P., Chaboureau, J.-P., Cheinet, S., Grabowski, W., Grenier, H., Jones, C. J., Koehler, M., Piriou, J.-M., Tailleux, R., and Tomasini, M. (2004). Modelling the diurnal cycle of deep convection over land with cloud-resolving models and single-column models. *Quart. J. Roy. Meteor. Soc.*, 130:3139–3172.
- Gunn, K. L. S. and Marhall, J. S. (1958). The distribution with size of aggregate snow flakes. *J.Meteor.*, 15:452–461.
- Guérémy, J.-F. (2011). A continuous buoyancy based convection scheme: one- and three dimensional validation. doi 10.1111/j.1600-0870.2011.00521.x. *Tellus*, 63A:687–706.
- Heymsfield, S. (1977). Precipitation development in stratiform ice clouds: a microphysical and dynamical study. *J. Atmos. Sci.*, 34:367–381.
- Hortal, M. (1996). Experiments with the linear gaussian grid at ecmwf. *Research Activities in Atmospheric and Oceanic Modelling*, 23:17–19.
- Hortal, M. and Simmons, A. (1991). Use of reduced gaussian grids in spectral models. *Mon. Wea. Rev.*, 119:1057–1074.
- Houghton, H. G. and Cramer, H. E. (1951). A theory of entrainment in convective currents. *J. Atmos. Sci.*, 8:95–102.
- Hourdin, F., Couvreux, F., and Menut, L. (2002). Parameterization of the dry convective boundary layer based on a mass flux representation of thermals. *J. Atmos. Sci.*, 59:1105–1123.
- Houze, R., Hobbs, P., Herzegh, P., and Parsons, D. (1979). Size distributions of precipitation particles in frontal clouds. *J. Atmos. Sci.*, 36:156–162.
- Joseph, J., Wiscombe, W., and Weinman, J. (1976). The Delta-Eddington approximation for radiative flux transfer. *J. Atmos. Sci.*, 33:2452–2459.
- Kain, J. and Fritsch, J. (1993). Convective parameterization for mesoscale models: The kain-fritsch scheme. *The Representation of Cumulus Convection in Numerical Models,Meteor. Monogr.,Amer. Meteor. Soc*, 24:165–170.

- Kessler, E. (1969). On the distribution and continuity of water substance in atmospheric circulation. In *Meteor. Monogr*, chapter 32, page 84. Amer. Meteor. Soc.
- Lacis, A. and Oinas, V. (1991). A description of the correlated k distribution method for modeling nongray gaseous absorption, thermal emission, and multiple scattering in vertically inhomogeneous atmospheres. *J. Geophys. Res.*, 96D:9027–9063.
- Lenderink, G. and Holtslag, A. (2004). An updated length-scale formulation for turbulent mixing in clear and cloudy boundary layers. *Quart. J. Roy. Meteor. Soc.*, 130:3405–3427.
- Lim, K.-S. S. and Hong, S.-Y. (2010). Development of an effective double-moment cloud microphysics scheme with prognostic cloud condensation nuclei (ccn) for weather and climate models. *Mon. Weather Rev.*, 138:1587–1612.
- Lin, Y.-L., Farley, R. D., and Orville, H. D. (1983). Bulk parameterization of the snow field in a cloud model. *J. Clim. Appl. Meteorol.*, 22:1065–1092.
- Lindner, T. and Li, J. (2000). Parametrization of the optical properties for water clouds in the infrared. *J. Climate*, 13:1797–1805.
- Lopez, P. (2002). Implementation and validation of a new prognostic large-scale cloud and precipitation scgeme for climate and data-assimilation purposes. *Quart. J. Roy. Meteor. Soc.*, 128:229–257.
- Lott, F. and Guez, L. (2013). A stochastic parameterization of the gravity waves due to convection and its impact on the equatorial stratosphere. *J. of Geophys. Res.*, 118:8897–8909.
- Lott, F., Guez, L., and Maury, P. (2012). A stochastic parameterization of non-orographic gravity waves, formalism and impact on the equatorial stratosphere. *Geophysical Res. Letters*, 39:–.
- Marshall, J. S. and Palmer, W. M. (1948). The distribution of raindrops with size. *J. Meteor.*, 50:165–166.
- Martin, G., Johnson, D., and Spice, A. (1994). The measurement and parameterization of effective radius of droplets in warm stratocumulus. *J. Atmos. Sci.*, 51:1823–1842.
- Milbrandt, J. and Yau, M. (2005). A multimoment bulk microphysics parameterization. part i and ii. *J. Atmos. Sci.*, 62:3065–3081.
- Mlawer, E., Taubman, S., Brown, P., Iacono, M., and Clough, S. (1997). Radiative transfer for inhomogeneous atmospheres: RRTM, a validated correlated-k model for the longwave. *J. Geophys. Res.*, 102D:16663–16682.
- Morrison, H., Curry, J. A., and Khvorostyanov, V. I. (2005). A new double-moment microphysics parameterization for application in cloud and climate models. part i: Description. *J. Atmos. Sci.*, ?:1665–1677.
- Morton, B.R.and Taylor, G. and Turner, J. (1956). Turbulent gravitational convection from maintained and instantaneous sources. *Proc.Roy.Soc.A*, 234:1–23.
- Murakami, M. (1990). Numerical modeling of dynamical and microphysical evolution of an isolated convective cloud: the 19 july 1981 ccope cloud. *J. Meteor. Soc. Japan*, 68:107–128.
- Nordeng, T. E. (1994). Extended versions of the convective parameterization scheme at ecmwf and their impact on the mean and transient activity of the model in the tropics. *ECMWF Technical Memorandum*, No. 206.
- Ou, S. and Liou, K. (1995). Ice microphysics and climatic temperature feedback. *Atmos. Res.*, 35:127–138.
- Parol, F., Buriez, J., Brogniez, G., and Fouquart, Y. (1991). Information content of AVHRR channels 4 and 5 with respect to the effective radius of cirrus cloud particles. *J. of Applied Meteorology*, 30:973–984.
- Pergaud, J., Masson, V., Malardel, S., and Couvreux, F. (2009). A parameterization of dry thermals and shallow cumuli for mesoscale numerical weather prediction. *Bound.-Layer Meteor.*, 132:83–106.

- Piriou, J.-M., Redelsperger, J.-L., Geleyn, J.-F., Lafore, J.-P., and Guichard, F. (2007). An approach for convective parameterization with memory: separating microphysics and transport in grid-scale equations. doi 10.1175/2007jas2144.1. *J. Atmos. Sci.*, 64:4127–4139.
- Plant, R. S. and Yano, J. (2016). *Parameterization of Atmospheric Convection.*
- Pruppacher, M. and Klett, J. D., . (1998). *Microphysics of clouds and precipitation.* Kluwer Academic Publishers.
- Redelsperger, J.-L., Mahé, F., and Carlotti (2001). A simple and general subgrid model suitable both for surface layer and free-stream turbulence. *Bound.-Layer Meteor.*, 101:375–408.
- Ricard, J. and Royer, J. (1993). A statistical cloud scheme for use in an agcm. *Ann. Geophysicae*, 11:1095–1115.
- Riehl, H. and Marcus, J. (1958). On the heat balance in the equatorial trough zone. *Geophysica*, 6:503–538.
- Rochas, M. and Courtier, P. (1992). La méthode spectrale en météorologie. *Note de Travail ARPEGE*, 30:58 pp.
- Rodgers, C. (1967). The radiative heat budget of the troposphere and lower stratosphere. *Planetary Circulation Project. Dept. of Meteorology. Mass. Instit. Techn.*, A2:99 pp.
- Rothman, L., Gamache, R., Goldman, A., Brown, L., Toth, R., Pickett, H., Poynter, R., Flaud, J., Camy-Peyret, C., Barbe, A., Husson, N., Rinsland, C., and Smith, M. (1987). The HITRAN database, 1986 edition. *Appl. Optics*, 26:4058–4097.
- Rothman, L., Gamache, R., Tipping, R., and et al., C. R. (1992). The hitran database, editions of 1991 and 1992. *J. Quant. Spectrosc. Radiat. Transfer*, 48:469–507.
- Rotstayn, L. (1997). A physically based scheme for the treatment of stratiform clouds and precipitation in large-scale models. i: Description and evaluation of the microphysical processes. *Quart. J. Roy. Meteor. Soc.*, 123:1227–1282.
- Ryo Onishi, K. T. (2012). A warm-bin-cold-bulk hybrid cloud microphysical model. *J. Atmos. Sci.*, 69:1474–1497.
- Sachidananda, H. R. and Zrnić, D. S. (1986). Differential propagation phase shift and rainfall rate estimations. *Radio Science*, 21:235–247.
- Saleeby, S. M. and Cotton, W. R. (2004). A large-droplet mode and prognostic concentration of cloud droplets in the colorado state university regional atmospheric modeling system (rams). part i: Module descriptions and supercell test simulations. *J. Appl. Meteor. and Climatology*, 43:182–195.
- Savijarvi, H. and Raisanen, P. (1997). Long-wave optical properties of water clouds and rain. *Tellus*, 50-A:1–11.
- Shettle, E. and Weinman, J. (1970). The transfer of solar irradiance through inhomogeneous turbid atmospheres evaluated by Eddington's approximation. *J. Atmos. Sci.*, 27:1048–1055.
- Siebesma, A. P., Soares, P., and J., T. (2007). A combined eddy-diffusivity mass-flux approach for the convective boundary layer. *J. Atmos. Sci.*, 64:1230–1248.
- Simmons, A. and Burridge, D. (1981). An energy and angular momentum conserving vertical finite-difference scheme and hybrid vertical coordinate. *Mon. Wea. Rev.*, 109:758–766.
- Simpson, J. (1971). On cumulus entrainment and one-dimensional models. *J. Atmos. Sci.*, 28:449–455.
- Simpson, J., Simpson, R. H., Andrews, D. A., and Eaton, M. A. (1965). Experimental cumulus dynamics. *Reviews of Geophysics*, 3:387–431.
- Simpson, J. and Wiggert, V. (1969). Models of precipitating cumulus towers. *Mon. Weather Rev.*, 97:471–489.
- Slingo, A. (1989). A GCM parameterization for the shortwave radiative properties of water clouds. *J. Atmos. Sci.*, 46:1419–1427.

- Smith, E. and Shi, A. L. (1992). Surface forcing of the infrared cooling profile over the tibetan plateau. Part I: Influence of relative longwave radiative heating at high altitude. *J. Atmos. Sci.*, 49:805–822.
- Smith, R. (1990). A scheme for predicting layer clouds and their water content in a general circulation model. *Quart. J. Roy. Meteor. Soc.*, 116:435–460.
- Soares, P. M. M., Miranda, P. M. A., Siebesma, A. P., and Teixeira, J. (2004). An eddy-diffusivity / mass-flux parameterization for dry and shallow cumulus convection. *Quart. J. Roy. Meteor. Soc.*, 130:3365–3383.
- Stommel, H. (1947). Entrainment of air into a cumulus cloud. *J. Meteorol.*, 4:91–94.
- Sun, Z. and Rikus, L. (1999). Parametrization of effective sizes of cirrus-cloud particles and its verification against observations. *Quart. J. Roy. Meteor. Soc.*, 125:3037–3055.
- Temperton, C. (1991). On scalar and vector transform methods for global spectral models. *Mon. Wea. Rev.*, 119:1303–1307.
- Thouron, O., Brenguier, J., Dubuisson, P., Geoffroy, O., Grini, A., Lac, C., and Sandu, I. (2017). Révision du schéma de transfert radiatif de méso-nh. *Internal Technical Report CNRM*, -:–.
- Tiedtke, M. (1989). A comprehensive mass flux scheme for cumulus parameterization in large-scale models. *Mon. Weather Rev.*, 117:1779–1800.
- T.Mauritsen, e. a. (2012). Tuning the climate of a global model. *J. Adv. Model. Earth Syst.*, 4:?
- Turner, J. (1962). The starting plume in neutral surroundings. *J. Fluid. Mech.*, 13:356–368.
- Verlinde, J., Flatau, P., and Cotton, W. (1990). Analytical solutions to the collection growth equation: Comparison with approximate methods and application to cloud microphysics parameterization schemes. *J. Atmos. Sci.*, 47:2871–2880.
- Warner, J. (1955). The water content in cumuliform cloud. *Tellus*, 7:449–457.
- Williamson, D. and Rosinski, J. (2000). Accuracy of reduced grid calculations. *Quart. J. Roy. Meteor. Soc.*, 126:1619–1640.
- Y., L. and Daum, P. (2003). Parametrization of the autoconversion process. part i: Analytical formulation of the kessler-type parameterizations. *J. Atmos. Sci.*, 61:1539–1548.
- Yanai, M., Esbensen, S., and Chu, J.-H. (1973). Determination of bulk properties of tropical cloud clusters from large-scale heat and moisture budgets. *J. Atmos. Sci.*, 30:611–627.